UNIVERSITY OF COPENHAGEN
DEPARTMENT OF MATHEMATICAL SCIENCES

PhD thesis

Jakob Günther

Quantum algorithms for molecular electronic structure

Date: June 2025

Advisor: Matthias Christandl, University of Copenhagen

This thesis has been submitted to the PhD School of The Faculty of Science, University of Copenhagen

Abstract

The behaviour of electrons lies at the heart of the formation of molecules, giving rise to complex chemical processes and structures up to the biomolecular scale. Studying this behaviour from first principles comes with several advantages: it allows for very accurate predictions, it is a universal approach for all molecular systems and, importantly, it results in a better understanding of chemical mechanisms. However, solving the underlying quantum many-body problem is notoriously difficult for classical computers, and an efficient, system-agnostic method is lacking. Quantum computing is a new paradigm that is naturally suited to numerically study complex quantum systems. In this thesis we describe how quantum computers can be used to solve molecular electronic structure problems. To lower the quantum resource requirements for computing electronic ground state energies, we introduce new methods and improve existing quantum algorithms.

In the first part of the thesis, we combine deterministic and randomized product formulas to reduce the cost of phase estimation, and we find that Hamiltonians describing electrons in molecules are particularly well suited for such an approach. Further algorithmic improvements include a novel analysis of the phase estimation signal, an optimized Hamiltonian representation, more efficient gate compilation and tighter Trotter error estimation. We find that resource estimates on chemical benchmark systems are significantly lower than previous estimates based on product formulas.

In the second part, a large-scale classical simulation of the partially randomized algorithm is performed on a fragment of a biologically relevant ligand, showing that algorithmic errors are orders of magnitude lower than rigorous error bounds suggest.

In the final part of this thesis, we improve upon the chemical model by introducing a method that goes beyond the ubiquitously used 'active-space' approximation. The method includes contributions to the ground state energy from virtual orbitals up to second order in perturbation theory. A key insight is that no additional qubits are needed to represent the virtual orbitals. Moreover, for a fixed active space numerical results suggest that the asymptotic total runtime scales sublinearly with the number of virtual orbitals.

Resumé

Elektronernes opførsel er fundamental for dannelsen af molekyler, hvilket giver anledning til komplekse kemiske processer og strukturer op til biomolekylær skala. At studere denne opførsel ud fra grundlæggende principper har fordele: det muliggør meget nøjagtige forudsigelser, det er en universel tilgang til alle molekylære systemer, og vigtigst af alt resulterer det i en bedre forståelse af kemiske mekanismer. Det er dog notorisk vanskeligt for klassiske computere at løse det underliggende kvante-mange-legeme-problem, og der mangler en effektiv, systemagnostisk metode. Kvanteberegning er et nyt paradigme, der er naturligt egnet til numerisk undersøgelse af komplekse kvantesystemer. I denne afhandling beskriver vi, hvordan kvantecomputere kan bruges til at løse det molekylære elektroniske strukturproblem. For at sænke kravene til kvanteressourcer til beregning af elektroniske grundtilstandsenergier introducerer vi nye metoder og forbedrer eksisterende kvantealgoritmer.

I den første del af afhandlingen kombinerer vi deterministiske og randomiserede produktformler for at reducere omkostningerne ved faseestimering, og vi finder, at Hamiltonoperatorer, der beskriver elektroner i molekyler, er særligt velegnede til en sådan tilgang. Yderligere algoritmiske forbedringer inkluderer en ny analyse af faseestimeringssignalet, en optimeret Hamilton-repræsentation, mere effektiv gate-kompilering og en bedre Trotter-fejlestimation. Vi finder, at ressourceestimater på kemiske benchmarksystemer er betydeligt lavere end tidligere estimater baseret på lignende metoder.

I anden del udføres en storskala klassisk simulering af den delvist randomiserede algoritme på et fragment af en biologisk relevant ligand, hvilket viser, at algoritmiske fejl er størrelsesordener lavere end strenge fejlgrænser antyder.

I den sidste del af denne afhandling introducerer vi en metode der går videre end den allestedsnærværende 'aktive rum'-approksimation og inkluder bidrag til grundtilstandsenergien fra virtuelle orbitaler på en perturbativ måde. En vigtig indsigt er, at en sådan metode ikke behøver yderligere qubits for at repræsentere de virtuelle orbitaler. Desuden tyder numeriske resultater for et fast aktivt rum på, at den asymptotiske samlede løbetid skalerer sublineært med antallet af virtuelle orbitaler.

Table of contents

Publication list

* Jakob Günther, Alberto Baiardi, Markus Reiher, and Matthias Christandl. "More Quantum Chemistry with Fewer Qubits". *Physical Review Research* 6.4 (2024), p. 043021

Mihael Erakovic, Freek Witteveen, Dylan Harley, Jakob Günther, Moritz Bensberg, Oinam Romesh Meitei, Minsik Cho, Troy Van Voorhis, Markus Reiher, and Matthias Christandl. "High Ground State Overlap via Quantum Embedding Methods". *PRX Life* 3.1 (2025), p. 013003

* Jakob Günther, Freek Witteveen, Alexander Schmidhuber, Marek Miller, Matthias Christandl, and Aram Harrow. "Phase estimation with partially randomized time evolution". *arXiv:2503.05647* (2025)

* Marek Miller, Jakob Günther, Freek Witteveen, Matthew S. Teynor, Mihael Erakovic, Markus Reiher, Gemma C. Solomon, and Matthias Christandl. "Phase2: Full-State Vector Simulation of Quantum Time Evolution at Scale". *arXiv:2504.17881* (2025)

Jakob Günther, Thomas Weymuth, Moritz Bensberg, Freek Witteveen, Matthew S. Teynor, F. Emil Thomasen, Valentina Sora, William Bro-Jørgensen, Raphael T. Husistein, Mihael Erakovic, Marek Miller, Leah Weisburn, Minsik Cho, Marco Eckhoff, Aram W. Harrow, Anders Krogh, Troy Van Voorhis, Kresten Lindorff-Larsen, Gemma Solomon, Markus Reiher, and Matthias Christandl. "How to Use Quantum Computers for Biomolecular Free Energies". *arXiv:2506.20587* (2025)

Publications marked by * are part of this thesis.

Preliminaries and notation

Postulates of quantum mechanics

Postulate 1 (State Space)**.** *The states of a quantum system live in a complex Hilbert space $\mathcal{H}$. States are denoted by $|\psi\rangle \in \mathcal{H}$ and the associated inner product is written as*

$$\langle\psi_1|\psi_2\rangle \qquad |\psi_1\rangle, |\psi_2\rangle \in \mathcal{H}. \tag{1}$$

(In this thesis we will assume that all states are normalized, $\langle\psi|\psi\rangle = 1$.)

Postulate 2 (Observables)**.** *Observables of a quantum system are given by self-adjoint operators on $\mathcal{H}$. The expectation value of an observable A with respect to a state $|\psi\rangle$ is given by the real number $\langle\psi|A|\psi\rangle$.*

Postulate 3 (Time evolution and the Hamiltonian)**.** *The time evolution of a quantum system is governed by the Schrödinger-equation*

$$i\frac{\mathrm{d}}{\mathrm{d}t}|\psi(t)\rangle = H|\psi(t)\rangle \tag{2}$$

where H is the Hamiltonian operator of the system. Consequently, the time-propagator is given by a unitary operator $U(t) = e^{-iHt}$ on $\mathcal{H}$, with the action

$$|\psi(t)\rangle = e^{-iH(t-t_0)}|\psi(t_0)\rangle. \tag{3}$$

Postulate 4 (Many-body quantum system)**.** *The Hilbert space of multiple quantum particles with associated single particle Hilbert spaces $\mathcal{H}_1, \ldots, \mathcal{H}_n$ is given by their tensor product $\bigotimes_{l=1}^{n}\mathcal{H}_l$. This naturally induces a group action of the permutation group S_n, and the state of identical particles occuring in nature either live in the symmetric (bosons) or antisymmetric subspace (fermions).*

Notation

- $\mathcal{H}$: Hilbert space
- $\mathcal{H}_1$: single-electron Hilbert space
- $\mathcal{H}_{\text{qc}}$: multi-qubit Hilbert space
- $|\psi\rangle$: a normalized vector in a Hilbert space
- H_{el}: electronic structure Hamiltonian
- E_k: k-th eigenvalue of Hamiltonian
- $|\psi_k\rangle$: k-th eigenstate of Hamiltonian
- P: Pauli operator or a tensor product of Pauli operators
- $\mathcal{F}$: The fermionic Fock space
- ϕ_p: spatial orbital
- $a_p, a_q^\dagger$: annihilation and creation operator
- γ_i: Majorana operator
- L: number of terms in Hamiltonian
- λ: sum of weights in Hamiltonian
- ε: precision
- $|\phi\rangle$: initial state for quantum phase estimation
- η: ground state overlap of initial state
- $\|\cdot\|$: operator norm (largest singular value)
- $\tilde{\mathcal{O}}(\cdot)$: big-O notation hiding polylogarithmic factors

Chapter 1

Introduction

> *The underlying physical laws necessary for the mathematical theory of a large part of physics and the whole of chemistry are thus completely known, and the difficulty is only that the exact application of these laws leads to equations much too complicated to be soluble.*
>
> Pauli Dirac, 1929 [6][1]

To accurately predict the behaviour of molecular systems from first principles it is necessary to apply our best theory of microscopic physics — quantum physics — to the basic constituents of molecules, nuclei and electrons. While the relevant equations and quantities in such a theory have been known for almost one hundred years now, the underlying quantum many-body problem is very hard to solve. Many approximate and heuristic methods have been developed that for some problems in certain parameter regimes give a qualitatively, and sometimes even quantitatively accurate answer. However, the path to exact solutions via efficient calculations on classical computers is obstructed by complexity-theoretic limits. Quantum computers represent and manipulate information in a fundamentally different and more powerful way than classical computers. While not all computational problems benefit from quantum computation, many problems related to the study of quantum systems experience enormous asymptotic quantum speed-ups. Consequently, quantum computing presents a promising way to predict and understand all types of atomistic matter from first principles. However, to date it is still unknown whether the resource requirements of quantum algorithms for molecular problems will be low enough for such methods to be

[1] Paul Dirac happens to be my academic great-great-great grandfather.

ever practically relevant. We will discuss the challenges present when trying to solve molecular problems using a quantum computer.

In this chapter we will first introduce the relevant concepts and main problems in quantum chemistry. Afterwards, we introduce the key concepts of quantum computing and explain how problems in quantum chemistry can be mapped to problems solvable using a digital quantum computer.

1.1 The electronic structure problem in chemistry

Quantum chemistry aims at predicting all chemical phenomena, like chemical binding, the geometry of a molecule and chemical reactions, in a reductionist way by applying the laws of quantum physics to electrons and nuclei. In this section we give a brief introduction to this field, and we explain why the electronic ground state plays such a central role when studying molecules computationally.

1.1.1 Many-electron quantum states

The spatial state space of a single electron is described by the Hilbert space of complex square-integrable functions over $\mathbb{R}^3$, and its elements are denoted by $|\psi\rangle \in L^2(\mathbb{R}^3)$. It is equipped with the inner product

$$\langle\psi_1|\psi_2\rangle = \int \psi_1^*(r)\psi_2(r)dr.$$

Electrons are spin-1/2 particles, so the full single electron Hilbert space is given by $L^2(\mathbb{R}^3) \otimes \mathbb{C}^2$. An element of this space is referred to as a *spin orbital* in the chemistry context (or just 'orbital'). When dealing with multiple identical quantum particles one needs to consider the correct particle-exchange symmetry. For N_{el} electrons, which are fermions, we therefore work with the antisymmetric subspace, denoted by $\Lambda^{N_{\text{el}}}(L^2(\mathbb{R}^3) \otimes \mathbb{C}^2)$. Hence, a state that combines N_{el} occupied spin orbitals $\{\varphi_1, \varphi_2, \ldots, \varphi_{N_{\text{el}}}\}$ is described by a *Slater determinant*

$$\psi(x_1, \ldots, x_{N_{\text{el}}}) = \frac{1}{\sqrt{N_{\text{el}}!}} \begin{vmatrix} \varphi_1(x_1) & \varphi_2(x_1) & \cdots & \varphi_{N_{\text{el}}}(x_1) \\ \varphi_1(x_2) & \varphi_2(x_2) & \cdots & \varphi_{N_{\text{el}}}(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_1(x_{N_{\text{el}}}) & \varphi_2(x_{N_{\text{el}}}) & \cdots & \varphi_{N_{\text{el}}}(x_{N_{\text{el}}}) \end{vmatrix},$$

where x_i is the combined spatial and spin variable of electron i. For numerical calculations it is necessary to discretise and truncate the spatial part $L^2(\mathbb{R}^3)$ to

a set of orthonormal functions $\{\phi_i\}_{i=1}^n$ to arrive at a finite-dimensional problem.[2] In computational chemistry one typically employs linear combinations of nuclei-centered Gaussian functions, since they efficiently capture relevant features of electronic states in molecules. The basis elements of the Hilbert space of the electron spin are denoted by $\chi_\uparrow, \chi_\downarrow$, and we are using letters σ, τ as indices for the spin variable. Then, let $\mathcal{H}_1 = \mathrm{span}(\phi_1, \ldots, \phi_n) \otimes \mathrm{span}(\chi_\uparrow, \chi_\downarrow)$ be the underlying $2n$-dimensional approximation to $L^2(\mathbb{R}^3) \otimes \mathbb{C}^2$ that is relevant for all practical purposes.

Instead of explicitly writing down the coordinate representation of a state, it is convenient to work in the second-quantized description, where the operators, and not the states, take account of the antisymmetry. In this picture, the relevant space is the fermionic *Fock space*, defined as the sum of antisymmetrized k-electron subspaces:

$$\mathcal{F} = \bigoplus_{k=0}^{2n} \Lambda^k(\mathcal{H}_1).$$

The canonical orthonormal basis of $\mathcal{F}$ is the set of *occupation number vectors*

$$|n_{1\uparrow} n_{1\downarrow} \ldots n_{n\uparrow} n_{n\downarrow}\rangle\,,$$

where $n_{i\sigma} \in \{0, 1\}$ indicates whether spin orbital (i, σ) is occupied or not. Operators are expressed in terms of creation and annihilation operators, $a^\dagger_{q\sigma}, a_{p\tau}$, which fulfil the fermionic anticommutation relations $\{a^\dagger_{q\sigma}, a_{p\tau}\} = \delta_{pq}\delta_{\sigma\tau}$.

1.1.2 The electronic structure Hamiltonian

The dynamics and energy levels of a molecular system are determined by the eigenstates and eigenenergies of the *molecular Hamiltonian*, which is the sum of kinetic and Coulomb-interaction terms of electrons and nuclei. We work in atomic units, $\hbar = m_e = e = 1$, where m_e is the mass of the electron and e is the charge of the electron. In coordinate representation the molecular Hamiltonian of N_{el} electrons and N_{nuc} nuclei reads

$$\begin{aligned} H_{\mathrm{mol}} &= -\sum_{I=1}^{N_{\mathrm{nuc}}} \frac{\Delta_I}{2M_I} - \sum_{i=1}^{N_{\mathrm{el}}} \frac{\Delta_i}{2} - \sum_{I=1}^{N_{\mathrm{nuc}}} \sum_{i=1}^{N_{\mathrm{el}}} \frac{Z_I}{|R_I - r_i|} + \sum_{i<j}^{N_{\mathrm{el}}} \frac{1}{|r_i - r_j|} + \sum_{I<J}^{N_{\mathrm{nuc}}} \frac{Z_I Z_J}{|R_I - R_J|} \\ &= T_{\mathrm{nuc}} + T_{\mathrm{el}} + V_{\mathrm{nuc,el}} + V_{\mathrm{el,el}} + V_{\mathrm{nuc,nuc}}. \end{aligned}$$

[2] In quantum chemistry such a set of orbitals is called a 'basis' (although it is not a basis of $L^2(\mathbb{R}^3)$).

Here, capital and lowercase indices correspond to nuclei and electrons, respectively. M_I and $Z_I \in \mathbb{N}$ are mass and nuclear charge number of nucleus I, Δ is the Laplace-operator and $r_i, R_I \in \mathbb{R}^3$ are the coordinates of electrons and nuclei, respectively.[3] Only for hydrogen-like atoms is an exact, closed-form expression for the spectrum of this operator known. The *Born-Oppenheimer approximation* simplifies the problem by separating the dynamics of electrons and nuclei. For fixed nuclear coordinates $\mathbf{R}$, one first solves for the eigenenergies of the *molecular electronic structure Hamiltonian*

$$H_{\text{el}} \equiv H_{\text{el}}(\mathbf{R}) = T_{\text{el}} + V_{\text{nuc,el}} + V_{\text{nuc,nuc}} + V_{\text{el,el}}.$$

This operator acts only on the Hilbert space of the electrons $\Lambda^{N_{\text{el}}}(\mathcal{H}_1)$, but it is parameterized by the nuclear coordinates via the Coulomb potential $V_{\text{nuc,el}}$ and the constant $V_{\text{nuc,nuc}}$. Consequently, the eigenenergies $E_{\text{el},i}(\mathbf{R})$ are functions of $\mathbf{R}$. In most scenarios, the gap between ground state energy and first excited state energy is so large that the thermal distribution of states at ambient temperature is to a very good approximation equal to the ground state. Thus, up to a good approximation the ground state energy $E_{\text{el},0}(\mathbf{R})$ forms the potential energy surface (PES) on which the nuclei move.[4] The time dynamics of the nuclei in the Born-Oppenheimer picture are governed by

$$H_{\text{nuc}} = T_{\text{nuc}} + E_{\text{el},0}(\mathbf{R}).$$

Many properties of a molecule are determined by the PES. For example, its minima correspond to the geometries of (meta-)stable molecular species and first-order saddle points correspond to barrier heights, encoding the necessary activation energy of a chemical reaction. In summary, the task of computing $E_{\text{el},0}(\mathbf{R})$ lies at the core of how molecular phenomena and processes are studied theoretically.

Given a set of spatial orbitals $\{\phi_p\}_{p=1}^n$ the electronic structure Hamiltonian in second quantization takes the form

$$H_{\text{el}} = \sum_{pq} \sum_{\sigma} h_{pq} a_{p\sigma}^\dagger a_{q\sigma} + \frac{1}{2} \sum_{pqrs} \sum_{\sigma\tau} h_{pqrs} a_{p\sigma}^\dagger a_{r\tau}^\dagger a_{s\tau} a_{q\sigma}$$

[3] Several approximations were used for H_{mol} already: nuclei are treated as point particles and relativistic effects are neglected, making the Hamiltonian spin-free.

[4] The ansatz of separating nuclei and electron dynamics breaks down if low-lying excited states and the ground state come close to each other. In these cases it is a common approach to solve the electronic Schrödinger equation for these low-lying states and use them as a starting point for improved nuclear solutions.

where σ, τ correspond to spin variables and the coefficients are defined as

$$h_{pq} = \int \phi_p^*(r) \left(-\frac{\Delta}{2} - \sum_I \frac{Z_I}{|r - R_I|} \right) \phi_q(r) dr$$
$$h_{pqrs} = \int\int \frac{\phi_p^*(r_1)\phi_r^*(r_2)\phi_s(r_2)\phi_q(r_1)}{|r_1 - r_2|} dr_1 dr_2.$$

We will from now on take the Hamiltonian H_{el} as given and for simplicity abbreviate $E_{\text{el},0}(\mathbf{R}) = E_0$.

Conceptually, the problem of computing the ground state energy E_0 is a simple linear algebra problem; H_{el} can be written as a matrix (for example in the occupation number basis) and we need to find its smallest eigenvalue. The problem is of course that this matrix has dimension $2^{2n} \times 2^{2n}$, so brute force diagonalization is possible only for small systems. The field of quantum chemistry has produced a huge variety of heuristic electronic structure methods to overcome this limitation, with different methods offering different trade-offs between computational difficulty and accuracy of the result.

The *variational principle* guides many electronic structure methods, it states that the expectation value of H_{el} for any state $|\psi\rangle$ is lower bounded by the exact ground state energy, $E_0 \leq \langle\psi|H_{\text{el}}|\psi\rangle$. A cornerstone of electronic structure theory is the *Hartree-Fock* method, it uses the variational principle to approximate the ground state by the lowest energy Slater determinant

$$E_{\text{HF}} = \min_{|\psi\rangle \text{ Slaterdet}} \langle\psi|H_{\text{el}}|\psi\rangle \,.$$

The minimization yields an orbital basis, the Hartree-Fock orbitals, with associated orbital energies, and the Hartree-Fock determinant corresponds to the Slater determinant in which the N_{el} lowest-energy orbitals are occupied. For quantitative results this is usually not accurate enough, however the Hartree-Fock orbitals and the Hartree-Fock state provide a good starting point for more sophisticated methods.

The *active-space* approach starts from the Hartree-Fock state, but allows for variable orbital occupations around the Fermi gap (the energy between highest occupied orbital and lowest unoccupied orbital in the Hartree-Fock state). These orbitals are referred to as active orbitals. Low-energy orbitals (core orbitals) are left occupied, while high-energy orbitals are left unoccupied. As a result, one arrives at a truncated version of H_{el} (active space Hamiltonian), which is small enough for exact diagonalization, while still describing ground states decently. It is desirable to use large active spaces to obtain more accurate results, and quantum computers might be able to deal with much larger active spaces than classical computers can. In chapter 4 we present a project that improves upon

the active-space approximation with a quantum algorithm for computing energy corrections perturbatively.

1.1.3 Symmetries of the electronic structure Hamiltonian

The electronic structure Hamiltonian H_{el} is invariant under various symmetry operations. The choice of the orthonormal orbital basis of the single-electron Hilbert space $\mathcal{H}_1$ is arbitrary, so unitary transformations of its basis leave H_{el} unchanged. For practical reasons one usually only considers unitary transformations among the spatial degrees of freedom. Formally, this symmetry is described by the action of the unitary group $U(n)$ on the n (spatial) orbitals, for $u \in U(n)$ defined via

$$\phi_p \mapsto \sum_{i=1}^{n} u_{ip}\phi_i$$

from which the transformation for creation and annihilation operators as well as the coefficients h_{pq}, h_{pqrs} follows

$$\begin{aligned} a^\dagger_{p\sigma} &\mapsto \sum_{i=1}^{n} u_{ip} a^\dagger_{i\sigma} & a_{p\sigma} &\mapsto \sum_{i=1}^{n} u^*_{ip} a_{i\sigma} \\ h_{pq} &\mapsto \sum_{i,j=1}^{n} u_{ip} u^*_{jq} h_{pq} & h_{pqrs} &\mapsto \sum_{i,j.k,l=1}^{n} u_{ip} u^*_{jq} u_{kr} u^*_{ls} h_{pqrs}. \end{aligned}$$

In the quantum chemistry literature such transformations are called *orbital rotations*. Let $\kappa = -\kappa^\dagger$ be an element of the Lie algebra $\mathfrak{u}(n)$ (the set of skew-hermitian matrices), then the unitaries can be parameterized through the exponential map [7], $u = \exp\left(\sum_{p,q=1}^{n} \kappa_{pq}\right) \in U(n)$. The corresponding representation on the electronic Fock space $\mathcal{F}$ is then given by the unitary operators $\exp\left(\sum_\sigma \sum_{p,q=1}^{n} \kappa_{pq} a^\dagger_{p\sigma} a_{q\sigma}\right)$. An important result about orbital rotations is the Thouless theorem [8], which states that Slater determinants are mapped to Slater determinants under orbital rotations.

The freedom to choose an orbital basis can be used to change the Hamiltonian coefficients in a way that simplifies computations. For many purposes it is useful to choose a basis in which the orbitals are as localized as possible. Fig. 1.1 illustrates the effect of orbital localization for a hydrogen chain. While the canonical Hartree-Fock orbitals are completely delocalized over the system, the extent of the localized orbitals is significantly reduced. Not only are localized orbitals easier to interpret (*e.g.* for assigning relevant orbitals in bond-breaking processes), but the coefficient tensors h_{pq}, h_{pqrs} become sparser, too. In chapter 2 we will

see that for some quantum algorithms the sum of absolute values of Hamiltonian coefficients is an important cost factor. Because of the sparsity, localized orbitals tend to be superior to Hartree-Fock orbitals in this context [9, 10].

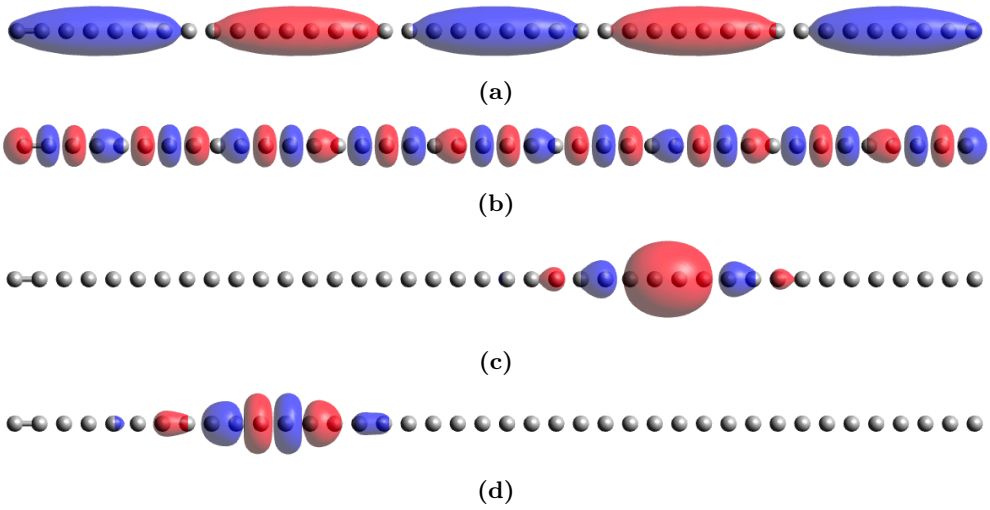


Figure 1.1: The shape of different types of orbitals for a hydrogen chain with 40 orbitals. (a) and (b): occupied and unoccupied Hartree-Fock orbitals, (c) and (d): occupied and unoccupied localized orbitals.

Another symmetry we want to mention relates to the electron number. The Hamiltonian H_{el} only contains particle conserving operators (creation and annihilation operators appear in pairs), hence it commutes with the particle number operator

$$N = \sum_{\sigma} \sum_{p=1}^{n} n_{p\sigma}$$

where $n_{p\sigma} = a_{p\sigma}^{\dagger} a_{p\sigma}$ are the orbital occupation number operators. For an N_{el}-electron state the particle number operator has eigenvalue N_{el}. Let $A = \sum_{\sigma} \sum_{p,q=1}^{n} g_{pq} a_{p\sigma}^{\dagger} a_{q\sigma}$ be an arbitrary single-electron term, then H_{el} can be modified without altering its action in the N_{el}-electron sector:

$$H_{\text{el}} \longrightarrow H_{\text{el}} + \alpha(N - N_{\text{el}}) + A(N - N_{\text{el}}) + (N - N_{\text{el}})A,$$

where α is some constant. The 1- and 2-electron parts of H_{el} are then shifted by $\alpha N - 2N_{\text{el}}A$ and $AN + NA$, respectively. This degree of freedom can also be used to reduce the sum weights in H_{el} [10, 11].

One can employ this *symmetry-shift* trick also for other symmetries of the molecular electronic structure Hamiltonian. These include the spin-projection S_Z, the total spin S^2 and point-group symmetries of the nuclear configuration [12].

1.2 Quantum computing

The elementary object of a digital quantum computer is a qubit, which is a two-dimensional complex Hilbert space, $\mathcal{H} \cong \mathbb{C}^2$, generalizing the notion of a classical bit. Over the last three decades, experimental physicists have come up with various ways to realize qubits. Among the most promising approaches are qubits based on superconducting circuits and implementations using electronic energy levels in atoms and ions. A quantum computer is a quantum system that is composed of multiple qubits, and whose time evolution is highly controllable [13]. The underlying Hilbert space of a quantum computer with n qubits is $\mathcal{H}_{\text{qc}} = \mathbb{C}^{2^n}$, so classical computers would need to store 2^n complex amplitudes to even represent a state in $\mathcal{H}_{\text{qc}}$. Operations on the quantum state are defined by a set of local unitaries, referred to as *gates*. Most quantum computing platforms can perform only single- and two-qubit gates, and more complex operations need to be decomposed into these native gates. Moreover, quantum computers can perform projective measurements of the quantum state, resulting in a single bit of classical information per qubit measured. The measurement defines a distinguished basis in $\mathbb{C}^2$, whose elements are labelled by $|0\rangle$ and $|1\rangle$. The corresponding n-qubit product basis is referred to as the *computational basis*, its elements are labeled by bit strings $|x_0 \dots x_{n-1}\rangle, x_i \in \{0, 1\}$.

A *quantum circuit* is sequence of quantum gates and single-qubit measurements. The circuit *width* refers to the number of qubits involved, and the circuit *depth* measures the number of layers of gates. Sometimes the set of qubits is divided into different registers, where for example a system register holds the state of a simulated system, while additional ancilla qubits are only used to facilitate operations on the system register. A quantum algorithm is a method of generating quantum circuits and processing the resulting measurement bit strings to solve a computational task. At any step in the quantum circuit, the current state can be written as $|\psi\rangle = \sum_{x\in\{0,1\}^n} c_x |x\rangle$. In contrast to the classical setting, one does not have full access to the information of $|\psi\rangle$, and a measurement will output only one of the bit strings x, according to the probability distribution $p_x = |c_x|^2$. Moreover, the measurement operation projects $|\psi\rangle$ onto $|x\rangle$ and is therefore inherently destructive. The challenge and art of quantum algorithm design lies in shaping the measurement probability distribution such that a bit string x encoding the solution to a problem is measured with high probability.

1.2.1 Quantum gates

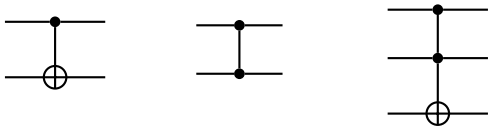
We now list some of the quantum gates that are frequently used in descriptions of quantum circuits. Relevant single qubits are the Pauli operations X, Y, Z, the

Hadamard gate H and the S- and T-gate

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \qquad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \qquad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$
$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \qquad S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}, \qquad T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$$

Some of the most frequently used entangling gates are the controlled-NOT, controlled-Z and Toffoli (or controlled-controlled-NOT) gates:

$$\begin{aligned} \text{CNOT} &= |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes X \\ \text{CZ} &= |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes Z \\ \text{Toffoli} &= (I \otimes I - |11\rangle\langle 11|) \otimes I + |11\rangle\langle 11| \otimes X. \end{aligned}$$



CNOT CZ Toffoli

The set of tensor products of Pauli matrices, $\{P_1 \otimes \cdots \otimes P_n | P_i \in \{I, X, Y, Z\}\}$, whose elements we refer to as *Pauli strings*, forms a basis of self-adjoint operators on $\mathcal{H}_{qc}$. When we additionally allow the Pauli strings to carry a phase factor $\alpha \in \{\pm 1, \pm i\}$, then this set forms the *Pauli group*, denoted by $\mathcal{P}_n$. Related to the Pauli group is the *Clifford* group, defined as the normalizer of $\mathcal{P}_n$,

$$\mathcal{C}_n := \{U \in U(n) : UPU^\dagger \in \mathcal{P}_n \forall P \in \mathcal{P}_n\}.$$

The Clifford group can be generated by the action of the single- and two-qubit gates $\{H, S, \text{CNOT}\}$. A quantum circuit that only contains Clifford gates can create superpositions of all computational basis states, yet it can be efficiently simulated with a classical computer, a result known as the Gottesman-Knill theorem [14].

The complex exponentials of Pauli strings are referred to as *Pauli rotations*. Since Pauli strings are involutions, $P^2 = I$, the expression for a Pauli rotation takes a simple form

$$R_P(\theta) = e^{-i\frac{\theta}{2}P} = \cos\left(\frac{\theta}{2}\right) I - i \sin\left(\frac{\theta}{2}\right) P.$$

In chapter 3 we describe a project in which this expression is used to accelerate classical simulations of quantum time evolution. Pauli rotations acting non-trivially on k qubits can be decomposed into $2k$ CNOTs, one single-qubit R_Z rotation and at most $2k$ Hadamard and S-gates. In Fig. 1.2 we show the circuit construction explicitly for $R_{XZY}(\theta)$. The controlled versions of Pauli rotations

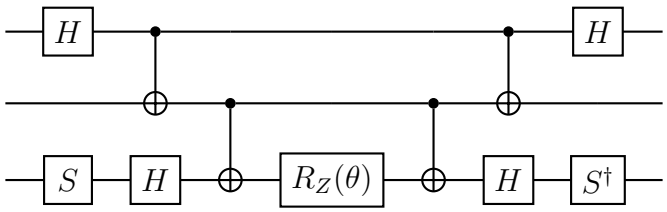


Figure 1.2: Decomposition of Pauli rotation $R_{XZY}(\theta)$.

are standard building blocks in quantum algorithms. Note here that instead of controlling every gate in the gate decomposition of a Pauli rotation, one only needs to control the central R_Z gate, since the outer parts are inverses of each other (see Fig. 1.2). A controlled R_Z gate is depicted in Fig. 1.3. Hence controlling a Pauli rotation only costs two additional CNOT gates and one additional R_Z gate, independent of the length of the Pauli string.

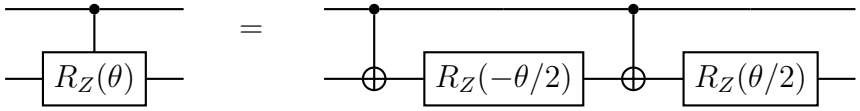


Figure 1.3: Decomposition of controlled-R_Z.

1.2.2 Gate compilation

Quantum algorithm descriptions are given in terms of a *universal gate set* from which all other unitaries can be constructed. Often two- or three-qubit entangling gates (CNOT, CZ or TOFFOLI for example) plus arbitrary single qubit Pauli rotations are used. However, one might only have a small, finite set of gates available, which must be used to generate the other gates.

Formally, we need a finite set of single-qubit gates $\mathcal{G} = (g_1, \ldots, g_m) \subset SU(2)$ that generates a dense subset in $SU(2)$, meaning that for every $U \in SU(2)$ and $\varepsilon > 0$ one can find a finite sequence g of elements in $\mathcal{G}$ such that $\|U - g\| \leq \varepsilon$. Naturally this raises the question of how long such a sequence needs to be. A slow convergence would directly translate into a higher compilation cost, resulting in a larger runtime overhead of all quantum algorithms. Fortunately, the

Solovay-Kitaev theorem resolves this issue, stating that it is possible to converge to arbitrary elements in $SU(2)$ fast [15]:

Theorem 1.2.1 (Solovay-Kitaev)**.** *Given a finite subset $\mathcal{G}$ of $SU(2)$ that is closed under inverses and that generates a dense subset of $SU(2)$, then there exists a constant $c > 0$ such that for every $U \in SU(2)$ and every $\varepsilon > 0$ one can find an ε-approximation to U which is a sequence of $l = \mathcal{O}\left(\log^c 1/\varepsilon\right)$ elements of $\mathcal{G}$.* [5]

The Solovay-Kitaev theorem assures that the use of finite universal gate sets is still efficient, in the sense that any universal set only requires polylogarithmic overhead. For the purpose of quantum algorithm design this justifies that we can largely ignore hardware-specific details of how universal gate sets are constructed in practice. Standard universal gate sets that are often used in theory are Clifford + T-gate or Clifford + Toffoli.

1.2.3 Quantum computing and noise

In order to implement gates and perform measurements quantum computers and their environment need to interact. However, the interaction with external degrees of freedom makes quantum computers inherently susceptible to noise. As will be shown in chapter 2, quantum circuits for chemistry applications easily have 10^9 gates or more. Assuming a uniform gate error rate p, then the probability of no error occuring is lower bounded by $(1-p)^{10^9} \geq 1-10^9 p$, suggesting error rates of at most $p \sim 10^{-9}$ are needed to guarantee error-free computations. Despite the progress in quantum hardware manufacturing, with gate fidelites reaching 99.95% and higher [16, 17, 18], such low error rates seem unrealistic to obtain just through hardware improvements. Instead, error rates must be decreased through *quantum error correction*, which are algorithmic schemes that protect quantum information by storing it redundantly and non-locally over multiple qubits. By concatenating several layers of error correction, the error rate can in principle be made arbitrarily small [19]. One distinguishes between *physical* (the actual hardware qubits) and *logical* qubits (the high-accuracy qubits emerging via error-correction). We will not discuss concrete error correction methods here, but simply note that a decrease in logical error rate comes at the cost of more physical qubits.

For quantum error correction schemes that appear to be relevant in practice the implementation of non-Clifford gates is significantly more challenging than for Clifford gates [20]. For this reason, for quantum algorithms that make use of

[5] Elements in $SU(d)$ and $U(d)$ only differ by a phase factor, which cannot be observed in quantum theory. For this reason it is sufficient that the Solovay-Kitaev theorem considers $SU(2)$ and not $U(2)$.

deep circuits — such that error correction is unavoidable — it is reasonable to only use the number of non-Clifford gates as a cost metric. Thus, when using a Clifford + T or Clifford + Toffoli universal gate set, then one reports the T-gate or Toffoli count. This is done for the quantum algorithm resource estimates in chapter 2.

1.3 Mapping fermions to qubits

We consider again the Fock space $\mathcal{F}$ of all many-electron states over $2n$ spin orbitals. The dimension of $\mathcal{F}$ is $\sum_{k=0}^{2n} \binom{2n}{k} = 2^{2n}$, hence it is isomorphic to the $2n$-qubit Hilbert space $\mathcal{H}_{qc} = (\mathbb{C}^2)^{\otimes 2n}$. Finding a suitable isomorphism, a *fermion-to-qubit mapping*, that maps fermionic states and operators to qubit counterparts is the first step in solving many-electron problems on a quantum computer. The main challenge in constructing fermion-to-qubit mappings is that local operators on $\mathcal{F}$ can in general not be mapped to local operators on $\mathcal{H}_{qc}$.

To accurately simulate a many-electron system on a quantum computer, we want a spectrum-preserving mapping from self-adjoint operators on $\mathcal{F}$ to self-adjoint operators on $\mathcal{H}_{qc}$. A natural basis of self-adjoint operators on $\mathcal{H}_{qc}$ is the set of Pauli strings, $\{P_1 \otimes \cdots \otimes P_{2n} | P_i \in \{I, X, Y, Z\}\}$. Pauli strings are unitary and self-adjoint, but the elementary operators on $\mathcal{F}$, the creation and annihilation operators $a_{p\sigma}, a_{p\sigma}^\dagger$, are neither unitary nor self-adjoint, and they are not even real-diagonalizable. To facilitate the search for suitable mappings, it is helpful to introduce *Majorana operators* $\gamma_{p\sigma,0}, \gamma_{p\sigma,1}$. They are related to the creation and annihilation operators via

$$\begin{aligned} \gamma_{p\sigma,0} &= a_{p\sigma} + a_{p\sigma}^\dagger \\ \gamma_{p\sigma,1} &= i\left(a_{p\sigma}^\dagger - a_{p\sigma}\right). \end{aligned}$$

Similar to Pauli operators, they are self-adjoint and unitary. Moreover, they satisfy the anticommutation relations

$$\{\gamma_{p\sigma,i}, \gamma_{q\tau,j}\} = 2\delta_{pq}\delta_{\sigma\tau}\delta_{ij}.$$

Therefore, any labelling of $2n$ pairwise anti-commuting Pauli strings defines a viable bijective fermion-to-qubit mapping. The electronic structure Hamiltonian

can be expressed in terms of the Majorana operators [9, 21]

$$\begin{aligned} H_{el} = &\left(\sum_p h_{pp} + \frac{1}{2} \sum_{pr} h_{pprr} - \frac{1}{2} \sum_{pr} h_{prrp} \right) I \\ &+ \frac{i}{2} \sum_{pq} \sum_{\sigma} \left(h_{pq} + \sum_r h_{pqrr} - \frac{1}{2} \sum_r h_{prrq} \right) \gamma_{p\sigma,0} \gamma_{q\sigma,1} \\ &- \frac{1}{8} \sum_{pqrs} \sum_{\sigma\tau} h_{pqrs} \gamma_{p\sigma,0} \gamma_{q\sigma,1} \gamma_{r\tau,0} \gamma_{s\tau,1}, \end{aligned} \tag{1.1}$$

and one arrives at the qubit Hamiltonian by simply replacing the Majorana operators with the corresponding anti-commuting Pauli operators.

We note that for resource estimates of some quantum algorithms, the only thing we need to know about $H_{\text{el}} = \sum_i h_i P_i$ is the sum of weights, $\lambda \equiv \sum_i |h_i|$. In these cases, it is not necessary to perform the fermion-to-qubit mapping explicitly, λ can be computed directly from (1.1).[6]

The mapping of the states follows from the mapping of operators: Given the mapping to Pauli strings $\gamma_{p\sigma,i} \to P_{p\sigma,i}$, the number operator of the spin orbital (p, σ) is mapped to $n_{p\sigma} \to \frac{1}{2}(I + iP_{p\sigma,0}P_{p\sigma,1})$. The $+1$ and -1 eigenspaces of $iP_{p\sigma,0}P_{p\sigma,1}$ correspond to states for which the spin orbital (p, σ) is occupied and unoccupied, respectively. Hence a Slater-determinant is mapped to the unique state that lives in the intersection of the $+1$ (-1) eigenspaces of $iP_{p\sigma,0}P_{p\sigma,1}$ corresponding to occupied (unoccupied) spin orbitals (p, σ). The resulting state is in general not a computational basis state. However, for the construction of mappings we describe next the additional property of mapping Slater determinants to computational basis states can be enforced.

We now present a simple way to describe a large family of fermion-to-qubit mappings [22, 23, 24]. Consider a *ternary tree*, *i.e.* a tree graph with at most three children per vertex. We will refer to vertices without children as *leafs*, and we call all other vertices *nodes*. Denote by m the number of nodes in the tree, then there are $2m + 1$ leafs. A set of $2m$ pairwise anti-commuting Pauli strings can be constructed as follows: For each node, label the three edges to its children by X, Y and Z from left to right. Distribute the labels $\{1, 2, \ldots, m\}$ to the m nodes (arbitrarily). Also, remove one of the leaf nodes; the remaining $2m$ leaf nodes represent the Pauli strings as follows: For each leaf node, trace back the edges through the ternary tree up to root node. Each edge label corresponds

[6] A small caveat: Due to the anticommutation relations and the index-permutation symmetries of the coefficient tensors h_{pq}, h_{pqrs} there are multiple terms corresponding to the same Pauli string in Eq. 1.1. In [9, 21] a minimal representation of H_{el} is derived, from which the exact value of λ can be computed.

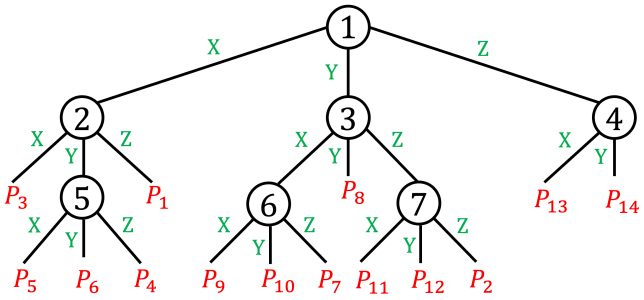


Figure 1.4: A ternary tree with $m = 7$ nodes. The Pauli string assignment leads, for example, to $P_5 = X_1Y_2X_5, P_2 = Y_1Z_3Z_7, P_8 = Y_1Y_3$.

to a Pauli operator, and the label of its parent node corresponds to the qubit the Pauli operator is acting on. See Fig. 1.4 for an example of the Pauli string assignment. Note that the maximal weight of any Pauli string (*i.e.* the number of non-identity terms, denoted by $|P_i|$) is equal to the depth of tree. All pairs of Pauli strings generated this way anti-commute by construction, because each pair has exactly one qubit on which the Pauli strings have different non-identity Pauli operators. This qubit is the one where the corresponding paths from the root to the leaf nodes splits into two; on qubits between the split and the root node the two Pauli strings act identically, while on qubits after the split they act on different qubits. Finally, since the spin orbital (p, σ) corresponds to two Majorana operators, the Pauli strings need to paired. In [24] it is shown how for every ternary tree this pairing can be performed in such a way that Slater determinants are mapped to computational basis states.

Let's discuss some of the mappings in more detail. The Jordan-Wigner (JW) mapping in Fig. 1.5 encodes the anticommutation relations in the most straightforward way via tails of Pauli Z operators (we combine spatial and spin indices into a single index):

$$\gamma_{p,0} \to Z_1 \otimes Z_2 \otimes \cdots \otimes Z_{p-1} \otimes X_p$$
$$\gamma_{p,1} \to Z_1 \otimes Z_2 \otimes \cdots \otimes Z_{p-1} \otimes Y_p.$$

The number operators take the form $n_p \to \frac{1}{2}(I - Z_p)$, and their $+1$ and 0 eigenspaces are given by the states for which qubit p is in state $|1\rangle$ and $|0\rangle$, respectively. Consequently, Slater-determinants are directly mapped to computational states with the same bit string, $|x_0x_1 \ldots x_{m-1}\rangle_{\text{Slater}} \to |x_0x_1 \ldots x_{m-1}\rangle_{\text{qubit}}$. In the Jordan-Wigner mapping, the average weight of Pauli strings in the Hamiltonian is $\mathcal{O}(m)$. By flipping the JW ternary tree horizontally (so that X and Z labels are swapped) we arrive at the *parity mapping*.

If one is interested in fermion-to-qubit mappings resulting in Pauli strings of low weight, then one might want to use ternary trees that have low depth and

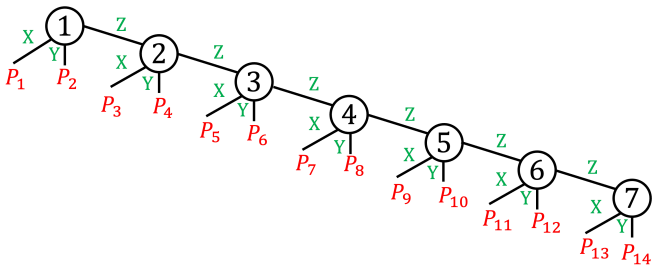


Figure 1.5: The ternary-tree representing the Jordan-Wigner mapping.

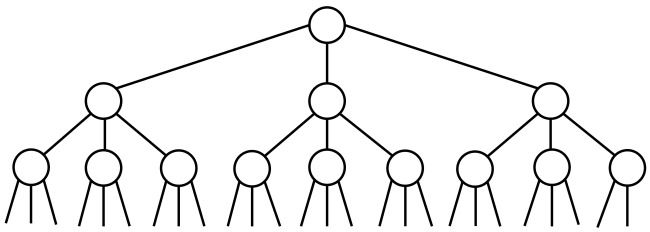


Figure 1.6: A low-depth ternary tree, resulting in Pauli strings of low weight.

that are filled densely. An example is given in Fig. 1.6. With this construction a maximal weight of $\lceil 2 \log_3 m \rceil$ can be obtained [22]. The *Bravyi-Kitaev* mapping [25] can be represented by a similar low-depth ternary tree [24], reaching Pauli weight $\mathcal{O}(\log_2 m)$.

How relevant is the Pauli weight as a cost factor in quantum algorithms? A standard access to $H_{\text{el}} = \sum_{i=1}^{L} h_i P_i$ on a quantum computer goes via implementing the Pauli rotations $e^{i\varphi P_j}$ for some angle φ. A quantum circuit that implements a Pauli rotation unitary is shown in Fig. 1.2. The number of 2-qubit gates in this circuit construction is $2(|P_i| - 1)$, while there is only a single non-Clifford gate, the central single-qubit R_Z-rotation. Hence, in the setting where 2-qubit gates are the most costly to implement on hardware, the Pauli weight is a crucial metric to consider. However, in a fault-tolerant scenario the number of non-Clifford operations is assumed to dominate the overall cost, so the Pauli weight is much less relevant.

Lastly, we want to mention that the low-depth ternary trees (as shown in Fig. 1.6) are only optimal when considering the cost-function $\sum_{i=1}^{L} |P_i|$. Here the Pauli weights $|P_i|$ are all weighted equally, which is sensible for algorithms that implement the Pauli rotations equally often, such as deterministic Trotter methods (see section 1.6). However, for randomized methods the Pauli strings are sampled by weight, $|h_i|/\lambda$, and it could make sense to design ternary trees such the weighted cost function $\sum_{i=1}^{L} |h_i||P_i|$ is minimized.

1.4 Quantum phase estimation

So far we have seen how a key problem in quantum chemistry (finding the electronic ground state energy) is set up, and how it is mapped to objects a quantum computer can work with. In this section we address the final step in solving the problem; we will present a quantum algorithm that can compute the eigenvalues of a Hamiltonian: quantum phase estimation (QPE).

Estimating an unknown eigenphase of a unitary U with precision $\varepsilon > 0$ at minimal cost is an important problem in quantum information processing, and in particular in the branch of quantum metrology. Various QPE algorithms have been proposed to solve this problem optimally in different settings. Let's assume that we can implement the unitary $U = e^{-iH}$, then the problem of estimating the (lowest) eigenvalue of H naturally fits into this framework. QPE protocols are also given an initial state $|\phi\rangle$ as input. Denote the eigenstates and eigenvalues of H by $\{|\psi_k\rangle\}_k$ and $\{E_k\}_k$, respectively, then we can expand $|\phi\rangle = \sum_k a_k |\psi_k\rangle$. Upon ℓ repeated actions of U, the state evolves to $\sum_k a_k e^{-i\ell E_k} |\psi_k\rangle$, thereby imprinting the spectral information of H into the phases. Quantum phase estimation algorithms process the states $U^\ell |\phi\rangle$, both with quantum and classical operations, in order to extract the relevant spectral information. A natural cost metric for these protocols is the number of times N the unitary U needs to be implemented.

There are general lower bounds on how efficiently any protocol can estimate an eigenphase: Assuming an exact eigenstate $|\psi_k\rangle$ can be prepared, then the unitary needs to be implemented at least $N = \mathcal{O}(1/\varepsilon)$ times in order to estimate E_k with precision $\varepsilon > 0$. This limit is called the *Heisenberg limit*, and it can be reached by either N sequential applications in a single circuit, or by N parallel applications of the unitary on N entangled registers [26]. On the other hand, if the unitaries are used in parallel without entanglement, then $N = \mathcal{O}(1/\varepsilon^2)$ is needed (in the quantum metrology language, this is the *shot-noise limit*).

In the setting of sequential applications of U, one might wonder whether it is possible to implement U^N at a lower cost than the $\mathcal{O}(N)$ depth from naive repetition. For example, if U is diagonal, then it acts by multiplying computational basis states $|x\rangle$ by a phase $e^{i\varphi_x}$. Implementing the eigenphases of U^N is then as easy or difficult as implementing the eigenphases of U, because one only needs to adjust $\varphi_x \to N\varphi_x$. However, having U in diagonal form and knowing how to compute its eigenphases defeats the purpose of phase estimation in almost all scenarios. Unless H belongs to a very specific set of Hamiltonians [27, 28], the cost of simulating the time-evolution e^{-itH} for a time t cannot scale sublinearly in t [29, 30] . This lower limit is also known as the *no-fast-forwarding theorem*.

1.4.1 Fully coherent phase estimation

The original quantum phase estimation algorithm, described in [31] performs the time evolution and the extraction of the eigenvalue coherently in one circuit (see Fig. 1.7). It contains m additional ancilla qubits and uses controlled-U a total of $2^m - 1$ times to attain accuracy $\mathcal{O}(2^{-m})$. Since the unitaries are implemented sequentially, the depth too is $2^m - 1$. The Hadamard gates on the ancilla register create a uniform superposition over all computational basis states $x_0 \dots x_{m-1} \equiv x \in \{0, \dots, 2^m - 1\}$. We expand the initial state in terms of the eigenstates $|\phi\rangle = \sum_{k=0}^{2^n-1} a_k |\psi_k\rangle$ and we define angles $\theta_k = \frac{E_k}{2\pi}$ with $\theta_k \in [0, 1]$. After the action of the controlled unitaries we get

$$\begin{aligned} & \frac{1}{2^{m/2}} \sum_{k=0}^{2^n-1} a_k \sum_{x\in\{0,1\}^m} |x_0 \dots x_{m-1}\rangle |\psi_k\rangle \\ \rightarrow \quad & \frac{1}{2^{m/2}} \sum_{k=0}^{2^n-1} a_k \sum_{x\in\{0,1\}^m} e^{-i\sum_{j=0}^{m-1} 2^j x_j E_k} |x_0 \dots x_{m-1}\rangle |\psi_k\rangle \\ = \quad & \frac{1}{2^{m/2}} \sum_{k=0}^{2^n-1} a_k \sum_{x=0}^{2^m-1} e^{-i2\pi x\theta_k} |x\rangle |\psi_k\rangle \end{aligned}$$

Finally, the quantum Fourier transform $QFT = 2^{-m/2} \sum_{y,z=0}^{2^m-1} \exp\{\frac{i2\pi yz}{2^m}\} |y\rangle\langle z|$ applied on the ancillary register transforms this state into

$$\frac{1}{2^m} \sum_{k=0}^{2^n-1} a_k \sum_{x,y=0}^{2^m-1} e^{\frac{i2\pi x}{2^m}(y-2^m\theta_k)} |y\rangle |\psi_k\rangle$$

Assuming the binary expansion of θ_k stops at, or before, m bits, the only terms surviving are the ones with $y = 2^m\theta_k$ and we end up with

$$\sum_{k=0}^{2^n-1} a_k |\theta_k\rangle |\psi_k\rangle$$

Thus, by measuring the ancilla register we obtain the eigenvalue $E_k = 2\pi\theta_k$ with probability $|a_k|^2$, and we project the state of the system onto the corresponding eigenstate $|\psi_k\rangle$. If the phases do not have an exact m-bit description, the precision of the outcome is $\varepsilon = 2^{-m}$, and since the cost is $\mathcal{O}(2^m)$ this quantum phase estimation version attains the Heisenberg limit $\mathcal{O}(\varepsilon^{-1})$. In order to boost the probability of obtaining a specific eigenstate, such as the ground state, one should aim for creating initial states that already have significant overlap with the target eigenstate. We will discuss the overlap aspect in more detail in section 1.5.

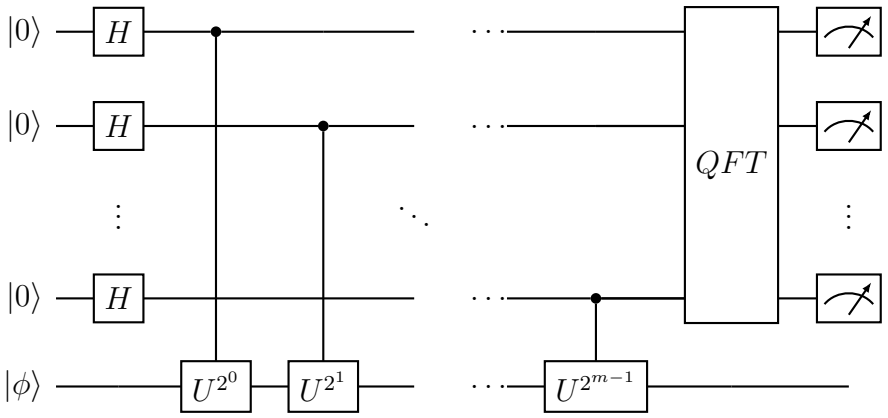


Figure 1.7: The fully coherent phase estimation algorithm [31, 32].

1.4.2 Hadamard test

The Hadamard test circuit is shown in Fig 1.8, it consists of the system register with initial state $|\phi\rangle$ and a single ancilla qubit. In contrast to the fully coherent phase estimation circuit it produces only a single bit of information per execution, which can be described by a binary random variable $\mathbf{Z}_\theta$. The state just before the measurement is

$$\frac{1}{2}\left(|0\rangle\left(I + e^{i\theta}U^\ell\right)|\phi\rangle + |1\rangle\left(I - e^{i\theta}U^\ell\right)|\phi\rangle\right)$$

The probability of measuring outcome $x \in \{1, -1\}$ is

$$\mathbb{P}[\mathbf{Z}_\theta = x] = \frac{1}{2}\left(1 + x\cos\theta\,\mathrm{Re}\langle\phi|U^\ell|\phi\rangle - x\sin\theta\,\mathrm{Im}\langle\phi|U^\ell|\phi\rangle\right)$$

and therefore the expectation value of $\mathbf{Z}_\theta$ is

$$\mathbb{E}[\mathbf{Z}_\theta] = \mathbb{P}[\mathbf{Z}_\theta = 1] - \mathbb{P}[\mathbf{Z}_\theta = -1] = \cos\theta\,\mathrm{Re}\langle\phi|U^\ell|\phi\rangle - \sin\theta\,\mathrm{Im}\langle\phi|U^\ell|\phi\rangle\,.$$

Combining results for angles $\theta \in \{0, -\pi/2\}$ yields

$$\mathbb{E}[\mathbf{Z}_0 + i\mathbf{Z}_{-\pi/2}] = \langle\phi|U^\ell|\phi\rangle = \sum_{k=0}^{2^n-1} |a_k|^2 e^{-i\ell E_k}.$$

There exist various phase estimation protocols that sample from this signal and perform classical post-processing in order to obtain information about the spectrum of H [33, 34, 35, 36, 37]. The common idea behind all variants is to perform a Fourier transformation from the time to the energy domain. Cost factors are the number of samples per circuit, and the maximal depth (*i.e.* largest ℓ). In

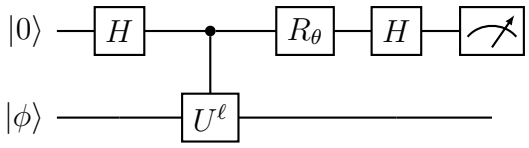


Figure 1.8: The Hadamard test circuit, $R_\theta = |0\rangle\langle 0| + e^{i\theta}\,|1\rangle\langle 1|$.

[33] the samples from the Hadamard test outcomes are used to approximate the cumulative distribution function (CDF)

$$C(x) = \sum_{k:E_k \geq x}^{2^n-1} |a_k|^2,$$

and a binary search-like algorithm is used to find the first jump of $C(x)$, thereby giving an estimate of the ground state energy. The method requires a maximal per-circuit cost of $\tilde{\mathcal{O}}(\varepsilon^{-1})$ and a total cost of $\tilde{\mathcal{O}}(\varepsilon^{-1}|a_0|^{-4})$, hence, it reaches the Heisenberg-limit. In [35] Hadamard test circuits with an alternating sequence of controlled unitaries and ancilla phase gates is used to compute ground state energies with a total cost of only $\tilde{\mathcal{O}}(\varepsilon^{-1}|a_0|^{-2})$, thereby improving the overlap dependence.

Moreover, several works based on sampling Hadamard test outcomes achieve reduced circuit depth, at the expense of an increase in total cost. For example, given a lower bound on the spectral gap, $\Delta = E_1 - E_0$, a maximal depth of $\tilde{\mathcal{O}}(\Delta^{-1})$ can be obtained, at a total cost of $\tilde{\mathcal{O}}(\varepsilon^{-2}|a_0|^{-2}\Delta)$ [36, 37]. Or, if the ground state overlap $|a_0|^2$ is larger than some constant $c > 0.5$ and a lower bound $c < \eta \leq |a_0|^2$ is known, then it is possible to reduce the maximal cost to $\mathcal{O}((1-\eta)\varepsilon^{-1})$, while the total cost increases to $\tilde{\mathcal{O}}((1-\eta)^{-1}\varepsilon^{-1})$ [38, 39]. Depth reduction approaches are potentially useful for "early fault-tolerant" quantum computers, *i.e.* devices that can run error-corrected circuits with a significant number of gates, but are still limited in their number of (logical) qubits and their maximal circuit depth. At that quantum hardware stage the reduction in circuit depth might be the only way to run phase estimation algorithms at all, and the quadratically increased total cost is still favourable compared to classical methods with exponential scaling.

In Fig 1.9 we show a quantum circuit that computes the same signal as the Hadamard test circuit, but without the control on the unitary U. For this to work $|0\rangle^{\otimes n}$ needs to be an eigenstate of U, and the corresponding eigenphase $e^{i\varphi^{(0)}}$ needs to be known. Under these assumptions the state before the measurement is

$$\frac{1}{2}\,|0\rangle\left(|0\rangle^{\otimes n} + e^{i(\theta-\varphi_0^\ell)}U_\phi^\dagger U^\ell\,|\phi\rangle\right) + \frac{1}{2}\,|1\rangle\left(|0\rangle^{\otimes n} - e^{-i(\theta-\varphi_0^\ell)}U_\phi^\dagger U^\ell\,|\phi\rangle\right).$$

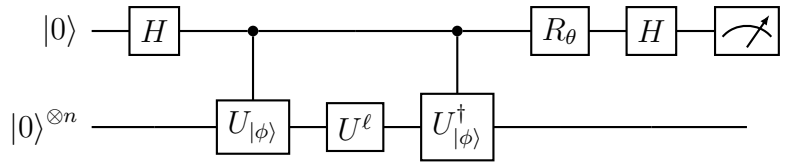


Figure 1.9: The Hadamard test with uncontrolled time evolution. The unitary $U_{|\phi\rangle}$ creates the initial state, $U_{|\phi\rangle}|0\rangle^{\otimes n} = |\phi\rangle$.

and the probability of measuring outcome $x \in \{1, -1\}$ is

$$\mathbb{P}[\mathbf{Z}_\theta = x] = \frac{1}{2}\left(1 + x\cos\left(\theta - \varphi_0^\ell\right)\mathrm{Re}\langle\phi|U^\ell|\phi\rangle - x\sin\left(\theta - \varphi_0^\ell\right)\mathrm{Im}\langle\phi|U^\ell|\phi\rangle\right)$$

which, up to the phase shift φ_0^ℓ, is identical to the outcome probability of the original Hadamard circuit. Shifting the control from U^ℓ to U_ϕ should eventually become advantageous for large ℓ. For many chemistry Hamiltonians it reasonable to assume that U_ϕ is significantly cheaper to implement than U^ℓ, because already simple states (for example the Hartree-Fock determinant) often have sufficient overlap with the ground state.

In chapter 2 we will describe a trick to effectively halve the required evolution time under U, if U is a Trotter circuit [40]. This trick requires a controlled implementation of U, which should be factored in when comparing the two Hadamard test circuits with and without controlled U. Additionally, if U is a product of Pauli rotations then its controlled version can be implemented at relatively low cost (see section 1.2).

Lastly, in Fig 1.10 we show a version of the Hadamard test circuit that allows for a trade-off between circuit width and depth. Assume $\ell = rj$ for positive integers r and j. The circuit starts with r initial states $|\phi\rangle$ in parallel, and instead of a single ancilla qubit it contains an r-qubit register, on which a GHZ-state, $(|0\rangle^{\otimes r} + |1\rangle^{\otimes r})/\sqrt{2}$ is created. The controlled unitaries U^j are each controlled on a different qubit of the GHZ-state, hence they can be performed in parallel. The final state before the measurement is

$$\frac{1}{\sqrt{2}}\left(U^\dagger_{\mathrm{GHZ}}|0\rangle^{\otimes r}|\phi\rangle^{\otimes r} + e^{i\theta}U^\dagger_{\mathrm{GHZ}}|1\rangle^{\otimes r}\left(U^j|\phi\rangle\right)^{\otimes r}\right).$$

The probability of measuring the uniform bitstrings with length r, $x^{(r)}$ for $x \in \{1, -1\}$, is

$$\mathbb{P}[\mathbf{Z}_\theta = x^{(r)}] = \frac{1}{2}\left(1 + x\cos\theta\,\mathrm{Re}\langle\phi|U^j|\phi\rangle^r - x\sin\theta\,\mathrm{Im}\langle U^j\rangle.\phi^r\right)$$

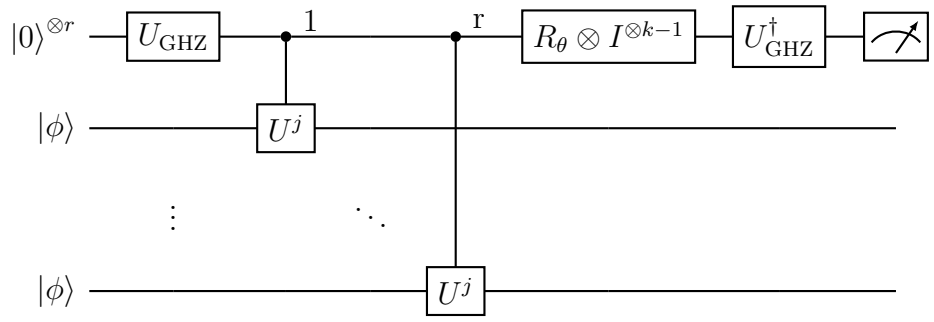


Figure 1.10: Trading space for time in the Hadamard test. U_{GHZ} is the unitary creating the GHZ state, $U_{\mathrm{GHZ}}|0\rangle^{\otimes r} = (|0\rangle^{\otimes r} + |1\rangle^{\otimes r})/\sqrt{2}$. The labels above the controls denote the index of the qubit in the GHZ state that is controlled on.

We can combine results for angles $\theta \in \{0, \pi/2\}$ gives

$$\mathbb{E}[\mathbf{Z}_0 + i\mathbf{Z}_{-\pi/2}] = \langle\phi|U^j|\phi\rangle^r ,$$

which contains the desired ground state signal $e^{i\ell E_0}$, but with reduced weight $|a_0|^{2r}$ instead of $|a_0|^2$. Thus, this approach incurs significant sampling overhead unless $\eta^r \geq (1-r(1-\eta))$ is still close to one, *i.e.* if $r \ll (1-\eta)^{-1}$. In the extreme case of $\eta = 1$, one can adjust width and depth of the circuit at no additional sampling cost, as long as the circuit footprint (width $\times$ depth) stays constant. Similar to the previous low-depth Hadamard test versions, this flexibility allows to adapt the quantum algorithm to hardware constraints. However, one should account for the fact that the initial state preparation cost increases by a factor of r.

1.5 Initial state preparation

In the previous section we saw how quantum phase estimation computes the ground state energy E_0 of a Hamiltonian H efficiently, if the initial state $|\phi\rangle$ has sufficient overlap with the ground state $|\psi_0\rangle$. Concretely, in terms of overlap $\eta \geq |\langle\psi_0|\phi\rangle|$, the runtime of quantum phase estimation is $\mathcal{O}(\eta^{-1})$. The expected overlap of an arbitrary vector $|\psi\rangle$ with the ground state is only $\mathcal{O}(2^{-n})$, where n is the number of qubits. To avoid the exponential overhead one might instead prepare a classical approximation to the ground state, in the hope that it has high ground state overlap. Note that, given the classical description of an approximate ground state, it is in general not possible to estimate η, as that requires access to $|\psi_0\rangle$. However, for up to 20 orbitals it is still possible to compute $|\psi_0\rangle$ by exact diagonalization. Furthermore, the density matrix renormalization group

(DMRG) method [41] can compute highly accurate ground state proxies even up to about 60 to 100 orbitals, depending on the complexity of the system [41, 42, 43]. Using results for the overlap of ground states for smaller system sizes gives insight into the quality of different types of initial states. We briefly mention three of these types.

Often the Hartree-Fock state, the simplest meaningful ground state approximation, yields surprisingly high overlap, exceeding $\eta \geq 0.9$. Ground states with that property are said to have a strong *single-reference* character. Especially closed-shell organic molecules close to their equilibrium configuration fall into this group. The implementation of the Hartree-Fock state is trivial and only amounts to preparing a computational basis state.

More challenging are systems that have multiple Slater determinants whose energies are similar, and whose contributions to the ground state are comparable. Such ground states have *multi-reference* character. Typical examples are molecular systems during bond-breaking processes, and systems with transition-metal atoms whose valence orbitals and electrons can lead to a complicated electronic structures [44]. A representative of this class is the iron-molybdenum cofactor (FeMoco), which is biologically relevant and difficult to study classically, and a benchmark application used for quantum algorithm resource estimates [3, 42, 40, 45, 46]. For these types of systems, more sophisticated initial states are needed to attain substantial overlap. Instead of only preparing the Hartree-Fock determinant, one might prepare a linear combination of Slater determinants, which is sometimes referred to as the 'Sum-of-Slaters' (SOS) approach [2, 47]. The cost of preparing such a state scales as $\mathcal{O}(D \log D)$, where D is the number of determinants in the sum.

Yet another option are matrix product states (MPS), which can efficiently represent a broad family of states by storing their coefficients in local tensors. Their expressiveness is determined by the dimension of the tensors, in particular by the *bond-dimension* χ. The cost of preparing an MPS in a quantum circuit scales as $\mathcal{O}(n\chi^2)$. DMRG is a heuristic method that computes MPS-based ground state approximations [48]. Empirically one observes that DMRG is reliable in converging towards the ground state as χ is increased [41, 49], however, it quickly becomes computationally very demanding as n and χ grow, therefore it is limited to small- to intermediate-sized systems.

We see that for SOS and MPS initial states the expressiveness (and hence the overlap) can be increased at the cost of a more expensive state preparation. Let's denote by $G_{\text{state-prep}}(\eta)$ the state preparation cost as a function of the overlap of the resulting initial state, and by G_{qpe} the remaining QPE cost in each circuit. We are interested in budgeting the optimal amount of cost to the state preparation part, such that the total cost is minimized. The number of circuit repetitions scales as $\mathcal{O}(\eta^{-1})$, and the cost per circuit is given by $G_{\text{state-prep}}(\eta) + G_{\text{qpe}}$, hence

the optimal value of η is determined by

$$\frac{d}{d\eta}\frac{1}{\eta}\left(G_{\text{state-prep}}(\eta)+G_{\text{qpe}}\right)=0.$$

The situation is of course different when we aim at minimizing the circuit depth, $G_{\text{state-prep}}(\eta)+G_{\text{qpe}}$. For standard QPE algorithms this suggests that only the simplest initial states should be prepared. However, as mentioned in the previous chapter, there are phase estimation variants whose maximal depth is upper bounded by $C_{\text{qpe}}(1-\eta)\varepsilon^{-1}$, for some constant C_{qpe}. With such a QPE method, the optimal value of η is determined by

$$\frac{d}{d\eta}\left(G_{\text{state-prep}}(\eta)+\frac{C_{\text{qpe}}(1-\eta)}{\varepsilon}\right)=0.$$

1.6 Realising unitaries from Hamiltonians

So far we assumed that we perform phase estimation on the perfect time evolution unitary $U=e^{-itH}$. In practice we usually work with approximations to e^{-iHt} or different unitaries based on H altogether. We now present three important constructions of unitary operators for a given Hamiltonian:

1.6.1 Deterministic product formulas

Let H be given as a sum of elementary terms, $H=\sum_{j=1}^{L}H_j$, such that the time evolution unitaries e^{-itH_j} of each term can be implemented exactly. A typical example is a linear combination of Pauli strings, $H=\sum_{j=1}^{L}h_jP_j$. The time evolution under H can then be approximated via the (first-order) Trotter approximation

$$e^{-itH}\approx\prod_{j=1}^{L}e^{-itH_j}\equiv S_1(t).$$

The approximation error is bounded in operator-norm by $\|e^{-itH}-S_1(t)\|\leq Ct^2$, where C is a constant depending on H. By repeating the Trotter unitary for smaller step sizes $\delta=t/m$, the error can be reduced to

$$\left\|e^{-itH}-S_1(\delta)^m\right\|\leq Cn\delta^2=\frac{Ct^2}{m},$$

so with $m=\mathcal{O}(t^2\varepsilon^{-1})$ the error is smaller than ε. For the purpose of phase estimation we can directly consider $S_1(\delta)$ as the unitary we are performing phase

estimation on. Since $S_1(\delta)$ approximates $e^{-i\delta H}$ and not e^{-iH}, the phase estimation precision must be rescaled, $\varepsilon \to \varepsilon\delta$.

Higher order Trotter formulas combine several first-order Trotter sequences S_1 in a nested fashion [50]

$$\begin{aligned} S_2(t) &= S_1(t/2)S_1^\dagger(-t/2) \\ S_{2p}(t) &= S_{2(p-1)}^2(u_p t)S_{2(p-1)}((1-4u_p)t)S_{2(p-1)}^2(u_p t) \end{aligned}$$

where $u_p = 1/(4 - 4^{1/(2p-1)})$. Their error scaling improves to $\|e^{-itH} - S_{2p}(t)\| \leq Ct^{1+1/2p}$, however at the cost of an overhead factor $2 \cdot 5^{p-1}$ in the constant C.

The constant C can be rigorously upper bounded in terms of nested commutators of the Hamiltonian terms, the upper bounds for first and second order are given by [51]

$$\begin{aligned} \text{1st order:} \qquad C \leq & \frac{t^2}{2} \sum_{i=1}^{L} \left\| \left[\sum_{j=i+1}^{L} H_j, H_i \right] \right\| \\ \text{2nd order:} \qquad C \leq & \frac{t^3}{12} \sum_{i=1}^{L} \left(\left\| \left[\sum_{k=i+1}^{L} H_k \left[\sum_{j=i+1}^{L} H_j, H_i \right] \right] \right\| \right. \\ & \left. + \frac{1}{2} \left\| \left[H_i, \left[H_i, \sum_{j=i+1}^{L} H_j \right] \right] \right\| \right). \end{aligned}$$

It was shown that these general bounds are tight [51], *i.e.* there exist Hamiltonians for which the Trotter error saturates this bound. However, for molecular electronic structure Hamiltonians it has been observed that the bounds are very loose [3, 40, 52, 53, 54]. Additionally, challenging chemistry Hamiltonians that are interesting quantum computing targets easily have $L = 10^6$ and more terms, making the evaluation of such bounds difficult. In chapter 2 and chapter 3 we take a heuristic approach and extrapolate exactly computed Trotter errors to larger system sizes.

1.6.2 Randomized product formulas

Let H be given as a decomposition $H = \sum_{i=1}^{L} h_i H_i$, such that the terms are normalized, $\|H_i\|_\infty \leq 1$, and the sign of h_i is absorbed by H_i. Furthermore, define $\lambda = \sum_{i=1}^{L} h_i$ and $p_i = h_i/\lambda$. The qDRIFT method [55] draws r terms H_i according to the probability distribution defined by the p_i, resulting in a list of indices $\{i_1, i_2, \ldots, i_r\}$. Define $\tau = t\lambda/r$, then qDRIFT approximates the time

evolution e^{-iHt} by the random unitary

$$U(\tau, \{i_j\}) = \prod_{j=1}^{r} \exp\big(-i\tau H_{i_j}\big).$$

The evolution of a state (density matrix) ρ for a single sample of this stochastic operation is described by the quantum channel

$$\mathcal{E}_\tau[\rho] = \sum_{j=1}^{L} p_j e^{-iH_j\tau} \rho e^{iH_j\tau}.$$

The error with respect to the exact time evolution $\mathcal{U}[\rho] = e^{-itH}\rho e^{itH}$ can be bounded in terms of the diamond distance [55]:

$$\|(\mathcal{E}_\tau)^r - \mathcal{U}\|_\diamond = \max_\rho \|((\mathcal{E}_\tau)^r \otimes I_n)[\rho] - (\mathcal{U} \otimes I_n)[\rho]\|_1 \leq \frac{2\lambda^2 t^2}{N},$$

where $\|\cdot\|_1$ denotes the trace norm (sum of singular values). The trace norm upper bounds differences in expectation values; given an operator O and states ρ, σ that are ε-close in trace norm, then

$$|\mathrm{Tr}[O\rho] - \mathrm{Tr}[O\sigma]| \leq \mathrm{Tr}[|O(\rho - \sigma)|] = \|O(\rho - \sigma)\|_1 \leq \varepsilon \|O\|.$$

In the last line Hölder's inequality was used, and $\|O\|$ is the usual operator norm (largest singular value of O). If O is a POVM element then $\|O\| \leq 1$. Consequently, the bound on the diamond distance $\|(\mathcal{E}_\tau)^r - \mathcal{U}\|_\diamond$ allows us to bound differences in measurement statistics.

The application of the Trotter method and qDRIFT to QPE are discussed in detail in chapter 2. An important difference between the two methods is that Trotter explicitly depends on the number of terms L, while qDRIFT depends on the cumulative weight λ. Hence, if H consists of a few high weight terms then Trotter should be better, while for many small weight terms qDRIFT should have the edge. This observation motivates constructing a method that combines the strength of both, and this will be the main contribution of the work presented in chapter 2.

1.6.3 Quantum walk operator

Let H be an n-qubit Hamiltonian given as a linear combination of unitaries, $H = \sum_{j=1}^{L} \alpha_j U_j$, $\alpha_j > 0$, and define $\lambda = \sum_{j=1}^{L} \alpha_j$. Consider two quantum

registers containing $d = \lceil \log_2(L) \rceil$ and n qubits, respectively, and define the operations PREPARE and SELECT by

$$\text{PREPARE} \, |0\rangle^{\otimes d} = \sum_{j=0}^{L-1} \sqrt{\frac{\alpha_j}{\lambda}} \, |j\rangle \equiv |\mathcal{L}\rangle$$

$$\text{SELECT} = \sum_{j=0}^{L-1} |j\rangle\langle j| \otimes U_j.$$

The quantum walk operator is defined as

$$W = (2 \, |\mathcal{L}\rangle\langle\mathcal{L}| \otimes I - I)\text{SELECT}$$

and it has eigenvalues $e^{\pm i \arccos(E_k/\lambda)}$ [56, 57]. This method of encoding the spectrum of H into an operator is known as *qubitization*. It is a key ingredient for an asymptotically optimal way of implementing the time evolution unitary e^{-iHt}, with cost $\mathcal{O}(t + \log(\varepsilon^{-1}))$ [58]. However, it was soon realised that one could perform phase estimation directly on the quantum walk operator W [56, 57]. Running quantum phase estimation on W yields the values $\arccos(E_k/\lambda)$, from which the eigenvalues E_k can be computed. In contrast to the Trotter unitary, no algorithmic errors need to be controlled. However, the implementation of PREPARE and SELECT requires complex multi-qubit operations, and often a non-negligible number of ancilla qubits. The quantum walk operator is part of a framework known as Quantum Signal Processing [59]. With the Quantum Signal Processing machinery a broad class of unitaries U based on H can be realised. Many modern quantum algorithms for energy estimation of electronic structure Hamiltonians are based on this method [46, 45, 60].

1.7 Complexity theoretic considerations

The main question we want to address in this section is "Which ground state problems in chemistry are classically difficult yet quantumly tractable?". Computing the ground state energy of the electronic structure Hamiltonian appears to be a very specific problem, and one might expect that insights from the perspective of (quantum) complexity theory are limited. However, it turns out that the electronic structure problem is closely related to a family of problems that lies at the heart of our understanding of the computational power of quantum computers, the LOCAL-HAMILTONIAN problem. In the following, we will introduce key concepts from classical and quantum complexity theory, and discuss results that are relevant to the hardness of solving molecular problems using a quantum computer. The conclusions of this section are not constructive, in the

sense that they don't yield practical methods to solve a problem. Instead, the complexity theoretic results are useful to set boundaries on our expectations of what quantum computers can and cannot solve, and when they have an edge over classical computers. We will see that the question of quantum advantage for chemistry is quite subtle, as it involves an interplay of several parameters that could make the problem easy or hard. Fig. 1.11 gives an overview of the complexity classes that are discussed in this section.

Complexity theory studies how difficult it is to solve computational problems, often by relating problems to each other to reach statements such as 'problem A is at least as hard as problem B'. Problems can be classified into different classes depending on how many computational resources (*e.g.* time, memory) are needed to solve them. The computational resources are defined in a sufficiently general way such that all types of classical computing are equivalent, and the type of physical hardware is not important.[7] It is widely believed that quantum computers are fundamentally more powerful than classical machines. This raises questions: How much more powerful are quantum computers than their classical counterparts? Which are the problems that not even a quantum computer can solve efficiently? We will discuss general results and explain how the ground state energy problem fits into the picture.

A computational problem refers to a family of problem instances, each of which is encoded in a bit string of length n, providing a natural scaling parameter. For example, for graph problems n may be the related to number of vertices in a graph, for problems about numbers n is the number of bits needed to represent a number, and for questions about a family of Hamiltonians n might be related to the number of qubits a Hamiltonian acts on. We denote the problem family by a set $L \subset \{0,1\}^*$, often called a *language*. One often considers *decision* problems, for which L is partitioned into yes and no instances, $L = L_{yes} \cup L_{no}$. Given $x \in L$, the task is to determine whether $x \in L_{yes}$ or $x \in L_{no}$. Computational complexity theory focuses exclusively on the *asymptotic* scaling ($n \to \infty$) of computational resources needed to decide a language. *Complexity classes* are sets of languages that have some scaling behaviour in common.

A fundamental classical complexity class is P, containing the problems that can be decided on a classical computer in poly(n) time. Closely related is the complexity class NP, containing those problems for which there is a short proof of $x \in L_{yes}$ for every yes instance. The proof (often referred to as a *witness*) needs to be verifiable by a classical computer in poly(n) time. Clearly, $\mathsf{P} \subset \mathsf{NP}$, because for problems in P no additional proof is needed. NP contains numerous optimisation-related problems to which no efficient algorithms are known. A prototypical

[7] Two types of computers are equivalent if they can simulate each other, up to some overhead that must not scale more than polynomially.

problem in NP is 3-SAT: Given a boolean formula in n boolean variables x_i with 3 variables per clause, does a satisfying assignment exist? Checking whether an assignment evaluates to true can be done in poly(n) time. However, to find a satisfying assignment, one might need to check 2^n different assignments. 3-SAT is actually NP-complete, meaning that it is at least as hard as all other problems in NP, but is itself still in NP. If it can be decided in poly(n) time, then so can all other problems in NP. The difficulty of 3-SAT reflects a general intuition: being able to recognise a solution to a problem is often much easier than finding a solution. Still, a major open question is whether $\mathsf{P} \stackrel{?}{=} \mathsf{NP}$, with strong indications that NP is strictly larger than P.

In a probabilistic computational model one allows the classical devices to make random choices, which leads to the probabilistic extensions of P and NP, namely BPP and MA. A family of problems is said to be in BPP if there exists such a probabilistic classical device which in polynomial time, and with probability at least 2/3, outputs the correct solution. MA contains those problems for which valid (invalid) witnesses of a yes instance can be verified (rejected) in polynomial time by a probabilistic classical computer with probability at least 2/3.

The quantum complexity classes BQP and QMA are the natural extensions of BPP and MA to the quantum computing setting, they were first defined in [61] and [62]. We give a definition of BQP, which can be viewed as the class of problems that can be solved efficiently by quantum computers.

Definition 1 (BQP)**.** *A promise-problem $L = L_{yes} \cup L_{no} \subset \{0,1\}^*$ is in BQP if there exists a family of efficiently constructable polynomial-time quantum circuits $\{Q_n\}_{n\in\mathbb{N}}$ such that for any $x \in \{0,1\}^n$:*

- $x \in L_{yes} \implies Q_n$ *accepts* $|x\rangle$ *with probability* $\geq 2/3$
- $x \in L_{no} \implies Q_n$ *accepts* $|x\rangle$ *with probability* $\leq 1/3$

The exact values 1/3 and 2/3 do not matter so much as they can be amplified to values close to 0 and 1 by repeating poly(n) times. It was shown that classical circuits can be simulated with quantum circuits, using only polynomial overhead, and as a consequence, P$\subset$BQP. A famous problem that is not known to be in BPP but is in BQP$\cap$NP is FACTORING, due to Shor's algorithm [63]. FACTORING is likely not as hard as the ground state energy problems we discuss below, but its classical verifiability means that it is easy to check whether a quantum hardware implementation of Shor's algorithm ran correctly or not.

The class QMA describes the set of problems whose yes instances can be verified efficiently with high probability using a quantum computer. Analogous to the classical classes P and NP, it is widely believed that BQP is strictly smaller than QMA. Consequently, showing that a problem is QMA-complete suggests that

not even a quantum computer can solve it efficiently in general. The formal definition of QMA is [64]:

Definition 2 (QMA)**.** *A promise-problem $L = L_{yes} \cup L_{no} \subset \{0,1\}^*$ is in QMA if there exists a family of efficiently constructable polynomial-time quantum circuits $\{Q_n\}_{n\in\mathbb{N}}$ such that for any $x \in \{0,1\}^n$:*

- *$x \in L_{yes} \implies$ there exists a $p(n)$-qubit witness state $|\psi_x\rangle$ such that Q_n accepts $(|x\rangle, |\psi_x\rangle)$ with probability $\geq 2/3$.*
- *$x \in L_{no} \implies$ for every $\mathrm{poly}(n)$-qubit witness state $|\psi\rangle$, Q_n accepts $(|x\rangle, |\psi\rangle)$ with probability $\leq 1/3$.*

The inclusion MA $\subset$ QMA follows from the fact that the quantum setup can simulate the classical setup. A probabilistic classical computer can be simulated by a quantum computer, and a classical witness can be represented by a computational basis state. Less obvious is the inclusion QMA $\subset$ PSPACE [65], where PSPACE is the set of problems decided by a classical computer with $\mathrm{poly}(n)$ space.

A major branch of quantum complexity is the field of Hamiltonian complexity theory. At its center lies the k-Local-Hamiltonian (lh) problem [65, 66]:

Definition 3 (k-local hamiltonian)**.** *Given a Hamiltonian $H = \sum_{i=1}^{r} H_i$ with $\|H_i\| \leq 1$, and each H_i acting on at most k qubits. Let E_0 be the ground state energy of H, then for $b > a$ decide whether:*

- *(yes case) $E_0 < a$*
- *(no case) $E_0 > b$,*

under the promise that one of them is the case.[8]

Kitaev showed that this problem is QMA-complete for $k = 5$ and $b - a \geq 1/\mathrm{poly}(n)$ [65]. This problem is in QMA because the witness $|\psi\rangle$ can be chosen to be the ground state of H, from which $\langle\psi|H|\psi\rangle$ can be computed to precision $b-a$ in $1/\mathrm{poly}(b-a) = \mathrm{poly}(n)$ time. For QMA-hardness it needs to be shown that the verification procedure via $\mathrm{poly}(n)$-sized quantum circuits Q_n on input $|\psi\rangle$ can be encoded in a ground state of a local Hamiltonian. In short, this is achieved via the construction of a history state, a state that (in superposition) records the correct sequence of changes in the quantum circuit, and by constructing a local Hamiltonian that penalizes changes in states that do not follow the correct evolution of Q_n on input $|\psi\rangle$. The QMA-completeness of k-Local-Hamiltonian

[8] The problem is formulated as a decision problem, but it can be used to estimate E_0 using binary search.

proves a deep connection between quantum many-body problems and complexity theory. It says that estimating the ground state energy of a quantum many-body system (a central problem in quantum chemistry and condensed matter physics) is at least as hard as any problem that a quantum computer can verify. Moreover, it is unlikely that this problem can be solved with a quantum computer in general.

Many variations of the k-local-hamiltonian problem have been studied, revealing connections to even more classes than QMA. In the following we will discuss some of the variants of k-local-hamiltonian that are especially relevant to the task of computing molecular ground state energies.

Interactions

Definition 3 includes a very broad class of Hamiltonians, many of which might be hard to physically realize. Does the problem become easier with physically motivated restrictions on H? In many cases the answer turns out to be "no". For example, it was shown that k-local hamiltonian is QMA-complete whenever $k \geq 2$ [67], even when the interactions are 2-local on a 2D lattice [68]. Apart from geometric constraints, restrictions on the types of interactions H_i in the Hamiltonian were considered. It was shown that various important spin-lattice models are QMA-complete, among them the 2D Bose-Hubbard model [69], the 2D Fermi-Hubbard model [70, 71], the 2D Heisenberg model [70] and a generalised transverse field Ising model [72]. Relevant to the ground state problem in quantum chemistry is the electronic-structure problem, *i.e.* lh for Hamiltonians describing interacting electrons in an external potential, expressed in second quantization in a specific orbital basis. By a reduction to the Fermi-Hubbard model, it has been shown that also this problem is QMA-complete [71]. Consequently, assuming QMA$\neq$BQP (which is strongly believed to be the case), the central problem studied in this thesis cannot be solved efficiently on all problem instances without further assumptions.

Spectral gap

An important quantity in the study of quantum many-body systems is the spectral gap, the energy difference between first excited and ground state, $E_1 - E_0$. Not only does it significantly influence the low-temperature behaviour of materials and molecules, it also has an impact on the difficulty of computing ground state energies. For example, for 1D systems with nearest-neighbour interactions the promise of a constant gap makes the problem classically tractable [73, 74]. In [75] the hardness for lh was investigated for the setting of an inverse exponential promise gap $b - a \geq 1/\exp(n)$. It turns out that the hardness of this problem, lh$(1/\exp)$, is amplified to be PSPACE-complete. Interestingly, if additionally

a lower bound on the spectral gap of only $E_1 - E_0 \geq 1/\exp(n)$ is promised, then LH($1/\exp$) is not PSPACE-complete anymore [75]. Furthermore, it is known that QMA-hardness of LH with a constant lower bound on the spectral gap cannot be proven via the standard technique of history states [76]. Taken together, these results show that an asymptotic lower bound on the spectral gap can be a powerful assumption, and it might also lower the hardness of the ELECTRONIC-STRUCTURE problem. In fact, chemical systems are often found to be gapped [77]. However, since the interactions in molecules take place in 3D and are not nearest neighbour, previous results on spectral gap promises do not apply to the molecular problem setting. Thus, the consequences of a lower bound on the spectral gap for the ELECTRONIC-STRUCTURE problem have not been resolved.

Guiding state

As discussed in the previous section, a key ingredient to the quantum phase estimation approach to computing ground state energies is the initial state $|\phi\rangle$. If the overlap η between ground state and initial state is at least $\eta = 1/\operatorname{poly}(n)$, then QPE can calculate the ground state energy in polynomial time, and the ground state energy problem is in BQP. Such an initial state is called a *guiding state* in the quantum complexity literature, and the variant of LOCAL-HAMILTONIAN with this additional input is the GUIDED LOCAL-HAMILTONIAN (GLH) problem. Clearly, it cannot be guaranteed that for all ELECTRONIC-STRUCTURE problems a guiding state with $\eta \geq 1/\operatorname{poly}$ can be found, as otherwise BQP=QMA. The expected overlap of a random state is only 2^{-n} due to the dimension of the Hilbert space. Nonetheless, it is observed that for the vast majority of molecular systems, classical heuristics such as Hartree-Fock or DMRG result in significant overlap, as discussed in section 1.5. These problem instances appear to be good candidates for a quantum advantage. However, it is not obvious whether classical computers could not also take advantage of the existence of efficiently findable guiding states. In [78, 79] the hardness of GLH was investigated for different regimes of precision. If only constant *relative* precision is demanded (with respect to $\|H\|$), then for any constant overlap $\eta \geq 0$, the problem becomes classically tractable, GLH(CONST)$\in$ BPP [78]. On the other hand, for inverse polynomial precision (with respect to $\|H\|$), which is the relevant setting when calculating ground state energies with chemical accuracy, the problem is BQP-hard for any $\eta \in (0, 1 - 1/\operatorname{poly}(n))$.[9] That is, even for guiding states that are very good approximations to the ground state, the problem remains intractable classically. Thus, these results suggest that a potential quantum advantage comes from achieving higher precision.

[9]These results assume classical sampling access to the guiding state, which for example applies to sparse states and matrix-product states.

Related to the study of guiding states, [80] investigated the hardness of LH under the additional assumption that the ground state is *succinct*, more concretely assuming that a classical poly-sized circuit exists that can compute amplitudes of the ground state. This notion of SUCCINCT states encompasses Matrix-Product States (MPS), fermionic Gaussian states and similar representations [80]. It was shown that this variant of LH, denoted by LH(SUCCINCT), is MA-complete, so its hardness drops from being quantumly hard to only being classically hard. This result motivated the following question: Do *strong* guiding states exist, which combine succinctness and a condition on the overlap, such that the GUIDED LOCAL-HAMILTONIAN problem becomes classically tractable, even in the regime of inverse-polynomial precision?[80]

It should be stressed that the complexity theoretic classifications discussed in this section concern the *worst-case* scaling for problems. There are problems that are in NP and therefore formally intractable, but heuristic methods are very effective at finding good approximate solutions to most practically relevant problem instances. A good example of this is the task of finding the Hartree-Fock energy, which for the electronic structure Hamiltonian was shown to be NP-complete [71]. Nonetheless, for almost all chemical systems this problem is solved routinely, with relatively low cost, using classical heuristics. Similarly, the QMA-hardness of ELECTRONIC-STRUCTURE does not imply that quantum advantage for chemistry problems is ruled out. Instead, it points to the importance of problem specific properties, such as lower bounds on the spectral gap, the existence of guiding states and the structure of interactions in the Hamiltonian. Such characteristics are often based on a chemical understanding of the problem at hand, and they are necessary to make complexity theoretic statements precise. Further closing the gap between complexity theoretic generality and the specifics of quantum chemistry is still an unresolved challenge. Progress in this direction is necessary to delineate the region of chemistry problems for which quantum advantage can be expected. BQP-completeness of the GUIDED LOCAL-HAMILTONIAN problem is currently the strongest indication for a quantum advantage in chemistry applications.

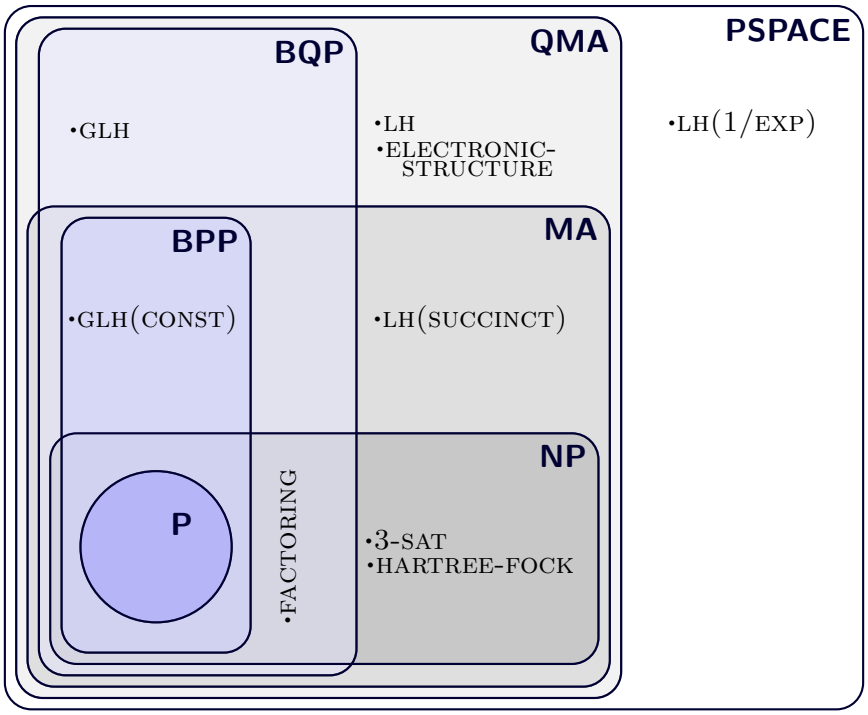


Figure 1.11: Complexity classes and problems mentioned in the main text. None of the inclusions are strict as not even $\mathsf{P} \overset{?}{\neq} \mathsf{PSPACE}$ has been resolved.

Chapter 2

Phase estimation with partially randomized time evolution

In the introduction we have seen how deterministic and randomized Hamiltonian simulation methods can be used to compute the ground state energy of molecular electronic structure Hamiltonians. The two approaches shine in different settings, depending on the distribution of weights in the Hamiltonian. This chapter investigates how both methods can be combined in a complementary way. We will see that for Hamiltonians emerging from molecules the distribution of weights has the right characteristics to benefit from such a treatment.

Key technical contributions of the manuscript are: (i) a partially randomized product formula with a novel analysis in the context of the Hadamard test circuit (ii) improved representations of chemistry Hamiltonians (iii) a reduced QPE overhead for randomized methods (iv) lower depth QPE circuits for randomized methods in the high overlap regime, at no increase in total cost (v) concrete resource estimates for chemical benchmark systems, showing large cost reductions compared to previous estimates based on product formulas.

Apart from section 2.10, this chapter is a reproduction of arxiv:2503:05647. In section 2.10 we present a few results that did not make it into the first version of the manuscript.

Phase estimation with partially randomized time evolution

Abstract:
Quantum phase estimation combined with Hamiltonian simulation is the most promising algorithmic framework to computing ground state energies on quantum computers. Its main computational overhead derives from the Hamiltonian simulation subroutine. In this paper we use randomization to speed up product formulas, one of the standard approaches to Hamiltonian simulation. We propose new partially randomized Hamiltonian simulation methods in which some terms are kept deterministically and others are randomly sampled. We perform a detailed resource estimate for single-ancilla phase estimation using partially randomized product formulas for benchmark systems in quantum chemistry and obtain orders-of-magnitude improvements compared to other simulations based on product formulas. When applied to the hydrogen chain, we have numerical evidence that our methods exhibit asymptotic scaling with the system size that is competitive with the best known qubitization approaches.

Authors:
Jakob Günther, Freek Witteveen, Alexander Schmidhuber, Marek Miller, Matthias Christandl and Aram Harrow

Reference: arXiv:2503.05647
Date: March 2025

2.1 Introduction

Among the most promising applications of quantum computing is the calculation of ground state energies in quantum many-body physics and quantum chemistry. While in general it is hard for a quantum computer to estimate the ground state energy, quantum phase estimation (QPE) provides an efficient algorithm when a state with significant overlap to the ground state is given. The complexity of the

resulting algorithms for chemistry depends largely on the complexity of simulating the Hamiltonian time evolution as a quantum circuit, as QPE requires time simulation of time $\mathcal{O}(\varepsilon^{-1})$ to compute the energy to accuracy ε. The dominant cost of performing QPE is then given by the cost of Hamiltonian evolution and its scaling with time and system size. State-of-the-art resource estimates for realistic examples from chemistry show that the computational cost of performing QPE is still formidable [81, 45, 82, 46]. This means that for realizing useful simulations of quantum chemistry, further reductions in the circuit size required to perform Hamiltonian simulation are necessary. This is particularly relevant for devices with a limited number of logical qubits and limited circuit depth. See [83] for a general discussion of QPE for quantum chemistry.

Deterministic product formulas, also known as Trotterization, have the advantage of being conceptually simple and to require little or no ancilla qubits for their implementation as a quantum circuit. In many situations, their performance can be better than what can be expected from rigorous bounds [52, 81] or can even be proven to have good scaling, e.g. in systems with spatially local interactions [51, 84, 85]. There also exist powerful post-Trotter Hamiltonian simulation methods [59, 58]; while they have better scaling, methods based on product formulas are still of great interest due to the small number of ancilla qubits required and their good performance for Hamiltonians of interest [81].

This work is motivated by quantum chemical Hamiltonians. For such a Hamiltonian, defined on an active space of N spatial orbitals (corresponding to $2N$ qubits in second quantization), the number of terms in the Hamiltonian scales as $L = \mathcal{O}(N^4)$. Trotter product formulas have a computational cost which scales with L. While this leads to polynomial scaling in N, the resulting circuits rapidly grow impractically large.

An alternative is provided by *randomized product formulas*, in particular qDRIFT [55]. Here, the number of gates no longer depends on L, but rather on λ, the sum of the interaction strengths in the Hamiltonian. If one wants to simulation Hamiltonian evolution for time t, this requires $\mathcal{O}(\lambda^2 t^2)$ gates. So, while this gets rid of the dependence on L, the downside is that the scaling is quadratic in the time (and λ). In particular, since QPE with target accuracy ε requires Hamiltonian simulation for times up to $\mathcal{O}(\varepsilon^{-1})$, using randomized product formulas leads to phase estimation costs scaling with ε^{-2} rather than the optimal Heisenberg-limited scaling of ε^{-1}. Prior work uses an analysis which suggests that combining QPE with qDRIFT leads to an unfavorable scaling of the total cost with the accuracy of the computation [55]. In particular, a naive analysis leads to a scaling of order ε^{-4} or an undesirable dependence on the probability of error [46]. Using a different randomization scheme, however, as well as a different phase estimation scheme, one can restore the scaling to ε^{-2} [86].

A natural generalization of randomized product formulas is to consider *par-*

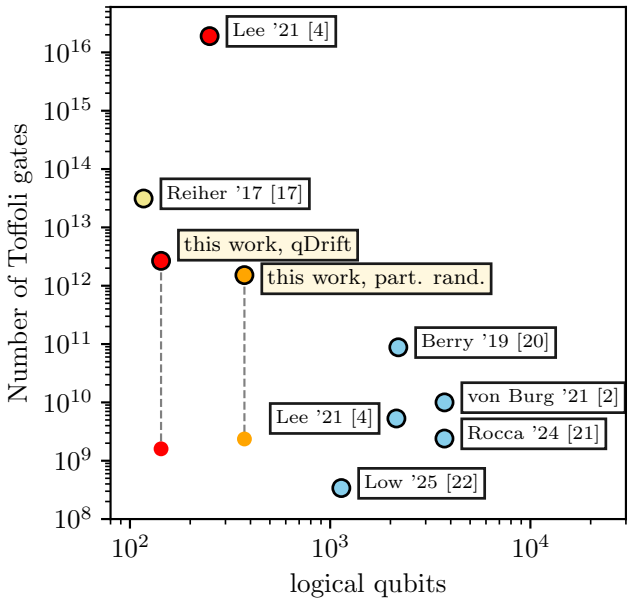


Figure 2.1: Total Toffoli gate count and logical qubit estimates for ground state energy estimation for FeMoco using the active space of [40], with target error $\varepsilon = 0.0016$ and assuming an initial state equal to the ground state. The blue-colored data points use qubitization [60, 45, 46, 90, 91], the yellow data point is the most optimistic estimate from [40] using a deterministic product formula, the red data points use randomized product formulas from [46] and this work, the orange data point uses partial randomization. We also show the Toffoli depth (i.e. the maximum number of Toffoli gates per circuit), assuming a high ground state overlap state and choosing $\xi = 0.1$, using (partially) randomized product formulas. The vertical dashed lines connect the Toffoli depth to the total Toffoli count, to illustrate how multiple circuit runs are used.

tially randomized product formulas [87, 88, 89], which are effective if there is a relatively small number of dominant terms in the Hamiltonian, and a long tail of small terms (which are nevertheless too large to be ignored altogether). Such behavior is typical for quantum chemistry Hamiltonians; see Section 2.6 and Section 2.9.3 for a detailed discussion.

2.1.1 Results and organization

We start by setting up notation for the relevant class of Hamiltonians and single-ancilla phase estimation in Section 2.2 and introduce background material on the Trotter formalism in Section 2.3. We discuss randomized product formulas in Section 2.4. Our first contribution is an improved analysis for the cost of randomized product formulas for single-ancilla QPE, achieving reduced cost compared to previous estimates [46, 86, 55] and reduced maximal circuit depth. This result

is stated as Theorem 2.4.1. In Fig. 2.1 we show the resulting resource estimates for the challenge benchmark problem FeMoco [40] and compare with previous work. We see multiple order of magnitude improvements in the performance of the randomized product formulas. Our improvement compared to [46] by more than two orders of magnitude breaks down as follows. We find a value of λ which is a factor 5 smaller, reducing the cost by a factor 25. The overhead of the phase estimation routine is reduced by a factor of 38 (while improving accuracy guarantees). Indeed, [46] finds a requirement of approximately $305\lambda^2\varepsilon^{-2}$ Pauli rotations, while we require approximately $8\lambda^2\varepsilon^{-2}$ Pauli rotations. Finally, as explained in the Section 2.9.5, we give a compilation from Pauli rotations into Toffoli gates using a factor 10 fewer gates by combining techniques from [92, 93, 46, 86].

Fig. 2.1 assumes an exact eigenstate, and aims to minimize total resources. More realistically, we should assume we start with a state with (high) ground state overlap. Our methods can accommodate such states, but using an exact eigenstate facilitates comparison with the previous methods. If one can prepare a state with high ground state overlap, using robust phase estimation in combination with randomized product formulas has the effect of using multiple circuits of smaller size. This can be useful for early fault-tolerant devices and paralellization. To make this concrete, let us again consider the example of FeMoco. If one indeed can prepare exact eigenstates, or states with suffixiently high ground state overlap, one can trade the gate count per circuit further against using more circuits. If we choose the trade-off parameter $\xi = 0.1$ in Theorem 2.4.1, we need 4741 circuits, with a maximum of 6.4×10^8 Pauli rotations per circuit (as also indicated in Fig. 2.1). This choice of ξ is also compatible with a state with ground state overlap $\eta > 0.9$, and it has been argued that for a larger active space of FeMoco achieving a squared overlap of appproximately 0.95 is realistic (but the cost of preparing this state is significant) [94]. As is also clear from Fig. 2.1, the partially randomized method (which scales better with the precision ε) benefits less from this reduced circuit depth.

The main contribution of this work is the study of partially randomized product formulas. In Section 2.5 we propose a modification of the scheme of Hagan and Wiebe [89]. We give a simple error analysis for our scheme when used in single-ancilla QPE, improving on the bounds that would be obtained by analyzing the product formula and the QPE separately. As an example of a significant cost improvement from partially randomized product formulas, we show in Fig. 2.2 cost estimates for the hydrogen chain with a minimal basis (a standard benchmark system to explore the scaling of the thermodynamic limit). We observe an improvement of more than an order of magnitude for partial randomization compared to fully randomized product formulas, which in turn outperform deterministic product formulas. Additionally, when varying both the required accuracy ε and the number of hydrogen atoms N, for the fully randomized method we

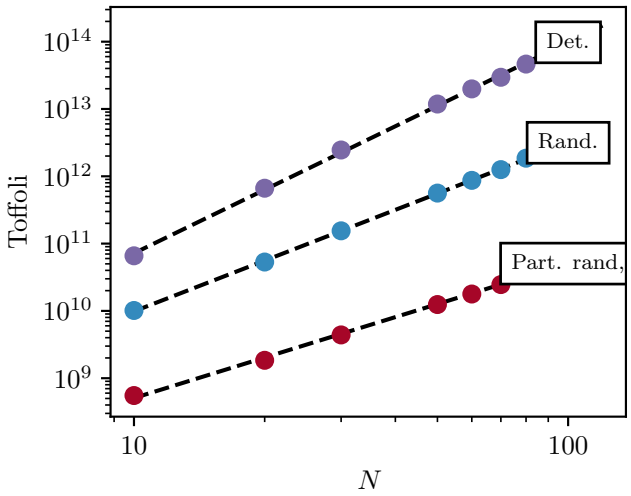


Figure 2.2: Gate counts for ground state energy phase estimation for precision $\varepsilon = 0.0016$ for an N atom hydrogen chain with a minimal basis set, using deterministic, randomized and partially randomized product formulas.

have a scaling of $\mathcal{O}(N^{2.5}\varepsilon^{-2})$, while for the partially randomized product formula we get $\mathcal{O}(N^{2}\varepsilon^{-1.7})$. While the scaling with ε is suboptimal, the scaling with N is competitive with qubitization using hypertensor contraction [46], which has a scaling of $\mathcal{O}(N^{2.1}\varepsilon^{-1})$. For FeMoco, we observe a modest improvement compared to fully randomized product formulas (see Fig. 2.1). The increased number of ancilla qubits is due to the compilation strategy from Pauli rotations to Toffoli gates.

For electronic structure problems, we show that partial randomization can be used to improve the performance of product formulas which decompose the Hamiltonian based on a *factorization* of the Coulomb tensor [95, 45]. For our quantum chemistry resource estimates, we perform a numerical estimate of the Trotter error for the hydrogen chain and a benchmark set of small molecules, and optimize of the representation of the Hamiltonian for improved performance.

2.1.2 Comparison to previous work

We first discuss previous work on phase estimation using randomized product formulas. Previous state-of-the art resource estimates for QPE using qDRIFT were given in [46]. That work presents three resource analyses of phase estimation using qDRIFT, based on diamond norm bounds for the qDRIFT error. Directly requiring mean squared error ε^2 lead to a scaling of the number of gates of $\mathcal{O}(\lambda^4\varepsilon^{-4})$, which is excessively expensive. The data point on Fig. 2.1 is their improvement which uses a less stringent error measure, namely a 95% confi-

dence interval of width ε. Finally, [46] gives a third approach, where the goal is to use a Hodges-Lehmann estimator on multiple samples of the QPE outcome. This reduces the phase estimation overhead by an order of magnitude, but relies on the non-rigorous assumption that the qDRIFT error is symmetric. For reference, Fig. 2.1 also compares with state-of-the-art estimates based on qubitization. These have better scaling in the precision and better overall performance, but do require more ancilla qubits in order to implement linear combinations of unitaries.

For single-ancilla phase estimation diamond norm bounds are not needed, and this was used in [86]. That work additionally proposed a modified randomized product formula, which we call Randomized Taylor Expansion (RTE). This has the property that it can be used to give an unbiased estimator of the time evolution unitary in single-ancilla QPE. Our analysis extends this idea to qDRIFT, which can be seen as only keeping the first order in the Taylor expansion. We also note that if one can prepare a state with high ground state overlap, one can break up the computation in multiple smaller circuits. We can compare our estimates for FeMoco to the ones in [86]. The algorithm in [86] is based on computing the cumulative density function of the spectrum, $C(x)$, and finding the ground state energy as the first jump of this function. Here, we aim for error ε with failure probability at most 0.1. Assuming an exact eigenstate and finding a crossing point for $C(x) < 0.3$ or $C(x) > 0.7$, leads to an estimate of approximately 4500 circuits with an *average* number of 7.9×10^9 Pauli rotations. Here, we have used the results from Figure 2 in [86], and rescaled to use our optimized value of λ. We see that robust phase estimation requires shorter depth circuits (without increasing the total required number).

Partial randomization schemes have also been studied previously. There are various different proposals [96, 87, 88, 89, 97, 98]. Our scheme is a modification of that proposed in [89]. The difference is that we replace qDRIFT by the RTE technique from [86] to 'unbias' the product formula, which is helpful when using single-ancilla QPE. Indeed, this means that when analyzing QPE, we do not need to use diamond norm bounds (as derived in [89]), which would lead to worse constant factors. Since we find that in practice partial randomization tends to lead to (modest) constant improvements, it is important to have a sharp cost analysis.

2.2 Single-ancilla quantum phase estimation

We consider Hamiltonians which are a sum of local terms $H = \sum_l H_l$. A special case of interest is when

$$H = \sum_{l=1}^{L} H_l = \sum_{l=1}^{L} h_l P_l, \qquad h_l \in \mathbb{R} \tag{2.1}$$

where $P_l \in \pm\{I, X, Y, Z\}^{\otimes N_q}$ are Pauli operators on N_q qubits. Our main motivation derives from molecular electronic structure Hamiltonians in second quantization, as we will discuss in Section 2.6 onward. In that case, we have $N_q = 2N$, where N is the number of molecular orbitals, and $L = \mathcal{O}(N^4)$. We define the *weight* of the Hamiltonian in Eq. (2.1) to be

$$\lambda = \sum_l |h_l|. \tag{2.2}$$

The eigenvalues of H and the associated eigenvectors are denoted by E_k and $|\psi_k\rangle$, respectively, where $E_0 \leq E_1 \leq E_2 \leq \ldots$. Our task is to compute the ground state energy E_0 up to precision $\varepsilon > 0$. If we know a *guiding state* $|\psi\rangle$ with significant overlap to the ground state $|\psi_0\rangle$, we can use a quantum computer to approximate E_0 by applying QPE to the time evolution operator and the initial state: $e^{-itH}|\psi\rangle$. For many problems involving the electronic structure of molecules, it is feasible to find good guiding states using established methods see [99, 100, 47, 2, 94, 44] for elaborate discussion and further references.

We consider versions of QPE based using only a single ancilla qubit, which use the *Hadamard test*, illustrated in Fig. 2.3. Peforming QPE using only the Hadamard test (without a quantum Fourier transform circuit) has an extensive history [101, 102, 103, 104, 105, 33].

We start with the guiding state $|\psi\rangle$, and a single ancilla qubit is set to $|+\rangle = H|0\rangle$. Having applied U, controlled on the ancilla qubit, we measure the ancilla with respect to the basis $|0\rangle \pm e^{i\theta}|1\rangle$. This gives a random variable $\boldsymbol{Z}_\theta$ which has outcomes ± 1. The Hadamard test has expectation values

$$\mathbb{E}\boldsymbol{Z}_\theta = \Re e^{i\theta} \langle\psi| U |\psi\rangle .$$

In particular, letting $\boldsymbol{X}$ and $\boldsymbol{Y}$ denote the random variables obtained from $\theta = 0, \pi/2$ and setting $\boldsymbol{Z} = \boldsymbol{X} + i\boldsymbol{Y}$, we have

$$\mathbb{E}\boldsymbol{Z} = \langle\psi| U |\psi\rangle .$$

We will focus on the case when $U = U(t) := e^{-itH}$.

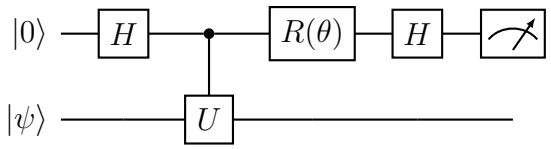


Figure 2.3: The Hadamard test for estimating $\boldsymbol{Z}(\theta)$. Here $R(\theta) = |0\rangle\langle 0| + e^{i\theta}\,|1\rangle\langle 1|$ is a phase gate.

If we denote by $\boldsymbol{Z}_\theta(t)$ the outcomes of the Hadamard test using $U = U(t)$, then

$$\begin{aligned} g(t) := \mathbb{E}\boldsymbol{Z}(t) &= \langle\psi|\exp(-iHt)\,|\psi\rangle \\ &= \sum_k c_k \exp(-iE_k t), \end{aligned} \tag{2.3}$$

where $c_k = |\langle\psi|\psi_k\rangle|^2$. We can approximate $g(t)$ by taking multiple independent samples $\boldsymbol{Z}^{(n)}(t)$ and take the mean $\overline{\boldsymbol{Z}}(t) = \frac{1}{N}\sum_{n=1}^{N}\boldsymbol{Z}^{(n)}(t)$.

If we think of $g(t)$ as a time signal, then the phase estimation routine will constitute a signal processing transformation to compute the lowest frequency of $g(t)$ (corresponding to the energy E_0) [105], provided that we have some guarantee on the overlap of $|\psi\rangle$ with the ground state; we assume a lower bound $c_0 \geq \eta$. With appropriate signal processing methods, one can find the value of E_0 with accuracy ε using m circuits with time evolution for times $t_1, \ldots, t_m$. This can be done such that the maximal time evolution $t_{\max} = \max\{t_1, \ldots, t_m\}$ and the total time over all circuit runs $t_{\text{tot}} = t_1 + t_2 + \cdots + t_m$ both scale as ε^{-1}. This *Heisenberg scaling* is known to be optimal [106]. When using a Hamiltonian simulation method to simulate the time evolution, $t_{\max}$ is what determines the maximal number of gates per circuit and t_{tot} determines the total gate count. We measure the error of the QPE by its (root) mean square error. In our resource estimates we consider two different scenarios. In the first scenario, we are given access to the exact eigenstate $|\psi_0\rangle$ (so $c_0 = 1$). In the more realistic second scenario, we consider a state $|\psi\rangle$ with an overlap bound $\eta < 1$.

There are various ways to extract the ground state energy by sampling from $\boldsymbol{Z}(t)$ which achieve Heisenberg scaling [104, 33, 39]. A particularly elegant method is *robust phase estimation* [107, 103, 26, 38]. For now, we assume that the Hadamard test is implemented using *exact* time evolution, and we measure the cost in terms of the time evolution required. This method assumes that η is sufficiently large. The idea of robust phase estimation is to estimate the time evolution signal at times $t = 2^m$ for $m = 1, \ldots, M$ with $M = \lceil \log \varepsilon^{-1} \rceil$. For each m, the expectation value of the Hadamard test outcome $\boldsymbol{Z}(2^m)$ is

$$g(2^m) = \sum_k c_k \exp(-i2^m E_k).$$

We repeat the test N_m times and obtain the average $\overline{\boldsymbol{Z}}(2^m)$. The integer M corresponds to the number of precision bits in the estimate of E_0. With each round we update a guess θ_m for E_0. Given an outcome $\overline{z}(2^m) = r_m \exp(-i\phi_m)$ of the random variable $\overline{\boldsymbol{Z}}(2^m)$ and an estimate θ_{m-1} for E_0 from the previous round, the new estimate θ_m is the number which is compatible with the value ϕ_m in the sense that $2^m\theta_m$ equals ϕ_m modulo 2π, and is close to the estimate for θ_{m-1} from the previous round. That is, $\theta_m = 2^{-m}(\phi_m + 2\pi k)$ for the integer k such that this is closest to θ_{m-1}. It can be proven [107, 103, 26, 38] that for $\eta \geq 4 - 2\sqrt{3} \approx 0.54$ this gives a correct algorithm with maximal evolution time $t_{\max} = \mathcal{O}(\varepsilon^{-1})$ and total evolution time $t_{\text{tot}} = \mathcal{O}(\varepsilon^{-1})$. In molecular electronic structure problems, in most cases it is possible to reach this ground state overlap regime [99, 47, 2, 94]. In situations with smaller values of η, there exist various other single-ancilla phase estimation schemes [33, 108, 39]. The methods in this work apply directly to these phase estimation schemes as well. The robust phase estimation scheme has the advantage that if the overlap is close to 1, one can reduce the maximal required time evolution, at the cost of increasing the total time evolution [38]. Indeed, if $\eta = 1 - \alpha$ for small α, we may take a trade-off parameter $\Omega(\alpha) = \xi \leq 1$ (i.e. we can take ξ small if the ground state overlap is large), and obtain

$$t_{\max} = \mathcal{O}\left(\frac{\xi}{\varepsilon}\right), t_{\text{tot}} = \mathcal{O}\left(\frac{1}{\xi\varepsilon}\right). \tag{2.4}$$

This result is due to [38]; for completeness we provide a precise statement and proof in Theorem 2.9.5.

2.3 Deterministic product formulas

The previous section involved approximating e^{-iHt} as a subroutine. In this section, we explain deterministic product formulas, also known as the Trotter method, one of the leading approaches to Hamiltonian simulation, on which our work will build. We refer to [51] for an extensive discussion and references. Given a Hamiltonian $H = \sum_{l=1}^{L} H_l$ consisting of L terms, we may approximate for a short time evolution for time δ as

$$U(\delta) \approx S_1(\delta) := e^{-i\delta H_L} e^{-i\delta H_{L-1}} \cdots e^{-i\delta H_2} e^{-i\delta H_1}.$$

This is a first order product formula. A second order product formula is

$$S_2(\delta) := \left(e^{-\frac{i}{2}\delta H_1} \cdots e^{-\frac{i}{2}\delta H_L}\right)\left(e^{-\frac{i}{2}\delta H_L} \cdots e^{-\frac{i}{2}\delta H_1}\right)$$

and higher order Suzuki-Trotter product formulas can be recursively defined by

$$S_{2k}(\delta) = (S_{2k-2}(u_k\delta))^2 S_{2k-2}((1-4u_k)\delta)(S_{2k-2}(u_k\delta))^2$$

for $u_k = (4 - 4^{1/(2k-1)})^{-1}$. For the Trotter-Suzuki product formula of order $2k$, $S_{2k}(\delta)$ requires $2L5^{k-1}$ unitaries of the form $e^{-i\varphi H_l}$. Compiling $e^{-i\varphi H_l}$ into elementary gates gives a quantum circuit. For example, if $H_l = h_l P_l$ are Pauli operators, $e^{-i\varphi H_l}$ is a Pauli rotation, which can straightforwardly be compiled into elementary gates.

In general, we write $S_p(\delta)$ for a p-order product formula; meaning that $S_p(\delta)$ is a product of time evolutions along the H_l. The order refers to the order of approximation in the sense that the error between $U(\delta)$ and $S_p(\delta)$ is $\mathcal{O}(\delta^{p+1})$. Time evolution for arbitrary time t can be approximated by dividing t into a sufficiently large number of intervals r and discretize $U(t)$ into r *Trotter steps*

$$U(t) = U(t/r)^r \approx S_p(t/r)^r$$

with time step $\delta = t/r$.

We will now discuss the error due to the approximation of $U(\delta)$ by $S_p(\delta)$, which we will refer to as the *Trotter error*, in more detail. If one wants to simulate $U(\delta)$, a good error measure is the operator norm difference, which scales as

$$\|U(\delta) - S_p(\delta)\|_{\text{op}} \leq C_{\text{op}} \delta^{p+1}.$$

Here we have written the prefactor

$$C_{\text{op}} = C_{\text{op}}(p, \{H_l\})$$

explicitly. We refer to this constant (and similar parameters, with respect to different error measures, defined below) as the *Trotter constant*. While constant with respect to the choice of step size δ, it does depend on the order p and the Hamiltonian and its decomposition into terms H_l (and in that way on the system size). This dependence is important.

For the purpose of phase estimation for ground state energies, we can consider a different error metric. For fixed δ, we may write $S_p(\delta) = \exp\left(-i\delta\tilde{H}\right)$ for some effective Hamiltonian $\tilde{H}$ which is close to H for small δ. If we apply QPE to the unitary $S_p(\delta)$, this is equivalent to performing QPE to the effective Hamiltonian $\tilde{H}$, and we can find an estimate for $\tilde{E}_0$, the ground state energy of $\tilde{H}$ (note that $\tilde{H}$ depends on δ as well). The outcome of the QPE now has a *bias* due to the Trotter error, and in order to obtain an estimate of E_0 to precision ε, we need that $|E_0 - \tilde{E}_0| = \mathcal{O}(\varepsilon)$. The scaling of this bias is

$$|E_0 - \tilde{E}_0| \leq C_{\text{gs}} \delta^p \tag{2.5}$$

for a Trotter constant $C_{\text{gs}} = C_{\text{gs}}(p, \{H_l\})$. This means that $\delta = \mathcal{O}((C_{\text{gs}}^{-1}\varepsilon)^{1/p})$ suffices for precision ε. Phase estimation needs $\mathcal{O}(\delta^{-1}\varepsilon^{-1})$ applications of $S_p(\delta)$

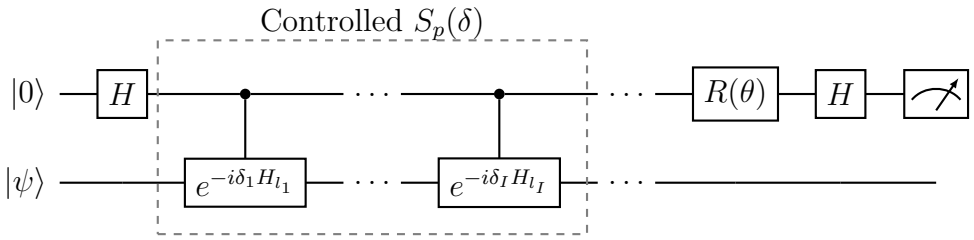


Figure 2.4: Performing the Hadamard test on $S_p(\delta)^s$ allows one to estimate $\langle\psi| S_p(\delta)^s |\psi\rangle = \langle\psi| \exp\left(-i\delta s\tilde{H}\right) |\psi\rangle$.

to estimate $\tilde{E}_0$ to precision $\mathcal{O}(\varepsilon)$. Together, this yields a scaling of $\mathcal{O}(C_{\mathrm{gs}}^{\frac{1}{p}}\varepsilon^{-1-\frac{1}{p}})$ Trotter steps.

The operator norm error can be bounded in the following way [51]:

$$\|U(\delta) - S_p(\delta)\|_{\mathrm{op}} \leq \alpha_p(\{H_l\})\delta^{p+1} = \mathcal{O}((\lambda\delta)^{p+1}). \tag{2.6}$$

Here $\alpha_p(\{H_l\})$ is an expression in terms of operator norms of nested commutators of the terms H_l in the Hamiltonian, and λ is the weight defined in Eq. (2.2). One can then use these to bound

$$|E_0 - \tilde{E}_0| = \mathcal{O}\left(\delta^{-1}\|U(\delta) - S_p(\delta)\|_{\mathrm{op}}\right) = \mathcal{O}(\lambda^{p+1}\delta^p).$$

The commutator bounds are often quite sharp [81, 51]; a disadvantage of such bounds is that the number of commutator terms scales as $\mathcal{O}(L^{p+1})$ and may be practically infeasible to compute even for modest system sizes [54].

In Section 2.6, we numerically estimate the Trotter error constant for ground state energy errors, and see that it is typically relatively small. This is an important aspect of product formula algorithms for phase estimation, since it means that Trotter methods may in practice use larger step sizes δ than one can rigorously justify and can be implemented using fewer resources than rigorous upper bounds suggest. Because of the relevance of the Trotter error for Hamiltonian simulation, this has been studied extensively in previous work; see [54, 109, 52, 40, 110] for empirical and rigorous Trotter error bounds for phase estimation for quantum chemistry. Additionally, there is theoretical evidence that for most states the Trotter error is significantly lower than the worst-case bounds [111, 112, 113].

2.4 Phase estimation with random compilation for time evolution

Two major downsides to Trotter product formulas are their scaling with L, and a difficult to rigorously control Trotter error. One can avoid both these features using *randomized product formulas* [55, 86, 114, 98, 115, 116]. Here we discuss two randomized methods of implementing an approximation to the time evolution $U(t)$: qDRIFT [55] and a randomized Taylor series expansion (RTE) [86].

We now assume that the Hamiltonian consists of a linear combination of Pauli operators P_l, since for these randomized methods we will use that $P_l^2 = I$. We will denote by

$$V_l(\phi) = e^{-i\phi P_l}$$

a rotation along the Pauli operator P_l, and refer to these as Pauli rotations. Absorbing signs into the Pauli operators (i.e. replacing $P_l \mapsto \mathrm{sgn}(h_l)P_l$), we may write the Hamiltonian as

$$H = \lambda \sum_{l=1}^{L} p_l P_l, \quad p_l = \frac{|h_l|}{\lambda}$$

where the numbers p_l form a probability distribution p. For convenience of notation we will now assume that $\lambda = 1$; it can be reinstated by rescaling the time variable by λ. To approximate time evolution for time t, we again divide t into r steps, with a step size τ so that $\tau r = t$. The qDRIFT approximation to time evolution is given by sampling random variables $\boldsymbol{l_1}, \ldots, \boldsymbol{l_r}$ according to p, and given outcomes $l_1, \ldots, l_r$, apply the unitary

$$V_{l_r}(\varphi) \cdots V_{l_1}(\varphi), \quad \text{where } \varphi = \arctan(\tau).$$

That is, we apply a sequence of equiangular Pauli rotations, and which ones we apply is sampled proportionally to the weight of that Pauli operator in the Hamiltonian. If $\mathcal{S}_\tau$ denotes the quantum channel which with probability p_l applies $V_l(\varphi)$, the above procedure leads to the composition $\mathcal{S}_\tau^r$.

This method was proposed in [55]. In order to measure the accuracy of the approximation of the exact time evolution, we have to compare the exact time evolution unitary with the quantum channel $\mathcal{S}_\tau$. The natural measure to compare two quantum channels is the diamond norm. We let $\mathcal{U}_t$ denotes the quantum channel corresponding to applying the time evolution unitary $U(t)$ mapping $\rho \mapsto U(t)^\dagger \rho U(t)$. It was shown in [55] that if we break the total time t into r shorter steps of time τ, so $\tau r = t$, then we have the diamond norm bound

$$\|\mathcal{U}_t - \mathcal{S}_\tau^r\|_\diamond \leq 2r\tau^2 \exp(2\tau) \approx 2t^2/r.$$

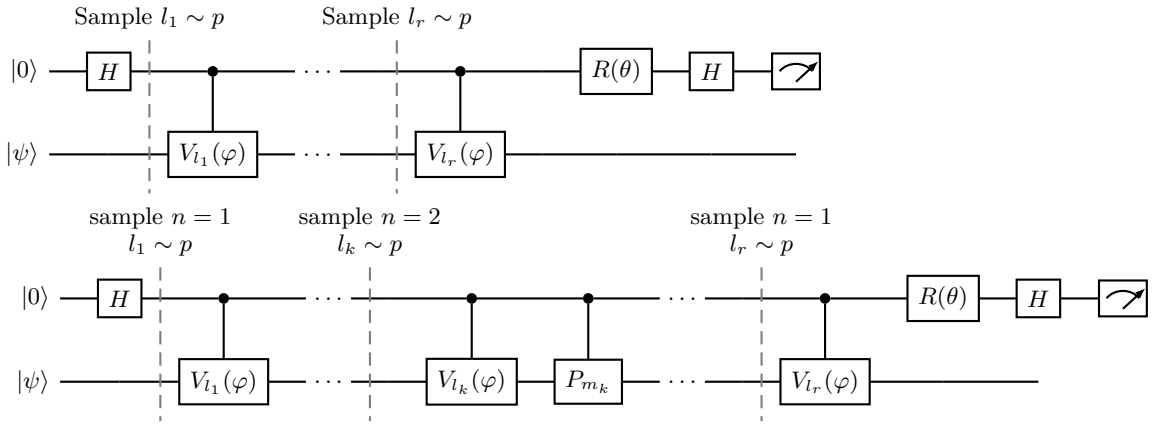


Figure 2.5: Hadamard test for qDRIFT and for RTE. For qDRIFT, $\tau = t/r$ and $\varphi = \arctan(\tau)$. RTE is similar, but now for each gate one samples an order n of the Taylor expansion. If $n = 1$, one samples l according to p as for qDRIFT. If $n > 1$ (which happens with probability $\mathcal{O}(\tau^2)$), the rotation is over a different angle φ_n and there is an additional controlled Pauli operator in the circuit.

This means that choosing a number of steps $r \geq 2t^2\varepsilon^{-1}$, or reinstating the dependence on λ, $r \geq 2\lambda^2 t^2 \varepsilon^{-1}$, suffices to approximate time evolution for time t to error ε in diamond norm. In particular, there is no dependence on L, but only on λ. This is potentially beneficial if L is large and there is a large number of terms with small coefficients in the Hamiltonian.

This can then be used to simulate time evolution in quantum phase estimation. Since phase estimation requires time evolution for time $t = \Theta(\varepsilon^{-1})$, this would lead to a scaling of $r = \mathcal{O}(\varepsilon^{-2})$ Pauli rotations. However, directly applying the diamond norm bounds with qDRIFT in the standard QPE algorithm leads to suboptimal results [55, 46]. In general, one can ask whether using qDRIFT leads to more efficient phase estimation procedures than using Trotterization. There is no unambiguous answer: the randomized method scales worse with ε, and the relation between L and λ depends on the Hamiltonian. An advantage for qDRIFT is most likely if L is large, which is the case for molecular systems using atomic orbitals. Here, [55] found an advantage, but using very loose upper bounds for the Trotter error. In [46], for FeMoco it was found that qDRIFT was much less efficient than the Trotter estimate with optimistic Trotter error estimates from [40] (see Fig. 2.1).

We will now argue that previous work has overestimated the cost of doing phase estimation using qDRIFT. A key reason is that using an analysis based on diamond norm bounds is suboptimal, as also observed in [86]. Similar to the discussion of Trotter error, what we care about eventually is not an accurate simulation of the full time evolution (corresponding to small diamond norm),

but rather the accuracy of the ground state energy as extracted from the phase estimation procedure. In particular, for phase estimation methods based on the Hadamard test, what is relevant is the expectation value of the outcome of the Hadamard test. If we apply the Hadamard test to a quantum channel which applies unitary U_m with probability p_m to initial state $|\psi\rangle$, the resulting random variable $\boldsymbol{Z}$ has expectation value

$$\mathbb{E}\boldsymbol{Z} = \sum_m p_m \langle\psi| U_m |\psi\rangle .$$

The expectation value is over two sources of randomness: the sampling of the unitary U_m and the randomness from the measurement at the end of the circuit. This means that what matters for the Hadamard test is the linear combination of unitaries $\sum_k p_m U_m$ rather than the quantum channel $\rho \mapsto \sum_m p_m U_m \rho U_m^\dagger$. Let $\boldsymbol{Z}(\tau, r)$ be the random variable obtained from applying a Hadamard test to r rounds of qDRIFT with interval τ. Consider the Taylor expansion of a short time evolution:

$$\begin{aligned} e^{-i\tau H} &= I - i\tau \sum_l p_l P_l + \mathcal{O}(\tau^2) \\ &= \sum_l p_l (I - i\tau P_l) + \mathcal{O}(\tau^2) \\ &= \sqrt{1+\tau^2} \sum_l p_l V_l(\phi) + \mathcal{O}(\tau^2) \end{aligned} \tag{2.7}$$

where we have used that $I - i\tau P_l = \sqrt{1+\tau^2} V_l(\phi)$ for $\phi = \arctan(\tau)$. Alternatively, we can use

$$\sum_l p_l V_l(\varphi) = \frac{I - i\tau H}{\sqrt{1+\tau^2}}. \tag{2.8}$$

Using this, one can show (see Section 2.9.1 for details) that, when using an initial state $|\psi\rangle$ with squared overlap c_k with the energy E_k eigenstate, the expectation value of $\boldsymbol{Z}(\tau, r)$ has the exact expression

$$g(\tau, r) := \mathbb{E}\boldsymbol{Z}(\tau, r) = \sum_k \frac{c_k}{(1+\tau^2)^{r/2}} (I - i\tau E_k)^r. \tag{2.9}$$

If we choose a sufficiently small step size τ, we get

$$g(\tau, r) \approx \sum_k c_k e^{-\tau^2 r (1 - E_k^2)/2} e^{-i\tau r E_k}.$$

This expression is similar to $g(t)$ (the exact time signal of Eq. (2.3)) if we take $t = \tau r$, with the difference that there is a *damping factor* $\exp(-\tau^2 r(1 - E_k^2)/2)$. So, if we want to deduce information about the phases E_k, and we do not want to increase the number of required samples, we need to choose τ and r such that $\tau^2 r$ is $\mathcal{O}(1)$. This leads to a scaling of $r = \Omega(t^2)$. One can reduce this value of r, but this comes at an (exponential) sampling overhead. The expectation value in Eq. (2.9) is an *exact* expression, so it is more amenable to analysis than the result of deterministic product formulas, where one has to bound the Trotter error. The original proposal in [55] uses an angle $\varphi = \tau$; this only changes the expectation values to order $\mathcal{O}(\tau^2)$ and does not change the error analysis; here we choose $\varphi = \arctan(\tau)$ for consistency with the Taylor expansion.

An alternative random product formula for Hamiltonian simulation was introduced in [86], which we call a *randomized Taylor expansion* (RTE). Whereas qDRIFT, as per Eq. (2.7), can be seen as a first-order expansion of $\exp(-iH\tau)$ as an LCU, one can also take the full Taylor expansion and normalize the coefficients to a probability distribution. This gives an LCU

$$e^{-i\tau H} = B(\tau) \sum_m b_m U_m \tag{2.10}$$

where the b_m form a probability distribution, and the U_m are unitaries which are a composition of one Pauli rotation and one Pauli operator and $B(\tau)$ is a normalization factor; see Section 2.9.1 for details and a derivation. Up to normalization, qDRIFT precisely corresponds to only keeping the lowest order term. The probability of sampling a higher order term is $\mathcal{O}(\tau^2)$. The normalization factor $B(\tau)$ is approximately $\exp(\tau^2)$ for small τ and can be efficiently computed. Taking r rounds thus gives a LCU

$$e^{-i\tau r} = B(\tau)^r \sum_{m_1,\dots,m_r} \underbrace{b_{m_1} \dots b_{m_r}}_{q_{\vec{k}}} \underbrace{U_{m_1} \cdots U_{m_r}}_{W_{\vec{k}}} \tag{2.11}$$

where the $q_{\vec{k}}$ are a probability distribution over $\vec{k} = (m_1, \dots, m_r)$, and the $U_{\vec{k}}$ are unitaries consisting of r Pauli rotations interspersed with Pauli operators. Performing a Hadamard test leads to a signal

$$\mathbb{E}\boldsymbol{Z}(\tau, r) = \frac{1}{B(\tau)^r} \sum_k c_k e^{-iE_k \tau r}.$$

In particular, if we want to estimate $g(t)$, one can do so using this random variable for $\tau r = t$. The factor $B(\tau)^r \leq \exp(\tau^2 r)$ determines the required number of samples. To keep this constant, one should have $\tau^2 r = \mathcal{O}(1)$ and hence $r = \Omega(t^2)$, as for qDRIFT. The advantage of this approach is that, opposed to qDRIFT, it gives an *unbiased* estimator for the signal $g(t)$.

Since we have exact expressions for the expectation value of Hadamard tests for both qDRIFT and RTE, we can now use these to analyze single-ancilla QPE schemes. We do so for robust phase estimation. The following result is stated and proven as Theorem 2.9.5 in Section 2.9.2 and follows straightforwardly from the analysis of robust phase estimation in [38, 103, 26]. In contrast to the approach in [55, 46], there is no dependence on a failure probability of the algorithm. We consider the case with large ground state overlap. If $c_0 \geq \eta = 1 - \alpha$, we can choose a parameter $\xi = \Omega(\alpha)$ to reduce the maximal circuit size.

Theorem 2.4.1. *Given a guiding state $|\psi\rangle$ with ground state overlap lower bounded by $\eta \geq 4 - 2\sqrt{3} \approx 0.54$ and $\xi = \Omega(\alpha)$ for $\eta = 1 - \alpha$, robust phase estimation using random product formulas gives an estimate of E_0 with root mean square error ε using circuits with at most*

$$r_{\max} = \mathcal{O}\left(\frac{\xi^2\lambda^2}{\varepsilon^2}\right)$$

Pauli rotations per circuit and a total of

$$r_{\text{tot}} = \mathcal{O}\left(\frac{\lambda^2}{\varepsilon^2}\right)$$

Pauli rotations.

We also numerically estimate the constant prefactors, and for $\eta = 1, \xi = 1$ we find $r_{\text{tot}} \leq 8\lambda^2\varepsilon^{-2}$ using qDRIFT. For $\eta = 1$, since we lose Heisenberg scaling, if one does not count the cost of preparing the exact ground state one might as well just measure the ground state energy by using $\mathcal{O}(\lambda^2\varepsilon^{-2})$ copies of the ground state. However, the robust phase estimation procedure uses only $\mathcal{O}(\log(\lambda\varepsilon^{-1})^2)$ copies of the ground state (when choosing $\xi = 1$), and crucially it still works for ground state overlap $\eta < 1$ sufficiently large. Note additionally that the *total* cost does not depend on the trade-off parameter ξ, in contrast to the trade-off in Eq. (2.4). However, choosing small ξ does come at a cost of a higher number of circuits, and in particular of the number of initial state preparation circuits. For randomized methods it is advantageous to choose $\xi \sim 1-\eta$ as small as possible (if the state preparation cost is not too high), since the total cost of the Hamiltonian simulation does not increase under the depth reduction. This is in contrast to the exact time evolution model, where a reduction in maximal evolution time comes at the cost of an increasing total evolution time, as in Eq. (2.4), due to the larger number of circuit repetitions. The suboptimal scaling of $\mathcal{O}(\varepsilon^{-2})$ of randomized methods is a significant disadvantage. However, as we see in molecular electronic structure examples, it can still outperform deterministic product formulas. This advantage increases further when one uses depth reductions for states with high ground state overlap.

2.5 Partially randomized product formulas

We have seen that randomization can be useful for Hamiltonian simulation in the context of phase estimation. However, it has the clear disadvantage of scaling quadratically with λ and ε^{-1}, whereas deterministic product formulas may have a better dependence on ε^{-1}. Deterministic product formulas are disadvantageous if L, the number of terms in the Hamiltonian, is large. Random product formulas are helpful if there is a tail consisting of many terms in the Hamiltonian of small weight.

A natural proposal is to treat a subset of terms in a random fashion, and a subset of terms deterministically. There exist various such proposals, with different ways of implementing the partial randomization [87, 97, 88, 89]. Here, we modify the scheme of [89] to a version which has the advantage of being particularly easy to analyze for the purpose of single-ancilla phase estimation. Again, the difference in the analysis is that we do not require a diamond norm bound on the resulting quantum channel, but only need to evaluate the expectation value of the resulting Hadamard test.

The partial randomization is based on a decomposition

$$H = \underbrace{\sum_{l=1}^{L_D} H_l}_{=H_D} + \underbrace{\sum_{m=1}^{M} h_m P_m}_{=H_R} \tag{2.12}$$

where we assume that the P_m square to identity and where H_D contains the terms we treat deterministically, and H_R the terms which are treated by a random product formula. The idea is to use a deterministic Trotter formula to the decomposition of H into the terms H_l and H_R, and to apply a random product formula to the decomposition of H_R into terms $h_m P_m$. In order for this to yield an improvement, one aims for a decomposition such that

$$\lambda_R = \sum_m |h_m| \ll \lambda \qquad \text{and} \qquad L_D \ll M. \tag{2.13}$$

To analyze how such methods will perform, we now briefly summarize two lessons from the previous two sections. Deterministic product formulas lead to a *bias* in the ground state energy E_0 due to the Trotter error. A small step size is required in order to keep this bias below the desired precision. For randomized product formulas, one can make the resulting signal such that the ground state energy is unbiased, but the *amplitude* of the signal is damped and one incurs a sampling overhead. Here, a sufficiently small step size is required in order to keep this sampling overhead of constant size. Partially randomized product formulas

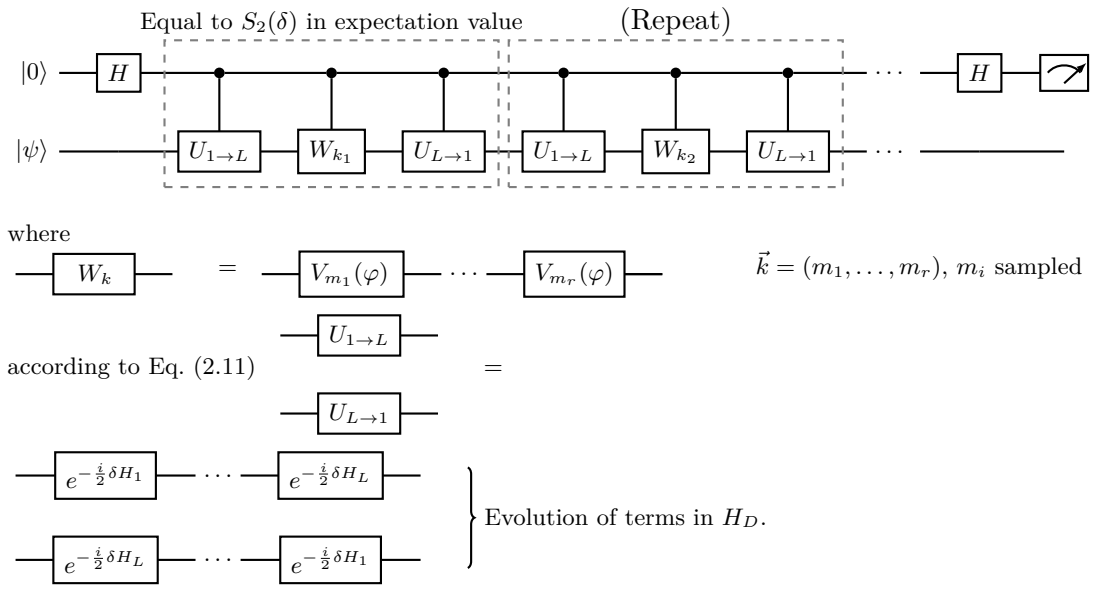


Figure 2.6: Hadamard test using partially randomized product formula. Here, we illustrate the second order deterministic method, based on Eq. (2.14).

will have both a bias (from the deterministic decomposition) and a damping (from implementing the randomized part of the Hamiltonian) in the resulting signal. The error analysis straightforwardly combines these elements.

Let $S_p(\delta)$ be a deterministic product formula with respect to the decomposition of H into the terms H_l and H_R. This means that $S_p(\delta)$ is a product of unitaries of the form $e^{-i\gamma H_l}$ (which we assume to have some decomposition into elementary gates) and terms of the form $e^{-i\gamma H_R}$ for some γ. As in Eq. (2.5), this gives an effective Hamiltonian with ground state $\tilde{E}_0$ with error $\left|E_0 - \tilde{E}_0\right| \leq C_{\text{gs}}\delta^p$. Here, the Trotter constant $C_{\text{gs}} = C_{\text{gs}}(p, \{H_l\}_{l=1}^{L_D}, H_R)$ depends on the order p as well as on the Hamiltonian, decomposed into terms H_l for $l = 1, \ldots, L_D$ and treating H_R as a single term. Each of the $e^{-i\gamma H_R}$ appearing in the formula we will then simulate using RTE, using r steps per Trotter step.

To make this concrete, we take the second order Suzuki-Trotter formula, in which case we get $S_2(\delta)$ given by

$$e^{-\frac{1}{2}i\delta H_1} \cdots e^{-\frac{1}{2}i\delta H_{L_D}} e^{-\delta i H_R} e^{-\frac{1}{2}i\delta H_{L_D}} \cdots e^{-\frac{1}{2}i\delta H_1}. \tag{2.14}$$

If we normalize H_R/λ_R, we need to do time evolution for time $\delta\lambda_R$, which we

break up into r steps. Using RTE as in Eq. (2.11) we expand

$$\begin{aligned} e^{-i\delta H_R} &= (e^{-i\tau H_R/\lambda_R})^r \quad \text{for } \tau r = \delta\lambda_R \\ &= B\sum_k q_k W_k \end{aligned}$$

as a LCU, which gives a LCU for $S_2(\delta)$. Here, each W_k is a composition of r Pauli rotations and Pauli operators.

Since this gives an unbiased LCU for the exact time evolution, it is straightforward to show (see Lemma 2.9.3) that using RTE and repeating s times in a Hadamard test has expectation value

$$\mathbb{E}\boldsymbol{Z}(\delta, r, s) = \frac{1}{B(r,\delta)^s} \langle\psi| S_p(\delta)^s |\psi\rangle$$

where

$$B(r,\delta) \leq \exp\left(\mathcal{O}\left(\frac{\lambda_R^2\delta^2 s}{r}\right)\right).$$

In order to perform phase estimation, we need to choose δ such that the Trotter error of $S_p(\delta)$ is smaller than ε. That means we should choose a Trotter step size $\delta = \mathcal{O}((C_{\text{gs}}\varepsilon)^{\frac{1}{p}})$ and maximal depth $s = \mathcal{O}(C_{\text{gs}}^{\frac{1}{p}}\varepsilon^{-1-\frac{1}{p}})$ for phase estimation. In order to keep the number of samples constanty that are required to estimate $\langle\psi| S_p(\delta)^s |\psi\rangle$ to constant precision, we need

$$B(r,\delta)^s \leq \exp\left(\mathcal{O}\left(\frac{\lambda_R^2\delta^2 s}{r}\right)\right)$$

to be constant. Since $\delta s = \mathcal{O}(\varepsilon^{-1})$, this can be achieved by taking the total number of rotations in the randomized part as $rs = \mathcal{O}(\lambda_R^2\varepsilon^{-2})$. Together, this leads to the following result.

Theorem 2.5.1. *Let H as in Eq. (2.13), and suppose that the Trotter error constant for H (written as a sum of the H_l and treating H_R as a single term) is C_{gs}, then phase estimation to precision ε using the partially randomized product formula requires $\mathcal{O}(L_D C_{\text{gs}}^{\frac{1}{p}}\varepsilon^{-1-\frac{1}{p}})$ evolutions along the H_l and $\mathcal{O}(\lambda_R^2\varepsilon^{-2})$ Pauli rotations.*

Gate counts with constant factors are given in Section 2.9.5. A subtlety here is that the Trotter constant C_{gs} depends on the choice of partitioning. This complicates the task of finding an optimal distribution of terms to H_R or H_D. One can bound C_{gs} using a commutator bound in the usual way (see [89] for

such a bound). For our numerical estimates we will take a different approach: we estimate the Trotter error for the fully deterministic method, and use the resulting Trotter error constant. In that case, it is clear that to minimize the cost, the optimal decomposition assigns the L_D largest weight terms to H_D for some choice of L_D, and the remainder to H_R. Suppose that H has L terms and weight λ. Clearly, partial randomization is most useful if there is a small number $L_D \ll L$ of terms with large weight (so including these in H_D means that $\lambda_R \ll \lambda$); see Fig. 2.7 for an example of a distribution of weights for a quantum chemistry Hamiltonian.

2.6 Electronic structure Hamiltonians

We start with a molecular electronic structure Hamiltonian in second quantization:

$$\sum_{pq,\sigma} h_{pq} a^\dagger_{p\sigma} a_{q\sigma} + \frac{1}{2} \sum_{pqrs,\sigma\tau} h_{pqrs} a^\dagger_{p\sigma} a^\dagger_{r\tau} a_{s\tau} a_{q\sigma}, \tag{2.15}$$

where $a^\dagger_{p\sigma}$, $a_{q\sigma}$ are fermionic creation and annihilation operators and h_{pq}, h_{pqrs} are integrals arising from a calculation using the underlying orbital basis (see 2.9). The indices p, q, r, s denote the spatial degrees of freedom, whereas σ, τ are reserved for spin. The Hamiltonian acts on the space of $2N$ spin-orbitals, and we fix the number of electrons n. And a choice of fermion-to-qubit mapping transforms it into a linear combination of Pauli operators, acting on $2N$ qubits with $L = \mathcal{O}(N^4)$ terms gives a representation of H as a linear combination of Pauli operators

$$H = \sum_l H_l = \sum_l h_l P_l. \tag{2.16}$$

We work with active spaces of molecular orbitals. An alternative choice is a plane wave basis, which has the advantage that the Hamiltonian has fewer terms and the Trotter error can be more easily controlled. This is particularly useful for materials [110, 51, 85, 117, 84]. For this work we restrict to active spaces based on molecular orbitals since plane wave bases need a much larger number of orbitals to reach similar accuracy. While in principle there are $\mathcal{O}(N^4)$ nonzero coefficients in Eq. (2.15), many are extremely small and can be truncated. For large spatially extended systems, one expects that the number of significant terms mostly scales quadratically in N. Generally, one expects polynomial decay of the weight, with a tail of terms which decays exponentially as in Fig. 2.7. See Section 2.9.3 for more discussion.

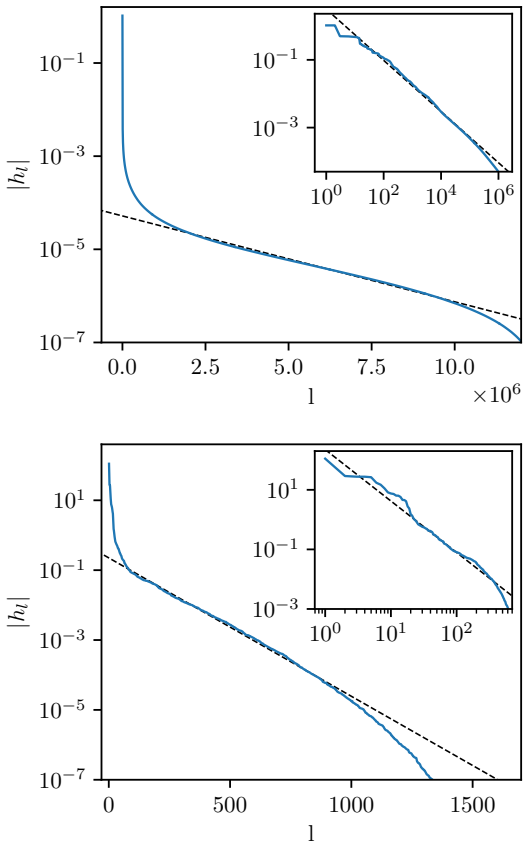


Figure 2.7: Distribution of the weights of terms of the FeMoco Hamiltonian in the Pauli decomposition (left) and the 1- and 2-electron terms in the double factorization representation (right). The dashed lines show an exponential fit to the tail (main plot) and a power law for the largest terms (the log-log inset plot).

As benchmark systems, we use a linear chain of hydrogen atoms, which is a toy model for a system of increasing size, and FeMoco, a standard challenge benchmark problem used for quantum computing resource estimates, as well as a collection of small molecules with varying active space sizes. Detailed descriptions can be found in Section 2.9.3.

The computational cost of implementing product formulas for H depends on various factors: for deterministic product formulas, it depends on L and the Trotter error; for randomized product formulas it depends on the weight λ, and for partially randomized methods it additionally depends on a choice of partitioning into deterministic and randomized terms. In order to assess the performance of phase estimation with partially randomized product formulas, in the following subsections we discuss numerical estimates of the Trotter error, and the decomposition into H_D and H_R.

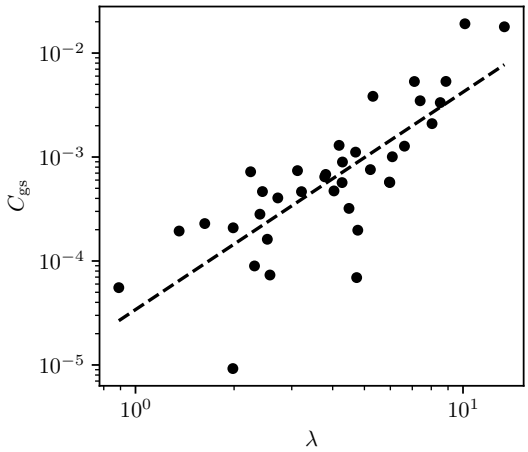


Figure 2.8: The Trotter error (on the ground state energy) scales as $C_{\mathrm{gs}}\delta^{p+1}$, here we plot the prefactor C_{gs} for the second order method for a variety of molecules and active spaces against λ.

2.6.1 Estimating Trotter error

In order to give a good comparison between deterministic and (partially) randomized product formulas, we start by evaluating the Trotter error, which is crucial for the performance of the deterministic product formula. We perform a detailed analysis of Trotter error with system size.

Apart from the practical difficulties in evaluating the error bounds based on commutators in Eq. (2.6), for the purpose of computing ground state energies bounding the ground state energy error in terms of the operator norm error can be a loose bound. This has been confirmed by numerical chemistry simulations for small molecules [52]. From a theoretical perspective, the operator norm of our matrices depends on how many orbitals we choose to include, which is a choice of the algorithm and not a fundamental physical feature of the problem; see [118] for more discussion of this.

We numerically estimate the Trotter error constant for ground state energy errors, and see that it is typically relative small, as shown in Fig. 2.8. See also [54, 109, 52, 40, 110] for previous work on empirical and rigorous Trotter error bounds for phase estimation for quantum chemistry. This is an important aspect of product formula algorithms for phase estimation, since it means that Trotter methods may in practice use larger step sizes δ than we can rigorously justify and can be implemented using fewer resources than rigorous upper bounds suggest.

As discussed in Section 2.3, there are various error measures and bounds one can use. For our purposes, we care about the ground state energy error. In order to estimate this error, we consider a benchmark set of small molecules with

varying active space size and at half filling, so the number of electrons n equals half the number of spin-orbitals $2N$. We compute the Trotter error constant numerically for the second order method with $p = 2$. As we show in Fig. 2.8, the Trotter error constant correlates well with λ. Crucially, even though the Trotter error constant scales with system size, it is typically rather small even for larger systems. These results corroborate similar findings in [52, 40].

Further numerical results are presented in Section 2.9.4. We estimate Trotter error scalings for the hydrogen chain, which has a natural scaling limit. In general we see that using the ground state energy error instead of an operator norm bound on the full unitary leads to about an order of magnitude improvement in the cost estimate. Additionally, for our chemistry benchmark systems we see no structural improvement by going to higher order Trotter-Suzuki product formulas.

2.6.2 Choice of decomposition for partial randomization in the Pauli representation

For the partially randomized product formulas, one additionally needs to choose a partition of the Hamiltonian into a deterministic and randomized part as in Eq. (2.12). We first discuss the case where we consider H in a Pauli representation, where the terms H_D and H_R are all Pauli operators (i.e. $H_l = P_l$ in Eq. (2.12)). The total cost of the partially randomized product formula circuit is the sum of the deterministic and randomized parts as in Theorem 2.5.1. Obviously, we want to choose a decomposition which minimizes this sum.

A subtlety is that the Trotter error of the scheme depends on the choice of decomposition, and which terms are treated randomly; i.e. $C_{\text{gs}} = C_{\text{gs}}(p, \{H_l\}, H_R)$. A (rigorous) approach to this is to use commutator bounds to estimate the Trotter error and to take this into account in the optimization problem. Such an approach is pursued in [89] and applies straightforwardly to our method as well. However, this leads to a complicated optimization problem. Additionally, for the type of problems we consider, the deterministic methods need to be taken with heuristic empirical error scaling in order to be competitive. As a heuristic, we simply bound the Trotter error of the partially randomized method by our estimate of the Trotter error of the fully deterministic method. This is not a rigorous treatment but appears to be reasonable in practice. We study the dependence of the Trotter error on the choice of decomposition numerically. The results can be found in Section 2.9.4; the conclusion is that when decreasing L_D, the Trotter error does not decrease significantly unless λ_R is a significant fraction of λ. This means that in practice the heuristic of using the Trotter error of the fully deterministic method should suffice; it is not valid when λ_R is close to λ but in this regime partially randomized product formulas will anyways not improve

much over fully randomized product formulas.

Once one assumes that the Trotter error constant C_{gs} may be taken constant when finding the optimal decomposition, it is clear that the optimal decomposition should simply include the L_D highest weight terms in H_D, and the remainder in H_R, for some choice of L_D. One can then easily numerically search for the optimal value of L_D which minimizes the cost in Theorem 2.5.1. In Fig. 2.2 we compare the gate counts for phase estimation of the hydrogen chain, comparing the deterministic, randomized (qDRIFT) and partially randomized methods. In terms of cost metric, we consider the number of non-Clifford gates (expressed as number of Toffoli gates) after compilation of the Pauli rotations. We also count the number of 2-qubit gates, see Section 2.9.5. In both cases, the partially randomized method achieves a reduction of cost, while scaling similar to the randomized method with increasing system size. We note that the optimal choice of decomposition depends on the cost metric employed (e.g. whether we measure two-qubit gates or non-Clifford gates).

Another interesting use for partial randomization is that one can improve the compilation to elementary gates. The randomized part consists of many rotations over the same angle, which is beneficial when compiling to Toffoli gates [119, 110]. For the deterministic part $H_D = \sum_l h_l P_l$, one can round the values of the h_l to values more favorable for compilation by transferring a small part of the weight to the randomized part, see Section 2.9.5. This is similar to previously known randomization strategies for compilation [120, 121, 122].

2.7 Different representations of the Hamiltonian

The representation in Eq. (2.16) is not unique. A significant degree of freedom is the choice of orbital basis. In Section 2.7.1 we discuss how this degree of freedom can be optimized to improve the performance of randomized product formulas. Moreover, for deterministic product formulas it can be beneficial to use different decompositions of the Hamiltonian based on a factorization of the Coulomb tensor. In Section 2.7.2 we argue that this approach benefits from partial randomization as well.

2.7.1 Weight reduction for the Hamiltonian

The performance of randomized product formulas depends on the value of the weight λ. Therefore, minimizing the value of λ is important for (partially) randomized product formulas. It is in fact relevant for all methods based on LCU,

such as qubitization. For this reason, it has been investigated extensively [9, 10]. We employ two techniques to significantly reduce the value of λ.

The first is that one can choose a different basis for the single-particle Hilbert space (i.e. a unitary transformation of the orbitals) [9]. For choosing a good basis of orbitals, a good starting point are localized orbitals, rather than the canonical orbitals derived from a Hartree-Fock calculation. While this choice already significantly reduces the value of λ, one can also use these as a starting point for further minimization [9]. One can directly optimize λ over parameterized orbital basis transformations via gradient-descent. Since this a high-dimensional optimization problem it is crucial to start from a good initial guess. Typically, using localized orbitals rather than the canonical orbitals derived from a Hartree-Fock calculation reduces the value of λ. In case orbital localization does not converge or does not yield a lower λ value, we find that the orbital rotation into the eigenbasis of the largest Cholesky vector of the Coulomb-tensor decomposition to be an inexpensive alternative. Besides the choice of orbitals, the Hamiltonian has a particle-number symmetry which can be exploited, and this leads to a second technique to find a different Hamiltonian with the same ground state energy, but with lower weight [11].

Applying these methods to FeMoco for the Reiher [40] and Li [42] choice of active space, we see a reduction of the value of λ from 1863 to 405 for the Reiher Hamiltonian and a reduction from 1511 to 568 for the Li Hamiltonian. We note that this also leads to a significant improvement of the performance of the sparse qubitization method of [60] to FeMoco. This leads to the results reported in Fig. 2.1; see Section 2.9.5 for details.

When we discussed the choice of decomposition for the partially randomized method, we assumed a fixed choice of orbitals and hence of the coefficients in the Pauli representation of the Hamiltonian. A reasonable choice is to take an orbital basis for which λ is small. However, a more complicated optimization is possible: one would want to find an orbital basis where most of the weight is concentrated in a small number of terms. We leave such optimization to future work.

2.7.2 Partial randomization with factorizations of the Hamiltonian

For deterministic product formulas, one does not need to restrict to the Pauli representation of the Hamiltonian. One general approach is based on factorizations of the Hamiltonian, where one can factorize the Coulomb tensor h_{pqrs}. This

gives a representation of the 2-body terms of the Hamiltonian as

$$\sum_{l=1}^{L} H_l = \sum_{l=1}^{L} (U^{(l)})^\dagger \underbrace{\Big(\sum_{p=1,\sigma}^{\rho_l} \lambda_p^{(l)} n_{p\sigma} \Big)^2}_{V^{(l)}} U^{(l)} \tag{2.17}$$

where for each l we have a different orbital basis, $U^{(l)}$ is the unitary implementing the change of orbital basis. See Section 2.9.3 for details. The $n_{p\sigma} = a_{p\sigma}^\dagger a_{p\sigma}$ are the number operators. After a Jordan-Wigner transformation, the Hamiltonian $V^{(l)}$ is diagonal in the standard basis. This can be used for a deterministic product formula [95], since one can implement $\exp(-i\delta H_l)$ by

$$\exp(-i\delta H_l) = (U^{(l)})^\dagger \exp\big(-iV^{(l)}\big) U^{(l)}.$$

Both the orbital basis changes and the diagonal Hamiltonian evolutions can be performed using $\mathcal{O}(N^2)$ gates using Givens rotations [123], so in total a single Trotter step needs $\mathcal{O}(LN^2)$ gates (both in terms of number of two-qubit gates and single-qubit rotations).

A decomposition of the form in Eq. (2.17) can be obtained through a *double factorization*, which we review in Section 2.9.3. Here, the first factorization is into a sum of l terms, the second factorization is the diagonalization of these in different orbital bases. This can be an improvement over the Pauli representation if the ranks L and ρ_l are small. Decompositions of the Hamiltonian as in Eq. (2.17), or similar decompositions are also essential to qubitization and other LCU based approaches to Hamiltonian simulation [60, 45, 46]. Typically, in factorized representations, one truncates a tail of terms which are so small that they do significantly affect the ground state energy. Partial randomization allows for much more flexible truncation schemes, and in that way can be used to improve product formulas based on Hamiltonian factorizations.

First of all, in Eq. (2.17), some of the terms H_l are rather small (but not so small that they can be completely ignored). As an example, the distribution of the sizes of the H_l for a double factorization of the FeMoco Hamiltonian is shown in Fig. 2.7. This shows, similar to the Pauli representation, a dominant set of terms obeying powerlaw decay, and a tail of terms with exponential decay. See [124] for a discussion of the scaling of ρ_l for spatially extended systems. Note that the number of terms is much smaller than in the Pauli representation (but the cost of implementing any of the $\exp(-i\delta H_l)$ is larger). We can apply partial randomization by keeping the L_D largest terms in the deterministic part. The remainder term is now not given as a linear combination of Paulis, but we can easily re-express the remainder terms $\sum_{l>L_D} H_l$ in terms of Pauli operators using some choice of orbital basis. In other words, we may use the partial randomization to truncate the rank L of the first factorization.

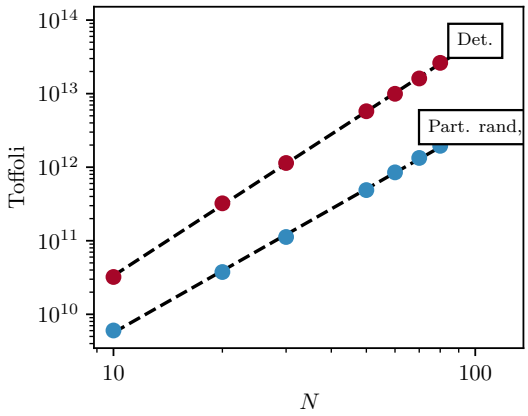


Figure 2.9: Toffoli gate counts for phase estimation of the hydrogen chain with N atoms using a factorized representation of the Hamiltonian.

One of the main bottlenecks of the factorization approach to Trotterization is the cost of implenting basis changes on the quantum computer, which scales quadratic in the number of orbitals. This cost can be reduced if the diagonal Hamiltonian $\sum_{p\sigma} \lambda_p^{(l)} n_{p\sigma}$ has only $\rho_l < N$ nonzero terms. In that case, the basis transformation can be arbitrary on the remaining $N - \rho_l$ orbitals. Such basis change unitaries can be implemented using $\mathcal{O}(N\rho_l)$ gates [123]. It turns out that for large systems, for each l, the λ_p can be truncated to ρ_l terms, where ρ_l is significantly smaller than N. Truncating small λ_p leads to constant ρ_l for large N; this effect, however, kicks in at rather large N [125, 124]. We can again use partial randomization to truncate the second factorization. Indeed, in this case for the diagonal Hamiltonian $V^{(l)}$ we only keep the terms for a subset of ρ_l orbitals for which $\lambda_p^{(l)}$ is relatively large. The remainder is assigned to the randomized part H_R. In Fig. 2.9 we show that this leads to a significant improvement for the hydrogen chain. Additionally, it resolves the issue that for the deterministic approach it is ambiguous what are good truncation thresholds which do not perturb the ground state energy too much. We note that for the hydrogen chain, the overall gate count using the Pauli representation, as shown in Fig. 2.2 is lower.

As a final comment, we note that for the randomized methods we assume the terms in the Hamiltonian are Pauli operators. However, for the derivations the only relevant property is that $P_l^2 = I$. This means that one can also use a Hamiltonian of the form $H = \sum_m h_m \tilde{P}_m$ where each $\tilde{P}_m$ is a Pauli with respect to a different choice of orbital basis. For example, one obtains such representations from the double factorization representation of the Hamiltonian. In that case, $V^{(l)}$ is a sum of terms proportional to $n_{p\sigma} n_{q\tau}$, which under the Jordan-Wigner mapping

can be expanded as Pauli operators acting on only two qubits. Therefore, the Hamiltonian can be expressed as a sum of terms of the form $\tilde{P}_m = (U^{(l)})^\dagger P_m U^{(l)}$, where $U^{(l)}$ is an orbital basis change unitary and P_m is a two-qubit Pauli operator. The cost of implementing $\exp\left(-i\varphi\tilde{P}_m\right) = (U^{(l)})^\dagger \exp(-i\varphi P_m)U^{(l)}$ is dominated by the cost of implementing the basis change, and since P_m only acts on two qubits, this cost is $\mathcal{O}(N)$. So, in cases where such a representation drastically reduces the weight λ of H, this may be a beneficial approach for randomized product formulas. Let λ be the weight of H in a Pauli representation, and let λ_{DF} be the weight in a double factorization representation. Since the cost scales quadratically in the value of λ, we have an improvement if $\lambda/\lambda_{\mathrm{DF}} = o(\sqrt{N})$. As an example, [46] reports for an arrangement for 4 hydrogen atoms with an increasing number of orbitals (as a toy model for the continuum limit), scalings of $\lambda = \mathcal{O}(N^{3.1})$ for the Pauli representation and $\lambda_{\mathrm{DF}} = \mathcal{O}(N^{2.3})$, so this approach would asymptotically improve the performance of random product formulas.

2.8 Conclusion

We performed a comprehensive analysis of quantum phase estimation using the Hadamard test and time evolution by product formulas for deterministic and (partially) randomized product formulas. Deterministic product formulas are conceptually simple and their scaling with the approximation error ε is close to optimal when using higher order methods. The main downside is their scaling with the number of terms in the Hamiltonian L, and a potentially complicated dependence on the Trotter error. Random product formulas have no dependence on the number terms, but rather on the weight λ. However, the scaling with ε^{-1} and λ is quadratic, which is suboptimal. In this work, we analyze phase estimation and Hamiltonian simulation in an integrated way. In particular, this allows us to obtain a sharper analysis of phase estimation for random product formulas. We also prove that the maximal depth can be reduced from ε^{-2} to $(1-\eta)^2\varepsilon^{-2}$ in regimes with large ground state overlap. These improvements do not increase the total number of gates, in contrast to other methods (which have better scaling with ε).

Our other main contribution is a modification of the partially randomized scheme of [89] and an analysis of its use in single-ancilla phase estimation. This approach combines advantages of deterministic and randomized product formulas. We demonstrated that this can lead to an improvement in the required resources for ground state energy estimation.

For a wide variety for benchmark molecular systems we compare the cost of different product formula Hamiltonian simulation methods for phase estimation. For deterministic product formulas, it is well known that (computable) rigorous

upper bounds can heavily overestimate the Trotter error, leading to much shorter Trotter steps (and hence deeper circuits) than necessary. We numerically analyze the empirical Trotter error, and show that while this gives a large improvement in the required resources, random product formulas can still outperform deterministic product formulas. Partial randomization can lead to a further reduction in cost.

Partially randomized product formulas for electronic structure Hamiltonians allow a large degree of freedom in the algorithm design. In particular, one can further optimize the representation of H by choosing different orbital bases for the deterministic and randomized Hamiltonians, or use the structure of the Coulomb tensor to optimize the cost. We think this is a promising direction for future explorations.

In this work we are counting both non-Clifford gates as well as two-qubit gates. It is unclear whether in practice the non-Clifford gates are necessarily the dominant cost in fault-tolerant quantum computation [126, 119]. In particular, it appears that magic state distillation may be cheaper for logical error rates around 10^{-9} [119], which means that trade-offs which reduce circuit depth may be extra useful. Product formulas (using many very small angle single-qubit rotations) are also a promising candidate for approaches which are only partially fault-tolerant [127]. The quantum circuits deriving from product formulas have the advantage of being structurally simple (they consist only of Pauli rotations if the Hamiltonian is a linear combination of Paulis), so it is straightforward to count or optimize other cost measures.

Together with various other developments in the algorithmic theory of product formulas, such as multiproduct formulas and polynomial interpolation techniques [128, 129, 130, 131], isometries to enlarged basis sets [132] this suggests that both for intermediate-term applications and fully fault-tolerant quantum computation, (partially randomized) product formulas remain a promising approach to ground state energy phase estimation.

Acknowledgements

We thank Li Liu for helpful discussions. We acknowledge funding from the Research Project “Molecular Recognition from Quantum Computing”. Work on ”Molecular Recognition from Quantum Computing” was supported by Wellcome Leap as part of the Quantum for Bio (Q4Bio) Program. We also acknowledge financial support from the Novo Nordisk Foundation (Grant No. NNF20OC0059939 ‘Quantum for Life’).

2.9 Appendices

2.9.1 Hadamard test using (partially) randomized time evolution

In this appendix we derive in detail the result of performing a Hadamard test on (partially) randomized product formulas. We briefly recall the setup and notation. The Hamiltonian is given as

$$H = \sum_l H_l$$

where for the randomized product formulas we will assume more specifically that

$$H = \sum_l h_l P_l$$

where the P_l are Pauli operators and $h_l \geq 0$ (using that signs can be absorbed in the Pauli operators). In fact, the only properties used in the derivations are the possibility to implement time evolution along P_l, and $P_l^2 = I$. We write $V(\varphi) = \exp(-i\varphi P_l)$. We may rescale the Hamiltonian by a factor

$$\lambda = \sum_l h_l \qquad \text{so} \qquad H = \lambda \sum_l p_l H_l$$

where the p_l form a probability distribution p. We may rescale time by a factor λ, and thereby assume that $H = \sum_l p_l P_l$ where the p_l form a probability distribution p. In the end, when determining the computational cost we have to account for the rescaling by rescaling the relevant parameters by λ. The Hamiltonian has energies E_k (which lie in $[-1, 1]$ after rescaling with λ) with eigenprojectors Π_k. We apply the Hadamard test with initial state $|\psi\rangle$, which has squared overlap $c_k = \langle\psi| \Pi_k |\psi\rangle$ wih the eigenspace of energy E_k. If E_k is nondegenerate, $\Pi_k = |\psi_k\rangle\langle\psi_k|$ and $c_k = |\langle\psi|\psi_k|\psi|\psi_k\rangle|^2$.

Hadamard test using qDRIFT

The qDRIFT method is to construct a unitary

$$U(\tau, r) = \prod_{i=1}^{r} \exp(-i\varphi H_{l_r})$$

where l_i is sampled according to p and φ depends on a choice of time step τ. This gives a random unitary, but it turns out that for appropriately chosen r and φ the

channel obtained by this process is close in diamond norm to the time evolution unitary $U(t) = \exp(-iHt)$ [55, 114]. This may then be used in combination with phase estimation to compute ground state energies.

In this work we analyze in detail the performance of phase estimation methods based on the Hadamard test using qDRIFT. The analysis is very similar to that of [86], which also uses Hadamard tests together with a randomized method for time evolution. The main difference is that in their approach the randomized time evolution method is based on a Taylor series expansion of the time evolution operator.

The fact that P_l squares to identity implies that for any $\varphi \in \mathbb{R}$ the operator

$$V_l(\varphi) := \cos(\varphi) I - i \sin(\varphi) P_l = \exp(-i\varphi P_l)$$

is unitary. We now choose $\tau \in [0, 1]$ and let $\varphi = \arctan(\tau)$. Let $\boldsymbol{X}(\tau, r)$ and $\boldsymbol{Y}(\tau, r)$ be the ± 1-valued random variables resulting from the following process:

1. Sample $l_1, \ldots, l_r$ from p.
2. Apply a Hadamard test to the unitary

$$U = V_{l_r}(\varphi) \cdots V_{l_1}(\varphi)$$

Let $\boldsymbol{Z}(\tau, r) = \boldsymbol{X}(\tau, r) + i\boldsymbol{Y}(\tau, r)$. We may now compute the expectation value of $\boldsymbol{Z}(\tau, r)$, which we denote by $g(\tau, r)$ to be the following signal

$$\begin{aligned}
g(\tau, r) :&= \mathbb{E}\boldsymbol{Z}(\tau, r) \\
&= \sum_{l_1, \ldots, l_r} p_{l_1} \ldots p_{l_r} \langle\psi| V_{l_r}(\varphi) \cdots V_{l_1}(\varphi) |\psi\rangle \\
&= \sum_{l_1, \ldots, l_r} p_{l_1} \ldots p_{l_r} \langle\psi| \frac{I - i\tau P_{l_r}}{\sqrt{1+\tau^2}} \cdots \frac{I - i\tau P_{l_1}}{\sqrt{1+\tau^2}} |\psi\rangle \\
&= (1+\tau^2)^{-r/2} \langle\psi| (I - i\tau \sum_{l_r} p_{l_r} P_{l_r}) \cdots (I - i\tau \sum_{l_1} p_{l_1} P_{l_1}) |\psi\rangle \\
&= (1+\tau^2)^{-r/2} \langle\psi| (I - i\tau \sum_{l} p_l P_l)^r |\psi\rangle \\
&= (1+\tau^2)^{-r/2} \langle\psi| (I - i\tau H)^r |\psi\rangle
\end{aligned}$$

which encodes the spectrum of the (normalized) Hamiltonian. We summarize this conclusion in the following lemma.

Lemma 2.9.1. *Let*

$$H = \sum_l p_l P_l$$

where $P_l^2 = I$ and the p_l are a probability distribution. Then the random variables $\boldsymbol{Z}$ from the measurement outcomes of a Hadamard test using qDRIFT with r Pauli rotations and step size τ has expectation value

$$\mathbb{E}\boldsymbol{Z}(\tau, r) = g(\tau, r) = (1+\tau^2)^{-r/2} \langle\psi| (I - i\tau H)^r |\psi\rangle .$$

This can be expressed in terms of the eigenstate overlaps and energies as

$$\begin{aligned} g(\tau, r) &= (1+\tau^2)^{-r/2} \sum_k c_k (1 - i\tau E_k)^r \\ &= \sum_k c_k \underbrace{\left(\frac{1+E_k^2\tau^2}{1+\tau^2}\right)^{\frac{r}{2}}}_{\geq \exp\left(-\frac{1}{2}(1-E_k^2)r\tau^2\right)} \exp[-ir \arctan(\tau E_k)] \end{aligned} \tag{2.18}$$

The goal is to extract the value of E_0 from this signal, choosing appropriate values of τ and r. Recall the signal obtained in Eq. (2.3) where we perform time evolution for time t. If we choose $r = \lceil t^2 \rceil$ and $\tau = t/r \approx t^{-1}$ we find that

$$g(\tau, r) = \sum_k d_k e^{-itF_k}$$

with

$$F_k = \tau^{-1} \arctan(\tau E_k) \approx tE_k, \qquad c_k \exp\left(-\frac{1}{2}(1-E_k^2)\right) \leq d_k \leq c_k \tag{2.19}$$

which approximates $g(t)$, with the difference that the amplitudes are modified by a factor at most $\sqrt{e}$.

Randomized Taylor expansion

In this section we review a refinement of qDRIFT due to Berta, Campbell and Wan [86], which we will call the *Randomized Taylor Expansion* (RTE) method. It proceeds in very similar fashion to qDRIFT, and in fact the above qDRIFT algorithm corresponds exactly to keeping only the first order term in the Taylor

expansion. We write a Taylor series expansion for $\exp(-i\tau)$ from which we get

$$\begin{aligned}
\exp(-i\tau H) &= \sum_{n=0}^{\infty} \frac{(-i\tau)^n}{n!} \left(\sum_l p_l P_l\right)^n \\
&= \sum_{n\,\text{even}} \frac{(-1)^{n/2}}{n!}\tau^n \left(I - \frac{i\tau}{n}\sum_l p_l P_l\right)\left(\sum_l p_l P_l\right)^n \\
&= \sum_{n\,\text{even}} \frac{(-1)^{n/2}}{n!}\tau^n \sum_l p_l \left(I - \frac{i\tau}{n+1} P_l\right)\left(\sum_l p_l P_l\right)^n \\
&= \sum_{n\,\text{even}} \frac{(-1)^{n/2}}{n!}\tau^n \sqrt{1+\frac{\tau^2}{(n+1)^2}} \sum_l p_l V_l(\varphi_n)\left(\sum_l p_l P_l\right)^n
\end{aligned}$$

where $\varphi_n = -\arctan(\tau/(n+1))$. This gives a decomposition

$$\exp(-i\tau H) = \sum_m c_m U_m$$

where m is a label consisting of an even integer n and $l, l_1, \ldots, l_n$

$$\begin{aligned}
c_m &= \frac{1}{n!}\tau^n \sqrt{1+\frac{\tau^2}{(n+1)^2}} p_l p_{l_1} \ldots p_{l_n} \geq 0 \\
U_m &= (-1)^{n/2} V_l(\varphi_n) P_{l_n} \cdots P_{l_1}.
\end{aligned} \tag{2.20}$$

Note that in principle, n can be arbitrary, but one can truncate to finite n at small error, see [86]. Again, for simulation of time t, we can break up in r steps, and for $\tau = t/r$

$$\begin{aligned}
\exp(-i\tau H)^r &= \sum_{m_1,\ldots,m_r} c_{m_1} \ldots c_{m_r} U_{m_r} \cdots U_{m_1} \\
&= \sum b_k W_k
\end{aligned}$$

where k is a label consisting of $m_1, \ldots, m_r$ and $b_k = c_{m_1} \ldots c_{m_r}$, and the unitary W_k is given by $U_{m_r} \cdots U_{m_1}$. If we now let

$$B = \sum_k b_k \qquad q_k = \frac{b_k}{B} \tag{2.21}$$

then sampling according to q_k and applying a Hadamard test to W_k with initial state $|\psi\rangle$ gives random variables $\boldsymbol{X}(\tau, r)$ and $\boldsymbol{Y}(\tau, r)$ such that for $\boldsymbol{Z}(\tau, r) = \boldsymbol{X}(\tau, r) + i\boldsymbol{Y}(\tau, r)$

$$\mathbb{E}\boldsymbol{Z}(\tau, r) = \frac{1}{B} \langle\psi| \exp(-itH) |\psi\rangle .$$

This means that we can estimate the value of

$$g(\tau r) = g(t) = \langle\psi| \exp(-itH) |\psi\rangle$$

to precision γ by estimating the expectation value of $\boldsymbol{Z}(\tau, r)$ to precision γ/B. In other words, if we want to compute $g(\tau, r)$ using this method we can do so at a sampling overhead of B^2/γ^2. One can bound [55]

$$B = \sum_k b_k = \left(\sum_m c_m\right)^r \leq \exp\bigl(\tau^2 r\bigr) \tag{2.22}$$

This means that one can estimate $g(t)$ with constant sampling overhead by taking the same choice of parameters as for qDRIFT with depth $r = \lceil t^2 \rceil$ and step size $\tau = t/r \approx t^{-1}$.

Given a sample $k = (m_1, \ldots, m_r)$ the resulting unitary W_k consists of the composition of r unitaries U_{m_i}, which itself consists of a single Pauli rotation together with a product of Pauli operators (which makes up a new Pauli operator), so we get a circuit with r Pauli rotations.

Lemma 2.9.2. *Let*

$$H = \sum_l p_l P_l$$

where $P_l^2 = I$ and the p_l are a probability distribution. Then the random variable $\boldsymbol{Z}$ resulting from Hadamard tests on RTE with r steps and step size τ has expectation value

$$\mathbb{E}\, B\boldsymbol{Z}(\tau, r) = g(\tau r) = \langle\psi| \exp(-i\tau r H) |\psi\rangle\,.$$

Here $B \leq \exp(\tau^2 r)$.

We now briefly comment on the difference between RTE and qDRIFT. First of all, if one chooses τ and r with $r = \Omega(t^2)$, they are very similar. At every step, the probability of sampling order $n \geq 2$ in RTE scales is $\mathcal{O}(\tau^2)$, so the expected number of steps where you sample $n \geq 2$ is $\mathcal{O}(r\tau^2) = \mathcal{O}(1)$ out of $r = \mathcal{O}(t^2)$ steps. The main difference is that RTE, with a Hadamard test, gives an *unbiased* estimator of the signal $g(t)$. This is convenient for analysis. On the other hand, we have also computed a simple expression for the exact signal resulting from qDRIFT, from which one can extract the same information about the energies. For qDRIFT, the damping depends on the energy as well; if E_0 is close to its minimal value of -1 (so the Hamiltonian is close to frustration free) this means that the damping factor is smaller.

Partially randomized product formulas

In Section 2.5 we have discussed a partially randomized scheme, which treats certain terms in the Hamiltonian deterministically, and the remainder randomly, and here we explain and analyze this scheme in more detail. The error analysis is particularly simple, where the signal from the partially randomized product formula has a bias from the deterministic decomposition, and a decreased amplitude due to the randomization. This intuition is illustrated in Fig. 2.10. We now recall the set-up. We write

$$H = \underbrace{\sum_{l=1}^{L_D} H_l}_{=H_D} + \underbrace{\sum_{m=1}^{M} h_m P_m}_{=H_R}, \qquad \lambda_R = \sum_m |h_m|. \tag{2.23}$$

Here, the P_m should square to identity, but the H_l need not. As before, we let

$$H_R = \lambda_R \sum_m p_m P_m$$

where the p_m form a probability distribution (and we have absorbed signs into the P_m).

We let $S_p(\delta)$ be a deterministic product formula for H, written as a sum of the H_l and treating H_R as a single term. This means we can write

$$S_p(\delta) = U_I \cdots U_1$$

where each U_i is either of the form $\exp(-\delta_i H_l)$ for some real δ_i, or $U_i = \exp(-\delta_i H_R)$. Here the number of terms is $I \leq (L_D + 1)N_{\text{stage}}$, where N_{stage} is the number of stages of the product formula. For the standard Trotter formula of order p, we have $N_{\text{stage}} = 2 \cdot 5^{k-1}$. For the second order Suzuki-Trotter formula, we get

$$S_2(\delta) = e^{-\frac{1}{2}i\delta H_1} \cdots e^{-\frac{1}{2}i\delta H_{L_D}} e^{-i\delta H_R} e^{-\frac{1}{2}i\delta H_{L_D}} \cdots e^{-\frac{1}{2}i\delta H_1}.$$

Let $U_{i_1}, \ldots, U_{i_N}$ for $N \leq N_{\text{stage}}$ denote the unitaries where we time evolve along H_R, for times $\delta_{i_1}, \ldots, \delta_{i_N}$. We have $\delta_{i_1} + \cdots + \delta_{i_N} = \delta$, and $\tilde{\delta} := |\delta_{i_1}| + \cdots + |\delta_{i_N}| = \mathcal{O}(\delta)$ (higher order methods require evolving backwards in time as well). Next, we will apply RTE to each of the U_{i_p}. To this end, we choose a number of steps r_{i_p} to simulate $\exp\bigl(-\delta_{i_p} H_R\bigr)$. We may write

$$\begin{aligned} U_{i_p} &= \exp\bigl(-i\delta_{i_p} H_R\bigr) = B_{i_p} \sum_k q_{i_p,k} W_{i_p,k} \\ B_{i_p} &\leq \exp\Bigl(-\delta_{i_p}^2 \lambda_R^2 / r_{i_p}\Bigr) \end{aligned} \tag{2.24}$$

as in Section 2.9.1, where each $W_{i_p,k}$ is a product of Paulis P_m and r_{i_p} Pauli rotations. Expanding each U_{i_p} in this fashion, and taking the s-th power of the result, gives a linear combination of unitaries

$$S_p(\delta)^s = B \sum_j q_j W_j$$

where

$$B = (B_{i_1} \dots B_{i_N})^s \leq \exp\left(s \sum_p \delta_{i_p}^2 \lambda_R^2 / r_{i_p} \right). \tag{2.25}$$

Each W_j consists of $s(r_{i_1} + \dots + r_{i_N})$ Pauli rotations from the expansion of the evolution along H_R, and at most $N_{\text{stage}} L_D s$ applications of $\exp(-\delta H_l)$ for some values of δ. Now, given δ and s, in order to keep the sample overhead constant, we choose $r_{i_p} = \lceil \lambda_R^2 \big| \delta_{i_p} \big| \tilde{\delta} s \rceil$, which gives

$$s \sum_p \delta_{i_p}^2 \lambda_R^2 / r_{i_p} \leq 1.$$

The total number of Pauli rotations applied in the randomized part of the resulting circuit is

$$r = s \sum_p r_{i_p} = s \sum_p \lceil \lambda_R^2 \delta_{i_p} \delta s \rceil \leq \lambda_R^2 \tilde{\delta}^2 s^2 + s N_{\text{stage}} = \mathcal{O}(\lambda_R^2 \delta^2 s^2).$$

The term sN_{stage} comes from having to round the values of the r_{i_p} to integers and is a small subleading contribution. We summarize this result as follows.

Lemma 2.9.3. *Let H be given as in Eq. (2.23) and let $S_p(\delta)$ be a deterministic product formula for H, written as a sum of the H_l and treating H_R as a single term. Then the random variables $\boldsymbol{Z}$ which results from the measurement outcomes of a Hadamard test, using the partially randomized product formula described above, has expectation value*

$$\mathbb{E}\, B \boldsymbol{Z}(\delta, s) = \langle \psi | \, S_p(\delta)^s \, | \psi \rangle \, .$$

Here $B = \mathcal{O}(1)$, and the circuit consists of at most $N_{\text{stage}} L_D s$ time evolution unitaries along one of the H_l, and $r = \mathcal{O}(\lambda_R^2 \delta^2 s^2)$ time evolution unitaries along one of the P_m.

We note that the second order Trotter-Suzuki formula has the benefit in this context that it only evolves in the positive time direction, so we have $\tilde{\delta} = \delta$. The conclusion of Lemma 2.9.3 is that from the Hadamard test with the partially

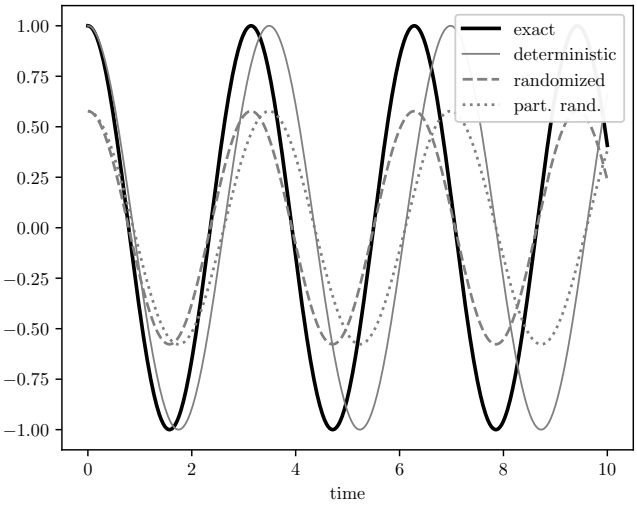


Figure 2.10: Intuition for the error analysis: deterministic product formulas give a discrete approximation which has a bias in the frequency, randomized product formulas give a damping of the amplitude; the partially randomized method has both.

randomized product formula, we get a signal which, up to a damping, implements the deterministic product formula which treats H_R as a single term. The resulting Trotter error can be bounded using any standard technique (such as commutator bounds).

Different schemes for composite random and deterministic product formulas have been proposed [87, 89, 133], treating a relatively small number of important terms using deterministic Trotterization and a tail of terms with small weight randomly. The scheme described above is the same as in [89], where we have replaced qDRIFT by RTE. Compared to the diamond norm bounds of [89], the randomized part of the product formula does not introduce any error. Additionally, in our analysis it is clear that if we are only intrested in ground state energies, $S_p(\delta)$ only needs to be accurate on the ground state, a significantly weaker requirement on the Trotter error. This allows for a sharper analysis of the performance of partially randomized product formulas for single-ancilla phase estimation.

A final comment is that a general potential disadvantage of (partially) randomized product formulas is that in principle for each Hadamard test one has to sample a new random circuit. In practice, it may be more convenient (e.g. for compilation purposes) to repeat the same circuit multiple times. For randomized product formulas [114] as well as partially randomized product formulas [98] it has been shown that this is in fact possible. These works show that with increasing circuit depth, the expectation value of observables (such as Hadamard test outcomes) rapidly concentrate with respect to the randomness of the sampled

circuit.

Halving the required time evolution

There is a basic trick to reduce the maximum required evolution time when using the Hadamard test for time evolution [40]. This assumes we have a symmetric Trotter formula (meaning that $S_p(-t) = S_p(t)^\dagger$), which is the case for the even order Suzuki-Trotter formulas. We write out the Trotter unitary in terms of the inidividual Hamiltonian time evolutions

$$S_p(\delta) = U_I(\delta_I) \cdots U_1(\delta_1)$$

where each unitary is time evolution along a local term H_l of the Hamiltonian. Now, for some arbitary unitary U, let $C_\pm U$ denote a controlled version

$$C_\pm U = |0\rangle\langle 0| \otimes U^\dagger + |1\rangle\langle 1| \otimes U$$

so the product formula, we have

$$C_\pm U_i(\delta_i) = |0\rangle\langle 0| \otimes e^{i\delta_i H_{l_i}} + |1\rangle\langle 1| \otimes e^{-i\delta_i H_{l_i}}.$$

By the symmetry property of the product formula, we have

$$C_\pm S_p(\delta) = C_\pm U_I(\delta_I) \cdots C_\pm U_1(\delta_1).$$

It is now easy to see that applying the Hadamard test to the unitary $(C_\pm S_p(\delta))^s$ gives expectation value $\langle\psi| S_p(\delta)^{2s} |\psi\rangle$. This is implemented by the circuit in Fig. 2.11. If the $U_i(\delta_i)$ are Pauli rotations, then it requires only Cliffords and one single-qubit rotation to implement $C_\pm U_i(\delta_i)$. Therefore, this trick reduces the required number of single-qubit rotations by a factor of 2.

This trick does *not* apply for randomized product formulas (which do not have the right symmetry property). However, it can be modified to apply for partially randomized product formulas. In that case, one proceeds as above, thinking of H as consisting of $H_1, \ldots, H_{L_D}$ and H_R as in Eq. (2.23). This gives a circuit with controlled applications of unitaries of the form $U = \exp(\pm i\delta H_R)$. Finally, we expand

$$C_\pm U = (|0\rangle\langle 0| \otimes U^\dagger + |1\rangle\langle 1| \otimes I)(|0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes U)$$

and separately apply RTE to U and $U^\dagger$ (with independent randomness). Altogether, this halves the gate count for the deterministic part of the partially randomized product formula (but not for the randomized part).

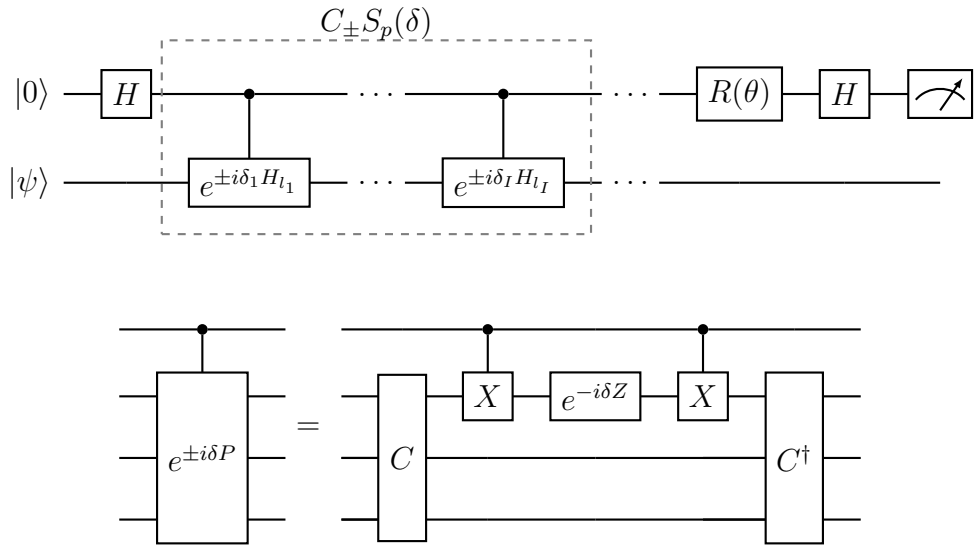


Figure 2.11: Performing the Hadamard test on $C_\pm S_p(\delta)^s$ allows one to estimate $\langle\psi| S_p(\delta)^{2s} |\psi\rangle$. If $U = e^{i\delta P}$ is a Pauli rotation, we can let C be a Clifford which is such that $CPC^\dagger$ acts as a Pauli Z on the first qubit, and use this to implement $C_\pm U$ with only one single-qubit rotation.

2.9.2 Robust phase estimation

As a phase estimation algorithm we consider *robust phase estimation* [107, 103, 26, 38]. This algorithm is particularly simple, and requires shorter depth if the guiding state is close to the ground state. Its main disadvantage is that it requires the ground state overlap η to be at least some constant.

The idea of robust phase estimation is to estimate the time evolution signal at times $t = 2^m$ for $m = 0, 1, \ldots, M$ with $M = \lceil \log \varepsilon^{-1} \rceil$. For each m the outcome of the Hadamard test $\boldsymbol{Z}(2^m)$ has expectation value

$$g(2^m) = \sum_k c_k \exp(-i2^m E_k).$$

We repeat N_m times and let $\overline{\boldsymbol{Z}}(2^m)$ denote the average. Given some outcome we compute an angle $\phi_m = \arg(\overline{Z}(2^m))$ Each m corresponds to one bit of precision for the estimate of E_0, i.e. ϕ_m is an approximation of $2^m E_0$ modulo 2π. The set of estimates of E_0 compatible with ϕ_m is $\{2^{-m}(2\pi k + \phi_m) : k = 0, \ldots, 2^m - 1\}$. Given a guess θ_{m-1} for E_0 after $m-1$ rounds, we let θ_m be given by

$$\theta_m = 2^{-m} \left(2\pi k + \phi_m\right))$$

for the integer $k = 0, \ldots, 2^m - 1$ such that the angles θ_{m-1} an θ_m are as close as possible. For two angles θ, ϕ we let $d(\theta, \phi)$ denote the distance between the

angles, so

$$d(\theta, \phi) = \min_{k \in \mathbb{Z}} |\theta - \phi + 2k\pi|.$$

The robust phase estimation algorithm is prescribed in detail in Algorithm 1. There, the angles ϕ_j are treated as input (and can for example be obtained by sampling Hadamard test measurement outcomes).

Algorithm 1: Robust phase estimation

Input: A sequence of angles ϕ_m, $m = 0, 1, \ldots, M$.
Output: A phase estimate θ.

$\theta_{-1} \leftarrow 0$
for $m = 0, 1, \ldots, M$ **do**
  $S_m = \{2^{-m}(2\pi k + \phi_m) : k = 0, \ldots, 2^m - 1\}$
  $\theta_m \leftarrow \arg\min_{\theta \in S_m} d(\theta, \theta_{m-1})$, $-\pi < \theta_m \leq \pi$
end for
$\theta \leftarrow \theta_M$

Rigorous bounds for robust phase estimation

We will now review the argument for the scaling of robust phase estimation, and argue for what happens when using randomized simulation methods. The algorithm gives an accurate estimate of a target E if the estimate ϕ_m in round m is withing distance at most $\frac{\pi}{3}$ of $2^m E$. If this is not the case, we will say that there is an error in round m. The Heisenberg scaling of robust phase estimation is achieved by making sure that the probability of error in early rounds is sufficiently small. The following results are derived in [107, 103, 26] and are the key to proving Heisenberg limited scaling as well as robustness of the phase estimation algorithm.

Lemma 2.9.4. *Let $\xi \leq 1$.*

1. *If in Algorithm 1, $d(\phi_m, 2^m E) < \frac{\pi}{3}$ for $m = 0, 1, \ldots, m$, then θ_m in Algorithm 1 is such that $d(\theta_m, E) \leq 2^{-m}\frac{\pi}{3}$. If $d(\phi_m, 2^m E) < \frac{\pi}{3}$ for $m = 0, \ldots, M-1$ and $d(\phi_M, 2^M E) \leq \xi$, then $d(\theta, E) \leq 2^{-M}\xi$.*

2. *Suppose that $\{\boldsymbol{\phi}_m\}_{m=0}^M$ are independent real random variables. such that for $m = 1, \ldots, M$ we have*

$$\mathbb{P}(d(\boldsymbol{\phi}_m, 2^m E) \geq \frac{\pi}{3}) \leq \xi^2 4^{-\alpha(M-m)}$$

for some $\alpha > 1$, and

$$\mathbb{E}\, d(\boldsymbol{\phi}_M, 2^M E)^2 \leq \xi^2.$$

Then applying Algorithm 1 to $\{\boldsymbol{\phi}_m\}_{m=0}^M$ gives an estimate θ of E with

$$\mathbb{E}\, d(\theta, E)^2 \leq C\xi^2 4^{-M}$$

Proof. The argument for the first claim is by induction. For $m = 0$, the result holds by assumption. Suppose that $d(\phi_m, 2^m E) < \frac{\pi}{3}$ and $d(\theta_{m-1}, E) \leq 2^{-m+1}\frac{\pi}{3}$. Then the candidate set S_m contains an estimate which is $(2^{-m}\frac{\pi}{3})$-close to E given by $2^{-m}(2\pi k + \phi_m)$. Let k' be the integer such that $\theta_m = 2^{-m}(2\pi k' + \phi_m)$, then we would like to show that $k = k'$. The distance between candidates in S_m is at least $2^{-m+1}\pi$, so it suffices to show that $d(\theta_{m-1}, 2^{-m}(2\pi k + \phi_m)) < 2^{-m}\pi$. We use the triangle inequality, and the induction hypothesis that $d(\theta_{m-1}, E) \leq 2^{-m+1}\frac{\pi}{3}$ to confirm this:

$$\begin{aligned} d(\theta_{m-1}, 2^{-m}(2\pi k + \phi_m)) &\leq d(\theta_{m-1}, E) + d(E, 2^{-m}(2\pi k + \phi_m)) \\ &< 2^{-m+1}\frac{\pi}{3} + 2^{-m}\frac{\pi}{3} = 2^{-m}\pi. \end{aligned}$$

In the last round, if $d(\phi_M, 2^M E) \leq \xi$, then $d(\theta, E) \leq 2^{-M}\xi$. Next, if for $l \leq m$ we have $d(\phi_l, 2^l E) < \frac{\pi}{3}$ (but possibly not for $l > m$), then the output $\theta = \theta_M$ satisfies $d(\theta_M, E) \leq 2^{-m+3}\frac{\pi}{3}$. This follows from the observation that the algorithm is such that $d(\theta_l, \theta_{l+1}) \leq 2^{-l}\pi$, so

$$d(\theta_M, E) \leq d(\theta_m, E) + \sum_{l=m}^{M-1} d(\theta_l, \theta_{l+1}) \leq 2^{-m}\frac{\pi}{3} + \underbrace{\sum_{l=m}^{M-1} 2^{-l}\pi}_{\leq 2^{-m+1}\pi} \leq 2^{-m+3}\frac{\pi}{3}.$$

We now consider the case where the angles $\boldsymbol{\phi_m}$ are random variables. Let p_m denote the probability that $m = 0, \ldots, M-1$ is the smallest value for which we have an error (so $d(\phi_m, 2^m E) \geq \frac{\pi}{3}$), and let p_M denote the probability there are no errors for $m = 0, \ldots, M-1$. Then we can bound the mean square error

$$\mathbb{E}d(\theta_M, E)^2 \leq \sum_{m=0}^{M-1} p_m \left(\frac{8\pi}{3 \cdot 2^{m-1}}\right)^2 + p_M 4^{-M}\mathbb{E}\, d(\boldsymbol{\phi}_M, 2^M E)^2.$$

We may bound p_m by the probability of having an error in round m, so $p_m \leq \xi^2 4^{-\alpha(M-m)}$, and $p_M \leq 1$ which gives

$$\begin{aligned}\mathbb{E}\, d(\theta_M, E)^2 &\leq \sum_{m=0}^{M-1} \xi^2 4^{-\alpha(M-m)} \left(\frac{8\pi}{3\cdot 2^{m-1}}\right)^2 + \xi^2 4^{-M} \\ &= \left(1 + \left(\frac{16\pi}{3}\right)^2 \sum_{m=0}^{M-1} 4^{-(\alpha-1)(M-m)}\right) \xi^2 4^{-M} \\ &\leq \left(1 + \frac{1}{4^{\alpha-1}-1}\left(\frac{16\pi}{3}\right)^2\right) \xi^2 4^{-M} = C\xi^2 4^{-M}\end{aligned}$$

for a constant C which only depends on α. □

In Line 6 we will give numerical estimates for the relevant constant factors. The angles ϕ_m are obtained from a Hadamard test, simulating time evolution for time 2^m. We will assume that we have a guiding state $|\psi\rangle$ with sufficiently high ground state overlap. We first repeat the following fact from [38]:

Suppose $Z \in \mathbb{C}$, $d_k \geq 0$ and $E_k \in \mathbb{R}$. If

$$\left|Z - \sum_k d_k \exp(iE_k)\right| \leq \beta, \qquad \frac{\beta + \sum_{k\geq 1} d_k}{d_0} \leq 1 \tag{2.26}$$

then we can bound

$$|Z - d_0 \exp(iE_0)| \leq \beta + \sum_{k\geq 1} d_k.$$

which implies that the sine of the angle between Z and $d_0 \exp(iE_0)$ satisfies $|\sin(d(E_0, \arg(Z)))| \leq (\beta + \sum_{k\geq 1} d_k)/d_0$ and hence

$$d(E_0, \arg(Z)) \leq \arcsin\left(\frac{\beta + \sum_{k\geq 1} d_k}{d_0}\right). \tag{2.27}$$

We run robust phase estimation on a signal estimated by a Hadamard test, and estimate the energy E_0 based on the outcome $\theta = \theta_M$. We consider two variants. For the randomized methods, we assume we have a normalized Hamiltonian $H = \sum_l p_l P_l$ (in general that means we have to rescale ε with λ).

- We estimate ϕ_j from applying a Hadamard test to e^{-iHt}, i.e. from the exact time evolution, for time $t = 2^m$. We use $2N_m$ samples to get an estimate Z_m of $g(2^m) = \langle\psi| e^{-iH2^m} |\psi\rangle$, and let $\phi_m = -\arg(Z_m)$. We estimate E_0 by θ_M. The total amount of time evolution we use is

$$t_{\text{tot}} = \sum_{m=0}^{M} 2N_m 2^m. \tag{2.28}$$

- We estimate ϕ_j from applying Hadamard tests using a randomized product formula (either qDRIFT or RTE) for time evolution $t = 2^m$ with $r_m = \Omega(2^{2m})$ Pauli rotations, and time step $\tau_m = t/r_m$. We use $2N_m$ samples to get an estimate Z_m of of the signal, and let $\phi_m = -\arg(Z_m)$. For RTE, we estimate E_0 by θ_M, and in the case of qDRIFT, we estimate E_0 by $\tau_M^{-1}\tan(\tau_M\theta_M)$. In both cases, the total number of Pauli rotations used is

$$r_{\text{tot}} = \sum_{m=0}^{M} 2N_m r_m. \tag{2.29}$$

The argument for the exact time evolution model in the following result follows [107, 103, 26] for 1 and [38] for 3. A difference compared to [38] is that for 2 we bound the mean square error, instead of bounding a success probability for an error guarantee. The application to randomized product formulas in 2 and 4 is a straightforward consequence of the analysis.

Theorem 2.9.5. *Suppose that we have a guiding state $|\psi\rangle$ with $c_0 \geq \eta > 4 - 2\sqrt{3} \approx 0.53$.*

1. *In the exact time evolution model, we obtain an estimate with root mean square error ε using $t_{\max} = \mathcal{O}(\frac{1}{\varepsilon})$, and $t_{\text{tot}} = \mathcal{O}(\frac{1}{\varepsilon})$.*

2. *When using RTE or qDRIFT, we obtain an estimate with root mean square error ε using maximal number of rotations per circuit $r_{\max} = \mathcal{O}(\frac{1}{\varepsilon^2})$ and total number of rotations $r_{\text{tot}} = \mathcal{O}(\frac{1}{\varepsilon^2})$.*

In the limit $\eta \to 1$, for $\xi > \arcsin\left(\frac{1-\eta}{\eta}\right)$, we can reduce the maximal depth.

3. *In the exact time evolution model, we obtain an estimate with root mean square error ε using $t_{\max} = \mathcal{O}(\frac{\xi}{\varepsilon})$, and $t_{\text{tot}} = \mathcal{O}(\frac{1}{\xi\varepsilon})$.*

4. *When using RTE or qDRIFT, we obtain an estimate with root mean square error ε using maximal number of rotations per circuit $r_{\max} = \mathcal{O}(\frac{\xi^2}{\varepsilon^2})$ and total number of rotations $r_{\text{tot}} = \mathcal{O}(\frac{1}{\varepsilon^2})$.*

Proof. We estimate X_m and Y_m using N_m Hadamard tests, and thus estimate $Z_m = X_m + iY_m$ using $2N_m$ Hadamard tests. For the case where we use a randomized product formula, we take $r_m = 2^{m+M}$. The expectation value of $\boldsymbol{Z}_m$ is given by

$$\mathbb{E}\boldsymbol{Z}_m = \sum_k d_k e^{-i2^m F_k}$$

where $F_k = E_k$ for the case of exact time evolution and RTE, and $F_k = 2^M \arctan\big(2^{-M}E_k\big)$ for qDRIFT. Note that if $|F_0 - \theta_M| \leq \varepsilon$, then

$$\begin{aligned}\big|E_0 - 2^M \tan\big(2^{-M}\theta_M\big)\big| &\leq 2^M \big|\tan\big(2^{-M}\theta_M\big) - \tan\big(2^{-M}F_0\big)\big| \\ &\leq \varepsilon + \mathcal{O}(2^{-2M}\varepsilon)\end{aligned}$$

so it suffices to get an accurate estimate for F_0. For qDRIFT, we assume for convenience that $|E_0| \geq |E_k|$ for $k \geq 1$ (which can always be achieved by shifting H with a constant factor).

For exact time evolution, $d_k = c_k$, for the randomized methods $e^{-1}c_k \leq d_k \leq c_k$ (while d_k also depends on m, this bound holds for all m). In particular, $d_0 \geq e^{-1}c_0 \geq e^{-1}\eta$. We let $0 < \beta = \eta(1+\sin\big(\frac{\pi}{3}\big)) - 1$ for the case with exact time evolution and $0 < \beta = e^{-\frac{1}{2}}(\eta(1+\sin\big(\frac{\pi}{3}\big)) - 1)$ for the randomized cases. This choice of β is such that Eq. (2.26) is satisfied

$$\arcsin\left(\frac{\beta + \sum_{k\geq 1} d_k}{d_0}\right) \leq \frac{\pi}{3}.$$

Now, by Eq. (2.27) and Hoeffding's inequality we see that for $\boldsymbol{\phi}_m = -\arg(\boldsymbol{Z}_m)$

$$\begin{aligned}\mathbb{P}\left(d(\boldsymbol{\phi}_m, F_0) \geq \frac{\pi}{3}\right) &\leq \mathbb{P}\left(|\boldsymbol{Z}_m - \mathbb{E}\boldsymbol{Z}_m| \geq \beta\right) \\ &\leq \mathbb{P}\left(|\boldsymbol{X}_m - \mathbb{E}\boldsymbol{X}_m| \geq \frac{\beta}{\sqrt{2}}\right) + \mathbb{P}\left(|\boldsymbol{Y}_m - \mathbb{E}\boldsymbol{Y}_m| \geq \frac{\beta}{\sqrt{2}}\right) \\ &\leq 4\exp\left(-\frac{N_m\beta^2}{4}\right).\end{aligned} \tag{2.30}$$

In particular, we find that by choosing $N_m = \beta^{-2}(2\log(\xi^{-1}) + \mathcal{O}(M-m))$ we can achieve

$$\mathbb{P}(d(\boldsymbol{\phi}_m, 2^m F_0) \geq \frac{\pi}{3}) \leq \xi^2 4^{-\alpha(M-m)}, \tag{2.31}$$

which is a necessary condition for the RPE algorithm to succeed (see Lemma 2.9.4). We now discuss the first part of the Theorem and take ξ constant. By Lemma 2.9.4 and Eq. (2.31) ε^2 is bounded as $\mathcal{O}(4^{-M})$. The total required evolution time is given by

$$t_{\text{tot}} = \sum_{m=0}^{M} 2N_m 2^m = \mathcal{O}(2^M)$$

and similarly $r_{\text{tot}} = \mathcal{O}(2^{2M})$. This results in $t_{\text{tot}} = \mathcal{O}(\varepsilon^{-1})$ and $r_{\text{tot}} = \mathcal{O}(\varepsilon^{-2})$ and proves statements 1 and 2 of the Theorem.

For 3 and 4, we choose $N_M = \mathcal{O}(\beta^{-2}\xi^{-2})$ samples in the last round. To bound the error, we bound by

$$\mathbb{E}\, d(\boldsymbol{\phi}_M, 2^M F_0)^2 \leq \mathbb{P}(|\boldsymbol{Z}_M - \mathbb{E}\boldsymbol{Z}_M| \geq \beta)\pi^2 + \mathbb{E}\left[d(\boldsymbol{\phi}_M, 2^M F_0)^2 \middle| \, |\boldsymbol{Z}_M - \mathbb{E}\boldsymbol{Z}_M| < \beta \right]. \tag{2.32}$$

The first term is bounded by Eq. (2.30), so $\mathbb{P}(|\boldsymbol{Z}_M - \mathbb{E}\boldsymbol{Z}_M| \geq \beta) = 4\exp(-\Omega(\xi^{-2})) = \mathcal{O}(\xi^{-2})$. For the second term, we use that for $\delta = |Z_M - \mathbb{E}\boldsymbol{Z}_M| < \beta$ we have

$$d(\phi_M, 2^M F_0) \leq \arcsin\left(\frac{\delta + (1-\eta)}{\eta}\right) = \mathcal{O}(\delta + (1-\eta))$$

(which for Lemma 2.9.4 we would like to be $\mathcal{O}(\xi^2)$). Here we use Eq. (2.27), $\eta^{-1}(\delta + (1-\eta)) \leq \frac{\pi}{3}$ and η is close to 1. This can be used to see that the second term in Eq. (2.32) can be bounded by

$$\mathbb{E}\, \arcsin\left(\frac{|Z_M - \mathbb{E}\boldsymbol{Z}_M| + (1-\eta)}{\eta}\right)^2 = \mathcal{O}\big(\mathbb{E}\,|\boldsymbol{Z}_M - \mathbb{E}\boldsymbol{Z}_M|^2 + (1-\eta)^2\big) = \mathcal{O}(\xi^2)$$

if $N_M = \Omega(\xi^{-2})$. By Lemma 2.9.4 and Eq. (2.31), this implies root mean square error $\varepsilon = \mathcal{O}(\xi 2^{-M})$. The maximal evolution time is $t_{\max} = 2^M = \mathcal{O}(\xi\varepsilon^{-1})$, and the randomized methods use $r_{\max} = r_M = \mathcal{O}(\xi^2\varepsilon^{-2})$, and

$$\begin{aligned} t_{\text{tot}} &= \sum_{m=0}^{M} 2N_m 2^m = \sum_{m=0}^{M} \mathcal{O}(\beta^{-2}(\log\big(\xi^{-1}\big) + (M-m))2^m) + \mathcal{O}(\xi^{-2}2^M) \\ &= \mathcal{O}(\xi^{-2}2^M) = \mathcal{O}(\xi^{-1}\varepsilon^{-1}). \end{aligned}$$

With the randomized methods, the total number of rotations is similarly given by

$$r_{\text{tot}} = \sum_{m=0}^{M} 2N_m r_m = \mathcal{O}(\xi^{-2}2^{2M}) = \mathcal{O}(\varepsilon^{-2}). \tag{2.33}$$

□

We briefly comment on the choice of depth r_m for the randomized product formulas, what choice minimizes the total cost, and what is the resulting difference between RTE and qDRIFT. Recall that the depth chosen determines how much the amplitude of the signal is damped, as in Eq. (2.18) and Eq. (2.22). If the relevant component of the signal is damped by a factor B, we need to scale the

number of samples by $\mathcal{O}(B^2)$. For RTE for time t and r rotations, we get a damping factor $B \leq \exp(t^2/r)$, which means that we have to take a number of samples scaling with $B^2 \leq \exp(2t^2/r)$ to estimate the signal to constant precision. The total cost scales as $B^2 r$, which is optimized by $r = 2t^2$. In particular, for $t = 2^m$, we should choose $r_m = 2^{2m+1}$. For qDRIFT, for our proof analysis above it was convenient to choose $r_m = 2^{M+m}$, since that means that the we have a constant value of F_k over different rounds m. Alternatively, one may take $r_m = 2^{2m}$, in which case in round m we have $2^m F_0 = 2^{2m} \arctan(2^{-m} E_0) = 2^m E_0 + \mathcal{O}(2^{-m})$. Since we only need to estimate $2^m E_0$ to constant precision in the m-th round, this does not make a significant difference (one has to be careful for small m, but in that case it is anyways cheap to take r_m larger). By Eq. (2.18), we have $c_0 \geq \exp(-t^2(1 - E_0^2)/2r) \geq \exp(-t^2/2r)$. In this case, optimizing the total resources means one should take $r = t^2$ (so this is a factor of 2 more efficient than RTE). One caveat is that the damping depends on E_k for qDRIFT, which makes a difference if we do not start with the ground state. This is beneficial if $|E_0| \leq |E_k|$ for all $k \neq 0$ with $c_k > 0$ (the excited states are damped more than the ground state).

Performance of robust phase estimation

In the previous section we saw that robust phase estimation achieves Heisenberg limited scaling, when choosing an appropriate number of samples for the Hadamard test. There, we did not attempt to optimize constant factors. In the special case where we start with an exact eigenstate, [26] shows a bound for the root mean square error of $\varepsilon \leq C_{\text{tot}} \pi / t_{\text{tot}}$ with $C_{\text{tot}} \approx 25$ using $N_m = \lceil N_M + (M - m)D \rceil$ samples with $N_M = 11$ and $D \approx 4.11$, based on the analysis in Lemma 2.9.4. This is in contrast to the optimal performance of QPE in general, which has optimal constant factor $C_{\text{tot}} = 1$ [134]. However, the rigorous bound of [26] is still loose, and for our resource estimates we use an empirically estimated constant. Numerical simulation shows that using the choice of parameters from [26], we achieve a scaling with a constant factor of $C_{\text{tot}} \approx 5$, see Fig. 2.12 (a). The empirical scaling with the *maximal* time evolution $t_{\max} = 2^M$ is $\varepsilon \approx C_{\max} \pi / t_{\max}$ for $C_{\max} \approx 0.08$. These results corroborate previous numerical estimates, see e.g. [135]. The total time evolution is given by

$$t_{\text{tot}} = \sum_{m=0}^{M} 2 N_m 2^m = \sum_{m=0}^{M} 2(N_M + D(M - m)) 2^m \leq 4(N_M + D) 2^M \tag{2.34}$$

$$= 4(N_M + D) t_{\max}. \tag{2.35}$$

This estimate only ignores a small subleading term, and gives a result which is consistent with the numerical relation between C_{tot} and $C_{\max}$ from Fig. 2.12.

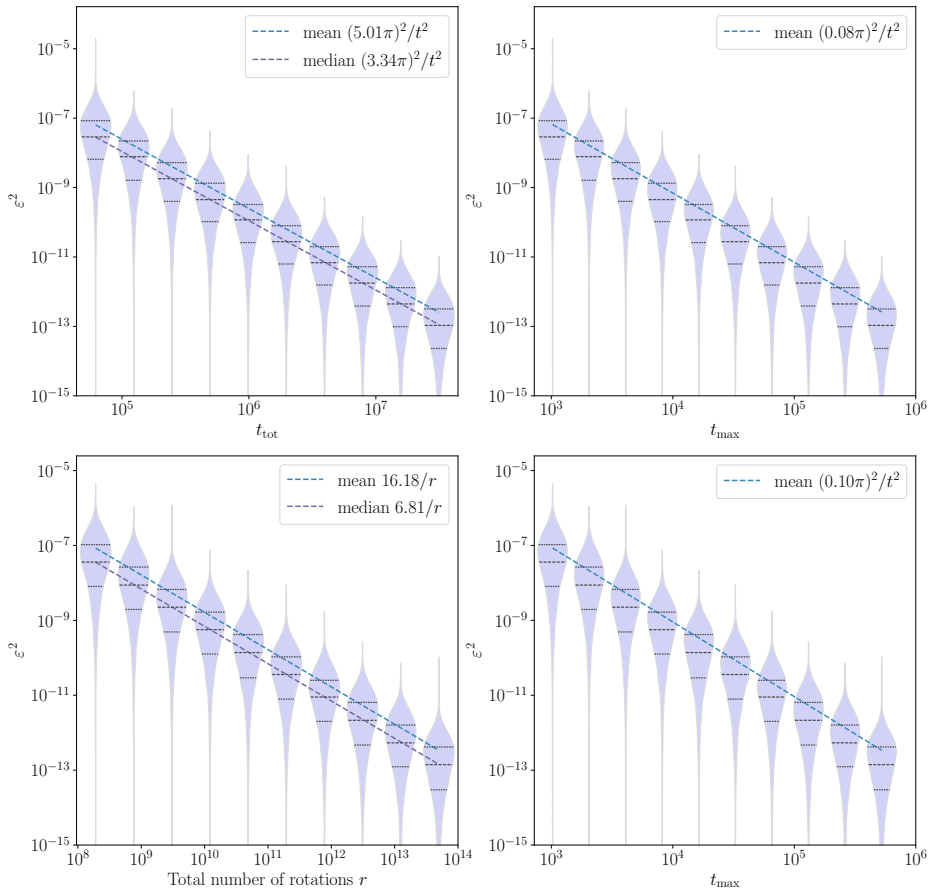


Figure 2.12: Estimation of the overhead of robust phase estimation. Distribution of the mean square error, 10^4 data points for each value of M. The top panels shows scaling of mean square error with total time evolution and maximal time evolution $t_{\mathrm{max}} = 2^M$, using $N_m = 11+4(M-m)$ samples in round m. The bottom panels show scaling of mean square error with total number of rotations according to Eq. (2.33) and maximal time evolution $t_{\mathrm{max}} = 2^M$ simulated, using depth $r_m = 2^{2m+1}$ and $N_m = \lceil e(11+4(M-n))\rceil$ samples in round m.

These estimates assume a perfectly accurate quantum computer. By design, the robust phase estimation procedure can tolerate a small amount of error. See [135] for a discussion of the error budget in the context of quantum chemistry computations.

We now discuss the total cost when using randomized product formulas. For RTE, using $\tau = 2^{-m-1}$ and $r = 2^{2m+1}$ in the m-th round requires a sampling overhead of e. We should now take N_m a factor e larger than in the deterministic case. For our numerical estimates, we take $N_m = e(11+4(M-m))$. The total

number of rotations is given to good approximation by

$$r_{\text{tot}} = \sum_{m=0}^{M} 2N_m 2^{2m+1} = \sum_{m=0}^{M} 2(N_M + D(M-m))2^{2m+1} \leq 16\left(\frac{N_M}{3} + \frac{D}{9}\right)2^{2M}$$
$$= 16\left(\frac{N_M}{3} + \frac{D}{9}\right)t_{\text{max}}^2 = 8\left(\frac{N_M}{3} + \frac{D}{9}\right)r_{\text{max}}\,. \tag{2.36}$$

In Fig. 2.12 (b) we numerically show that this indeed leads to a scaling $\varepsilon^2 \approx C'_{\text{tot}}/r_{\text{tot}}$ and $\varepsilon^2 \approx (C'_{\text{max}}\pi)^2/r_{\text{max}}$ where $C'_{\text{tot}} \approx 16$ and $C'_{\text{max}} \approx 0.2$ (note that the maximal number of rotations is $2t_{\text{max}}^2$). As discussed at the end of the previous subsection, for qDRIFT we save an additional factor of 2 for both r_{max} and r_{tot}, giving a total cost of approximately $r_{\text{tot}} = 8\varepsilon^{-2}$ for performing phase estimation with root mean square error ε.

Concretely, for the example of FeMoco discussed in the introduction, we have $\lambda = 405$ and $\varepsilon = 0.0016$. We should estimate the ground state energy of the normalized Hamiltonian to precision $\lambda^{-1}\varepsilon$. We choose $\xi = 0.1$, giving $M = \lceil \log_2(\xi\lambda\varepsilon^{-1})\rceil = 15$. For the last round, we do not necessarily need to take a power of two, but we can choose $2^{M-1} < K_M \leq 2^M$ and estimate $K_M E_0$. We let $K_M = \lceil \xi\lambda\varepsilon^{-1}\rceil$, then estimating $K_M E_0$ to precision ξ gives an estimate of E_0 to precision ε. This gives circuits with at most $K_M^2 = 6.4 \times 10^8$ Pauli rotations.

2.9.3 The electronic structure problem

The motivation of this work is to use quantum computing for Hamiltonians which arise in electronic structure problems in the Born-Oppenheimer approximation, meaning that we fix the location of nuclei, and treat the electron wavefunctions as the quantum degrees of freedom. For completeness, we describe the set-up, relevant to the numerical examples, in some detail here. We work in second quantization and we assume that we have chosen a set of orthonormal spatial orbitals labelled by an index $p \in [N]$ corresponding to a function $\phi_p \in L^2(\mathbb{R}^3)$. Additionally, there is a spin degree of freedom, so we end up with a set of spin-orbitals $\phi_{p,\alpha}$ for $p \in [N]$ and $\alpha = \uparrow, \downarrow$. For convenience, we will assume that the spatial orbitals are real-valued functions (but this is not essential to the arguments). The spin-orbitals span a single-particle space $\mathcal{H}_1 \cong \mathbb{C}^{2N}$. The Hilbert space we work with is the corresponding Fock space

$$\mathcal{H} = \bigoplus_{n=0}^{2N} \mathcal{H}_n \qquad \mathcal{H}_n = \mathcal{H}_1^{\wedge n}.$$

We denote the fermionic annihilation operator associated to orbital p and spin α by $a_{p\sigma}$. The size of the problem is determined by N, the number of orbitals, and n, the number of electrons.

Given a choice of orbitals, the Hamiltonian of the electronic structure problem is given by

$$H = T + V = \sum_{pq} \sum_{\sigma} h_{pq} a^\dagger_{p\sigma} a_{q\sigma} + \frac{1}{2} \sum_{pqrs} \sum_{\sigma\tau} h_{pqrs} a^\dagger_{p\sigma} a^\dagger_{r\tau} a_{s\tau} a_{q\sigma} \tag{2.37}$$

where the coefficients h_{pq} and h_{pqrs} result from performing appropriate integrals involving the orbital functions. Here and in the remainder of this work we use the convention that the spatial orbitals are labelled by $p, q, r, s, \ldots$ and spins by $\sigma, \tau, \ldots$. In Eq. (2.37), the first term T represents the kinetic energy and the interaction between the nuclei and the electrons

$$h_{pq} = \int_{\mathbb{R}^3} \phi_p(\vec{r}) \left(-\frac{1}{2}\nabla^2 - \sum_{i=1}^{m} \frac{Z_i}{\left\| \vec{r} - \vec{R}_i \right\|} \right) \phi_q(\vec{r}) d\vec{r}. \tag{2.38}$$

Here the $\vec{R}_i$ and Z_i are the (fixed) positions and the nuclear charge of the m nuclei. The last term corresponds to the electron-electron Coulomb interaction V

$$h_{pqrs} = \int_{\mathbb{R}^3} \int_{\mathbb{R}^3} \frac{\phi_p(\vec{r}_1)\phi_q(\vec{r}_1)\phi_r(\vec{r}_2)\phi_s(\vec{r}_2)}{\|\vec{r}_1 - \vec{r}_2\|} d\vec{r}_1 d\vec{r}_2. \tag{2.39}$$

The integrals depend on the choice of the basis of orbitals. Any unitary $u \in U(N)$ defines a basis change on the spatial part of the single-particle space, giving rise to a new set of orbitals by $\tilde{\phi}_p = \sum_q u_{pq}\phi_q$. It defines a particle-number and spin conserving unitary U on the Hilbert space $\mathcal{H}$ by defining a new set of fermionic creation operators $\tilde{a}^\dagger_p = \sum_q u_{pq} a^\dagger_q$. Typically, the set of orbitals used to define Eq. (2.37) is derived from a mean-field calculation, giving rise to a set of *canonical orbitals*. Often it is useful to perform a basis change so the orbitals are *localized* functions with exponentially decaying tails. In that case, from Eq. (2.39) we see that the density-density terms with $p = q$ and $r = s$ lead to contributions which decay polynomially in the distance between the spatial locations of the orbitals p and r, whereas for $p \neq q$ we have an exponential decay in the distance between the spatial locations of orbitals p and q. In Fig. 2.7 we indeed see a regime of terms with power law decay, and a regime of terms with exponential decay. For large spatially extended systems, one finds that many of the h_{pqrs} are extremely small (say, less than 10^{-10}) and can be safely ignored. In that case, the number of relevant terms scales as $\mathcal{O}(N^2)$. Details on electronic structure formalism and

computations can be found in [7]; see in particular section 9.12 for a detailed discussion of the scaling of the size of integrals h_{pqrs}.

The Hamiltonian H is mapped to $2N$ qubits by a fermion-to-qubit mapping. Various options are possible, but the key feature is that the terms in H get mapped to Pauli operators. Generally, one defines the Majorana fermionic operators by

$$\gamma_{p\sigma,0} = a^\dagger_{p\sigma} + a_{p\sigma}, \qquad \gamma_{p\sigma,1} = i(a^\dagger_{p\sigma} - a_{p\sigma})$$

and each $\gamma_{p\sigma,\alpha}$ should be mapped to a Pauli operator in a way that preserves the anticommutation relation $\{\gamma_{p\sigma,\alpha}, \gamma_{q\tau,\beta}\} = 0$ if $p\sigma, \alpha \neq q\tau, \beta$ (for example, using the Jordan-Wigner mapping, or Bravyi-Kitaev mapping). Ignoring a component which is proportional to the identity operator, the Hamiltonian can be written as [9]

$$H = \frac{i}{2} \sum_{pq} \sum_{\sigma} \left(h_{pq} + \sum_r h_{pqrr} - \frac{1}{2} \sum_r h_{prrq} \right) \gamma_{p\sigma,0} \gamma_{q\sigma,1} \tag{2.40}$$

$$- \frac{1}{8} \sum_{pqrs} \sum_{\sigma\tau} h_{pqrs} \gamma_{p\sigma,0} \gamma_{q\sigma,1} \gamma_{r\tau,0} \gamma_{s\tau,1} \tag{2.41}$$

which gives a representation of H

$$H = \sum_{l=1}^{L} h_l P_l \tag{2.42}$$

where the P_l are Pauli operators and the h_l are real coefficients.

Factorization of electronic structure Hamiltonians

The Coulomb interaction (i.e. the two-body term) is the part of the Hamiltonian that is the main contribution to the complexity of quantum simulation of H in Eq. (2.37). For example, simply Trotterizing over the sum as given in the description of the Hamiltonian, the one-body interactions give $\mathcal{O}(N^2)$ terms, while the two-body interaction contributes $\mathcal{O}(N^4)$ terms. Moreover, if the Hamiltonian *only* has one-body terms, the Hamiltonian is quadratic in the creation and annihilation operators, which may be simulated efficiently classically. Employing anticommutation relations the electronic structure Hamiltonian 2.37 can be written as

$$H = \sum_{pq} \sum_{\sigma} \underbrace{(h_{pq} - \frac{1}{2} \sum_r h_{prrq})}_{h'_{pq}} a^\dagger_{p\sigma} a_{s\sigma} + \frac{1}{2} \sum_{pqrs} \sum_{\sigma\tau} h_{pqrs} a^\dagger_{p\sigma} a_{q\sigma} a^\dagger_{r\tau} a_{s\tau} = T + V. \tag{2.43}$$

It is useful to look for simpler representations of the Coulomb interaction V in order to simplify both product formulas and block encodings of the Hamiltonian. A prominent example is *double factorization* [95, 60, 136, 45, 137, 138, 90].

The idea is to first consider h as a positive $N^2 \times N^2$ real matrix (the positivity follows from Eq. (2.39)), and decompose it as

$$h_{pqrs} = \sum_{j=1}^{L} (\mathcal{L}_{pq}^{(j)})^T \mathcal{L}_{rs}^{(j)} \tag{2.44}$$

where $\mathcal{L}^{(j)}$ is a real vector of dimension N^2, with entries labelled by orbital pairs pq. This is the 'first factorization'. The Coulomb tensor satisfies the symmetries $h_{pqrs} = h_{pqsr} = h_{qprs} = h_{qpsr} = h_{rspq} = h_{rsqp} = h_{srpq} = h_{srqp}$. These symmetries imply that we may take the $\mathcal{L}^{(j)}$ to be real and such that $\mathcal{L}_{pq}^{(j)} = \mathcal{L}_{qp}^{(j)}$. Therefore, we can reinterpret the $\mathcal{L}^{(j)}$ as *symmetric* $N \times N$ matrices. One may now apply a 'second factorization' and diagonalize each $\mathcal{L}^{(j)}$ using an orthogonal transformation. That is, we find $u^{(j)} \in O(N)$ and diagonal $N \times N$ matrices $\Lambda^{(j)}$ with real numbers $\lambda_k^{(j)}$ on the diagonal, such that

$$\mathcal{L}^{(j)} = (u^{(j)})^T \Lambda^{(j)} u^{(j)}. \tag{2.45}$$

Inserting this back into the Coulomb interaction term of the Hamiltonian, one finds

$$V = \frac{1}{2} \sum_{j=1}^{L} \left(\sum_{pq} \sum_{\sigma} \mathcal{L}_{pq}^{(j)} a_{p\sigma}^\dagger a_{q\sigma} \right)^2 = \frac{1}{2} \sum_{j=1}^{L} \left(\sum_{k=1}^{\rho_j} \lambda_k^{(j)} \sum_{pq} \sum_{\sigma} u_{kp}^{(j)} a_{p\sigma}^\dagger a_{q\sigma} u_{kq}^{(j)} \right)^2$$

The $N \times N$ matrices $u^{(j)}$ define a change of orbital basis, and they give rise to a corresponding orbital rotation $U^{(j)}$ on the full Hilbert space. The fermionic annihilation operators in the changed basis $\phi_k^{(j)} = \sum_p u_{kp}^{(j)} \phi_p$ are given by

$$a_{k\sigma}^{(j)} = \sum_p u_{kp}^{(j)} a_{p\sigma}$$

and the associated number operators are $n_{k\sigma}^{(j)} = (a_{k\sigma}^{(j)})^\dagger a_{k\sigma}^{(j)}$. Then,

$$V = \frac{1}{2} \sum_{j=1}^{L} \left(\sum_{k=1}^{\rho_j} \sum_{\sigma} \lambda_k^{(j)} n_{k\sigma}^{(j)} \right)^2 = \sum_{j=1}^{L} (U^{(j)})^T V^{(j)} U^{(j)}, \tag{2.46}$$

$$\text{for} \quad V^{(j)} = \frac{1}{2} \left(\sum_{k,\sigma} \lambda_k^{(j)} n_{k\sigma} \right)^2. \tag{2.47}$$

The main reason that this decomposition is useful is that h is well-approximated by a *low-rank* matrix. In the first factorization in Eq. (2.44) we may truncate the sum to contain only $L = \tilde{\mathcal{O}}(N)$ terms, instead of the maximal number of N^2. Secondly, numerical results suggest that in the limit of large N, for spatially extended systems, one can truncate ρ_j, the number of terms in the second factorization, to only $\tilde{\mathcal{O}}(1)$ terms instead of N [125, 124], although this only becomes relevant for large systems. Note that in this construction the choice of the $\mathcal{L}^{(j)}$ is not unique. Common choices are an eigenvalue decomposition or a Cholesky decomposition, but there is significant flexibility. There are various methods which aim to optimize the factorization in order to reduce the total circuit depth [137, 139, 90]. We discuss the associated gate counts for Trotterization using this low rank factorization in Section 2.9.5.

2.9.4 Benchmark systems

For FeMoco we use the model Hamiltonian first described by Reiher et al. [40]. The data specifying the Hamiltonian coefficients for both the hydrogen chain and for FeMoco is taken from [46]. For the hydrogen chain we use a STO-6G minimal basis set and set the distance between the hydrogen atoms at 1.4 Bohr. The number of electrons is $n = N_H$, the number of spin-orbitals is $2N_H$. Additionally, we use a collection of small molecules with varying active space sizes to estimate the Trotter error. These molecules are alanine $C_3H_7NO_2$, the chromium dimer Cr_2, an iron-sulfur cluster Fe_2S_2, hydrogen peroxide H_2O_2, imidazole $C_3H_4N_2$, naphtalene $C_{10}H_8$, nitrous acid HNO_2 and sulfurous acid H_2SO_3 at equilibrium geometries. The active space sizes vary from four to nine orbitals, and the active orbitals are chosen to be the closest ones to the HOMO-LUMO gap. Furthermore, for each active space Hamiltonian generated this way, the same λ optimization as described below is performed.

Optimizing the weight λ

As discussed in Section 2.7, it is useful to optimize the choice of representation to have small weight λ. Firstly, it is clear that the coefficients h_{pq} and h_{pqrs} defined by Eq. (2.38) and Eq. (2.39) depend on the choice of orbital basis. Typically, one starts from a set of canonical orbitals from a mean-field calculation, which determine the active space for the problem. However, given the subspace, one is free to choose a different basis, for which is the representation of H has desirable properties, such as a small weight λ. In [9] the effect of different basis changes on the value of λ was explored, with as conclusion that orbital localization (i.e. choosing an orbital basis for which the orbital functions are spatially localized) decreases the value of λ and that further reduction is possible by minizing over

the basis change. We perform such optimization for all benchmark systems, using gradient descent over the basis change unitaries to minimize the weight λ. The problem is not convex and needs a good starting point. One option is to use localized orbitals. From our numerics, we observe that this choice can be a local minimum for λ. A good alternative is to take the largest term in a Cholesky decomposition of the Coulomb tensor (cf. Eq. (2.44)) $\mathcal{L}^{(1)}_{pq}$. This defines an $N \times N$ symmetric matrix, and we can consider the orbital basis which diagonalizes this matrix, as in Eq. (2.45). This is cheap to compute, already leads to significantly reduced weight λ, and forms a good Ansatz to start the optimization.

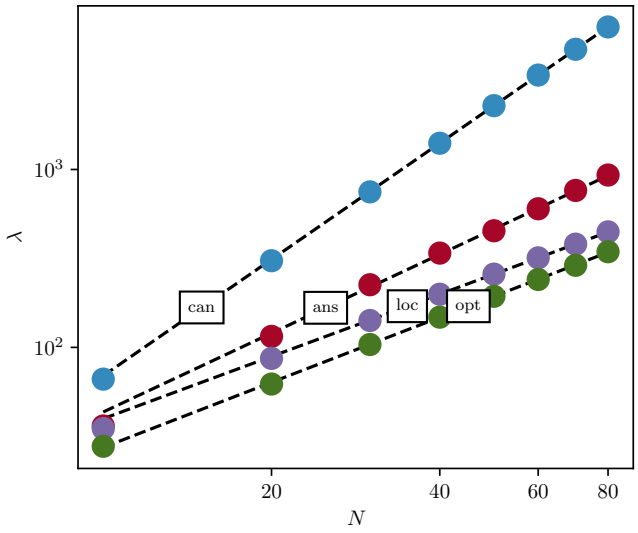


Figure 2.13: The weight λ of the Hamiltonian depends on its decomposition. Here we show the scaling of λ for the hydrogen chain with system size N_{H}, for canonical orbitals, localized orbitals, an ansatz based on the eigenvectors of the largest Cholesky vector, and an optimized representation from gradient descent over the space of orbital basis changes.

The Hamiltonian has a particle-number symmetry. That is, H commutes with the electron number operator $\hat{n} = \sum_{p,\sigma} a^\dagger_{p\sigma} a_{p\sigma}$, and we work in a subspace of fixed electron number n, which is set by the initial state. This feature can be exploited to define a different Hamiltonian, which equals H on the subspace of n-particles (possibly up to an additive constant factor) but with lower weight. This has been used in various ways, in particular to optimize the cost of LCU implementations of H [10, 90, 12]. For this work we use a variant of the method of Ref. [11]. Let $\hat{n} = \sum_{p\sigma} n_{p\sigma}$ be the number operator, and let n be the number of electrons for the ground state of H. Then we can also apply phase estimation to $H + F(\hat{n} - n) + (\hat{n} - n)F$ for any self-adjoint operator F, since this has the same spectrum when restricted to the n-particle subspace. In particular, one can take F to be a one-body operator, and use this to reduce the weight of the

two-body terms in H. We perform a joint numerical optimization, minimizing λ, over the choice of the symmetry shift and the choice of orbital basis, using the L-BFGS-B quasi-Newton optimization method. For the optimization, we restrict to an Ansatz where we decompose the Coulomb operator as in Eq. (2.46) and shift

$$V^{(j)} \to \frac{1}{2}\left(\sum_{k\sigma} \lambda_k^{(j)} n_{k\sigma} + f_j(\hat{n} - n)\right)^2$$

and optimize the parameters f_j. We show the resulting scaling of λ with N for the hydrogen chain in Fig. 2.13. Localized and optimized orbitals bases give a weight scaling as $\lambda = \mathcal{O}(N^{1.2})$.

Trotter error for electronic structure Hamiltonians

In this section we report in detail on our numerical analysis of Trotter error. We briefly recall our motivation for doing so. There are many ways to analytically bound the Trotter error. The tightest bounds are based on nested commutators. Nevertheless, even if these have the correct scaling, they are still loose by some constant factor [51]. Additionally, if the Hamiltonian has L terms, the commutator bound for the p-th order in principle may require as many as $\mathcal{O}(L^{p+1})$ operations, which is not feasible to compute exactly in practice, even for $p = 2$, for molecular electronic structure Hamiltonians with many orbitals.

For these reasons, we take a numerical approach also taken in [52, 40] and numerically compute the Trotter error for small systems, to then extrapolate to larger systems. Recall that if H is the Hamiltonian with ground state E_0, $\tilde{H}$ the effective Hamiltonian of the Trotterized time evolution with ground state $\tilde{E}_0$, $U(\delta)$ is the exact time evolution for time δ and $S_p(\delta)$ is a Trotter step with time δ, then we have

$$|E_0 - \tilde{E}_0| \leq C_{\text{gs}}\delta^p \quad \text{and} \qquad \|U(\delta) - S_p(\delta)\|_{\text{op}} \leq C_{\text{op}}\delta^{p+1}$$

for an order p product formula. Here, we let C_{gs} and C_{op} be the smallest constants for which the bound holds. Crucially, these values depend on the Hamiltonian, as well as the system size. The cost of the product formula scales with $C_{\text{gs}}^{1/p}$.

We estimate C_{gs} and C_{op} by numerically computing $U(\delta)$ and $S_p(\delta)$ for a range of values of δ, and fit a power law to the resulting ground state energy and operator norm errors. We find that for a wide range of δ we get an excellent fit (even though in principle for large δ there will be higher order corrections which are dominant), and use this to determine C_{gs} and C_{op}, leading to Fig. 2.8 in the main text. For the numerical examples, we use a symmetry-preserving

Bravyi-Kitaev mapping [140]. The Trotter error depends on the ordering of the terms; we choose a lexicographic ordering. See [141, 142] for a numerical study of the effect of the choice of fermion-to-qubit mapping and ordering on the Trotter error.

In Fig. 2.14 we compare C_{gs} and C_{op} for the small molecule benchmark set and the hydrogen chain for $p = 2$. We observe both correlate well (especially C_{op}) with λ. Additionally, we also check the correlation with the commutator

$$\alpha = \sum_{l_1,l_2} \|[H_{l_1}, H_{l_2}]\|_{\mathrm{op}} = \sum_{l_1,l_2} |h_{l_1} h_{l_2}| \, \|[P_{l_1}, P_{l_2}]\| \tag{2.48}$$

and find a comparable correlation.

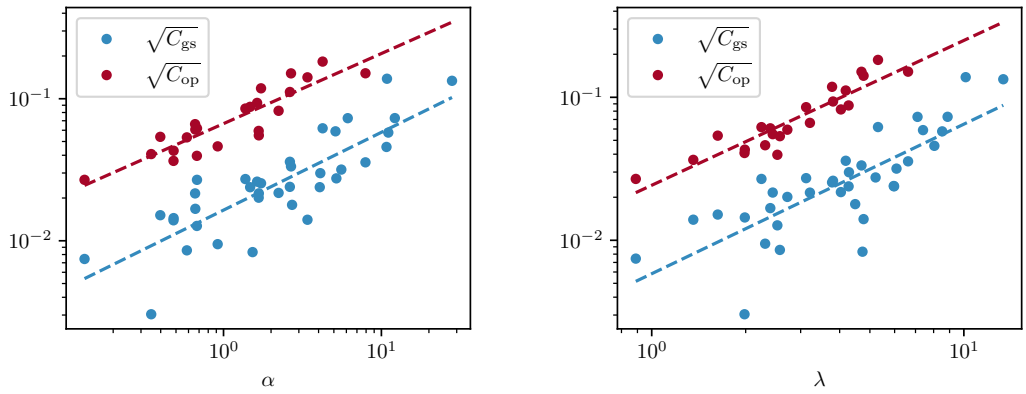


Figure 2.14: The left plot shows the Trotter constants for the operator norm C_{op} and the ground state energy C_{gs}, for the benchmark set of small molecules and its correlation with α. The right plot shows the same data, compared to the weights λ.

One can also use higher order product formulas, which leads to a better scaling in t, the time to be simulated. However, at fixed finite time t it is not clear whether this gives an improvement, since increasing the order also increases the number of stages N_{stage} of the product formula, and N_{stage} increases exponentialy with the order p for the Trotter-Suzuki product formulas. In Fig. 2.15 we compare the second and fourth order product formulas for the same benchmark systems. Here we plot the factor that determines the difference in cost, which is $N_{\mathrm{stage}} C_{\mathrm{gs}}^{1/p}(p, \{H_l\}) \varepsilon^{-1/p}$. The number of stages is $N_{\mathrm{stage}} = 2, 10$ for $p = 2, 4$ and we take $\varepsilon = 0.0016$. We do not observe a consistent improvement from using the fourth order method. Of course, for sufficiently small accuracy ε, the higher order method will eventually be superior. However, for this work we conclude that it suffices to use second order (which has the added benefit that it does need to evolve backwards in time, which is beneficial for the partially randomized product formula).

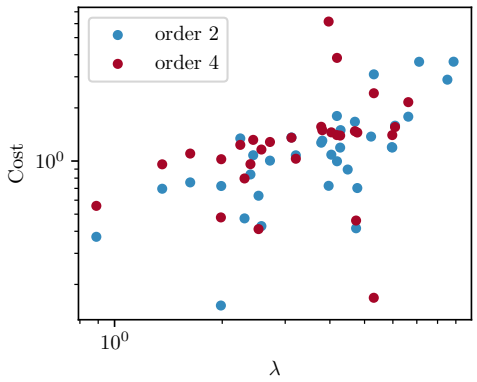


Figure 2.15: This figure shows the effect of the choice of order p of the product formula. We plot Cost $= PC_{\text{gs}}^{1/p}\varepsilon^{-1/p}$ for $p = 2$ and $p = 4$ Trotter-Suzuki product formulas and $\varepsilon = 0.0016$ for the benchmark set of small molecules and for the hydrogen chain.

We also briefly comment on the first order method: for the ground state energy it has *second order scaling*. This is an easy consequence of the fact that in perturbation theory, the first order correction $\langle\psi_0|\,[H_{l_1}, H_{l_2}]\,|\psi\rangle$ vanishes [143]. However, since the second order method only has two stages, the additional cost is not very high.

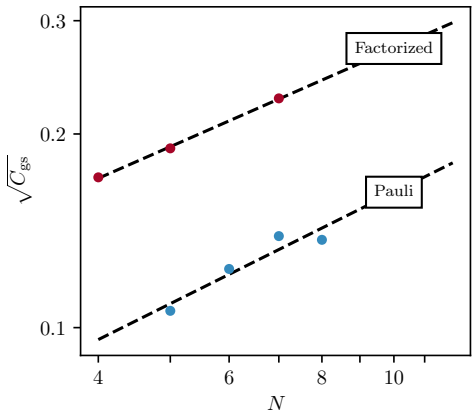


Figure 2.16: For order $p = 2$ the Trotter ground state error constant $\sqrt{C_{\text{gs}}}$ for the hydrogen chain with N atoms, both for the Pauli representation and the factorized representation.

Trotter error for partial randomization

For partially randomized product formulas, given

$$H = \sum_l h_l P_l$$

one groups together a subset of terms into H_R. The Trotterization error for the partially randomized method is the Trotterization error of dividing the Hamiltonian into terms $h_l P_l$ and keeping H_R as a single term. For the performance of the partially random method it is clearly of interest how the Trotter error depends on the choice of H_R. There are two intuitions of what could happen to the Trotter error. Firstly, if many terms are in H_R, we are discretizing less, and the Trotter error should become smaller. In particular, if all terms are in H_R, there is no Trotter error whatsoever. On the other hand, the Trotter error decreases if one takes very small steps. Grouping together many small terms into one larger term H_R means performing a relatively large step in the direction of H_R, which could *increase* the Trotter error.

In Fig. 2.17 we show numerically that for λ_R relatively small (the regime which is of most interest for partial randomization), the Trotter error remains essentially unchanged. For intermediate values of λ_R, the Trotter error in fact increases, but only moderately so. When λ_R tends to λ, the Trotter error decreases significantly.

These findings justify our choice to optimize the partially randomized methods using the Trotter error estimate for the fully deterministic method. Indeed, the regime where λ_R is a large fraction of λ will not lead to significant speed-up over the fully randomized method (so it does not help much that the Trotter error is smaller), and when λ_R is reasonably small, the Trotter error of the fully deterministic product formula is a very good approximation.

2.9.5 Detailed cost estimates and compilation

In this part of the Appendix we are giving a detailed account of how the gate counts are estimated for ground state energy estimation using the deterministic, randomized, or partially randomized simulation methods. For each method we are estimating the number of gate G, where we can either measure the number of (non-Clifford) Toffoli gates, or the number of two-qubit gates. We are given a Hamiltonian H, and wish to estimate the ground state energy to target precision ε. In these estimates we assume access to an exact eigenstate of H, but as discussed this assumption can easily be relaxed. The total cost depends on various parameters, some of which depend on H and some of which can be optimized. We give an overview of the relevant parameters in Table 2.1.

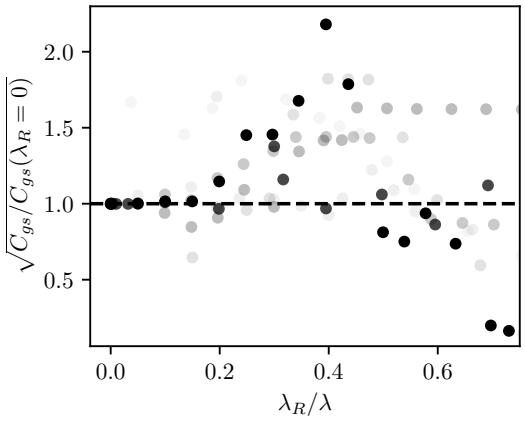


Figure 2.17: Trotter error as a function of the fraction of the randomized part for different active spaces of alanine and Fe_2S_2. The dashed line indicates the fully deterministic Trotter error, while the darkness of the data points corresponds to the size of the active space, with darker points corresponding to larger active space size.

Parameter	Notation
Phase estimation parameters	
ε	Target precision for the energy.
M	Number of rounds in phase estimation.
N_m	Number of samples in round m, $N_m = N_M + D(M - m)$.
Deterministic product formulas	
L	Number of terms in the Hamiltonian.
$\varepsilon_{\text{qpe}}, \varepsilon_{\text{trot}}$	Dividing error budget between phase estimation and Trotter error.
δ	Trotter step size.
p	Order product formula.
N_{stage}	Number of stages product formula.
C_{gs}	Trotter error constant for the ground state energy.
G_{det}	Number of elementary gates of one elementary evolution $\exp(-i\delta H_l)$.
Randomized product formulas	
λ	Weight of the Hamiltonian.
τ	Step size
φ	Rotation angle $\varphi = \arctan(\tau)$.
G_{rand}	Number of elementary gates to time evolve a single Pauli operator P_l.
B	Damping factor from the randomized product formula.
Partially randomized product formulas	
L_D	Number of terms treated deterministically.
λ_R	Weight of the terms treated randomly.
κ	Scaling factor for the number of rotations in randomized part.

Table 2.1: Relevant parameters for gate counts for of phase estimation of ground state energies using deterministic and (partially) randomized product formulas.

Gate counts for Trotterization

The Hamiltonian has a decomposition $H = \sum_{l=1}^{L} H_l$. We use a product formula of order p. For most of the resource estimates we use second order ($p = 2$). As explained in Section 2.3, we perform phase estimation on the Trotter unitary $S_p(\delta) = e^{-i\delta\tilde{H}} \approx e^{-i\delta H}$ for this task, computing the ground energy $\tilde{E}_0$ of the effective Hamiltonian $\tilde{H}$. The error between the ground states of H and $\tilde{H}$ behaves as $\varepsilon_{\text{trot}} = |E_0 - \tilde{E}_0| \leq C_{\text{gs}}\delta^p$. Additionally, when applying phase estimation, we obtain $\tilde{E}_0$ with mean square error ε_{qpe}. This error is unbiased and independent of the Trotter error, which means that in order get an estimate of precision ε for E_0 we should have

$$\varepsilon^2 = \varepsilon_{\text{qpe}}^2 + \varepsilon_{\text{trot}}^2. \tag{2.49}$$

To reach precision ε_{qpe} for $\tilde{E}_0$, we need to estimate the phase of $S_p(\delta)$ to precision $\varepsilon_{\text{qpe}}\delta$. We estimated in Section 2.9.2 that robust phase estimation needs on average $5\pi/(\varepsilon_{\text{qpe}}\delta)$ implementations of $S_p(\delta)$ to do so. The number of rounds in the phase estimation is

$$M = \left\lceil \log_2\left(\frac{0.08\pi}{\varepsilon_{\text{qpe}}\delta}\right)\right\rceil. \tag{2.50}$$

Let G_{det} be the average gate cost to implement a single evolution along the H_l, then we see that the total gate cost is given by

$$G = 5\pi N_{\text{stage}} L G_{\text{det}} \frac{C_{\text{gs}}^{1/p}}{\varepsilon_{\text{qpe}}\varepsilon_{\text{trot}}^{1/p}}.$$

Minimizing the cost under the error buget in Eq. (2.49) results in $\varepsilon_{\text{qpe}} = \varepsilon\sqrt{\frac{p}{p+1}}$, and trotter step size and optimal total cost

$$\delta = \left(\frac{\varepsilon}{C_{\text{gs}}}\right)^{1/p}\left(\frac{p}{1+p}\right)^{1/2p}, \tag{2.51}$$

$$G = 5\pi N_{\text{stage}} L G_{\text{det}} \sqrt{\frac{(p+1)^{1+1/p}}{p}} \frac{C_{\text{gs}}^{1/p}}{\varepsilon^{1+1/p}}. \tag{2.52}$$

We can reduce the cost by factor 2 using the method in Section 2.9.1. At this point, the cost is agnostic of the gate set used. For the Toffoli count we additionally need to approximate with a gate set of Cliffords and Toffolis, and consider the resulting synthesis error. Here we follow the analysis of [40, 110]. We decompose circuits into Cliffords plus arbitray single-qubit rotations, which can be

synthesized to error ε' at a cost of

$$G_{\text{rot}} = 1.14 \log_2 \left(\frac{1}{\varepsilon'} \right) + 9.2$$

T gates [144]. We convert from Toffoli to T gates using 1 Toffoli gate for every 2 T gates [145]. Let n_{rot} be the number of single qubit rotations per trotter stage, each approximated to error ε'. We then want to bound the synthesis error $\varepsilon_{\text{synth}}$, i.e. the difference in the ground state energies of the exact effective Hamiltonian and the effective Hamiltonian with imperfectly synthesized single-qubit rotations, denoted by H'. One can bound [40]

$$\left\| e^{-i\delta \tilde{H}} - e^{-i\delta H'} \right\| \leq N_{\text{stage}} n_{\text{rot}} \varepsilon',$$

so to first order in δ the ground state error is bounded by

$$\left| \tilde{E}_0 - E'_0 \right| \leq \left\| \tilde{H} - H' \right\| \leq N_{\text{stage}} n_{\text{rot}} \varepsilon' \delta^{-1}.$$

Consequently, to keep the synthesis error below $\varepsilon_{\text{synth}}$, we choose

$$\varepsilon' = \frac{\delta \varepsilon_{\text{synth}}}{N_{\text{stage}} n_{\text{rot}}} = \frac{\varepsilon_{\text{synth}}}{N_{\text{stage}} n_{\text{rot}}} \left(\frac{\varepsilon}{C_{\text{gs}} \sqrt{p+1}} \right)^{1/p}.$$

We do not optimize the total error budget with respect to $\varepsilon_{\text{synth}}$, instead, when computing the Toffoli count, we allocate $\varepsilon = 1.5\,\text{mH}$ and $\varepsilon_{\text{synth}} = 0.1\,\text{mH}$. Alternatively, if we measure two-qubit gates, we ignore the synthesis error, we let $\varepsilon = 1.6\,\text{mH}$ and set G_{det} to be the number of 2-qubit gates.

So far, we have assumed a general Hamiltonian $H = \sum_l H_l$. We now specialize to specific representations.

Pauli Hamiltonian

We start with the case where the Hamiltonian has a representation as a linear combination of Pauli operators, where $H_l = h_l P_l$, as in Eq. (2.42). The time evolution of a single term is then a Pauli rotation, $V_l(\delta h_l) = \exp(-i\delta h_l P_l)$. This can be implemented using a single-qubit rotation and $2(|P_l| - 1)$ CNOTgates to implement. Here we use $|P_l|$ to denote the number of qubits on which P_l acts nontrivially.

The number of Pauli rotations per stage simply equals the number of terms L. The worst-case scaling of the number of terms with the number of orbitals is $L = \mathcal{O}(N^4)$, but in practice many terms are sufficiently small to be truncated (as discussed in Section 2.9.3) and H is much more sparse. This depends on

the choice of single orbital basis as well. We find that minimizing the weight λ through orbital rotations and symmetry shifts typically also leads to more sparse representations of H. For our estimates, we truncate the smallest coefficients of H, such that the total weight $\sum_{l>L'} |h_l|$ of truncated terms is at most a small threshold 10^{-3}.

When counting two-qubit gates we can contract the innermost CNOTgates together with the single qubit rotation into a single 2-qubit gate, so the number of two-qubit gates per Pauli rotation is $2|P_l| - 3$. Therefore the number of 2-qubit gates per operation is $G_{\text{det}} = 2\left|\bar{P}\right| - 3$, where $\left|\bar{P}\right|$ denotes the average Pauli weight. Reductions in two-qubit gate cost are achieved by reducing L or by reducing the average Pauli weight. Again, minimizing λ also reduces L. The average Pauli weight depends on the choice of fermion-to-qubit mapping. On N spin orbitals the Bravyi-Kitaev mapping gives an average Pauli weight of $\mathcal{O}(\log(N))$, compared to $\mathcal{O}(N)$ for the Jordan-Wigner and parity mappings. We use a symmetry-conserving Bravyi-Kitaev mapping, which has the additional advantage of reducing the number of qubits by 2 by taking into account spin and particle number conservation [140]. Yet another degree of freedom is the choice of term ordering the ordering in a Trotter stage, which can lead to significant cancellations of 2-qubit gates [146]. We use the lexicographic ordering, which has a relatively large number of cancellations [141].

Factorized Hamiltonian

In this case, we have a Hamiltonian of the form

$$H = T + \sum_{l=1}^{L} (U^{(l)})^\dagger V^{(l)} U^{(l)}, \qquad V^{(l)} = \sum_{\sigma} \sum_{p=1}^{\rho_l} \lambda_p^{(l)} n_{p\sigma}.$$

We first discuss the two-body terms. In order to implement one stage in a Trotter step, we need to implement L basis changes $U^{(l)}(U^{(l-1)})^\dagger$ (which can be combined into a single orbital basis change) and evolutions of the form $\exp\left(-i\delta V^{(l)}\right)$. This can be done [95, 123] using fermionic swap networks. The orbital basis rotation $U^{(l)}$ for both spin values can be implemented with $2(\binom{N}{2} - \binom{N-\rho_l}{2}) = \mathcal{O}(\rho_l N)$ Givens rotations (which are two-qubit gates, and can be implemented using two arbitrary single-qubit gates), and $\exp\left(-i\delta V^{(l)}\right)$ can be implemented using $\binom{2\rho_l}{2}$ two-qubit gates (requiring one arbitrary single-qubit gate). This gives a total number of single-qubit rotations

$$\sum_{l=1}^{L} 4(\binom{N}{2} - \binom{N-\rho_l}{2}) + \binom{2\rho_l}{2} = \sum_{l=1}^{N} 4N(\rho_l - 1) + \rho_l.$$

The evolution along T is a single orbital basis change (requiring $2\binom{N}{2}$ Givens rotations). For this approach to be useful, one has to truncate small $\lambda_p^{(l)}$. For Fig. 2.9, where we compute the cost of this approach for the hydrogen chain, we truncate such that for each l, if we truncate $\lambda_p^{(l)}$ for $p > p_l$, we have $\sum_{p>p_l} |\lambda_p^{(l)}| \leq \varepsilon'$, where we choose $\varepsilon' = 10^{-3}$.

Gate counts for randomized product formulas

For randomized product formulas, the Hamiltonian is assumed to be given in a Pauli represention. The cost is determined by the weight λ and the target precision ε. We are given a Pauli Hamiltonian with some value of λ, and target precision ε. As discussed in Section 2.9.2, we have to perform time simulation for $t = 2^m$ and take N_m samples from a Hadamard test, with $m = 1, \ldots, M$ for $M = \lceil \log_2(\lambda \varepsilon^1) \rceil$, which gives a number of Pauli rotations as in Eq. (2.29) as

$$r = \frac{16.3\lambda^2}{\varepsilon^2}$$

Counting two-qubit gates is similar to the previous section. When counting Toffoli gates it turns out that it is helpful that all (or most) rotations are over the same angle, leading to a significantly smaller overhead in converting the number of rotations to the number of Toffoli gates.

The below approach combines ideas from [92, 110, 93, 86] in the analysis of the Toffoli gate requirements. To begin with, for qDRIFT, in every circuit *all* rotations are over the same angle. For RTE, the angle φ_n depends on the order n sampled of the term in the Taylor series. For small τ, the probability of sampling a term which is not of the lowest order scales with $\mathcal{O}(\tau^2)$. When performing time simulation for time t with $\tau = \Omega(t^{-1})$ and number of steps $r = \mathcal{O}(t^2)$ this means that we only expect $\mathcal{O}(1)$ Pauli rotations with an angle different than φ_0. This means that we expect long sequences of equiangular Pauli rotations which dominate the cost.

Additionally, any individual Pauli operator P_l with nonzero h_l in the Hamiltonian commutes with *most* of the other terms; it only anti-commutes with $\mathcal{O}(N^3)$ out of $\mathcal{O}(N^4)$ terms. For this reason, one expects relatively long sequences of commuting Pauli rotations in both qDRIFT and RTE, as observed in [86] (if the terms were selected uniformly, one would have average length $\mathcal{O}(\sqrt{N})$). Given such a sequence, $e^{i\varphi P_{l_1}} \cdots e^{i\varphi P_{l_K}}$ of length $K < 2N$, we may apply a Clifford to map

$$\{P_{l_1}, \ldots, P_{l_K}\} \to \{Z_1, \ldots, Z_K\}$$

to single-qubit Z-rotations on the first K qubits. We may now use Hamming weight phasing [92, 110, 147] to implement these rotations, as suggested for RTE in [86].

To reduce the cost even further, we can use a single catalyst phase state to implement the rotations [101, 93, 46]. We may choose the step size in the randomized product formula to our liking; in particular we may choose such that the rotations are of angle $2^{-J}\pi$ for integer J. Since we also want to choose the angle such that $\tau = \mathcal{O}(\varepsilon\lambda^{-1})$, we take $J = \lceil \log_2(\lambda\pi^{-1}\varepsilon^{-1})\rceil$. Let

$$|\phi\rangle = \frac{1}{2^{J/2}}(|0\rangle + e^{2\pi i 2^{-J}}|1\rangle)^{\otimes J} = \frac{1}{2^{J/2}}\sum_{j=0}^{2^J-1} e^{2\pi i j 2^{-J}}|j\rangle$$

be a phase state on J qubits. At the start of the computation, we initialize the ancilla phase estimation qubit, together with J qubits, in the state

$$\frac{1}{\sqrt{2}}\left(|0\rangle|+\rangle^{\otimes J} + |1\rangle|\phi\rangle\right) = (I\otimes \mathrm{QFT})\frac{1}{\sqrt{2}}(|0\rangle|0\rangle + |1\rangle|1\rangle)$$

where we label the basis for the J qubits by $\{0, 1, \ldots, 2^J - 1\}$. It (once) requires T gates or Toffoli gates to prepare this state (a constant cost which is negligible compared to the other contributions). The phase state $|\phi\rangle$ is such that if we add another register with $W \leq J$, and implement addition $|w\rangle|j\rangle \mapsto |w\rangle|j+w\rangle$, then

$$|w\rangle|\phi\rangle \mapsto e^{-2\pi i w 2^{-J}}|w\rangle|\phi\rangle$$

We can use J Toffoli gates, together with J ancilla qubits to perform the addition. The state $|+\rangle^{\otimes J}$ is invariant under addition, so we get that

$$\alpha|0\rangle|w\rangle|+\rangle^{\otimes J} + \beta|1\rangle|w\rangle|\phi\rangle \mapsto \alpha|0\rangle|w\rangle|+\rangle^{\otimes J} + \beta e^{-2\pi i w 2^{-J}}|1\rangle|w\rangle|\phi\rangle$$

To implement $\prod_{k=1}^{K} e^{i\pi 2^{-J} Z_k}$, on the first K qubits we first compute the Hamming weight on a register of $W = \lceil \log_2(K+1)\rceil$ qubits. We then apply the above trick to apply (controlled on the QPE ancilla) $e^{iw\pi 2^{-J}}$ to Hamming weight w, and we uncompute the Hamming weight. Note that if $|k\rangle$ has Hamming weight w, then

$$\left(\prod_{k=1}^{K} e^{i\pi 2^{-J} Z_k}\right)|k\rangle = e^{-i\pi(2w-K)2^{-J}}|k\rangle$$

so we have implemented the correct transformation up to a global phase $e^{-i\pi K 2^{-J}}$ which we can account for in classical postprocessing of the Hadamard test outcomes.

Computing and uncomputing the Hamming weight requires at most $K-1$ Toffoli gates applying the phase to the Hamming weight requires $J-1$ Toffoli gates. This means we use $1+(J-2)/K$ Toffoli gates per controlled Pauli rotation. The adders require respectively $(K-1)$ and $(J-1)$ ancilla qubits, giving in total $K+2J-2$ ancilla qubits.

For example, for FeMoco, we may take $K=10$ (choosing K not too large, in order to make sure that with high probability we get sequences of commuting terms of length K) and $J=17$, giving an average of 2.5 Toffoli gates per controlled Pauli rotation, and using 42 ancilla qubits.

Gate counts for partially randomized product formulas

We assume a given decomposition of H in L_D deterministic terms, and H_R with weight λ_R

$$H = \sum_{l=1}^{L_D} H_l + H_R.$$

The Trotter error concerns the decomposition into H as a sum of the H_l and treating H_R as a single term. For our estimates we use the second order Trotter-Suzuki product formula (so $p=2$ with $N_{\text{stage}}=2$). It is straightforward to extend to higher orders. Let G_{det} and G_{rand} be the (average) number of Toffolis used per unitary $\exp(-i\delta H_l)$ in the deterministic part, and per Pauli rotation for the randomized part of the Hamiltonian simulation. These can be different.

We choose the parameters ε_{qpe} and $\varepsilon_{\text{trot}}$, the time step δ for the deterministic product formula, and the number of rounds M for phase estimation as in Eq. (2.49), Eq. (2.50) and Eq. (2.51). For the compilation of the randomized part, it is convenient to choose time step τ such that we can use Pauli rotations over angles $\pi 2^{-J}$. For $m=1,\ldots,M$ we use $2N_m$ times the Hadamard tests with the partially randomized product formula to estimate (see Lemma 2.9.3)

$$B^{-1}\left\langle\psi\right| S_2(\delta)^{2^m}\left|\psi\right\rangle.$$

As explained at the end of Section 6, for RTE alone it is optimal to choose a total of $r=2t^2=2\lambda_R^2\delta^2 2^{2m}$ Pauli rotations. For the partially randomized method it is advantageous to allow for r to vary, so we introduce a factor $\kappa\geq 0$ and set $r=\kappa\lambda_R^2\delta^2 2^{2m}$, which gives a damping factor of $B\leq e^{1/\kappa}$. The number of samples N_m scales with the damping factor B^2, as discussed in Section 2.9.2. We use the observation in Section 2.9.1 to reduce the evolution time of the deterministic part by a factor of 2, so the deterministic product formula needs 2^{m-1} rather than 2^m

applications of $S_2(\delta)$. This gives a total number of gates

$$G = \sum_{m=0}^{M} 2N_m \left(G_{\mathrm{det}} N_{\mathrm{stage}} L_D 2^{m-1} + G_{\mathrm{rand}} \kappa \lambda_R^2 \delta^2 2^{2m} \right).$$

Note that if we take $L_D = 0$, this reduces (up to possibly different rounding) to the cost of phase estimation to precision $\varepsilon_{\mathrm{qpe}}$ using a randomized product formula, independent of the choice of δ. If we choose $N_m = N_M + D(M-m) = B^2(11 + 4(M-m)) = e^{2/\kappa}(11 + 4(M-m))$, the total number gates is bounded as in Eq. (2.34) and Eq. (2.36) by

$$\begin{aligned} G &\leq 2G_{\mathrm{det}} N_{\mathrm{stage}} L_D (N_M + D) 2^M + 8 G_{\mathrm{rand}} \left(\frac{N_M}{3} + \frac{D}{9} \right) \kappa \lambda_R^2 \delta^2 2^{2M} \\ &= 30 G_{\mathrm{det}} N_{\mathrm{stage}} L_D e^{2/\kappa} \frac{0.1\pi}{\delta \varepsilon_{\mathrm{qpe}}} + \frac{296}{9} G_{\mathrm{rand}} \kappa e^{2/\kappa} \frac{(0.1\pi\lambda_R)^2}{\varepsilon_{\mathrm{qpe}}^2} \end{aligned} \tag{2.53}$$

The optimal choice of κ solves a quadratic equation. Again, we require the total error to be bounded by

$$\varepsilon^2 \leq \varepsilon_{\mathrm{qpe}}^2 + \varepsilon_{\mathrm{trot}}^2 = \varepsilon_{\mathrm{qpe}}^2 + C_{\mathrm{gs}}^2 \delta^4$$

such that we substitute $\varepsilon_{\mathrm{qpe}} = \sqrt{\varepsilon^2 - C_{\mathrm{gs}}^2 \delta^4}$ in Eq. (2.53). Finally, we find the minimal cost by optimizing for L_D (which determines λ_R). In the following we discuss two strategies for decomposing the Hamiltonian into deterministic and randomized parts, and the associated gate counts.

Pauli decomposition

A natural scheme for the partially randomized method is to start from the Hamiltonian in the Pauli sum decomposition after the fermion-to-qubit mapping, $H = \sum_{l=1}^{L} h_l P_l$. We decompose into H_D and H_R by assigning the L_D terms with largest weight h_l to H_D, which makes it straightforward to find the optimal choice of L_D based on Eq. (2.53).

If we count two-qubit gates, as discussed for deterministic product formulas, the terms in the deterministic part can be ordered lexicographically in order to achieve cancellations of CNOTgates.

If we count Toffoli gates we can again apply the modified Hamming weight phasing to the randomized part, as presented in Section 2.9.5. We can additionally leverage this to reduce the gate synthesis cost of the deterministic part, as presented in above. The idea is that while in principle the deterministic part consists of terms $h_l P_l$ with arbitrary real coefficients h_l, we may round h_l to a

desired value and assign the remainder to H_R. This slightly increases the value of λ_R, but may significantly reduce the synthesis cost of time evolving along P_l in the deterministic part. This approach is similar to previous methods using randomization for compiling arbitrary-angle rotations [120, 121, 122]. We obtain further reductions by expanding the rounded h_l bitwise, and reorder terms so we can apply Hamming weight phasing.

We now explain this idea in detail. Consider the binary expansion of the rotation angle of the l-th term

$$\frac{h_l \delta}{\pi} = \sum_{J \geq 1} x_{l_i} 2^{-J} \quad \text{with} \quad x_{l_J} \in \{0, 1\}.$$

Grouping bits of the same significance and truncating to $n_{\max}$ bits of precision gives

$$\delta H = \sum_{J=1}^{n_{bit}} \underbrace{\sum_{l=1}^{L_J} \pi 2^{-J} P_l}_{\delta H_J} + \delta H_{\text{rem}},$$

where H_{rem} contains the remainding higher bits of precision. For the deterministic product formula, we now apply all the terms in H_J consecutively. This consists of L_J Pauli rotations with the same angle, so we can use Hamming weight phasing. Moreover, for the deterministic part we are free to choose the order of the Pauli rotations, and since any individual term anti-commutes with $\mathcal{O}(N^3)$ out of the total $\mathcal{O}(N^4)$ terms, one can group into long sequences of commuting terms. While decomposing into the H_J increases the number of terms, they are cheaper to implement, reducing the total cost.

We now comment on two subtleties which influence this approach. Firstly, one needs to group the terms in each H_J into commuting groups. Such groupings have been studied extensively for the sake of reducing measurement overheads when measuring energies [148, 149]. While one can typically find quite large groupings, this can be computationally intensive. For our compilation purposes, if we have groups of K commuting Pauli operatos, implementing a rotation over angle $\pi 2^{-J}$ requires $1+(J-2)/K$ Toffoli gates per Pauli. This means that groups of moderate size suffice to have a low overhead, which can be constructed using simple heuristics. As a lower bound, which we use for our numerical estimates, when considering Eq. (2.40), it is not hard to see that it is possible to group into groups of 12 commuting Pauli operators. The second subtlety is that the procedure of decomposing each term into several bits and reordering the terms will in principle affect the Trotter error. The effect of ordering on Trotter error has been studied numerically for molecular electronic structure Hamiltonians [142].

We expect that decomposing by bits of precision will not increase the Trotter error (since it decomposes into shorter time evolutions, so this should be beneficial). For simplicity we assume that the Trotter error is the same as for the fully deterministic product formula and leave a detailed numerical exploration to future work. For FeMoco, we find a total cost of 1.5×10^{12} Toffoli gates using this strategy, as shown in Fig. 2.1. Note that this approach increases the number of ancilla qubits, since we need to prepare phase states for different bits of precision J.

2.9.6 Scaling for the hydrogen chain

As a benchmark system we use the hydrogen chain. The result for $\varepsilon = 0.0016$ is shown in the main text in Fig. 2.2. We can also vary over ε as well, we can fit an expression of the form $CN^a\varepsilon^{-b}$ to the resulting cost. In Fig. 2.18 we show scalings with respect to N and ε (keeping the other variable fixed). We find a fit for the Toffoli cost of $\mathcal{O}(N^2\varepsilon^{-1.7})$.

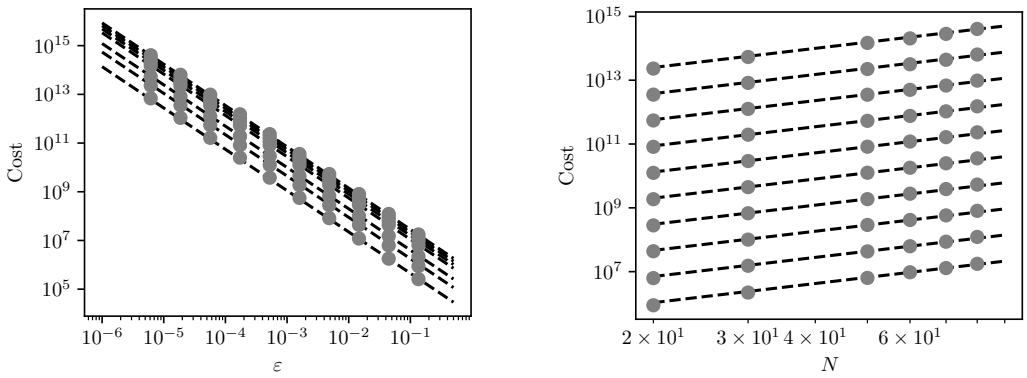


Figure 2.18: Toffoli cost for partially randomized product formulas with respect to varying N and ε, with a fit to a scaling $\mathcal{O}(N^2\varepsilon^{-1.7})$.

In Fig. 2.19 we show the number of CNOTgates required for phase estimation using deterministic, randomized and partially randomized product formulas. Note that this has a different scaling than the Toffoli count, since the number of two-qubit gates required per Pauli rotation scales with N. As for the Toffoli gate count, we observe a significant improvement using partial randomization.

2.9.7 Double factorized decomposition

If we truncate to L_D terms in the factorization, where the l-th term has rank at most ρ_l, then for the deterministic part we have cost as described in Section 2.9.5.

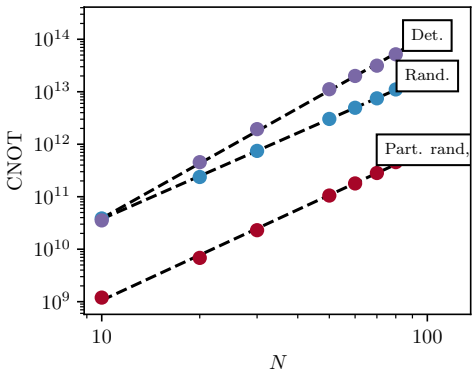


Figure 2.19: Two-qubit gate count in terms of CNOTgates required for the hydrogen chain for precision $\varepsilon = 0.0016$.

The cost of the randomized part remains the same. The performance of this approach for the hydrogen chain is shown in Fig. 2.9.

End of arxiv:2503:05647 - Phase Estimation with Partially Randomized Time Evolution

2.10 Results beyond the arxiv submission

We conclude this chapter with a few results of analyses that were performed after the first arxiv version was submitted. In particular, we performed the resource estimates for the FeMoco system also in the larger active space Hamiltonian proposed by Li [42].

First, in Fig. 2.20 we show an updated version of the FeMoco resource estimate plot (*c.f.* Fig. 2.1) for the Reiher Hamiltonian. For the partially randomized result two corrections were made. First, the number of ancilla qubits necessary for the Hamming weight phasing was initially overcounted so the number of logical qubits is now lowered. Second, due to an error in our resource estimate code, the effect of the depth reduction was not accounted for correctly in the deterministic part, hence the total cost in Fig. 2.20 is about 50% higher than initially thought. Moreover, a wrong constant prefactor in the code for qDRIFT was corrected, which now leads to a reduction of the total cost by about a factor of two. As a result, the qDRIFT costs are lower than the cost of the partially randomized method. However, as we will see below, for the more accurate Li Hamiltonian the partially randomized outperforms qDRIFT in terms of total cost.

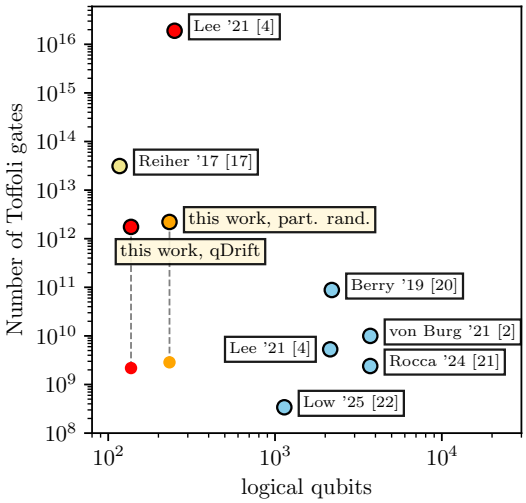


Figure 2.20: An updated version of the resource estimate plot of the Reiher Hamiltonian of FeMoco, see Fig. 2.1.

The graphs in figure 2.21 give additional insight into the performance of the partially randomized method. Subfigures 2.21a and 2.21c show how the total cost changes as the deterministic part in the partitioning increases (see Eq. (2.12)). To compare decomposition strategies with different types of terms (Pauli strings, Pauli strings bitwise via Hamming weight phasing and Cholesky vectors in double-

factorization), we use the cost of the deterministic part on the x-axis to quantify the "number of deterministic terms" in a decomposition-agnostic way.

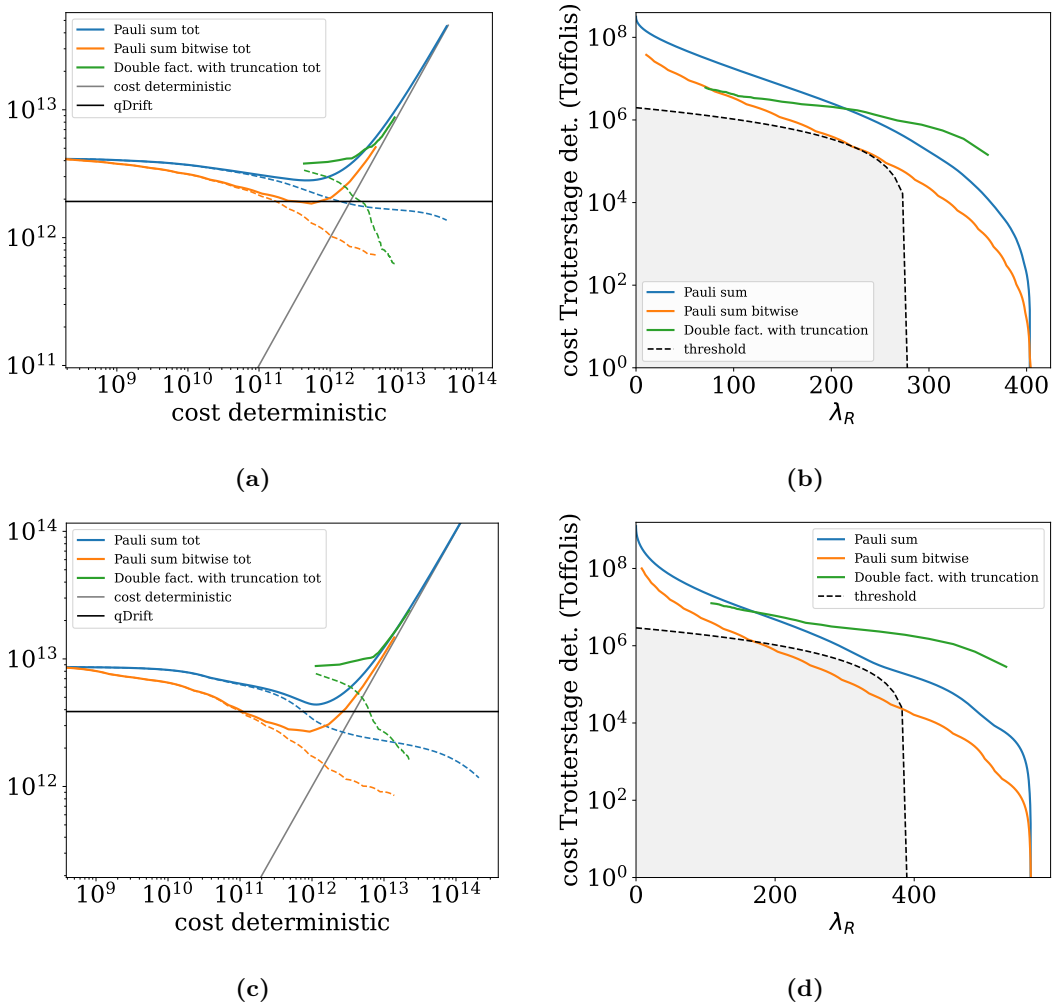


Figure 2.21: Analysis of the partitioning into deterministic and randomized parts, (a) and (b) for the Reiher Hamiltonian, (c) and (d) for the Li Hamiltonian. A value of $\xi = 0.1\pi$ was used. (a), (c) Total cost as a function of the size of the deterministic part (measured in implementation cost) for different decomposition schemes. The horizontal line denotes the qDrift cost, the shaded area depicts the region in which the partially randomized method is cheaper than qDrift. The dashed line shows the cost of the randomized part. (b), (d) The cost of the deterministic part in a single Trotter step as a function of the remaining λ_R value. The shaded area depicts the region in which the partially randomized method outperforms qDrift.

Let us consider the asymptotics of the total cost first. As a larger part of the Hamiltonian is implemented deterministically, the cost of the randomized part (dashed lines) decreases, and the total cost approaches the deterministic cost (grey slope) from above. On the other hand, as the deterministic part decreases,

the total cost plateaus and should approach the cost of the fully randomized method. Since the randomized part in the partially randomized method is implemented via the RTE method, the difference between the qDRIFT cost (black line) and the value approached by the total cost curves amounts to the difference between qDRIFT and RTE. As described in the Appendix (in the paragraph after the proof of Theorem 2.9.5), the number of samples for qDRIFT is smaller by a factor of two, which explains the differences seen in the plots here. Note that the cost of the fully randomized method is significantly lower than the fully deterministic method. This would change if ε was smaller or if the the Hamiltonian had a smaller number of terms L.

The Pauli sum decomposition (blue curve) reaches a shallow minimum that slightly improves over the full RTE method, however, not by enough to outperform the qDRIFT result. In comparison, applying the bitwise decomposition together with the Hamming-weight phasing technique to the Pauli sum decomposition (section 'Pauli decomposition' in the Appendix) consistently lowers the total cost. The minimal cost reaches the qDRIFT cost for the Reiher Hamiltonian, and slightly outperforms it for the Li Hamiltonian.

In subfigures 2.21b and 2.21d we show the cost of a partition for a given decomposition strategy from a different perspective. Each partition into randomized and deterministic parts comes with a different value of λ_R, so the x-axis in 2.21b and 2.21d is simply scanning over all partitions. Note that the cost of implementing the randomized part depends only on λ_R. The cost of implementing the deterministic part corresponding to each λ_R depends on the decomposition strategy, this is shown by the colored lines. For each λ_R we show (dashed line) the difference between the qDRIFT cost for the full Hamiltonian and the cost of the randomized part. In other words, the dashed line shows the threshold below which the partially randomized method would outperform qDRIFT. Due to the factor two cost advantage for qDRIFT, the threshold starts at a lower value λ_R than the full λ.

The bitwise Hamming weight phasing technique lowers the cost of the Pauli sum decomposition by up to one order of magnitude. Interestingly, the double factorized decomposition has a better scaling than the Pauli sum decompositions for $\lambda_R \to 0$. However, the large constants factors appearing in the cost of this decomposition make it worse than the Pauli sum decomposition overall.

Lastly, in Table 2.2 we present additional details regarding the optimal partition of the partially randomized method. The total cost G_{tot} increases as ξ decreases, since the deterministic cost of the partially randomized method scales as $G_{\text{det}}\ \xi^{-1}$. Note that the weight of the deterministic terms relative to the total weight $(1-\lambda_R/\lambda)$ is between $24\%-45\%$ with less than 1% of the terms. This data underlines that the main aim of the partially randomized method was reached: That implementing the small fraction of high weight terms deterministically can

result in more efficient phase estimation.

	Reiher (54o)			Li (76o)		
ξ	0.314	0.1	0.05	0.314	0.1	0.05
G_{tot}	$1.8 \cdot 10^{12}$	$2.2 \cdot 10^{12}$	$2.3 \cdot 10^{12}$	$2.7 \cdot 10^{12}$	$3.4 \cdot 10^{12}$	$3.7 \cdot 10^{12}$
$G_{\text{det}}/G_{\text{tot}}$	0.29	0.16	0.13	0.34	0.21	0.20
L_D	$1.2 \cdot 10^5$	$2.1 \cdot 10^4$	$9.5 \cdot 10^3$	$1.7 \cdot 10^5$	$3.5 \cdot 10^4$	$2.0 \cdot 10^4$
L_D/L	0.0083	0.0014	0.00064	0.0046	0.00096	0.00053
$1 - \lambda_R/\lambda$	0.45	0.29	0.24	0.56	0.41	0.35

Table 2.2: Optimized parameters of the partially randomized method (Pauli sum decomposition with Hamming weight phasing) for the Reiher [40] and the Li [42] Hamiltonians of FeMoco. Here, $\xi = \arcsin((1-\eta)/\eta)$ is the depth reduction parameter, G_{tot} and G_{det} are total and deterministic cost, L_D is the number of terms in the deterministic part, L is the total number of terms, and λ_R and λ are the sum of weights of the randomized part and the total Hamiltonian, respectively.

Chapter 3

phase2: full-state vector simulation of quantum time evolution at scale

There are many quantum algorithms that have heuristic components. An obvious example in the context of chemistry is the variational quantum eigensolver, which relies on an accurate wavefunction ansatz and the success of a classical optimization problem [150, 151]. But even the resource estimates of Trotter-based phase estimation algorithms contain heuristic elements, namely the estimation of the Trotter error. It is empirically observed that analytic bounds of the Trotter error are larger than their actual values, often by orders of magnitude, leading to inflated resource estimates. Computing the errors exactly is as hard as computing the ground state energy, so in practice empirical estimates need to be used. A strategy to obtain reasonable estimates is to compute the exact Trotter error for system sizes and extrapolate the result. Therefore, until sufficiently powerful quantum computers become available, it is useful to perform classical simulations of quantum algorithms to give a realistic account of a quantum algorithm's actual performance. In this chapter we present such a simulation method and use it to get improved Trotter error estimates.

The main contributions of this paper are: (i) a fast full-state vector simulation software (ii) a large-scale simulation of the time-evolution of a chemical Hamiltonian with up to 40 qubits, and (iii) the computation of the Trotter error for the partially randomized method, showing significantly lower values than the analytic upper bounds suggest.

This chapter is a reproduction of arXiv:2504:17881.

phase2: full-state vector simulation of
quantum time evolution at scale

Abstract:
Large-scale classical simulation of quantum computers is crucial for benchmarking quantum algorithms, establishing boundaries of quantum advantage and exploring heuristic quantum algorithms. We present a full-state vector simulation algorithm and software implementation designed to perform HPC simulation of layers of rotations around a string of Pauli operators. We demonstrate robust scalability of the simulation method on large distributed CPU and GPU systems. Our distributed computation harnessed up to 16384 CPU cores and 512 NVIDIA H100 GPUs, using 32 TB of memory. The simulator significantly outperforms other high-performance libraries, showing a typical speedup of 10−100 for large-scale multi-GPU workloads. As a first application of our approach, we report a numerical experiment aimed at simulating exactly Hamiltonian dynamics of up to 40 qubits to investigate the Trotter error for a quantum chemistry problem. Bounding the Trotter error is important for evaluating the cost of quantum algorithms for chemistry, and rigorous bounds are often conservative, as our simulations confirm. Our software, specifically designed for quantum time evolution applications, is also well equipped to manage circuits that utilize standard gate sets.

Authors:
Marek Miller, Jakob Günther, Freek Witteveen, Matthew S. Teynor, Mihael Erakovic, Markus Reiher, Gemma C. Solomon and Matthias Christandl

Reference: arxiv:2504:17881
Date: April 2025

3.1 Introduction

The field of quantum computation is approaching the exciting era of practically useful quantum computations which are out of reach or highly challenging for

classical hardware and conventional algorithms. This necessitates the development of optimized classical high-performance simulators for quantum computing, for two reasons. Firstly, classical simulators set a benchmark against which to compare potential quantum advantage. The second reason is that many of the quantum algorithms which are most likely to be run on near-term hardware with limited resources have no performance or accuracy guarantees. This makes evaluating their *empirical* performance particularly challenging, especially in the regime where the computations are highly quantum mechanical.

There are various paradigms for classical simulation of quantum computers, based on tensor network representations [152, 153, 154, 155, 156], stabilizer decompositions [157] and Pauli propagation [158, 159]. For a review, further references and a comparison of different methods, see [160]. These methods have different properties, but all either have exponential scaling cost in the quantum circuit size, or make significant approximations (or both). This work reports on an implementation of the most direct classical simulation method, full state vector simulation. Here, all non-zero amplitudes of the quantum state are stored in memory, and updated with each operation the quantum algorithm performs. The obvious disadvantage of this approach is that its memory requirements scale exponentially in the number of qubits. The key advantage, compared to other approaches, is that it simulates to numerical precision the complete process the quantum computer performs. Moreover, it returns all amplitudes of the final state, and intermediate amplitudes are also available. Since the main goal of classical simulation of quantum computers is the validation of unknown or uncontrollable errors in the quantum computation, using approximate methods for classical simulation has the significant drawback that these introduce additional (and generally difficult to control) errors. This means that full state vector simulation plays an important role in benchmarking quantum algorithms, as well as in benchmarking alternative (approximate) classical simulation algorithms which can handle larger system sizes. In conclusion, although the number of qubits for which state vector simulation can be performed is limited by exponential scaling, it is an important tool. For practical use, optimized implementations which reduce the total computational resources are crucial. Various state vector circuit simulators are available, including open-source options such as Qiskit [161] and QuEST [162], as well as proprietary software packages like NVIDIA's cuQuantum [163] library. A recent overview of state vector simulators with benchmark data can be found in [164]. This work introduces a new implementation, called `phase2`, improving and complementing previous implementations.

The way `phase2` achieves improved performance is by restricting the set of operations to *Pauli rotations* and optimizing their implementation. That is, the simulator can implement unitaries of the form $\exp(i\phi P)$ where P is a Pauli operator and ϕ an arbitrary angle. Single-qubit and two-qubit Pauli rotations already

form a universal gate set, so `phase2` can simulate any circuit, with a small conversion overhead. The simulator also natively implements rotations along arbitrary high-weight Pauli operators. These naturally occur in many relevant applications. A prime example are simulations of electronic structure in chemistry, material science and physics. There, the Hamiltonian of the system can be represented as a linear combination of Pauli operators. For this reason, Pauli rotations naturally appear in most Variational Quantum Eigensolver (VQE) approaches for quantum chemistry, such as Ansatzes based on unitary coupled cluster theory [150], the Hamiltonian variational Ansatz [165], and ADAPT-VQE [166]. Beyond applications in simulating quantum physics, variational algorithms for optimization problems, notably the Quantum Approximate Optimization Algorithm (QAOA) also often use Pauli Hamiltonians to encode the problem instance, and the resulting quantum circuits are naturally expressed in terms of Pauli rotations [167]. With these examples, it is fair to say that most proposals for heuristic quantum algorithms are based on Pauli rotation circuits. For these Ansatzes detailed benchmarking of their performance and optimization landscape is crucial.

In this work, we showcase the simulator on a different use case. The most fundamental primitive in quantum algorithms for simulating quantum physics is Hamiltonian simulation. This can be used in quantum phase estimation to estimate energies, or it can be used directly to simulate dynamic properties. There are various quantum algorithms for Hamiltonian simulation. The simplest approach is based on Trotterization, or product formulas. These come with a discretization error, which can be controlled rigorously. However, the error bounds can be loose, and to evaluate the expected performance in practice it is important to empirically study its scaling for realistic systems. This issue has been analyzed extensively in previous numerical studies [54, 81, 53, 40], showing that the rigorously derived upper bounds for standard first- and second-order Trotter formulas can be overly conservative, differing from empirical measurements by several orders of magnitude. Product formulas lead to deep quantum circuits, scaling with the number of terms in the Hamiltonian. Since Hamiltonians in quantum chemistry have a large number of terms, we use a partial randomization scheme [3] to deal with large systems. We have used the software to simulate Hamiltonian dynamics of up to 40 qubits for a sequence of active spaces of a ruthenium complex. This study investigates the Trotter error resulting from non-standard Hamiltonian partitioning in a partially randomized Trotter product formula, using extensive HPC simulations.

3.2 Results

We have developed an exact simulation method to simulate the application of sequences of Pauli rotations on large quantum systems. See Section 3.2.1 for the full description of our simulation technique. We demonstrate near-optimal scaling of our simulation algorithm on CPU and GPU clusters (see Section 3.2.2), as well as its efficiency and performance that surpasses existing implementations by several orders of magnitude (see Section 3.2.3). We conducted a numerical experiment to examine the Trotter error of the time evolution given by the Hamiltonian representing the ruthenium complex NKP-1339 (see Section 3.2.4 and Section 3.4.4). The section Section 3.2.5 presents in detail the results of the computation.

3.2.1 Software implementation

Our method employs a straightforward simulation technique in which the quantum state $|\psi\rangle$ of a n-qubit system is represented as a contiguous array of 2^n complex probability amplitudes. Explicitly storing all amplitudes has the advantage of providing the highest precision of numerical calculation. However, for the number of qubits larger than $n = 30$, it requires significant computational resources: fast random access memory and processing power greater than that provided by modern personal computers. We give an optimized implementation for applying circuits consisting of Pauli rotations $\exp(i\phi P)$. This is particularly relevant for Trotter approximations of time evolution of Hamiltonians in chemistry and material science. As a benchmark problem, we use the simulator for deterministic (Trotter) and (partially) randomized product formulas for Hamiltonian simulation [3]. Here the time evolution along a Hamiltonian $H = \sum_l \alpha_l P_l$ which is a linear combination of Pauli operators is approximated by a sequence of Pauli rotations along the P_l over small angles. See Section 3.4.3 below for an outline of the quantum algorithms simulated in this study.

Our goal is to study the accuracy of quantum algorithms capable of obtaining numerical data with precision comparable to that of classical methods in computational chemistry and biology. In this context, other techniques, such as tensor network contraction [168], would be unsatisfactory due to their inherent errors, confounding the study of the precision of quantum algorithms. Using various optimization strategies, which we discuss in detail below, we successfully simulated Hamiltonian time evolution of a chemically relevant 40-qubit system on a multi-GPU cluster. This large-scale simulation required more than 32 TB of random access memory and was distributed on 512 NVIDIA H100 GPU devices. A similar simulation with 36 qubits was performed on a CPU cluster using up to 16 384 CPU cores and 1 TB of distributed memory. As a special case of the Schrödinger algorithm for simulating quantum computers [169, 170], we call our

technique: *full-state vector simulation of quantum time evolution* and provide a formal model in Section 3.4.1.

Starting from the initial state $|\psi\rangle$, the simulation algorithm proceeds by applying the sequence of the unitary operators $e^{i\varphi\tilde{P}}$ on $|\psi\rangle$, where $\varphi \in \mathbb{R}$, and $\tilde{P} = P_1 \otimes P_2 \otimes \ldots \otimes P_n$, $P_k \in \{I, \sigma_x, \sigma_y, \sigma_z\}$, $k = 1, 2, \ldots n$. We call the operator $\tilde{P}$ a *Pauli string*, and the unitary $e^{i\varphi\tilde{P}}$, a *Pauli rotation*. Observing that for every Pauli string $\tilde{P}$, $e^{i\varphi\tilde{P}} = \cos\varphi\, I + i \sin\varphi\, \tilde{P}$, we can effect the action of the unitary on the array of amplitudes in two steps by (i) a transformation, given by $\tilde{P}$, similar to transposition of the array, and (ii) by taking a linear combination of elements. This is a consequence of the particular structure of $\tilde{P}$ as a direct sum of 2-by-2 matrices, for all n and the Pauli operators $P_1, P_2, \ldots, P_n$. (See Section 3.4.1 for further details, as well as for proofs of other mathematical facts stated in this section.)

Drawing from previous work on high-performance simulation of quantum circuits, notably the QuEST software library [162, 171], we developed a system geared toward a large-scale distributed workload, with the representation of the quantum state array divided evenly between a power-of-two number of concurrent processes. In parallel computing, inter-process communication is typically achieved with *Message Passing Interface* (MPI). We refer to concurrent processes as MPI *tasks*. We copy QuEST's approach to efficient synchronization between the tasks by keeping buffers of size equal to the size of the local partition of the distributed state. Although this method effectively doubles the amount of memory required for the simulation, it does not increase overall data transfer and has the benefit of greatly simplifying the addressing of the elements of the distributed array. This in turn makes it possible for multiple unitaries of the form $e^{i\varphi\tilde{P}}$ to be applied in a single simulation step, see Eq. (3.1) below.

We then proceed to optimize the simulation algorithm by observing that: (i) every Pauli string $= P_1 \otimes P_2 \otimes \ldots \otimes P_n$, for $n \leq 64$, can be efficiently represented as two 64-bit integers in such a way that the result of the action of $\tilde{P}$ on a computational basis state $|i\rangle$, $i = 0, 1, \ldots 2^n - 1$, is obtained by using only a minimal number of logical instructions: AND, XOR, and bit parity. This effectively makes processing elements of the array of amplitudes as fast as memory access time. (ii) The particular form of the unitary $e^{i\varphi\tilde{P}}$ as a direct sum of 2-by-2 matrices reduces the data transfer required for distributed simulation to pairwise exchanges between MPI tasks, occurring simultaneously for all tasks. For a multi-GPU simulation, the same matrix structure enables the execution of the entire unitary in parallel using compact CUDA kernels. This approach offers the benefit of eliminating the requirement of a global synchronization or locking system to maintain the coherence of distributed states across concurrent tasks. (iii) In order to minimize further the data transfer, we group similar Pauli

rotations using the following identity:

$$\prod_{l=1}^{L} e^{i\varphi_l \tilde{P}_l \otimes \tilde{Q}} = \prod_{l=1}^{L} e^{i\varphi_l \tilde{P}_l} \otimes \frac{I+\tilde{Q}}{2} + \prod_{l=1}^{L} e^{-i\varphi_l \tilde{P}_l} \otimes \frac{I-\tilde{Q}}{2}, \tag{3.1}$$

where $\varphi_l \in \mathbb{R}$, $\tilde{P}_l$ and $\tilde{Q}$ are Pauli strings, $l = 1, 2, \ldots, L$. If two terms, say $e^{i\varphi_1 \tilde{P}_1 \otimes \tilde{Q}}$ and $e^{i\varphi_2 \tilde{P}_2 \otimes \tilde{Q}}$, share a common suffix $\tilde{Q}$ of length m, such that 2^m is the number of MPI tasks participating in the computation, only one pairwise exchange of local array partitions is needed to achieve the action of the concatenated unitary $e^{i\varphi_1 \tilde{P}_1 \otimes \tilde{Q}} e^{i\varphi_2 \tilde{P}_2 \otimes \tilde{Q}}$ on the distributed state. In practice, for a real-life Hamiltonian that consists of 10^5 terms and which we used in our numerical experiment, we observe frequent grouping of several thousand similar terms, with the simulation state divided among as many as $2^{10} = 1024$ MPI tasks. This approach greatly reduces the data transfer overhead, and hence the overall simulation time. See Section 3.2.3 section below.

We have developed a simulation system designed to facilitate efficient computation on both CPU and GPU clusters within a typical HPC execution environment. Our simulator implementation, `phase2`, provides a library of numerical routines to perform effective operations on a distributed array that represents a quantum state. The library includes all three quantum algorithms presented in [3]. In addition, we have written a non-interactive application: `ph2run` that coordinates the computation, provides a command-line user interface, and manages input and output operations. Both the library and the associated application are implemented in standard C, utilizing `MPI` [172] for inter-process communication and `HDF5` [173] for managing parallel I/O. The GPU version of the library employs custom CUDA kernels, which were compiled, optimized, and linked with `nvcc` for the NVIDIA Hopper architecture: `sm_90a`. For GPU computation, our strategy involves a single MPI task per GPU, with one CPU core acting as a host that controls the CUDA device, and GPU devices communicating via MPI directly using the CUDA Unified Memory Model [174]. We automate the computation through standard Unix shell scripting and `Makefiles`.

3.2.2 Scalability

The simulation algorithm exhibits effective weak and strong scaling [175, 176]. Fig. 3.1 demonstrates weak scaling of the implementation. The simulation speed, measured as the average number of Pauli rotations per second per CPU core (or per a GPU device in the case of the GPU-based computation) decreases proportionally to the size of the distributed quantum state, which scales like 2^n, where n is the number of qubits. The figure indicates that the state distribution technique is efficient up to the largest simulated system size. As the simulation speed varies

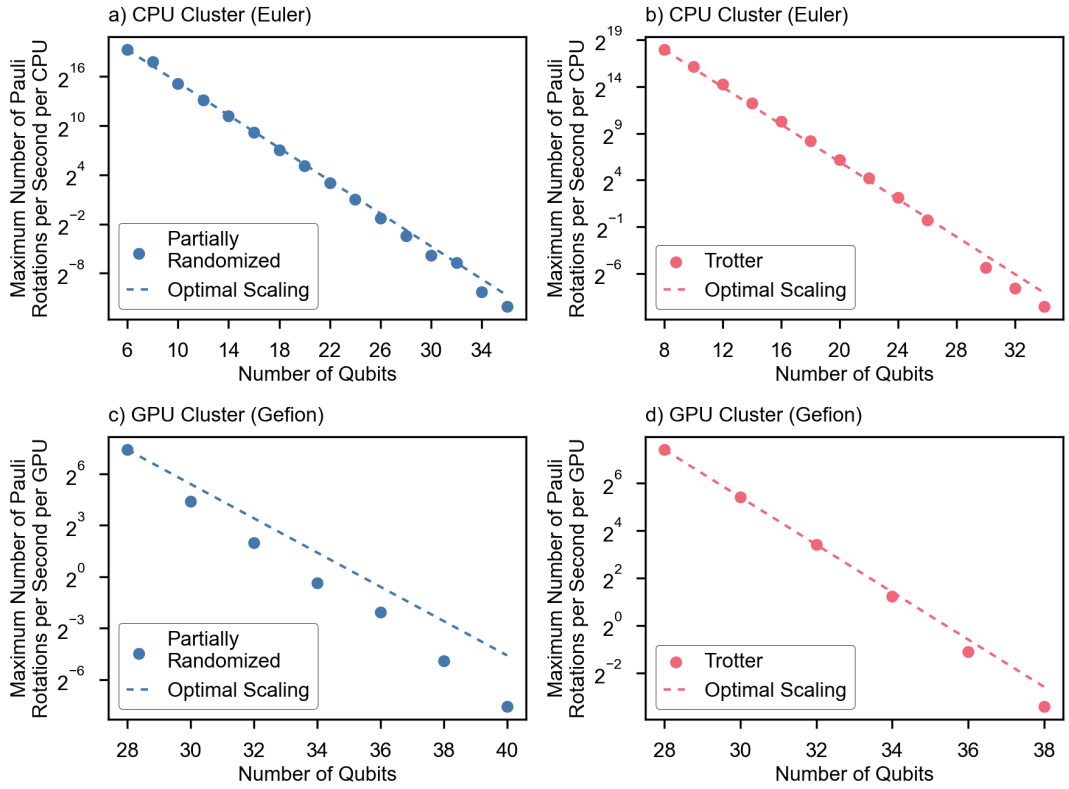


Figure 3.1: Weak scaling of the simulation algorithm. The simulation speed, measured by the best (maximum) number of Pauli rotations per second per number of compute units (CPUs or GPUs) should ideally scale inversely proportionally to the problem size: 2^n, where n is the number of qubits. Top row, a) and b), CPU simulation (Euler); bottom row: c) and d) GPU simulation (Gefion). Blue: partially randomized algorithm, red: Trotter (deterministic) algorithm. Dashed lines indicate optimal scaling with the slope coefficient: -1.00.

greatly between simulation runs due to factors such as (i) Hamiltonian structure, (ii) the choice of the quantum algorithm: explicit Trotterization vs. randomized, and of the simulation parameters, (iii) cluster interconnect speed and node occupancy, in the figure we use as the measure of the computation speed only the fastest case. On the other hand, Fig. 3.2 illustrates strong scaling: for a fixed size of the computational task, determined by the size of the distributed quantum state *and* the total number of Pauli rotations, increasing the number of CPU cores or GPU devices results in a nearly linear speedup on the log-log scale. In other words, using twice as many CPU cores or GPU devices decreased the simulation time by a factor of two, indicating that computational resources are allocated and utilized efficiently. The strong scalability of our simulation technique makes it an adequate approach for a large-scale simulation. As the

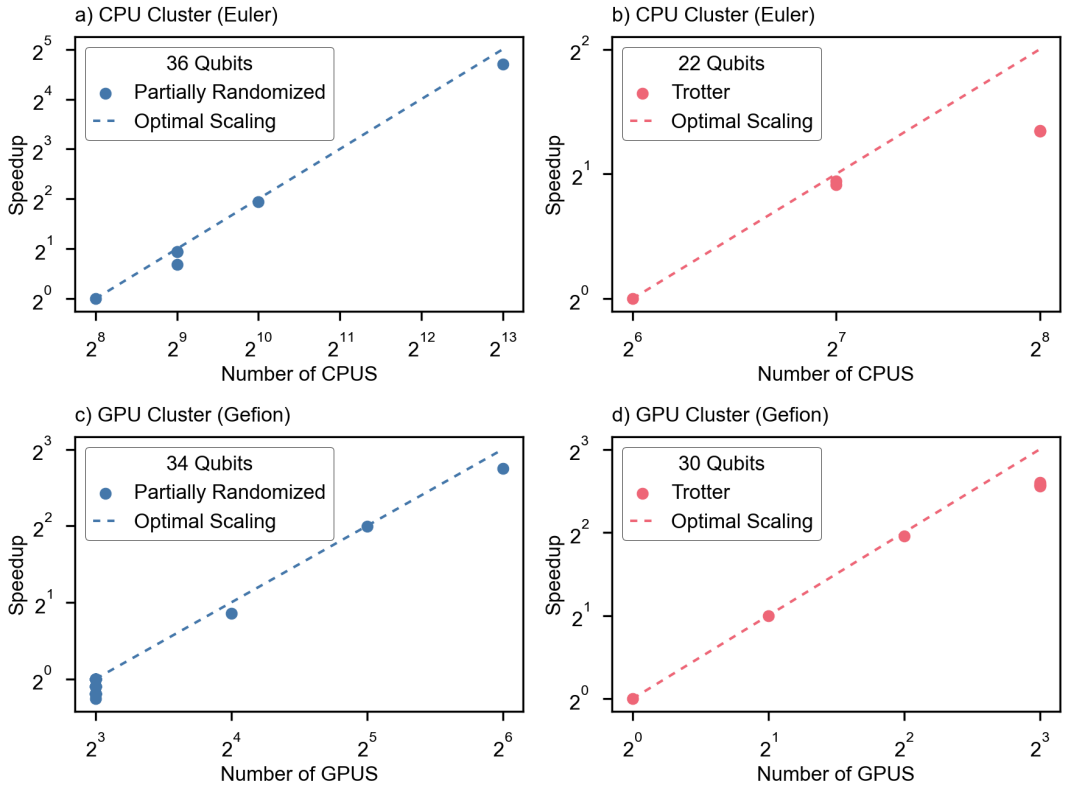


Figure 3.2: Strong scaling of the simulation algorithm. Each panel shows speedup of the total simulation time for a fixed problem size: the number of qubits and the number of Pauli rotations, with varying amount of parallel computational resources: CPU cores or GPU devices. Top row, a) and b), CPU simulation; bottom row, c) and d), GPU-based computation. Blue: Partially Randomized, red: Trotter (deterministic) algorithm. Dashed lines indicate optimal scaling.

computational power of supercomputer facilities increases over time, our simulation algorithm will be able to handle even larger system sizes and more complex simulations with improved efficiency.

3.2.3 Performance

With the optimizations discussed so far, our computational problem becomes *distributed-memory bound*. Memory and network speed are left as the primary factors that affect the total simulation time. Nevertheless, our implementation achieved remarkable efficiency in utilizing the processing power of the latest hardware. For example, for the system of $n = 28$ qubits, the implementation of the partially randomized quantum algorithm was capable of performing on average 169 Pauli rotations per second while running on a single NVIDIA H100 GPU.

Given the size of the simulation state in this case to be 8 GB, the algorithm processed the array of amplitudes at a rate of 1.32 TB/s, which is consistent with the upper limit of the GPU memory bandwidth, as stated in the manufacturer's specifications of this hardware model: 3.35 TB/s.

We assess the relative performance of the simulation algorithm by comparing our implementation with the quantum circuit simulation library, QuEST. We chose QuEST as a performance reference due to (i) similarities in design, since our work was directly inspired by QuEST, as well as (ii) QuEST being among the fastest software packages available to simulate large quantum circuits [164], and (iii) QuEST capabilities to perform distributed simulation in the HPC execution environment described above. We emphasize that although our implementation clearly appears to be faster, an assessment like the benchmark we discuss here ignores other aspects of both software packages, such as their functionality and scope, where it is QuEST that excels thanks to its comprehensive and user-friendly API. Our benchmark involves measuring the time needed to perform a group of $L = 1$, 10, or 100 Pauli rotations that share a common suffix $\tilde{Q}$ as in Eq. (3.1), for the number of qubits $n = 16, 17, \ldots, 31$, and the number of CPU cores: 1, 8, and 128. We also performed an analogous measurement for the GPU-based computation, with the number of devices: 1 and 8. Refer to Section 3.4.2 for a detailed comparison summary. The measurements presented in Fig. 3.3, show a significant speedup of our implementation over QuEST. We observe a reduced simulation time in nearly all runs: the larger the workload, the more pronounced the difference. In the case where there is no grouping of similar terms, i.e. $L = 1$, the increase in the computation speed is of the order of 1 to 10. In the case of grouping of up to $L = 100$ Pauli strings, we observe a drastic speedup of more than 100, and in the most efficient case of $n = 31$-qubit 8-GPU simulation, the speedup is more than 800. See Table 3.1 in Section 3.4.2 for the exact measurement results. This discrepancy can easily be explained by QuEST's lack of some of the optimizations based on Eq. (3.1), which have been implemented in our software specifically with Hamiltonian time evolution in mind. Without these improvements, each Pauli rotation involved both MPI and the CUDA device synchronization step, impeding the performance for larger simulation states.

The simulation software we introduced, along with our strategy for large-scale distributed computation of Hamiltonian time evolution and significant theoretical advances of quantum algorithms developed to analyze Hamiltonian dynamics [3], greatly enhances the simulation capabilities of these algorithms compared to state-of-the-art methods. This advancement enables us to explore a novel simulation regime of up to 40 qubits, which we will demonstrate now as a practical application of our software.

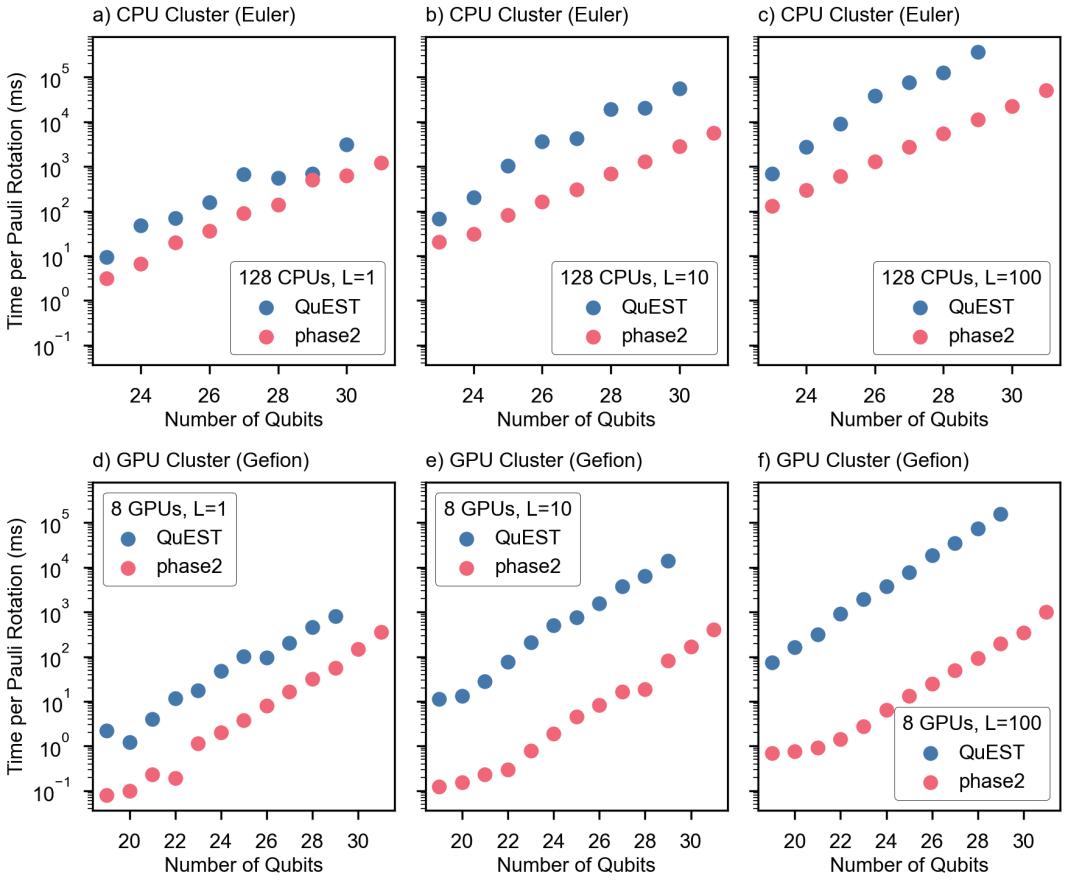


Figure 3.3: Comparing the performance of the simulation software `phase2` with QuEST. Plots show the average time [ms] needed to execute a single Pauli rotation for different group sizes L as in Eq. (3.1). a) and d): $L = 1$, b) and e): $L = 10$, c) and f): $L = 100$. Blue: QuEST, red: `phase2`. Number of CPUs: 128 (top row), number of GPUs: 8 (bottom row).

3.2.4 Experimental setup

The numerical calculation to estimate the Trotter error constant for a Hamiltonian model of a ruthenium ligand NKP-1339 in a drug-protein complex [177, 178], was performed between January and March 2025 on the two HPC facilities: (i) a CPU cluster Euler at ETH Zürich, and (ii) a NVIDIA GPU cluster Gefion, operated by Danish Centre for AI Innovation (DCAI) and located in Copenhagen. On Euler, the simulation of a system of 36 qubits used up to 16 384 CPU cores (AMD EPYC 9654 CPU) and 8 TB of memory, distributed among 64 nodes. The Gefion architecture consists of hybrid servers (NVIDIA DGX nodes) equipped with 2 Intel Xeon Platinum 8480C processors and 8 NVIDIA H100 GPUs. For our simulation, we used up to 64 DGX nodes (512 GPUs), and 32 TB

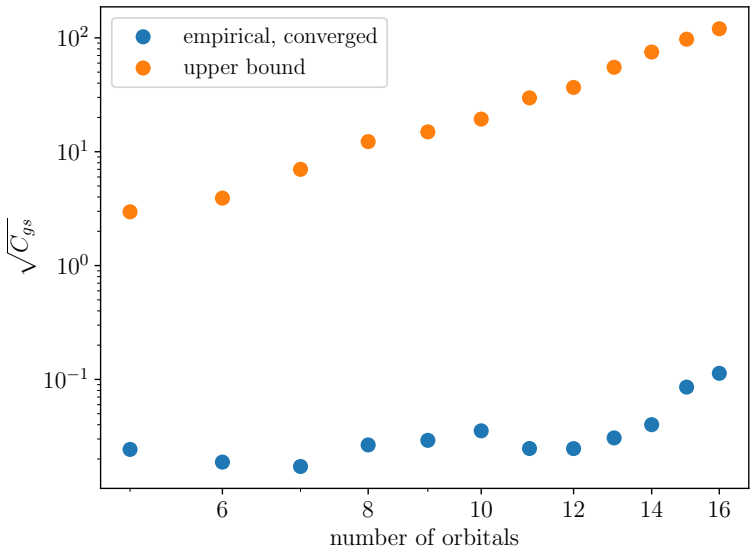


Figure 3.4: Scaling of the Trotter error factor as a function of system size (the number of qubits equals twice the number of orbitals). More details are provided in table 3.2.

of GPU memory to simulate the time evolution of a 40-qubit Hamiltonian. See Section 3.4.4 for the description of the simulation input data.

The study leveraged the two supercomputers in a complementary way: the Euler cluster, with its large number of cores, was better suited for parallelizing multiple smaller simulation runs and for calibrating the entire simulation system of up to 36 qubits. In contrast, Gefion, having fewer but faster GPU devices, was more efficient for larger workloads, and hence the main numerical experiment was performed largely on Gefion. The simulation work was divided into several hundred jobs that typically lasted up to 24 hours. The complete calculation, including the testing and deployment phase, took approximately $1.33 \cdot 10^7$ CPU $\cdot$ h and $5 \cdot 10^4$ GPU $\cdot$ h. An approach to improve the efficiency of the calculation by leveraging Gefion's hybrid architecture and distributing the simulation workload among both GPUs and CPUs for the same computation proved unsatisfactory. This was largely due to the distributed computation speed being bound by the network bandwidth, and hence utilizing additional compute units within the same server node brought no significant increase in simulation speed.

3.2.5 Simulation

Following the validation of the presented algorithm and its implementation, we now present an example where we used the simulator to estimate the performance of quantum computers. We consider *product formula* approaches to Hamiltonian

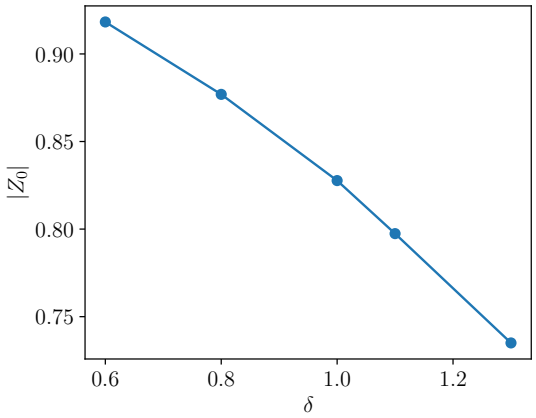


Figure 3.5: Amplitude of the signal $Z_0 = \langle\psi| e^{i\delta\tilde{H}(\delta)} |\psi\rangle$ for various times δ for the 40 qubit simulation of a 20 orbital active space, performing $1.26 \cdot 10^4$ Pauli rotations for each data point.

simulation. Here a time evolution $\exp(iHt)$ for $H = \sum_l \alpha_l P_l$ is discretized in small time steps, and performing $\exp(i\phi P_l)$ for small angles ϕ (exactly the native operations of the simulator). However, this does introduce a discretization error. This error scales with the system size. For electronic structure, the system size is measured in terms of the number of (spatial) orbitals, where N orbitals correspond to a Hilbert space of $2N$ qubits. We measure the error in terms of the ground state energy error, which is the relevant metric when using Hamiltonian simulation as a subroutine for ground state energy estimation using quantum phase estimation.

For the numerical results we use a Hamiltonian which models a ruthenium ligand complex, see section Section 3.4.4 for details. To deal with the large number of terms in the Hamiltonian, we use a partial randomization scheme [3]. We simulate quantum phase estimation to reproduce the ground state energy as it would be computed by a quantum computer. In Fig. 3.4, we present one of the main results of the simulation: the scaling of the Trotter error with system size, up to 32 qubits (16 orbitals). This scaling is captured by a parameter C_{gs}, defined in section Section 3.4.3. For quantum algorithms using product formula simulations, the cost scales with $\sqrt{C_{gs}}$. We compare the results with an easy-to-compute analytical upper bound. We find that our numerically computed values of $\sqrt{C_{gs}}$ are around 100 times smaller than the theoretical upper bound. Consequently, for the task of ground-state energy estimation, our results imply significantly lower cost compared to the rigorous but loose upper bound on C_{gs}. The empirically computed data points in Fig. 3.4 show that C_{gs} generally increases with system size, as expected, but only moderately so. The numerical calculation exhibits a less regular relation than then the theoretical scaling of the upper bound, making it rather challenging to find precise estimates of C_{gs} based on the upper bound

alone.

We also performed simulations for larger system sizes, corresponding to 17, 18, 19 and 20 orbitals, or 34, 36, 38 and 40 qubits, respectively. In this regime, simulating phase estimation with reasonable precision required quantum circuits exceeding our computational resources. However, we could still perform time evolution using a partially randomized product formula and find that the simulator produces accurate results. For the 40-qubit simulations, we performed a distributed computation on 64 NVIDIA DGX H100 nodes, with 512 H100 GPUs having simultaneous access to the simulation state, performing $1.26 \cdot 10^4$ Pauli rotations. This posed significant challenges for the numerical implementation. In Fig. 3.5 we show the resulting amplitude for short time values using only a single time step in the discretization method. The large dimension of the state space, 2^n, implies that an incorrect implementation of $e^{i\delta\tilde{H}(\delta)}$ would with high probability result in an amplitude very close to zero. We see that this is not the case and that the amplitude has the correct expected behavior. This validates the correctness of the simulation even in the largest analyzed case. See Section 3.4.5 for more details on the analysis of the simulation of the ruthenium ligand.

3.3 Discussion

For the purpose of simulating exactly Hamiltonian time evolution of complex biomolecules using quantum computing techniques, we have developed and optimized a software package geared toward distributed large-scale computation on CPU and GPU clusters. We have demonstrated the scalability of our implementation up to the largest systems of 40 qubits, given by the available computational resources. The performance of our simulation system, thanks to several highly specific optimizations, compares favorably to the simulator library QuEST, one of the best of its class [164].

In [171] the benefit of expressing the Pauli rotation as a direct sum of linear transformations for distributed computing was discussed. In our work, this mathematical property of Pauli operators was utilized as one of the key optimizations. The main speedup of our implementation, however, was obtained by combining it with a new approach based on Eq. (3.1), together with a first-of-its-kind efficient multi-GPU implementation.

Our empirical findings show that the observed error is much lower than suggested by the theoretical bounds. This discrepancy highlights the potential for executing similar computations using current quantum hardware, implying enhanced feasibility and reduced complexity. Additionally, when considering future simulations, it may be beneficial to adopt an inverse approach: by conducting

simulations with a greater permissible error, one could align with the theoretical limits while incurring substantially lower simulation cost.

The challenge of accurately simulating time evolution driven by real-life Hamiltonians associated with biological processes necessitates the precision achievable only through a full-state vector simulation of quantum circuits. Due to the fact that the circuit depth typically surpasses the capabilities of current general-purpose software simulators, we have developed a specialized implementation of the quantum algorithms discussed in [3] that significantly reduces the simulation time by a factor of 10-100. Compared with conventional classical techniques for estimating the ground state energy of chemical systems, our method ensures accuracy guarantees that we anticipate will be a cornerstone in the application of quantum computers.

3.4 Methods

3.4.1 Mathematical model

For a natural number $n = 2, 3, \dots 64$, let $\mathcal{H}$ be the Hilbert space of the system of n qubits, $\mathcal{H} = (\mathbb{C}^2)^{\otimes n}$, $\dim \mathcal{H} = 2^n$. A pure state $|\psi\rangle \in \mathcal{H}$ is represented as an array of 2^n complex probability amplitudes: $[a_i] = [a_0, a_1, \dots, a_{2^n-1}]$, $a_i \in \mathbb{C}$, $i = 0, 1, \dots 2^n - 1$, such that $|\psi\rangle = \sum_{i=0}^{2^n-1} a_i |i\rangle$. Using the C99 type: `_Complex double` to store complex numbers in computer memory, the total space needed for the state is $2 \cdot 8 \cdot 2^n = 2^{n+4}$ bytes.

For a given number m, $1 \leq m < n$, let $M = 2^m$ be the number of MPI processes (tasks) that participate in the computation. Let also $N = 2^{n-m}$. Our technique is to allocate one CPU core or one GPU device per task. The array $[a_i]$ is then divided into M parts of equal size: $A_0, A_1, \dots A_{M-1}$, containing consecutive amplitudes. For example, $A_0 = [a_0, a_1, \dots a_{N-1}]$, and $A_{M-1} = [a_{2^n-N}, \dots, a_{2^n-1}]$. We denote by $A_k(i)$ the i-th element of the k-th part A_k. Each MPI task also allocates a buffer B_k, $k = 0, 2, \dots, M-1$, of size equal to the size of A_k. Hence, the total amount of memory required for the simulation is 2^{n+5} bytes. Dividing the Hilbert space into $n-m$ "lower" and m "upper" qubits: $\mathcal{H} = (\mathbb{C}^2)^{\otimes n-m} \otimes (\mathbb{C}^2)^{\otimes m}$, by writing a computational basis state $|i\rangle$ as: $|i\rangle = |i_0 i_1 \dots i_{n-m}\rangle\, |i_{n-m+1} \dots i_n\rangle$, where $(i_n i_{n-1} \dots i_1 i_0)_2$ is the binary representation of i, $i = 0, 1, \dots 2^n - 1$, we can write the quantum state $|\psi\rangle$ in the form:

$$|\psi\rangle = \sum_{k=0}^{M-1} \sum_{i=0}^{N-1} A_k(i)\, |i\rangle |k\rangle = \sum_{k=0}^{M-1} \sum_{i=0}^{N-1} a_{i+k\cdot 2^{n-m}} |i\rangle |k\rangle. \tag{3.2}$$

Now it is easy to see that a unitary operator $U \otimes I_m$, acting as an identity on the upper qubits, can be implemented by the same matrix operation per-

formed by all MPI tasks in parallel on their respective partitions A_k. If U_{ij} are matrix elements of the unitary U in the computational basis, $U_{ij} = \langle i|U|j\rangle$, $i, j = 0, 1, \ldots, N-1$, we have

$$U \otimes I_m|\psi\rangle = \sum_{k=0}^{M-1}\sum_{j=0}^{N-1} A_k(j)\, U|j\rangle|k\rangle = \sum_{k=0}^{M-1}\sum_{i,j=0}^{N-1} U_{ij}A_k(j)\,|i\rangle|k\rangle. \tag{3.3}$$

Hence, the action of $U \otimes I_m$ on $|\psi\rangle$ amounts to a local transformation $[A_k(i)] \mapsto [\sum_j U_{ij}A_k(j)]$ of the partitions of the array by each MPI task in parallel.

On the other hand, to implement a unitary $I_{n-m} \otimes U'$, the tasks need to exchange their partitions with each other. In a special case where $U' = \tilde{P} = P_1 \otimes P_2 \otimes \ldots \otimes P_m$, $P_k \in \{I, \sigma_x, \sigma_y, \sigma_z\}$, $k = 0, 1, \ldots, m$, because $\tilde{P}^2 = I_m$ and $\tilde{P}^\dagger = \tilde{P}$, the space $(\mathbb{C}^2)^{\otimes m}$ splits into 2^{m-1} 2-dimensional orthogonal subspaces, invariant under the action of $\tilde{P}$. To prove this statement, let us note first that it is clearly true, if $\tilde{P}$ is a diagonal operator in the computational basis, i.e. when all P_k are either I or σ_z. Furthermore, it is easy to see that if the statement holds true for two Pauli strings $\tilde{P}_1$ and $\tilde{P}_2$ such that $\tilde{P}_1\tilde{P}_2 = \tilde{P}_2\tilde{P}_1$, it also holds for $\tilde{P}_1\tilde{P}_2$, as Pauli strings are Hermitian operators. Hence, we can reduce the proof to the special case when all $\tilde{P}_k$ are equal to the identity operator except for exactly one of them, e.g. $\tilde{P} = \sigma_x \otimes I \otimes \ldots \otimes I$. In that case, the statement can be verified explicitly by writing out the matrix elements of $\tilde{P}$ in the computational basis. By this reasoning, to achieve the action of $I_{n-m} \otimes \tilde{P}$ on the state $|\psi\rangle$, the task k needs to obtain only the amplitudes from the task k' such that $\tilde{P}|k\rangle = \omega_k|k'\rangle$, where $\omega_k \in \{1, i, -1, -i\}$, and *vice versa*. Note that in the case where $\tilde{P}$ is a diagonal operator, it might as well happen that $k' = k$. In any case, the map $k \mapsto k'$ is a bijection for every Pauli string, and hence the action of $I_{n-m} \otimes \tilde{P}$ reduces to a pairwise exchange of partitions A_k and $A_{k'}$ for $k = 0, 1, \ldots, 2^m - 1$. More precisely,

$$I_{m-n} \otimes \tilde{P}|\psi\rangle = \sum_{k=0}^{M-1}\sum_{i=0}^{N-1} \omega_k A_k(i)\,|i\rangle|k'\rangle = \sum_{k=0}^{M-1}\sum_{i=0}^{N-1} \overline{\omega_k} A_{k'}(i)\,|i\rangle|k\rangle, \tag{3.4}$$

where the last step follows from swapping the indices k and k', since $\tilde{P}$ is a bijection, and from the fact that $\omega_{k'} = \omega_k^{-1} = \overline{\omega_k}$, the complex conjugation, since $\tilde{P}^2 = I$. The task k can store the amplitudes $A_{k'}$, multiplied by the factor $\overline{\omega_k}$, in their buffer B_k. This is the main mechanism that allows us to implement efficiently the action of a Pauli rotation $e^{i\varphi\tilde{P}}$, $\varphi \in \mathbb{R}$, on the distributed state $|\psi\rangle$.

A further enhancement is to observe that the network traffic between MPI tasks can be reduced to a single buffer exchange in the case where a sequence of

Pauli rotations share the same Pauli string on upper qubits as in Eq. (3.1):

$$\prod_{l=1}^{L} e^{i\varphi_l \tilde{P}_l \otimes \tilde{Q}} = \prod_{l=1}^{L} e^{i\varphi_l \tilde{P}_l} \otimes \frac{I+\tilde{Q}}{2} + \prod_{l=1}^{L} e^{-i\varphi_l \tilde{P}_l} \otimes \frac{I-\tilde{Q}}{2}.$$

where $\tilde{P}_l = P_{l,1} \otimes \ldots \otimes P_{l,n-m}$ are Pauli strings that act on the lower qubits and $\tilde{Q} = Q_1 \otimes Q_2 \otimes \ldots \otimes Q_m$ is a Pauli string on the upper qubits, which is constant for $l = 1, 2, \ldots, L$. This fact can be easily proved by induction over L. Indeed, as a consequence of the properties of Pauli strings: $\tilde{P} = \tilde{P}^\dagger$, $\tilde{P}^2 = I$, $\tilde{Q} = \tilde{Q}^\dagger$, $\tilde{Q}^2 = I$, from the definition of the matrix exponent we have

$$\begin{aligned} e^{i\varphi\tilde{P}\otimes\tilde{Q}} &= \cos\varphi I \otimes I + i\sin\varphi\tilde{P}\otimes\tilde{Q} = \\ &= \left(\cos\varphi I + i\sin\varphi\tilde{P}\right)\otimes\frac{I+\tilde{Q}}{2} + \left(\cos\varphi I - i\sin\varphi\tilde{P}\right)\otimes\frac{I-\tilde{Q}}{2} \\ &= e^{i\varphi\tilde{P}}\otimes\frac{I+\tilde{Q}}{2} + e^{-i\varphi\tilde{P}}\otimes\frac{I-\tilde{Q}}{2}, \end{aligned} \quad (3.5)$$

for every Pauli string $\tilde{P}$ and $\varphi \in \mathbb{R}$. This proves Eq. (3.1) for $L = 1$. The inductive step follows immediately from multiplying the right-hand sides of Eqs. (3.1) and (3.5).

If $|\psi'\rangle = (I_{n-m} \otimes \tilde{Q})|\psi\rangle$ is a state obtained by a one-time pairwise buffer exchange associated with $\tilde{Q}$, and stored in the auxiliary buffers B_k, the final state can be obtained by a linear combination of states:

$$\prod_{l=1}^{L} e^{i\varphi_l \tilde{P}_l \otimes \tilde{Q}} |\psi\rangle = \frac{1}{\sqrt{2}} \left(\prod_{l=1}^{L} e^{i\varphi_l \tilde{P}_l} \otimes I_m \frac{|\psi\rangle + |\psi'\rangle}{\sqrt{2}} + \prod_{l=1}^{L} e^{-i\varphi_l \tilde{P}_l} \otimes I_m \frac{|\psi\rangle - |\psi'\rangle}{\sqrt{2}} \right). \quad (3.6)$$

The result of Eq.(3.6) can be obtained in three steps: (i) For $|\psi\rangle$ distributed among partitions $[A_k]$, $k = 0, 1, \ldots, M-1$, load the state $|\psi'\rangle$ into the auxiliary buffers $[B_k]$. (ii) By taking an in-place linear combination, load the state $\frac{|\psi\rangle+|\psi'\rangle}{\sqrt{2}}$ into $[A_k]$ and $\frac{|\psi\rangle-|\psi'\rangle}{\sqrt{2}}$ into $[B_k]$. (iii) Perform the action of $\prod_{l=1}^{L} e^{\pm i\varphi_l \tilde{P}_l} \otimes I_m$ by all MPI tasks in parallel on lower qubits and store the linear combination in $[A_k]$.

By choosing an efficient representation of Pauli strings in computer memory, we can perform the action of $\tilde{P} = P_1 \otimes P_2 \otimes \ldots \otimes \tilde{P}_{n-m}$ on lower qubits at a rate comparable to the memory access time for all elements of the local array $[A_k]$. Generally, because by assumption $n \leq 64$, we can represent an n-qubit Pauli string as two 64-bit integer numbers: $\tilde{P} \approx (p_1, p_2)$, $0 \leq p_1, p_2 < 2^{64}$, such that the n-th bit of p_1 is set if and only if the Pauli operator P_n is *not* diagonal, $P_n \in \{\sigma_x, \sigma_y\}$, whereas the n-th bit of p_2 is set whenever $P_n \in \{\sigma_y, \sigma_z\}$. For example, for the Pauli string $\sigma_x \otimes I \otimes \sigma_y$, $p_1 = (101)_2 = 5$, and $p_2 = (100)_2 = 4$.

With this representation, for a computational basis state $|i\rangle$, $i = 0, 1, \ldots, 2^n - 1$, we obtain immediately that $\tilde{P}|i\rangle = \omega|j\rangle$, where $j = p_1 \oplus i$, and the symbol $\oplus$ stands for 'exclusive OR', and ω is a 4th root of unity, $\omega \in \{1, i, -1, -i\}$. The value of ω is computed from the bit parity of $p_1 \wedge p_2$ (the number of i's, i.e. the number of σ_y operators in the string $\tilde{P}$), and $p_2 \wedge i$ (the number of "minuses": when either σ_y or σ_z act on the n-th qubit state: $|1\rangle$).

In order to further improve the efficiency of the simulation algorithm, one needs to address the fact that at this stage the computation is memory bound. A way forward would be to pipeline memory access and hide memory and network latency with overlapping computational steps [179]. For example, with a straightforward modification to the present implementation, one could execute more than one Pauli rotation in the time required to access all amplitudes of the simulated state by traversing the array in a different than linear order. Before such efficiently pipelined computation is possible, however, challenges remain in how to streamline non-local access to a distributed array in a way that would be as efficient as the algorithm outlined above.

3.4.2 Benchmarking methodology

The benchmark compares our implementation with QuEST version 3.7.0 and measures the time needed to perform a group of $L = 1$, 10, or 100 Pauli rotations that share a common suffix $\tilde{Q}$ as in Eq. (3.1). The number of qubits ranges from $n = 16$ to $n = 31$, and the number of CPU cores is 1, 8, and 128. A similar measurement was performed for the GPU-based computation, with the number of GPU devices: 1 and 8.

It should be noted that although we strive to make the computation using our software and QuEST as comparable as possible, the two versions do not perform exactly the same calculation intrinsically despite our best effort. For example, the implementation of the Pauli rotation $e^{i\varphi\tilde{P}}$, given by the function `statevec_multiRotatePauli()` as of QuEST version 3.7.0 is less efficient than ours, even in the most straightforward single-CPU case. The same benchmark application is linked with four object files, each containing the compiled and optimized version of the core routine to perform Pauli rotations: using our implementation in the CPU and GPU case, and using QuEST compiled with MPI or CUDA support, using `cmake` compile flags: `-DDISTRIBUTED=ON` and `-DGPUACCELERATED=ON` respectively. Since as of version 3.7.0, QuEST lacks direct multi-GPU support, for our benchmark case involving 8 GPUs, QuEST routines perform the computation locally on individual GPUs, and the MPI data transfer is handled by the benchmark application, mimicking QuEST's approach for the CPU-based computation. We perform the same computation for our implementation and for QuEST, measuring the duration needed to apply the rotations on

the array of amplitudes. Because exact values of the quantum amplitudes are irrelevant for timing the execution, we initialize the simulation state with random values. This process is repeated ten times for each of the benchmark parameters, after which we compute the average. The measurements were performed on Gefion's single NVIDIA DGX H100 server with 2 Intel Xeon Platinum 8480C CPUs and 8 NVIDIA H100 GPUs.

Table 3.1 shows the ratio of the time to execute a group of length $L = 100$ of Pauli rotations for QuEST to the equivalent time for `phase2`, interpreted as a speedup achieved by the latter simulation technique.

no. of qubits	CPU: 1	CPU: 128	GPU: 1	GPU: 8
19	5.45	-	77.37	109.95
20	5.75	-	236.68	211.79
21	5.57	-	174.64	335.14
22	5.66	-	183.62	636.79
23	4.81	5.22	187.38	708.95
24	4.10	9.15	291.96	580.37
25	7.65	14.53	293.80	582.29
26	-	29.53	350.94	744.18
27	-	28.36	490.78	714.81
28	-	22.96	335.45	783.17
29	-	33.36	-	801.95

Table 3.1: Speedup of performing a group of Pauli rotations for $L = 100$, calculated as the ratio between the time needed by QuEST vs. `phase2`. Columns correspond to benchmarking the CPU implementation for the number of MPI tasks: 1 or 128, and the GPU implementation with 1 or 8 devices.

3.4.3 Hamiltonian Simulation Description

The algorithm we test is an improved method for computing molecular ground state energies on a quantum computer, combining the Robust Phase Estimation algorithm (RPE) with Hamiltonian simulation implemented via partially randomized product formulas [3]. An integrated analysis, see Section 3.4.5, improves error bounds significantly compared to previous work. To the authors' knowledge, it is the first application of partially randomized Hamiltonian simulation methods to molecular Hamiltonians. Our numerical optimizations provide improved resource estimates, particularly in the low-qubit regime. Moreover, our approach benefits from reduced circuit depth in the presence of high-overlap initial states, as often encountered in chemistry problems [3].

In general, we consider a Hamiltonian of the form

$$H = \sum_{l=1}^{L} h_l \tilde{P}_l \tag{3.7}$$

where $h_l \in \mathbb{R}$ and $\tilde{P}_l$ are tensor products of Pauli operators on N qubits. We denote by $\lambda = \sum_{l=1}^{L} |h_l|$ the sum of absolute values of coefficients. The goal is to find its ground-state energy, E_0, within some precision $\varepsilon > 0$. We denote by $|\psi_l\rangle$ the l-th eigenstate of H, and let $|\psi\rangle$ be an initial state with ground state overlap $|\langle\psi_0|\psi\rangle|^2 \geq \eta$, for $1 \geq \eta > 0$. Phase estimation protocols solve the ground-state energy task by extracting spectral information from the phase during Hamiltonian time evolution. They achieve the optimal scaling in terms of precision, requiring a total evolution time $\mathcal{O}(\varepsilon^{-1})$ in order to resolve the ground state energy to precision ε (Heisenberg scaling). A standard way to implement the time evolution is to discretize it into a sequence of Trotter steps, which approximate the exact time evolution for small steps sizes δ:

$$e^{i\delta H} \approx \prod_{l=1}^{L} e^{i\delta h_l \tilde{P}_l} = e^{i\delta \tilde{H}(\delta)}. \tag{3.8}$$

This describes the first-order Trotter formula, which can be generalized to higher orders [51]. These come with a higher overhead, but have improved asymptotic scaling in terms of ε. In this work, we focus on the second-order method. For the purpose of ground-state energy estimation, we are interested in constraining the difference between the ground-state energy of the effective Hamiltonian $\tilde{H}(\delta)$ and that of H. For δ small enough, the Trotter error on the ground state energy behaves as

$$\varepsilon_{\text{trott}} = C_{gs}\delta^2, \tag{3.9}$$

where C_{gs} is a Hamiltonian-dependent constant (so it depends on system size). It is desirable to take δ large, because the total number of Trotter steps scales as $\mathcal{O}(\varepsilon^{-1}\delta^{-1})$. On the other hand, to keep $\varepsilon_{\text{trott}}$ sufficiently small, we require $\delta < \sqrt{\varepsilon/C_{gs}}$. Consequently, an accurate estimate of C_{gs} is crucial in order to choose the optimal step size δ and by this to use only as few Trotter steps as necessary. One disadvantage of fully deterministic Trotter product formulas is that their cost in terms of the number of Pauli rotations scales explicitly with L, the number of terms. For electronic structure Hamiltonians from chemistry, we have $L \sim N^4$, where N is the number of spin orbitals (or qubits).

Randomized methods, such as qDRIFT and its generalizations [55, 86] are an alternative to Trotter formulas. Their runtime does not explicitly depend on L, but instead scales with λ^2. The drawback is that they come with a worse

scaling in terms of precision (ϵ^{-2} vs. $\epsilon^{-3/2}$ for the 2nd order Trotter). This approach outperforms the deterministic method if many terms with small weights are present.

Both approaches are combined in the partially randomized method, in which high-weight terms are treated deterministically and small-weight terms are sampled randomly [3]. More precisely, the Hamiltonian is split into a deterministic and randomized part,

$$H = H_D + H_R = \sum_{l=1}^{L_{det}} h_l P_l + \sum_{l=L_{det}+1}^{L} h_l P_l \tag{3.10}$$

such that $L_{det} \ll L$ and $\lambda_R \ll \lambda$. The total cost will be of the form

$$C_{tot} = C_D + C_R = a\frac{L_D}{\varepsilon\delta} + b\frac{\lambda_R^2}{\varepsilon^2}, \tag{3.11}$$

where a and b are constants relating to the phase estimation protocol. Similarly to the fully deterministic Trotter formula, the step size δ cannot be too large, as otherwise $\varepsilon_{\text{trott}}$ will be larger than the targeted precision. Thus, the value of $\varepsilon_{\text{trott}}$ (and hence C_{gs}) is a parameter that affects the total cost C_{tot}, with large C_{gs} leading to increased cost. The question we are addressing in this numerical study is: How does the value of C_{gs} compare against its analytic upper bound? The split as in Eq. (3.10) differs from standard Trotter formulas in that it contains one term H_R consisting of many individual terms of the Hamiltonian. Consequently, it remains uncertain *a priori* whether the understanding of the behavior of C_{gs} in the conventional Trotter framework can be applied to the partially randomized context.

3.4.4 Hamiltonian preparation

The Hamiltonian data used for the simulation represents an embedded fragment of the biologically active ruthenium complex NKP-1339 [177, 178] in the binding pocket of the protein GRP78 [180] that is based on a hybrid model embedding protocol developed in Refs. [181, 182]. To obtain this hybrid model, two-level QM/QM/MM embedding was employed, where the large QM region in the initial QM/MM model consisted of the entire NKP-1339 complex [181], which was embedded in the protein environment parametrized with the Amber ff99sb*-ILDN force field [183, 184]. The technical details of this QM/MM electrostatic-embedding model have been described in detail in Ref. [181]. The electronic structure of this large QM region was described [181] with Kohn–Sham density functional theory using Perdew, Burke, and Ernzerhof's exchange–correlation

functional PBE [185] with Grimme's D3 dispersion correction [186] and Becke–Johnson damping [187] using the def2-SVP basis set [188].

In the second-level quantum-in-quantum embedding, the Hamiltonian for a smaller QM region (so-called *quantum core* [182]) embedded in the larger QM region of the QM/MM model was obtained by Huzinaga-type projection embedding, performed with the Serenity quantum chemistry program, [189, 190]) along the lines of what we have previously reported on for a different system in Ref. [182] (see Fig. 3.6 for a molecular structure of the QM regions in the QM/QM/MM model). For this work, we performed active orbital space selection on the set of occupied valence orbitals assigned to the small QM region and on all virtual orbitals of the large QM region using the autoCAS algorithm [191, 192, 193]. The autoCAS algorithm selects active orbitals based on an approximate density matrix renormalization group calculation [194] with bond dimension 250 using the QCMaquis program [195] (interfaced with the PySCF program [196] for the calculation of the integrals in the Hartree-Fock molecular orbital basis). Different active spaces were determined on the basis of the largest single-orbital entropy values obtained from the autoCAS calculation. The Hamiltonians obtained in this way describe the system with 5 to 20 spatial orbitals, mapped via the canonical Jordan-Wigner transformation to a quantum register of 10 to 40 qubits, respectively. The numbers of terms of the mapped Hamiltonian range from 875 (10 qubits) to 229 140 (40 qubits). The symmetry-shift technique [10], specifically a variant described in [3], was used to reduce the value of λ.

To demonstrate that guiding states with high overlap can be efficiently prepared and to generate the reference data, DMRG calculations were converged for the constructed Hamiltonians with a bond dimension set to 1024, which was found to be sufficient to ensure convergence of the DMRG energies for the active spaces considered. The ordering of the orbitals on the DMRG lattice was determined from the Fiedler vector of the mutual information [197] obtained from the autoCAS calculation [192, 193]. The matrix product state (MPS) wavefunctions obtained from these converged DMRG calculation (which will be referred to as the reference MPSs) were taken as a good approximation for the exact ground states for the overlap estimation. The guiding state preparation strategy involved the reference wavefunction expansion into the Slater determinants basis and taking into consideration only a small number of the determinants with the largest contribution to the expansion. The most significant determinants and the corresponding coefficients were calculated from the reference MPSs with the sampling-reconstruction of the complete active space (SR-CAS) algorithm [198].

Finally, the active Hamiltonians are partitioned into a deterministic and a randomized part. The optimal split is found by minimizing C_{tot} in Eq. (3.10) with respect to L_D, with the constraint that $\varepsilon_{\text{trott}} = C_{gs}\delta^2$ must not be larger than ε (we are using $\varepsilon = 0.001$ throughout this work). This poses a problem: We want

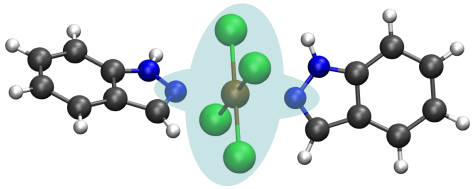


Figure 3.6: Molecular structure of NKP-1339, a drug molecule that is a ruthenium complex taken from a QM/MM hybrid model of the full drug-target protein complex studied in Ref. [181]. The fully-embedded small QM region used for the Hamiltonian construction is indicated in transparent blue. Carbon atoms are indicated in black, hydrogen atoms in white, nitrogen atoms in blue, chlorine atoms in green, and the ruthenium atom (center) in brown.

to study C_{gs} corresponding to the optimal partitioning, but finding the optimal partitioning requires us to know C_{gs}. We side step this 'chicken-or-egg' problem by employing a numerical fit of C_{gs} to λ from [3] to set the partitioning. Although this fit is purely heuristic, it is a good proxy that allows us to probe reasonable Hamiltonian partitionings, for which the exact value of C_{gs} is then computed. See Table 3.2 for the resulting sizes of the deterministic parts (corresponding to a term order with decreasing magnitude of the coefficients).

3.4.5 Data analysis

For a general Hamiltonian decomposition, $H = \sum_\alpha H_\alpha$, an upper bound on the Trotter error C_{gs} is given by [54]

$$C_{gs} \leq 4 \sum_{\alpha=1} \|H_\alpha\| \left(\sum_{\beta:[H_\alpha,H_\beta]\neq 0} \|H_\beta\| \right)^2 , \tag{3.12}$$

where $\|\cdot\|$ denotes the operator 2-norm. Adapted to the partially randomized decomposition of Eq. (3.10), we get the upper bound

$$C_{gs} \leq 4 \sum_{l=1}^{L_{det}} |h_l| \Big(\sum_{k:[\tilde{P}_l,\tilde{P}_k]\neq 0}^{L_{det}} |h_k| \Big) + \lambda_R(\lambda-\lambda_R)^2 + (\lambda-\lambda_R)\lambda_R^2, \tag{3.13}$$

where we assumed that all deterministic terms fully anti-commute with H_R. The resulting upper bound for the active space Hamiltonians of the ruthenium ligand are plotted in Fig. 3.4.

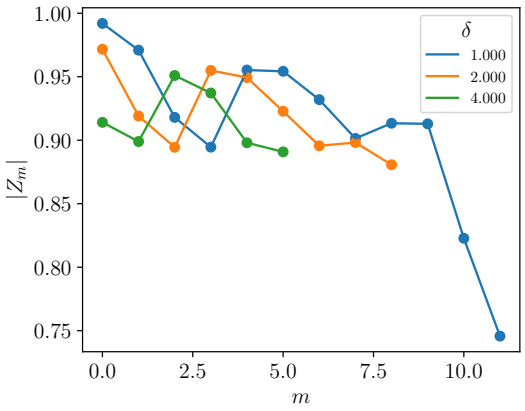
To estimate C_{gs} empirically, several step sizes δ are chosen, and the ground state energies of the corresponding effective Hamiltonians $\tilde{H}(\delta)$ is computed. The latter is done by classically simulating quantum phase estimation. We use the Robust Phase Estimation (RPE) protocol for which the signal is computed by exactly simulating the partially randomized time evolution [103]. We utilize ground states and their corresponding energies, calculated via DMRG, as exact references. Given that we have an accurate ground state energy E_0, it suffices to determine the ground state energy of $\tilde{H}(\delta)$ with enough precision to distinguish it from the exact E_0. The larger the difference (*i.e.*, the Trotter error $\varepsilon_{\text{trott}}$) is, the fewer phase estimation rounds are needed to resolve it. Since the Trotter error behaves as $\varepsilon_{\text{trott}} = C_{gs}\delta^2$, choosing large step sizes δ reduces the simulation depth. In Fig. 3.7, we illustrate the numerical analysis to extract C_{gs}.

The numerical simulator is used to compute the time signal Z_m of the 2nd order partially randomized method for each RPE round m,

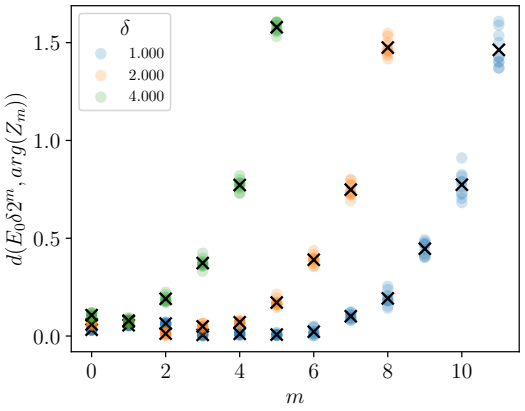
$$Z_m = \langle\psi| \, e^{i\delta\tilde{H}(\delta)2^m} \, |\psi\rangle \,, \tag{3.14}$$

where $e^{i\delta\tilde{H}(\delta)}$ is the Trotter step unitary corresponding to the partitioning in Eq. (3.10). Given a step size δ, a maximal RPE round M is chosen sufficiently large such that the estimate of $\varepsilon_{\text{trott}}$ is converged. If in the end $\varepsilon_{\text{trott}}$ is not converged, the procedure is repeated with a higher maximal M. The deterministic terms are ordered lexicographically, which speeds up their execution via the optimization described in Eq. (3.1) and below. We remark that the Trotter error depends on the ordering of terms, but previous work did not find a systematic differences for the lexicographic ordering compared to other orderings [141].

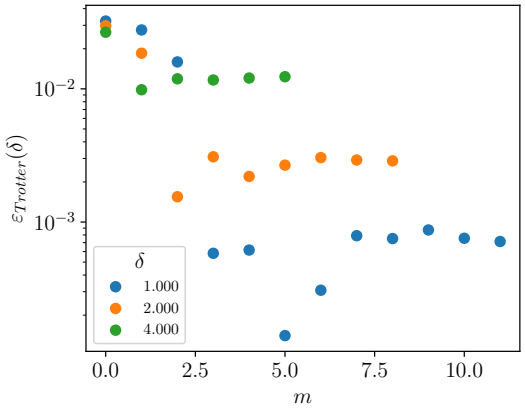
For the randomized part, we follow the derivation in [3] and introduce a parameter $\kappa = \frac{\delta}{0.2\pi}$, and set the number of random samples per Trotter stage is to $r = \lceil \kappa\lambda_R^2\delta^2 2^M \rceil$. To simulate RPE round m, $2r2^m$ terms are sampled from the randomized part H_R, which will then be executed in the 2^m Trotter steps (another factor of 2 appears because we are using the 2nd order Trotter method). Unlike the deterministic part, the terms in the randomized part are not reordered since the independence of the randomized terms is necessary to implement $e^{-i\delta H_R}$ correctly [3]. As a consequence, randomized terms are computed on average more slowly. For example, for the 20 orbital system the average speed difference may be as large as 100 times, so the randomized part of the Trotter step becomes the computational bottleneck. For systems larger than 14 orbitals we reduced the number of randomized terms by a factor of up to 3 to reduce the runtime cost. Reducing the number of samples lowers the amplitude of the signal. However, since the success of the RPE algorithm relies on estimating the complex phase of the signal accurately, and not its magnitude, we find that a signal with dampened amplitude still allows us to compute the energy correctly. This is illustrated in Fig. 3.8 for a 14-orbital active space.



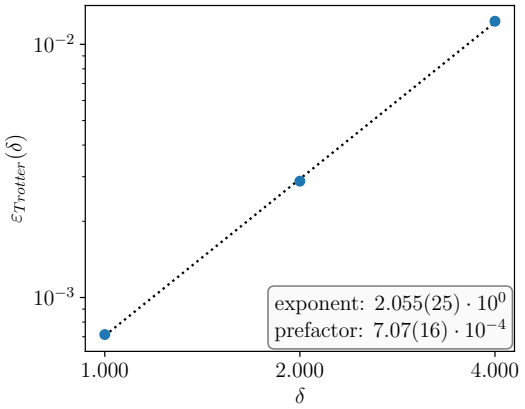
(a) Amplitude of the time signal $|Z_m|$ as a function of the RPE round m with step size δ.



(b) Phase difference between exact and partially randomized time evolution. Circles represent independent circuit runs, crosses denote averages.



(c) Convergence of Trotter error $\varepsilon_{\text{trott}}$ during RPE algorithm.

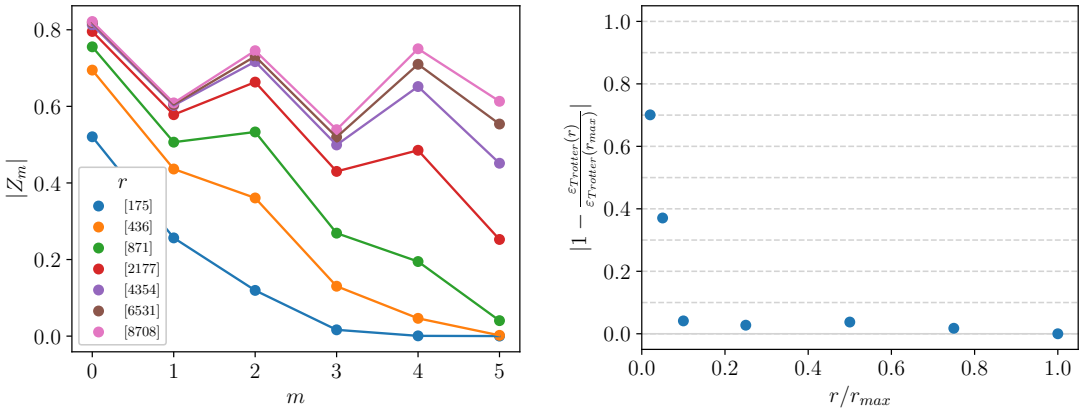


(d) Power law fit of Trotter error $\varepsilon_{\text{trott}}(\delta)$ to step size δ.

Figure 3.7: Numerical analysis to extract the Trotter error prefactor C_{gs} from the simulated time signals Z_m.

Data availability

The source code of the simulator software: `phase2-v0.13.1`, published under BSD-3-Clause license, is available at: `https://doi.org/10.5281/zenodo.15188683`. The Hamiltonian data (ruthenium complex NKP-1339), simulation results, and benchmark data are available at: `https://doi.org/10.5281/zenodo.15188556`.



(a) Amplitude of the time signal $|Z_m|$ as a function of the RPE round m for different numbers of random samples r.

(b) The relative error in $\varepsilon_{\text{trott}}$ as a function of r, relative to the estimate at the maximum number of samples r_{max}.

Figure 3.8: The effect of reducing the number of random samples r, here for the 14-orbital active space of the ruthenium ligand, using a Trotter step size of $\delta = 4$.

no. orb.	no. tot.	no. det.	λ	λ_R/λ	RPE m	C_{gs} fit	a fit	C_{gs} bound	max circ.
5	875	155	2.6	0.079	18	$5.90 \cdot 10^{-4}$	2.01	$8.80 \cdot 10^{0}$	$1.25 \cdot 10^{9}$
6	1800	170	3.1	0.11	15	$3.53 \cdot 10^{-4}$	2.02	$1.53 \cdot 10^{1}$	$2.05 \cdot 10^{8}$
7	3355	277	4.6	0.12	13	$2.95 \cdot 10^{-4}$	1.98	$4.91 \cdot 10^{1}$	$1.75 \cdot 10^{7}$
8	5758	362	6.4	0.15	12	$7.07 \cdot 10^{-4}$	2.06	$1.51 \cdot 10^{2}$	$1.46 \cdot 10^{7}$
9	9249	429	7.1	0.18	12	$8.51 \cdot 10^{-4}$	2.06	$2.23 \cdot 10^{2}$	$4.60 \cdot 10^{7}$
10	14156	574	8.2	0.20	14	$1.25 \cdot 10^{-3}$	2.14	$3.74 \cdot 10^{2}$	$3.10 \cdot 10^{8}$
11	20813	895	10.8	0.21	13	$6.12 \cdot 10^{-4}$	2.40	$8.82 \cdot 10^{2}$	$1.01 \cdot 10^{9}$
12	29512	970	12.4	0.22	11	$6.10 \cdot 10^{-4}$	2.09	$1.35 \cdot 10^{3}$	$5.10 \cdot 10^{7}$
13	40755	837	16.2	0.23	10	$9.39 \cdot 10^{-4}$	2.27	$3.04 \cdot 10^{3}$	$2.04 \cdot 10^{7}$
14	54876	872	19.8	0.23	8	$1.60 \cdot 10^{-3}$	1.90	$5.64 \cdot 10^{3}$	$4.90 \cdot 10^{6}$
15	72071	1053	23.4	0.24	9	$7.34 \cdot 10^{-3}$	2.27	$9.48 \cdot 10^{3}$	$6.56 \cdot 10^{6}$
16	92968	1322	27.0	0.24	8	$1.28 \cdot 10^{-2}$	2.24	$1.45 \cdot 10^{4}$	$2.83 \cdot 10^{6}$
17	119087	1873	31.0	0.27	6	-	-	$2.36 \cdot 10^{4}$	$1.37 \cdot 10^{6}$
18	150088	2396	36.0	0.28	4	-	-	$3.82 \cdot 10^{4}$	$2.19 \cdot 10^{5}$
19	186561	2853	41.5	0.29	3	-	-	$5.98 \cdot 10^{4}$	$9.56 \cdot 10^{4}$
20	229140	3284	48.9	0.29	1	-	-	$9.83 \cdot 10^{4}$	$1.26 \cdot 10^{4}$

Table 3.2: Total and deterministic terms of the Hamiltonians, fitting parameters of the Trotter error ($\varepsilon_{\text{trott}} = C_{gs}\delta^a$), and the maximum circuit simulated, in terms of number of Pauli rotations.

Acknowledgments

The authors would like to thank Moritz Bensberg for his valuable assistance with the preparation of the Hamiltonian input data, and Anders Krogh for help with securing computational resources. Moreover, we are grateful to Olivier Byrde and the Euler team at ETH Zürich for their help in tailoring the accessibility of the Euler cluster to the needs of this work. We also gratefully acknowledge support of the Danish Centre for AI Innovation (DCAI) which hosts the Gefion supercomputer, where part of the computations were done in the frame of the pilot project "Large-scale distributed simulation of quantum algorithms for quantifying molecular recognition processes".

We acknowledge funding from the Research Project "Molecular Recognition from Quantum Computing". Work on "Molecular Recognition from Quantum Computing" was supported by Wellcome Leap as part of the Quantum for Bio (Q4Bio) Program. MC, JG, FW, MM also acknowledge financial support from the Novo Nordisk Foundation (Grant No. NNF20OC0059939 "Quantum for Life"). MST and GCS acknowledge support from the Novo Nordisk Foundation, (Grant No. NNF22SA0081175, NNF Quantum Computing Programme and Grant No. NNF20OC0060019, Solid-Q). MM and MC acknowledge support from the novoSTAR Programme by NovoNordisk A/S.

Author contributions

MM designed and wrote the simulation software. MC, JG, MM, FW conceptualized the research project. JG, MM, MST conducted the numerical experiment and analyzed data. ME prepared Hamiltonian data. MC, MR, GCS acquired computational resources. ME, JG, MM, FW wrote the first version of the manuscript. All authors reviewed the manuscript.

End of arxiv:2504:17881 — phase2: Full-State Vector Simulation of Quantum Time Evolution at Scale

Chapter 4

More quantum chemistry with fewer qubits

So far we have not questioned the quality of the electronic structure Hamiltonian that is passed on to the quantum computer to deal with. Recall that the Hamiltonian is the result of a real-space discretisation into a finite set of orbitals, and a subsequent truncation to an active-space Hamiltonian. Despite all the chemical insight and experience that goes into deciding on a good active space, there will always be some error due to the lack of interactions with core and virtual orbitals. In this section we investigate how parts of the missing ground state energy can be recovered, by proposing a second-order multireference perturbation theory quantum algorithm.

The key contributions are: (i) a general second-order perturbation theory quantum algorithm based on a Hamiltonian time evolution access model, with rigorous error bounds (ii) application of the algorithm to the active space Hamiltonian in quantum chemistry, resulting in a method that takes contributions of virtual orbitals into account without the need to represent them as qubits (iii) numerics on realistic chemistry systems.

Apart from section 4.8 this chapter is a reproduction of *Phys. Rev. Research* ***6****, 043021*. Additional improvements are discussed in section 4.8.

More quantum chemistry with fewer qubits

Abstract:
Quantum computation is one of the most promising new paradigms for the simulation of physical systems composed of electrons and atomic nuclei, with applications in chemistry, solid-state physics, materials science, or molecular biology. This requires a truncated representation of the electronic structure Hamiltonian using a finite number of orbitals. While it is, in principle, obvious how to improve on the representation by including more orbitals, this is usually unfeasible in practice (e.g., because of the limited number of qubits available on a quantum computer) and severely compromises the accuracy of the obtained results. Here, we propose a quantum algorithm that improves on the representation of the physical problem by virtue of second-order perturbation theory. In particular, our quantum algorithm evaluates the second-order energy correction through a series of time-evolution steps under the unperturbed Hamiltonian. An important application is to go beyond the active-space approximation, allowing to include corrections of virtual orbitals, which is known as multireference perturbation theory. Here, we exploit that the unperturbed Hamiltonian is diagonal for virtual orbitals and show that the number of qubits is independent of the number of virtual orbitals. This gives rise to more accurate energy estimates without increasing the number of qubits. Moreover, we demonstrate numerically for realistic chemical systems that the overall runtime of our method has highly favourable scaling in the number of virtual orbitals compared to previous work. Numerical calculations also confirm the necessity of the multireference perturbation theory energy corrections to reach accurate ground state energy estimates. Our perturbation theory quantum algorithm can also be applied to Symmetry-Adapted Perturbation Theory. As such, we reduce the quantum hardware requirements for quantum chemistry by leveraging perturbation theory.

Authors:
Jakob Günther, Alberto Baiardi, Markus Reiher and Matthias Christandl

Reference: Phys. Rev. Research **6**, 043021
Date: October 2024

4.1 Introduction

Among the proposed applications of quantum computing, the simulation of physical systems (such as molecules or materials) of increasing size that are unreachable by traditional approaches seems to be one of the most promising ones [199, 200, 201, 202, 203, 204]. In the context of quantum chemistry, a particularly important task is the delivery of highly accurate ground state energies, as, for example, accurate electronic energies are required to yield a reliable electronic contribution to the free energies of chemical transformations. On future fault-tolerant quantum computers, quantum phase estimation (QPE) is a rigorous approach to approximate the exact energy of a physical system within a specified error range, provided that the state preparation step can be accomplished with a sufficiently high overlap of the initial state with the target state. Although it is debatable whether good trial states can be found for all chemically relevant systems [100], recent complexity theoretic results suggest practical quantum advantage for the ground state energy estimation problem in the presence of a good trial state [**gharibianImprovedHardnessResults2022**].

However, any quantum computer will provide only a limited hardware framework (e.g., in terms of the number of qubits available for the representation of the quantum state) for the encoding of the discretized Hamiltonian that describes the physical system. Usually, the discretization is considered in occupation number vector space (rarely in real space [205]) and, hence, determined by the number of one-particle functions (e.g., orbitals of electronic structure or modals of vibrational structure) chosen for the construction of the many-particle basis states. The more orbitals are used to construct the second-quantized Hamiltonian, the more accurate this representation becomes and, hence, the more accurate the approximation of the energy of the physical system will be.

In view of the finite size of a quantum computer, the number of orbitals that they can represent is likely to be always too small, except for very small systems. This requires one to use a reduced-dimensional representation of the physical system that, in turn, then compromises accuracy. For instance, active-orbital-space methods in electronic structure theory such as complete active space self-consistent field (CASSCF) [206, 207, 208, 209], density matrix renormalization group (DMRG) [210, 211, 48, 212, 213, 214, 215, 216, 217, 218, 41], or full configuration interaction quantum Monte Carlo (FCIQMC) [219, 220] introduce a reduced space of orbitals in which the many-electron wave function is constructed, leaving aside the far larger number of remaining virtual orbitals (describing the so-called dynamic correlation). However, these virtual orbitals eventually account for a huge number of tiny energy contributions that, taken together, cannot be neglected in a calculation of the total energy. The standard traditional procedure

to assess this dynamic correlation energy is multi-reference perturbation theory (MRPT), usually taken to the second-order (MRPT2) for feasibility reasons [221, 222, 223].

It is natural to pursue a similar strategy on a quantum computer in order to deal with the fact that, though a number of K orbitals is required for the proper description of a physical system, only k orbitals with $k \ll K$ can be used on the machine (note that k spin orbitals will usually be mapped onto $\mathcal{O}(k)$ qubits). It has been argued in this context that the dynamic correlation energy may be added by classical computation of traditional approaches [40, 45]. However, instead of relying on classical approaches, we derive here a quantum algorithm that computes perturbative energy corrections.

In this work, we consider a perturbative treatment subsequent to an energy measurement directly on the quantum computer in a "diagonalize-then-perturb" fashion [40] to yield accurate electronic energies. Specifically, we introduce a quantum algorithm based on perturbation theory for estimating consecutive energy corrections to the ground state energy of the total Hamiltonian $H + V$ of a physical system, without solving the full, K-orbital quantum many-body problem on a quantum computer, but instead the model k-orbital Hamiltonian H. In regimes, where perturbation theory is expected to be a good approximation, this drastically reduces the quantum hardware requirements for quantum chemistry calculations.

We note that the algorithm we propose may also be considered a sanity check of the quantum model that is supposed to represent the physical model. That is, it may serve as a way to assess the discretization error.

The algorithm that we propose is general with respect to the specific perturbation theory application: It can be employed as an approach to parallel traditional MRPT2 calculations, but it also allows one to assess intermolecular interaction energies through symmetry-adapted perturbation theory (SAPT) [224].

In traditional computing, the unperturbed problem can be of mean-field (Hartree–Fock) type (the second-order energy correction is then an estimate for the correlation energy in Møller–Plesset PT2) or of complete-active-space type as in CASSCF (where the energy correction then approximates the lacking dynamic correlation energy). In any such case, the zeroth-order model in traditional computing is limited by known drawbacks (such as mean-field ansatz in Hartree–Fock and limited active space size of CASSCF). We emphasize that quantum computing will allow one to produce a more accurate zeroth-order model as it does not suffer from the curse of dimensionality of CASSCF.

The MRPT2 energy corrections computed in our approach are based on the Dyall-Hamiltonian as the unperturbed Hamiltonian, and the resulting energy corrections are equivalent to those of the totally uncontracted NEVPT2 method in the frozen core approximation. As such it is size-extensive and free of the

intruder-state problem [225]. Computing the totally uncontracted NEVPT2 energy is classically very expensive, therefore only approximations to it (partially- and strongly-contracted NEVPT2 energies) are computed in practice.

Previous works using quantum computing for perturbative methods in quantum chemistry include a variational quantum eigensolver (VQE) method for the first-order SAPT corrections as described by Malone et al. [226]. Their method was extended to second-order corrections by Loipersberger et al. [227], while Tammaro et al. [228] and Krompiec et al. [229] show how MRPT2 energies can be obtained with the help of a VQE subroutine. By constrast, we consider algorithms for fault-tolerant quantum computers.

The first steps towards first- and second-order perturbation theory for fault-tolerant quantum computers were taken by Mitarai et al. [21]. Key steps in their approach, such as reference state preparation and implementation of the reduced resolvent, is achieved with a quantum signal processing (QSP) approach. In comparison, we leverage an alternative implementation of the reduced resolvent and exploit certain features of the unperturbed Hamiltonian to reduce the complexity of the method. Specifically, for MRPT, we exploit the fact that H is diagonal for orbitals not included in the active space while, for SAPT, we use that H is the sum of two commuting Hamiltonians H_A and H_B. The authors of [21] give an extensive account of resources estimates for MRPT calculations on polyacenes. However, their partitioning of the full Hamiltonian into an unperturbed part and a perturbation is done in way that is not standard in quantum chemistry, and numerical evidence for the accuracy of this choice of partitioning is lacking. Furthermore, the runtime dependence on K, determined by the norm of perturbation coefficients, $\|v\|_{2/3}^2$, suffers from poor scaling, leading to highly impractical resource estimates in [21]. Details regarding this issue are discussed in Section 4.4.

In conclusion, although work on MRPT approaches for quantum computing has been done previously, they either rely on variational quantum eigensolver (VQE) subroutines that lack performance guarantees, or they do not take into account the specifics of the unperturbed MRPT Hamiltonian, thus exhibiting a poor scaling with K, both in the number of qubits and the overall runtime.

Lastly, we note that recently a fault-tolerant quantum algorithm for computing the first-order SAPT corrections was proposed [230]. This approach is based on a QSP-algorithm for expectation-value estimation, tailored to the specific form of the SAPT interaction operator. To obtain an at least qualitatively correct description of intermolecular interactions, it is necessary to include dispersion and induction terms, which appear only in the second-order SAPT correction. The second-order correction is computationally significantly more demanding as it involves the perturbation V twice, and additionally, the appearance of the reduced resolvent operator poses a non-trivial technical challenge. Our proposed quantum

algorithm applied to SAPT can overcome these challenges.

This work is organized as follows. In Section 4.2, we set the stage by reviewing the essential background for Rayleigh-Schrödinger perturbation theory and, in Section 4.3, we introduce our new quantum algorithm for calculating perturbative second-order energy corrections. In Section 4.4 we apply our algorithm to MRPT and illustrate the accuracy and scaling with realistic chemical systems. The application to SAPT is discussed in Section 4.5. Lastly, in Section 4.6, we finish with a conclusion and an outlook.

4.2 Perturbation theory

To introduce some basic notation, we briefly review standard (Rayleigh-Schrödinger) perturbation theory, whose energy corrections are of general applicability to different perturbation theory frameworks. The goal of perturbation theory is to estimate eigenstates and eigenenergies of a Hamiltonian $H + V$, using the eigenstates and eigenenergies of H as a starting point. The underlying Hilbert space is finite dimensional and has dimension D. We will only focus on the energies in our discussion.

The eigenenergies $E_j(\lambda)$ of the Hamiltonian $H + \lambda V$ are formally expanded in a Taylor series around $\lambda = 0$ [231, 232]

$$E_j(\lambda) = \sum_{k=0}^{\infty} \lambda^k E_j^{(k)} \,. \tag{4.1}$$

We will solely focus on the ground state ($j = 0$) of the total system $H + V$ and drop the index j from now on. The zeroth-order term $E^{(0)}$ is the energy of the unperturbed ground state $|\Phi_0\rangle$ of H, which we assume to be unique. Although the expansion is not guaranteed to converge for $\lambda = 1$ [233], low-order corrections, in particular $E^{(1)}$ and $E^{(2)}$, often yield valuable improvements to $E^{(0)}$. The k-th order correction, $E^{(k)}$, can be expressed as a sum of expectation values containing products of operators including V and the so-called reduced resolvent operator R_0, that is defined as

$$\begin{gathered} R_0 = \Pi_0 (E_0 I - H)^{-1} \Pi_0 = \sum_{i=1}^{D-1} \frac{|\Phi_i\rangle\langle\Phi_i|}{E_0 - E_i} \\ \text{with} \quad \Pi_0 = (I - |\Phi_0\rangle\langle\Phi_0|) \,. \end{gathered} \tag{4.2}$$

Here, E_i is the energy of the i-th excited state of H, $|\Phi_i\rangle$. The expressions for $E^{(k)}$ can symbolically be generated by the bracketing technique [234]. The three

lowest-order terms are:

$$\begin{aligned} E^{(1)} &= \langle V \rangle \equiv \langle \Phi_0 | V | \Phi_0 \rangle \\ E^{(2)} &= \langle V R_0 V \rangle \\ E^{(3)} &= \langle V R_0 V R_0 V \rangle - \langle V \rangle \left\langle V R_0^2 V \right\rangle \end{aligned} \tag{4.3}$$

where all expectation values are calculated with respect to the zeroth-order ground-state wave function $|\Phi_0\rangle$. The k-th order energy correction is a sum over expectation values in which V appears at most k times and R_0 at most $k-1$ times.

Lastly, we note that it is possible to calculate perturbative corrections also for approximate ground states of H. For an approximate ground state $|\widetilde{\Phi}_0\rangle$ with $\widetilde{E}_0 = \langle \widetilde{\Phi}_0 | H | \widetilde{\Phi}_0 \rangle$ we need the projectors

$$P = |\widetilde{\Phi}_0\rangle\langle\widetilde{\Phi}_0|\,, \quad Q = I - P$$

and partition the unperturbed Hamiltonian as

$$\begin{aligned} H =& H' + V' \qquad \text{with} \\ H' =& PHP + QHQ \qquad \text{and} \qquad V' = PHQ + QHP\,. \end{aligned}$$

Then $|\widetilde{\Phi}_0\rangle$ is an eigenstate of H' with eigenvalue $\widetilde{E}_0$ and the perturbation is taken to be $V' + V$.

4.3 Quantum algorithm for perturbation theory

In this section, we introduce our quantum algorithm for computing Rayleigh-Schrödinger perturbation theory energy corrections up to second order. However, we note that the techniques we will present can be generalized to higher order energy corrections. For simplicity of presentation we assume that $D = 2^d$, where d is the number of qubits. We consider a decomposition of H and V into a linear combination of Pauli strings $\sigma_i = \sigma_{i,1} \otimes \sigma_{i,2} \otimes \cdots \otimes \sigma_{i,d}$ acting on the qubits,

$$V = \sum_{i=1}^{L_V} v_i \sigma_i, \qquad H = \sum_{i=1}^{L_H} h_i \sigma_i\,, \qquad v_i, h_i \in \mathbb{R}\,. \tag{4.4}$$

We note that for many physical models mapped to such a qubit Hamiltonian, the number of terms L_V and L_H is significantly smaller than 4^d, the total number of Pauli strings on d qubits. For example, the states of a spin-1/2 lattice system with

d sites can be represented by d qubits, and a lattice Hamiltonian that includes only nearest-neighbour interaction has a number of terms that scales linearly in d. A relevant example for our work is a many-electron system expressed in d spin orbitals. It can be represented by d qubits based, for instance, on the Jordan-Wigner encoding, and the number of terms of the full electronic Hamiltonian with pairwise Coulomb-interaction scales as $\mathcal{O}(d^4)$.

Based on the decomposition of V into Pauli operators, $E^{(1)}$ can be easily calculated by summing expectation values of the σ_i,

$$E^{(1)} = \sum_{i=1}^{L_V} v_i \langle \sigma_i \rangle ,$$

which reduces to standard measurements of Pauli operators. The expression of the second-order energy correction $E^{(2)} = \langle V R_0 V \rangle$ implies that $E^{(2)}$ can be computed as the expectation value of $V R_0 V$ over $|\Phi_0\rangle$. The key challenge here is to find an efficient implementation of R_0. One could break R_0 into terms as done for V and H in Eq. (4.4), and then measure $\langle V R_0 V \rangle$ term by term. To find such a decomposition one could expand $(E_0 I - H)^{-1}$ as a power series in H. However, since the number of terms in H^k grows as L_H^k this approach suffers from a poor scaling. By regarding R_0 as a function of the spectrum of H, it is natural to implement R_0 through a block-encoding via quantum signal processing, which is the strategy Mitarai et al. [21] followed. While their technique could be employed for the applications we consider in Sections 4.4 and 4.5, it is in general not obvious how it can be used to exploit an underlying simplified structure that the unperturbed Hamiltonian might possess, e.g. if H is the sum of two commuting Hamiltonians. We therefore take a different path and express R_0 as a linear combination of time-evolution unitaries $U(t_n)$ under the unperturbed Hamiltonian for a series of time-steps t_n. The decomposition of R_0 into unitaries $U(t_n)$ is achieved by utilizing the Fourier series of R_0, with respect to the spectral variable E. $E^{(2)}$ can then be evaluated as a sum of expectation values $\langle \sigma_i U(t) \sigma_j \rangle$ at a series of time-steps t_n, where σ_i and σ_j are Pauli strings arising from V acting on the left and on the right of the resolvent, respectively. The quantities $\langle \sigma_i U(t) \sigma_j \rangle$ will be evaluated using a quantum computer. In Sections 4.4 and 4.5, we will show how the terms $\langle \sigma_i U(t) \sigma_j \rangle$ simplify for specific unperturbed electronic Hamiltonians.

We will now explain our approach by first discussing the Fourier-transformed resolvent, then applying it to the expression of the second-order energy correction, and afterwards presenting the quantum circuit for computing $\langle \sigma_i U(t) \sigma_j \rangle$. Key parameters in the discussion are the spectral range $R = E_{D-1} - E_0$ and the

spectral gap $\Delta = E_1 - E_0$ of H. First, we write the resolvent as:

$$R_0 = \sum_{i=1}^{D-1} |\Phi_i\rangle\langle\Phi_i| \left. \frac{1}{E_0 - E} \right|_{E=E_i} \tag{4.5}$$

i.e., as a function of the spectral variable E evaluated at the energies of the excited states E_i. The general idea is to define a function $g_{\Delta,R}(E)$ that evaluates to $-\frac{1}{E}$ in the interval $[\Delta, R]$ and that is zero at $E = 0$ such that

$$R_0 = \sum_{i=0}^{D-1} g_{\Delta,R}(E_i - E_0) \, |\Phi_i\rangle\langle\Phi_i| \, , \tag{4.6}$$

and then express $g_{\Delta,R}(E)$ in the time domain via a Fourier series. In Appendix 4.7.1 we design a function $g_{\Delta,R}(E)$ which itself and with all its derivatives vanishes at $E = 0$ and at $E = R + \Delta$. With these properties, $g_{\Delta,R}(E) : [0, R + \Delta] \to \mathbb{R}$ can be extended to a smooth periodic function with period $R + \Delta$. Because of smoothness and periodicity, the Fourier series of $g_{\Delta,R}(E)$ converges uniformly by the Dirichlet-Jordan test [235]. The Fourier series of $g_{\Delta,R}(E)$ is the $(R + \Delta)$-periodic function

$$\begin{aligned} &\sum_{n\in\mathbb{Z}} \beta_{\Delta,R,n} e^{-iEt_n} \quad \text{with} \quad t_n := \frac{2\pi n}{R+\Delta} \\ \text{and} \quad &\beta_{\Delta,R,n} := \frac{1}{R+\Delta} \int_0^{R+\Delta} g_{\Delta,R}(E) e^{iEt_n} dE \, , \end{aligned} \tag{4.7}$$

such that the resolvent can be expressed as

$$R_0 = \sum_{n\in\mathbb{Z}} \sum_{i=0}^{D-1} \beta_{\Delta,R,n} e^{-i(E_i - E_0)t_n} \, |\Phi_i\rangle\langle\Phi_i| \, . \tag{4.8}$$

We introduce the propagator under H, $U(t) = e^{-iHt}$, to rewrite (4.8) as

$$R_0 = \sum_{n\in\mathbb{Z}} \beta_{\Delta,R,n} e^{iE_0 t_n} U(t_n) \, , \tag{4.9}$$

which leads to the following expression for $E^{(2)}$,

$$\begin{aligned} E^{(2)} &= \langle V R_0 V \rangle \\ &= \sum_{n\in\mathbb{Z}} \beta_{\Delta,R,n} e^{-iE_0 t_n} \langle V U(t_n) V \rangle \, . \end{aligned} \tag{4.10}$$

In practice, we truncate the series to a finite number of terms N, and define

$$R_{0,N} := \sum_{|n|\leq N} \beta_{\Delta,R,n} e^{iE_0 t_n} U(t_n) \tag{4.11}$$

$$\begin{aligned} E_N^{(2)} &:= \langle V R_{0,N} V \rangle \\ &= \sum_{|n|\leq N} \beta_{\Delta,R,n} e^{iE_0 t_n} \langle V U(t_n) V \rangle \ . \end{aligned} \tag{4.12}$$

Hence, we decomposed an approximation of R_0 into a linear combination of N unitaries, and found a corresponding approximation $E_N^{(2)}$ to $E^{(2)}$. It is worth noting that

$$e^{iE_0 t} \langle V U(t) V \rangle = \left\langle e^{iHt} V e^{-iHt} V \right\rangle = \langle V(t) V \rangle$$

is simply the autocorrelation function of V at time t. We employ the decomposition of V into unitary operators in (4.4), so that $E_N^{(2)}$ is expressed as

$$E_N^{(2)} = \sum_{|n|\leq N} \beta_{\Delta,R,n} e^{iE_0 t_n} \sum_{i,j=1}^{L_V} v_i v_j \langle \sigma_i U(t_n) \sigma_j \rangle \tag{4.13}$$

Equation (4.13) is the essential expression for our method, and describes how an approximation to $E^{(2)}$ can be obtained by computing the overlaps $\langle \sigma_i U(t_n) \sigma_j \rangle$ on a quantum computer.

The overall accuracy ϵ of our algorithm is limited by the order of truncation, N, and the finite accuracy of the overlap estimation. The required truncation order for a desired accuracy ϵ depends on the convergence rate of the Fourier series of $g_{\Delta,R}$. In Appendix 4.7.1, Lemma 4.7.1, we show that the Fourier coefficients decay super-algebraically (i.e. faster than any polynomial) in n,

$$|\beta_{\Delta,R,n}| \leq \frac{61R}{\Delta^2} \exp\left(-\sqrt{\frac{|n|\Delta}{256R}} \right) .$$

This result is then used to prove that the truncation error is bounded by

$$\begin{aligned} |E^{(2)} - E_N^{(2)}| &\leq \frac{\epsilon}{2} \\ \text{if} \quad N &\geq \left\lceil \frac{2132R}{\Delta} \ln^2 \left(\frac{5.3 \cdot 10^4 R \|v\|_1^2}{\Delta^3 \epsilon} \right) \right\rceil , \end{aligned} \tag{4.14}$$

where $\|v\|_1 = \sum_{i=1}^{L_V} |v_i|$. The proof is provided in Appendix 4.7.2.

Equation Eq. (4.14) shows that, on top of the global accuracy ϵ, knowledge of the parameters Δ, R and $\|v\|_1$ is required in order to get the truncation order

N. Note that $R \leq 2\|h\|_1$, as shown in Eq. (4.62) in Appendix 4.7.3. With the definition of H and V (see Eq. (4.4)) $\|h\|_1$ and $\|v\|_1$ can readily be computed. For the gap Δ, a realistic lower bound needs to be supplied. Once the truncation order N is known, the target quantity to compute is given by Eq. (4.13).

The key step of our method is the estimation of the overlaps using a quantum computer. To perform this estimation efficiently, we build on quantum amplitude estimation. Quantum overlap estimation is closely related to the quantum amplitude estimation problem, and in Appendix 4.7.2, we describe how a recent version of the quantum amplitude estimation algorithm, the Iterative Quantum Amplitude Estimation algorithm (IQAE) [236, 237] can be employed for quantum overlap estimation. The results in this part of the Appendix build up to Lemma 4.7.7:

Lemma 4.5. *Given a confidence level $1-\alpha \in (0,1)$, a target accuracy $\epsilon > 0$, a linear combination of d-qubit unitaries $V_U = \sum_{i=1}^{L} u_i U_i$ and a d-qubit state-preparation unitary U_0, it is possible to obtain an estimate X to the overlap $\langle\Phi_0|V_U|\Phi_0\rangle$ such that*

$$\mathbb{P}[|X - \langle\Phi_0|V_U|\Phi_0\rangle| < \epsilon] \geq 1-\alpha$$

with 1 ancillary qubit, a total $\mathcal{O}\left(\epsilon^{-1}\|u\|_{2/3} \ln\frac{L}{\alpha}\sqrt{\ln\frac{1}{\alpha}}\right)$ applications of U_0 and $\mathcal{O}\left(\epsilon^{-1}|u_i|^{2/3}\|u\|_{2/3}^{1/2} \ln\frac{L}{\alpha}\sqrt{\ln\frac{1}{\alpha}}\right)$ applications of controlled-U_i for each i=1,…,L, where $\|u\|_{2/3} = \left(\sum_{i=1}^{L}|u_i|^{2/3}\right)^{3/2}$.

The most expensive subroutines of our quantum algorithm are the time evolution $U(t)$ and the state-preparation U_0. Apart from QPE, various algorithms have been proposed to prepare the ground state of a Hamiltonian [21, 238, 239, 240] , e.g. by eigenvalue filtering. Furthermore, as argued at the end of Section 4.2, perturbative corrections can be calculated also for approximate eigenstates, opening up the possibilities to different types of state preparation subroutines U_0. Therefore, in order to keep the algorithm general, we treat U_0 as a black-box, and report the total number of calls to U_0 required to compute an estimate to $E^{(2)}$ with ϵ precision. Moreover, we will state the total time for which the evolution under H needs to implemented in order to compute the estimate to $E^{(2)}$.

The overall cost of our method is shown with Lemma 4.7.8, with the result that for H and V as in Eq. (4.4) computing an estimate to $E^{(2)}$ with precision ϵ requires implementing the Hamiltonian evolution under H for a total time upper bounded by

$$\widetilde{\mathcal{O}}\left(\frac{\|h\|_1^{3/2}\|v\|_{2/3}^2}{\Delta^{7/2}\epsilon}\right), \tag{4.15}$$

and a total number of calls to U_0 upper bounded by

$$\widetilde{\mathcal{O}}\left(\frac{\|h\|_1^{3/2}\|v\|_{2/3}^2}{\Delta^{5/2}\epsilon}\right). \tag{4.16}$$

Here, the notation $\widetilde{\mathcal{O}}$ denotes the suppression of logarithmic factors, and $\|v\|_{2/3} = (\sum_{i=1}^{L_V} |v_i|^{2/3})^{3/2}$. Furthermore, Lemma 4.7.8 implies that the quantum algorithm requires $d+1$ qubits.

We note that our approach may be optimized in terms of Δ and $\|h\|_1$. For example, our implementation of $g(E)$ as a trigonometric polynomial is not unique and could be optimized in order to reduce the dependence on Δ and R. Furthermore, for Hamiltonians H considered in practice, the upper bound $R \leq 2\|h\|_1$ might turn out to be very loose.

4.4 Application to multireference perturbation theory

So far, we kept our derivation entirely generic, so that it can be applied to any method based on second-order perturbation theory. In this section we shift the focus to the problem of finding the electronic ground state energy of a molecular system. Multi-electron states are in practice constructed from a discrete and finite set of single-electron states, *i.e.* a set of so-called molecular spin orbitals, and via fermion-to-qubit mappings of molecular spin orbitals to qubits the electronic structure problem can be tackled using a quantum computer.

If the number of qubits available is much smaller than the number of molecular spin orbitals one desires to include in the description of a chemical system, a subset of so-called 'active orbitals' must be chosen which fit the number of qubits, yielding a reduced-dimensional problem. This reduction in problem size necessarily incurs some errors, however, as we will explain in this section, it is possible to use our quantum algorithm for second-order perturbation theory to reduce this error. In this regard, our approach parallels classical multi-reference perturbation theory methods for adding dynamical correlation effects on top of a complete active space solution.

Classical active space methods include complete active space configuration interaction (CASCI), CASSCF and DMRG, and well-established methods for including dynamical correlation effects based on second-order perturbation theory

exist, such as complete active space and N-electron valence perturbation theory of second order, CASPT2 and NEVPT2, respectively. These perturbation theory methods differ in the choice of the zeroth-order Hamiltonian.

The starting point is an orthonormal single-electron basis of dimension K, $\mathcal{I} = \{\phi_1, \ldots, \phi_K\}$, spanning the single-electron Hilbert space $\mathcal{H}_1(\mathcal{I})$. We consider a system of N_{el} electrons, so that the full Hilbert space is given by $\Lambda^{N_{el}}(\mathcal{H}_1(\mathcal{I}))$. The quantities of interest are the eigenenergies of the full electronic Hamiltonian

$$H_{el} = \sum_{p,q \in \mathcal{I}} h_{pq} a_p^\dagger a_q + \frac{1}{2} \sum_{p,q,r,s \in \mathcal{I}} g_{pqrs} a_p^\dagger a_q^\dagger a_r a_s \tag{4.17}$$

for a fixed number of N_{el} electrons, and where the fermionic creation and annihilation operators corresponding to ϕ_i are denoted by $a_i^\dagger$ and a_i. The coefficients h_{pq} and g_{pqrs} are the standard one- and two-electron integrals in Hartree atomic units[7]

$$\begin{aligned} h_{pq} &= \int \overline{\phi_p}(\mathbf{x}) \left(-\frac{\nabla^2}{2} - \sum_{I=1}^{N_{nuc}} \frac{Z_I}{|\mathbf{R}_I - \mathbf{x}|} \right) \phi_q(\mathbf{x}) d\mathbf{x}, \\ g_{pqrs} &= \int \int \frac{\overline{\phi_p}(\mathbf{x}_1)\overline{\phi_q}(\mathbf{x}_2)\phi_r(\mathbf{x}_2)\phi_s(\mathbf{x}_1)}{|\mathbf{x}_1 - \mathbf{x}_2|} d\mathbf{x}_1 d\mathbf{x}_2 \,, \end{aligned} \tag{4.18}$$

where $\mathbf{x}$ denotes spatial and spin variables, N_{nuc} is the number of nuclei, and Z_I and $\mathbf{R}_I$ are their charge numbers and positions, respectively. We assume that the one-electron basis $\mathcal{I}$ is constructed from molecular spin orbitals that diagonalize the Fock operator F, i.e.

$$F = \sum_{p \in \mathcal{I}} \epsilon_p a_p^\dagger a_p \,,$$

where ϵ_p are the orbital energies. The Fock operator results from a mean-field approximation of the electronic Hamiltonian of (4.17). Although Hartree-Fock does not yield quantitatively accurate predictions, it is considered a good starting point for more accurate methods [7].

Methods aiming at improving upon Hartree-Fock results face the problem that the full Hilbert space has dimension $\binom{K}{N_{el}}$. When employing standard atom-centered basis sets, the size of the single-particle basis, K, is roughly proportional to, and at least an order of magnitude larger than, the number of electrons. Therefore, the dimension of the Hilbert space scales exponentially in the number of electrons, which is the reason why representing general electronic wave functions on a classical computer becomes unfeasible.

Many methods in quantum chemistry deal with this issue by considering only a small, manageable subset of the full, many-electron Hilbert space. In the class of active-space methods this smaller subset is defined on the level of the single-electron basis. Active-space methods partition the single-particle basis into three

parts, the inactive (or core) space $\mathcal{I}_{core}$, the space of active orbitals $\mathcal{I}_{act}$ and the virtual space $\mathcal{I}_{virt}$, such that $\mathcal{I} = \mathcal{I}_{core} \cup \mathcal{I}_{act} \cup \mathcal{I}_{virt}$ (see Figure 4.1). We will use indices a, b, c, d, e, f for active orbitals, v, w, x, y for virtual orbitals and p, q, r, s for arbitrary orbitals. The number of active orbitals will be denoted by k. The active space is defined as the span of N_{el}-electron Slater determinants where all virtual orbitals are unoccupied and all core orbitals are occupied. This leads to a reduced problem size of $N_{el}^{act} = N_{el} - |\mathcal{I}_{core}|$ electrons in k molecular spin orbitals. On this subspace, the Hamiltonian H_{el} (see (4.17)) can be associated with the CAS Hamiltonian H_{CAS}, which acts on the Hilbert space $\Lambda^{N_{el}^{act}}(\mathcal{H}_1(\mathcal{I}_{act}))$ and reads

$$H_{CAS} = \sum_{a,b \in \mathcal{I}_{act}} h'_{ab} a_a^\dagger a_b + \frac{1}{2} \sum_{a,b,c,d \in \mathcal{I}_{act}} g_{abcd} a_a^\dagger a_b^\dagger a_c a_d \,.$$

Here, h'_{ab} includes the mean-field potential of the core electrons, which the active electrons are subjected to in the active space approximation. Note that the energy of the core electrons themselves is not part of H_{CAS}. The ground state wave function and energy of H_{CAS} are denoted by $|\Phi_0^{CAS}\rangle$ and E_0^{CAS}, respectively. For increasing k and N_{el}^{act}, classical implementations of active space methods face the same dimensionality issue as before, because the dimension of $\Lambda^{N_{el}^{act}}(\mathcal{H}_1(\mathcal{I}_{act}))$ is $\binom{N_{el}^{act}}{k}$. The promise of fault-tolerant quantum computers for chemistry is to target larger active spaces with the help of quantum algorithms like QPE. Nonetheless, every active space computation, whether classical or quantum, is necessarily an approximation, and it is desirable to improve on that.

Multireference perturbation theory (MRPT) methods are employed to go beyond the active space approximation, accounting for Slater determinants not included in the active space. The second-order MRPT energy correction will contain contributions from Slater determinants associated with variable occupation numbers of core, active and virtual orbitals. In fact, it will contain contributions associated with so-called single and double excitations from the core and active spaces to the active and virtual spaces (by single and double excitations we refer here to a Slater determinant that differs from the reference one by the occupation of one or two spin orbitals, respectively). For the sake of clarity, we introduce an approximation and keep the core orbitals occupied. We refer to this approximation as "frozen-core approximation". In this setting the Hilbert space for the perturbed and unperturbed Hamiltonians is $\Lambda^{N_{el}^{act}}(\mathcal{H}_1(\mathcal{I}_{act} \cup \mathcal{I}_{virt}))$. We note that in many situations this approximation is justified and used in practice. Extending our methods to include also electron excitations from the core orbitals is conceptually trivial. The full Hamiltonian in the frozen-core approximation is

$$H'_{el} = \sum_{p,q \in \mathcal{I}_{act} \cup \mathcal{I}_{virt}} h'_{pq} a_p^\dagger a_q + \frac{1}{2} \sum_{p,q,r,s \in \mathcal{I}_{act} \cup \mathcal{I}_{virt}} g_{pqrs} a_p^\dagger a_q^\dagger a_r a_s + E_{core} \,. \tag{4.19}$$

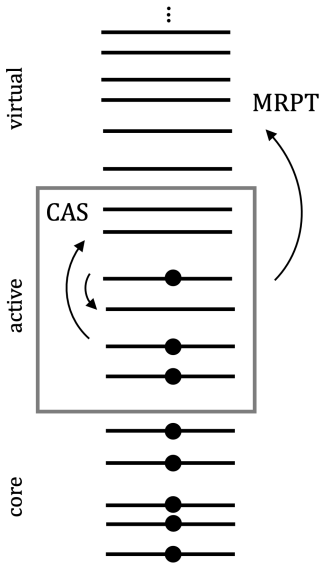


Figure 4.1: Illustration of occupations of core, active and virtual orbitals. Lines and circles symbolize orbitals and electrons, respectively.

Compared to the full Hamiltonian Eq. (4.17), the core electrons enter Eq. (4.19) only through the single electron coefficients h'_{pq} and the energy-shift E_{core}, the energy of the core electrons. We define the unperturbed Hamiltonian H as the sum of H_{CAS}, the Fock operator on the virtual space, and E_{core} (note that this choice yields the Dyall Hamiltonian in the frozen-core approximation [241]) as

$$\begin{aligned} H &= H_{CAS} + F_{virt} + E_{core} \\ F_{virt} &= \sum_{v \in \mathcal{I}_{virt}} \epsilon_v a_v^\dagger a_v \,. \end{aligned} \tag{4.20}$$

The unperturbed reference function can easily be constructed from the ground-state wave function of H_{CAS}, $\left|\Phi_0^{CAS}\right\rangle$, as $|\Phi_0\rangle = \left|\Phi_0^{CAS}\right\rangle \wedge |0 \dots 0\rangle$, which is an eigenfunction of H with energy $E_0^{CAS} + E_{core}$. In practice, the active space is chosen such that the virtual orbitals all have sufficiently high orbital energies to ensure that $|\Phi_0\rangle$ is also the ground-state wave function of H. The expression for V follows from $V = H'_{el} - H$,

$$V = \sum_{\substack{p,q \in \mathcal{I}_{act} \cup \mathcal{I}_{virt} \\ \{p,q\} \not\subset \mathcal{I}_{act}}} h'_{pq} a_p^\dagger a_q + \frac{1}{2} \sum_{\substack{p,q,r,s \in \mathcal{I}_{virt} \cup \mathcal{I}_{act} \\ \{p,q,r,s\} \not\subset \mathcal{I}_{act}}} g_{pqrs} a_p^\dagger a_q^\dagger a_r a_s - F_{virt} \,.$$

One could directly apply the general perturbation theory algorithm to H and V and obtain the asymptotic cost from equations (4.15) and (4.16). We briefly

discuss the scaling of this approach with respect to the large number of molecular spin orbitals K, and we assume that $K \gg k$. It is evident that the number of terms in V scales as $\mathcal{O}(K^4)$. Since V appears once in the first-order correction $E^{\mathrm{MRPT(1)}}$ and twice in the second-order correction $E^{\mathrm{MRPT(2)}}$, the calculations of $E^{\mathrm{MRPT(1)}}$ and $E^{\mathrm{MRPT(2)}}$ involve $\mathcal{O}(K^4)$ and $\mathcal{O}(K^8)$ expectation values, respectively. More specifically, the dependence on K in the cost of the computation of $E^{\mathrm{MRPT(2)}}$ in equations (4.15) and (4.16) is determined by $\|v\|_{2/3}^2$, apart from logarithmic factors. For the approach of using the full difference $V = H'_{el} - H$ as the perturbation we will denote this quantity by $\|v_{\mathrm{diff}}\|_{2/3}^2$. As described later in this section, for a realistic chemical system we numerically observe that $\|v_{\mathrm{diff}}\|_{2/3}^2$ scales as $\mathcal{O}(K^{6.2})$, which would incur considerable cost for the method.

We note that directly using the full difference $V = H'_{el} - H$ as the perturbation is very similar to the approach by Mitarai et al. [21] when they applied their QSP perturbation theory algorithm to multireference perturbation theory. Their unperturbed, active space Hamiltonian is defined after the fermion-to-qubit mapping and is therefore different compared to the CAS- or Dyall-Hamiltonian. Nonetheless, all terms of the full Hamiltonian H'_{el} are taken into account, either in the unperturbed Hamiltonian or in the perturbation. The perturbation norm $\|v\|_{2/3}^2$, as defined for their partitioning into H and V, obeys the same scaling in K as $\|v_{\mathrm{diff}}\|_{2/3}^2$. Highly impractical numbers for $\|v\|_{2/3}^2$ lead [21] to conclude that their perturbation theory method does not reduce the computational cost of solving chemical systems. Noteworthy, their method is not accompanied by a reduction in the number of qubits.

In this light we set the following two criteria: First, computing $E^{\mathrm{MRPT(2)}}$ should not require more qubits than is necessary to represent the active space orbitals plus a constant number of ancillary qubits. Second, we want to overcome the issue of impractically large values of $\|v_{\mathrm{diff}}\|_{2/3}^2$. In the following we explain how to accomplish these goals.

A first simplification of V is possible by noting that $a_v |\Phi_0\rangle = 0$ if a_v corresponds to a virtual orbital. Consequently, the expectation values $\langle a_p^\dagger a_q \rangle$ and $\langle a_p^\dagger a_q^\dagger a_r a_s \rangle$ calculated over the reference wave function will be non-zero only if $p, q, r, s \in \mathcal{I}_{act}$. However, none of the terms in V act only on the active orbitals, therefore the first-order energy correction is (within the frozen-core approximation) zero, i.e.,

$$E^{\mathrm{MRPT(1)}} = \langle V \rangle = 0. \tag{4.21}$$

Extending this reasoning to the (truncated) second-order energy correction,

$$E_N^{\mathrm{MRPT(2)}} = \sum_{|n| \leq N} \beta_{\Delta,R,n} e^{iE_0 t} \langle V U(t_n) V \rangle \ ,$$

by taking into account only the terms of V that are non-vanishing when acting on $|\Phi_0\rangle$, we can write

$$\begin{aligned}\langle VU(t_n)V\rangle &= \sum_{v,a,w,b} h'_{va}h'_{wb}\,\langle a_b^\dagger a_w U(t_n) a_v^\dagger a_a\rangle \\ &+ \frac{1}{2}\sum_{v,a,p,q,b,c} h'_{va} g_{pqbc}\,\langle a_c^\dagger a_b^\dagger a_q a_p U(t_n) a_v^\dagger a_a\rangle \\ &+ \frac{1}{2}\sum_{p,q,a,b,v,c} g_{pqab} h'_{vc}\,\langle a_c^\dagger a_v U(t_n) a_p^\dagger a_q^\dagger a_a a_b\rangle \\ &+ \frac{1}{4}\sum_{p,q,a,b,r,s,c,d} g_{pqab} g_{rscd}\,\langle a_d^\dagger a_c^\dagger a_s a_r U(t_n) a_p^\dagger a_q^\dagger a_a a_b\rangle \\ &=: T_1(t_n) + T_2(t_n) + T_3(t_n) + T_4(t_n)\end{aligned} \tag{4.22}$$

with

$$\begin{aligned} v, w &\in \mathcal{I}_{virt}\,, \quad a, b, c, d \in \mathcal{I}_{act} \\ p, q, r, s &\in \mathcal{I}_{act} \cup \mathcal{I}_{virt}\,, \quad \{p, q\}, \{r, s\} \not\subset \mathcal{I}_{act}\,. \end{aligned}$$

To keep the discussion of further reductions of these terms simple, here in the main text we will just consider the first of the four sums, $T_1(t_n)$. The full derivation for all four sums is analogous and is given in Appendix 4.7.4.

H commutes with the number operators $n_v = a_v^\dagger a_v$ of the virtual orbitals, $[H, n_v] = [\epsilon_v n_v, n_v] = 0$, therefore it conserves the number of electrons in each virtual orbital. The same holds for the propagator $U(t_n)$. Consequently, $\langle a_b^\dagger a_w U(t_n) a_v^\dagger a_a\rangle$ is non-vanishing only if creation and annihilation operators appear in pairs,

$$T_1(t_n) = \sum_{\substack{v\in\mathcal{I}_{virt}\\ a,b\in\mathcal{I}_{act}}} h'_{va}h'_{vb}\,\langle a_b^\dagger a_v U(t_n) a_v^\dagger a_a\rangle\ .$$

Next, we employ the time-dependent creation and annihilation operators in the Heisenberg picture, $a_p^\dagger(t) \equiv e^{iHt} a_p^\dagger e^{-iHt}$ and $a_p(t) \equiv e^{iHt} a_p e^{-iHt}$. Since H is diagonal for the virtual orbitals, the time dependence of their creation and annihilation operators is particularly simple, $a_v(t) = a_v e^{-i\epsilon_v t}$ and $a_v^\dagger(t) = a_v^\dagger e^{i\epsilon_v t}$.

Using this insight, we find

$$\begin{aligned} T_1(t_n) &= \sum_{\substack{v\in\mathcal{I}_{virt}\\ a,b\in\mathcal{I}_{act}}} h'_{va}h'_{vb}\,\langle a_b^\dagger e^{-iHt_n}e^{iHt_n}a_v e^{-iHt_n}a_v^\dagger a_a\rangle \\ &= \sum_{\substack{v\in\mathcal{I}_{virt}\\ a,b\in\mathcal{I}_{act}}} h'_{va}h'_{vb}e^{-i\epsilon_v t_n}\,\langle a_b^\dagger e^{-iHt_n}a_v a_v^\dagger a_a\rangle \\ &= \sum_{\substack{v\in\mathcal{I}_{virt}\\ a,b\in\mathcal{I}_{act}}} h'_{va}h'_{vb}e^{-i(\epsilon_v+E_{core})t_n}\,\langle a_b^\dagger e^{-iH_{CAS}t_n}a_a\rangle\ , \end{aligned}$$

where, in the last line, we exploited the anticommutation relations between fermionic second-quantization operators, as well as $a_v\,|\Phi_0\rangle = 0$. Furthermore, since the last expectation value only contained active orbitals, we rewrote e^{-iHt_n} as $e^{-i(H_{CAS}+E_{core})t_n}$.

Following these steps for all terms (see Appendix 4.7.4), we get

$$\begin{aligned} T_1(t_n) &= \sum_{a,b\in\mathcal{I}_{act}} v_{a,b}(t_n)\,\langle a_b^\dagger e^{-iH_{CAS}t_n}a_a\rangle \\ T_2(t_n) &= \sum_{a,b,c,d\in\mathcal{I}_{act}} v_{abc,d}(t_n)\,\langle a_d^\dagger e^{-iH_{CAS}t_n}a_a^\dagger a_b a_c\rangle \\ T_3(t_n) &= \sum_{a,b,c,d\in\mathcal{I}_{act}} v_{bcd,a}(t_n)\,\langle a_d^\dagger a_c^\dagger a_b e^{-iH_{CAS}t_n}a_a\rangle \\ T_4(t_n) &= \sum_{a,b,c,d\in\mathcal{I}_{act}} v_{ab,cd}(t_n)\,\langle a_d^\dagger a_c^\dagger e^{-iH_{CAS}t_n}a_a a_b\rangle \\ &\quad + \sum_{a,b,c,d,e,f\in\mathcal{I}_{act}} v_{abc,def}(t_n)\,\langle a_f^\dagger a_e^\dagger a_d e^{-iH_{CAS}t_n}a_a^\dagger a_b a_c\rangle\ , \end{aligned} \tag{4.23}$$

where the coefficients are defined by

$$\begin{aligned} v_{a,b}(t) &= \sum_{v\in\mathcal{I}_{virt}} h'_{va}h'_{vb}e^{-i(\epsilon_v+E_{core})t} \\ v_{abc,d}(t) &= \sum_{v\in\mathcal{I}_{virt}} h'_{vd}g_{vabc}e^{-i(\epsilon_v+E_{core})t} \\ v_{ab,cd}(t) &= \tfrac{1}{2}\sum_{v,w\in\mathcal{I}_{virt}} g_{vwab}g_{vwcd}e^{-i(\epsilon_v+\epsilon_w+E_{core})t} \\ v_{abc,def}(t) &= \sum_{v\in\mathcal{I}_{virt}} g_{vabc}g_{vdef}e^{-i(\epsilon_v+E_{core})t}\ . \end{aligned} \tag{4.24}$$

This reduction shares similarities with classical perturbation theory methods based on the Laplace-transform of the resolvent [242, 243]. In particular, the

method of [244] also achieves a reduction in the number of terms to be computed for MRPT2 energies by requiring only 3-RDM like terms instead of 4-RDM.

Finally, the creation and annihilation operators in Eq. (4.23) need to be mapped to Pauli strings acting on qubits resulting in an expression of the form

$$E_N^{\mathrm{MRPT(2)}} = \sum_{|n|\leq N} \beta_{\Delta,R,n} \sum_{i=1}^{L^{\mathrm{MRPT(2)}}} v_{ni}^{\mathrm{MRPT(2)}} \times \left\langle \sigma_{i,1} e^{-iH_{CAS}t_n} \sigma_{i,2} \right\rangle . \tag{4.25}$$

Equation (4.25) is not of the form of (4.13), since the perturbation coefficients $v_{ni}^{\mathrm{MRPT(2)}}$ depend on n, too. Following the complexity analysis of the general algorithm (Appendix 4.7.3), the coefficients enter the cost as

$$\|v^{\mathrm{MRPT(2)}}\beta\|_{2/3} = \left(\sum_{|n|\leq N} \sum_{i=1}^{L^{\mathrm{MRPT(2)}}} \left| \beta_{\Delta,R,n} v_{ni}^{\mathrm{MRPT(2)}} \right|^{2/3} \right)^{3/2} .$$

The perturbation coefficients $v_{ni}^{\mathrm{MRPT(2)}}$ derive directly from the coefficients in equation (4.24), and by moving the absolute values into the sums over the virtual orbitals we obtain an upper on the perturbation coefficients that is independent of n:

$$\begin{aligned}
&\left(\sum_i |v_{ni}^{\mathrm{MRPT(2)}}|^{2/3} \right)^{3/2} \\
&\leq \Bigg(2^{2/3} \sum_{a,b\in\mathcal{I}_{act}} \left(\sum_{v\in\mathcal{I}_{virt}} |h'_{va}h'_{vb}| \right)^{2/3} \\
&+ 2^{7/3} \sum_{a,b,c,d\in\mathcal{I}_{act}} \left(\sum_{v\in\mathcal{I}_{virt}} |h'_{va}g_{vbcd}| \right)^{2/3} \\
&+ 2^{2/3} \sum_{a,b,c,d\in\mathcal{I}_{act}} \left(\sum_{vw\in\mathcal{I}_{virt}} |g_{vwab}g_{vwcd}| \right)^{2/3} \\
&+ 4 \sum_{a,b,c,d,e,f\in\mathcal{I}_{act}} \left(\sum_{v\in\mathcal{I}_{virt}} |g_{vabc}g_{vdef}| \right)^{2/3} \Bigg)^{3/2} \\
&\equiv \|v^{\mathrm{MRPT(2)}}\|_{2/3} .
\end{aligned}$$

See Appendix 4.7.4 for details. As a consequence we can decouple $\|v^{\mathrm{MRPT(2)}}\beta\|_{2/3}$ and use the upper bound

$$\|v^{\mathrm{MRPT(2)}}\beta\|_{2/3} \leq \|v^{\mathrm{MRPT(2)}}\|_{2/3}\|\beta_{\Delta,R}\|_{2/3}$$

so that in the bounds of the general perturbation theory algorithm in Eqs (4.15) and (4.16) $\left\|v^{\mathrm{MRPT(2)}}\right\|_{2/3}$ plays the role of $\|v\|_{2/3}^2$ and $L^{\mathrm{MRPT(2)}}$ that of L_V^2.

Another parameter that entered the complexity analysis of the algorithm is R, the spectral range of H. In Eq. (4.15) and Eq. (4.16) this parameter has already been replaced by the upper bound $2\|h\|_1$. The Dyall Hamiltonian (see Eq. (4.20)), which we use as the unperturbed Hamiltonian, is acting on the virtual orbitals with a Fock operator, so its spectral range will depend on K. Therefore, if we were to simply use the spectral range of the Dyall Hamiltonian, this would incur additional K-dependence in the complexity of the method. However, we argue that in this specific application, R can be replaced by $R_{CAS} + \mathrm{const}$, where R_{CAS} is the spectral range of H_{CAS}. As a consequence, we will get rid of the K-dependence due to R. To see that this replacement is justified, consider the exponents in Eq. (4.23), which encode the energy via the Fourier transform, see Eq. (4.7) - Eq. (4.10). The operator $e^{-iH_{CAS}t}$ is acting in the expectation values of Eq. (4.23), which, together with the prefactor $e^{i(E_0-E_{core})t} = e^{iE_0^{CAS}t}$, results in an energy in the exponent of at most R_{CAS}. Moreover, the energies of at most two virtual orbitals contribute via the coefficients, so the largest energy that occurs is upper bounded by $R_{CAS} + 2\max_{v\in\mathcal{I}_{virt}} \epsilon_v$. Thus, since $R_{CAS} \leq 2\|h^{CAS}\|_1$, the 1-norm $\|h\|_1$ in Eq. (4.15) and Eq. (4.16) can be replaced by $\|h^{CAS}\|_1$, and we arrive at our main result:

Main Result: Computing $E^{\mathrm{MRPT(2)}}$ with precision ϵ requires $k+1$ qubits and implementing the evolution under H_{CAS} for a total time upper bounded by

$$\widetilde{\mathcal{O}}\left(\frac{\left\|h^{CAS}\right\|_1^{3/2}\left\|v^{\mathrm{MRPT(2)}}\right\|_{2/3}}{\Delta^{7/2}\epsilon}\right),$$

and a total number of calls to U_0 upper bounded by

$$\widetilde{\mathcal{O}}\left(\frac{\left\|h^{CAS}\right\|_1^{3/2}\left\|v^{\mathrm{MRPT(2)}}\right\|_{2/3}}{\Delta^{5/2}\epsilon}\right).$$

Explicit expressions for upper bounds on $\left\|h^{CAS}\right\|_1$ and on $\left\|v^{\mathrm{MRPT(2)}}\right\|_{2/3}$ are provided in appendix 4.7.4.

Numerical results: In the computation of all results the quantum chemistry software PySCF [245] was employed.

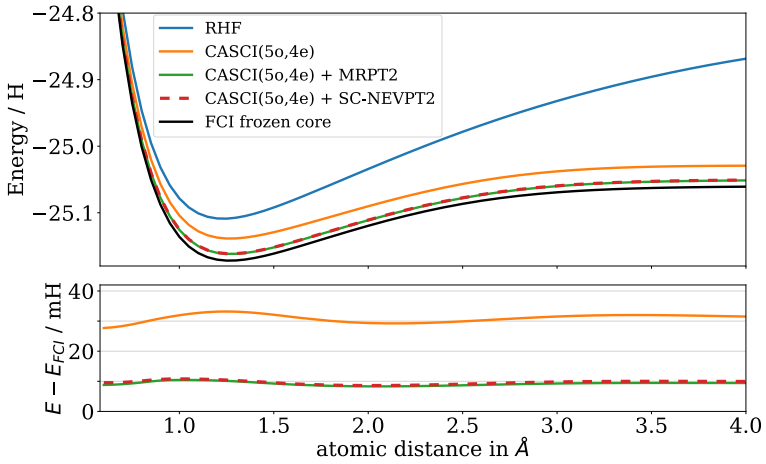


Figure 4.2: Potential energy surface scan of the B-H molecule with different methods: Restricted Hartree-Fock (blue), CASCI with 4 electrons in 5 spatial orbitals with and without the MRPT2 correction (orange and green, respectively) and Full-CI with frozen core electrons. Upper: absolute energies, lower: energies differences with respect to the Full-CI frozen core energies.

First, the impact of the MRPT2 corrections is demonstrated numerically with the dissociation curve of the singlet ground state of the B-H molecule in the 6-31G basis set, see Figure 4.2. Based on Hartree-Fock orbitals, a full-valence space active space with 5 spatial orbitals and 4 electrons was chosen, yielding the CASCI(5o,4e) curve. This corresponds to $k = 10$ active spin-orbitals and with the frozen-core approximation (the 1s-orbital of B is frozen) the full number of spin-orbitals is $K = 20$. Considering this relatively small system allows us to do a Full-CI calculation as an exact reference (red curve). The green curve shows how the MRPT2 correction improves the active space calculation by including contributions from the virtual orbitals. In fact it recovers about two thirds of the missing correlation energy of CASCI(5o,4e) compared to the frozen core Full-CI result, and as seen in Table 4.1, the error in the dissociation energy D_e is reduced from 3.1 mH to 0.8 mH. Despite the relatively small size of the basis set, this result highlights an important fact: To achieve 'chemical accuracy' (1.6 mH) it is necessary to go beyond the active space approximation and take into account dynamical correlation effects.

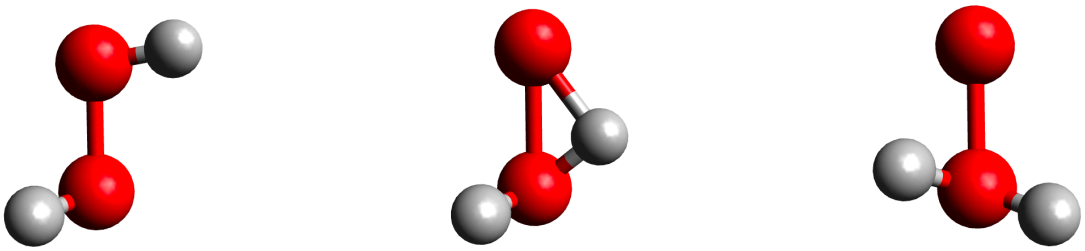
Furthermore, to investigate the effect of the MRPT2 correction on a larger chemical system with a barrier, we considered the interconversion reaction from oxywater (H_2OO) to hydrogen peroxide (HOOH) via a transition state (TS). This system has sparked interest among theoretical chemists in the 1990s, because it was debated whether oxywater was a stable minimum on the potential energy surface or not [246, 247]. Starting from a restricted Hartree-Fock (RHF)

calculation in the def2-TZVP basis set, we optimized the geometries of H_2OO, HOOH and the transition state at the CASCI(10o, 14e) level. This active space includes all valence orbitals and valence electrons, translating to the number of active and total spin-orbitals $k = 20$ and $K = 144$, respectively. At the resulting geometries, two different second-order perturbative energy corrections were calculated, the MRPT2-correction described in this work, denoted MRPT2, and a traditional multireference perturbation correction, (strongly contracted)-NEVPT2. All results are reported in Table 4.2. To put the change of barrier height of the transition state into perspective, the reaction rate k_{rate} for the reaction $H_2OO \to HOOH$ predicted by transition state theory is dominated by the scaling $k_{rate} \propto \exp\{-\Delta G_{TS}/(k_B T)\}$, where ΔG_{TS} is the difference in Gibbs free energy between the transition state and oxywater and k_B is the Boltzmann-constant. The change in Gibbs free energy is dominated by the change in electronic energy $\Delta E_{TS} = E_{TS} - E_{H_2OO}$, therefore we approximate $k_{rate} \propto \exp\{-\Delta E_{TS}/(k_B T)\}$. According to this scaling, the reduction in barrier height from 44.89 mHartree to 16.11 mHartree due to the MRPT2 correction increases k_{rate} by a factor of 10^5 at $T = 300$K. Despite the approximations involved in our argument, it is evident that an exact solution in the active-space approximation is not sufficient to obtain quantitative results. Although this is not a new insight in the field of traditional quantum chemistry, this fact is often disregarded in the literature on quantum computing for chemistry, where the active-space approximation has become a standard. The difference between MRPT2 correction and the SC-NEVPT2 energy correction is non-negligible for this bigger system. As mentioned before, the MRPT2 correction computed with our method is equivalent to the totally-uncontracted NEVPT2 correction. Therefore, the observed difference between MRPT2 and SC-NEVPT2 can be attributed to the approximations made in the definition of the SC-NEVPT2 correction.

Furthermore, for the H_2O_2 geometry we have investigated the scaling of $\|v^{\mathrm{MRPT2}}\|_{2/3}$ and $\|v_{\mathrm{diff}}\|_{2/3}^2$ with increasing number of virtual orbitals, the results are shown in Figure 4.4. The data points we compute are upper bounds to the relevant coefficient norms, but the bounds are tight up to a constant factor. Details of this technical aspect are provided in Appendix 4.7.4. The difference in order of magnitude between the two quantities is tremendous, for the highest

method	D_e in mH	E_{min} in H
CASCI(5o,4e)	108.7	-25.1388
CASCI(5o,4e) + SC-NEVPT2	111.1	-25.1616
CASCI(5o,4e) + MRPT2	111.0	-25.1619
Full CI frozen core	111.8	-25.1720

Table 4.1: Dissociation and minimum energies of BH.



(a) HOOH **(b)** TS **(c)** H_2OO

Figure 4.3: Geometries of (a) Hydrogen Peroxide (b) the transition state and (c) Oxywater.

method	HOOH	TS	H_2OO	ΔE_{TS}
RHF	-396015.58	-395792.72	-395869.61	76.89
CASCI(10o,14e)	-396117.08	-395891.00	-395935.89	44.89
CASCI(10o,14e) + SC-NEVPT2	-397181.21	-396951.40	-396971.30	19.89
CASCI(10o,14e) + MRPT2	-397201.63	-396975.35	-396991.46	16.11

Table 4.2: Energies in kJ/mol of Hydrogen Peroxide, the Transition State and Oxywater in the basis set def2-TZVP. All geometries are optimized at the CASCI(10o,14e) level, and for all methods the frozen-core approximation is used.

number of virtual orbitals considered, $K - k = 122$, $\|v^{\mathrm{MRPT2}}\|_{2/3}$ and $\|v_{\mathrm{diff}}\|_{2/3}^2$ differ by about 10 orders of magnitude. Fitting a power law to the data points resulted in exponents 0.8 and 6.2 for $\|v^{\mathrm{MRPT2}}\|_{2/3}$ and $\|v_{\mathrm{diff}}\|_{2/3}^2$, respectively, underlining the difference in scaling. In this regard, we conclude that our approach to compute dynamical correlation effects on a quantum computer constitutes a significant improvement over the previous algorithm of Mitarai et al [21]. Additionally, the scaling of $\|v_{\mathrm{diff}}\|_{2/3}^2$ and $\|v^{\mathrm{MRPT2}}\|_{2/3}$ with increasing number of active orbitals was examined by studying hydrogen chains in the 6-31G basis set. The ratio of active to virtual orbitals remains constant as the number of hydrogen atoms (and therefore of active orbitals) is increased. Data points and fits are shown in Figure 4.5, and the fitted power-law coefficients are given in Table 4.3. We observe that $\|v^{\mathrm{MRPT2}}\|_{2/3}$ outperforms $\|v_{\mathrm{diff}}\|_{2/3}^2$ by scaling lower by a quadratic factor in k. However, $\|v^{\mathrm{MRPT2}}\|_{2/3}$ still behaves as $k^{6.5}$, leaving open room for further improvements in the scaling of the method with k.

The true value of quantum computing in chemistry is for routine applications of multiconfigurational electronic structures, often found in 3d transition metal compounds. As an extreme case for such applications, we consider the FeMo-cofactor, the active site of the metalloenzyme nitrogenase [40, 46]. To describe the important static electron correlation in FeMoco, an active space of on the order of $k \sim 100$ active spin orbitals will be needed. However, about $K \sim 4000$ spin orbitals would be required in a triple-zeta basis set to describe the entire electronic structure (static and dynamic correlation) of FeMoco, where

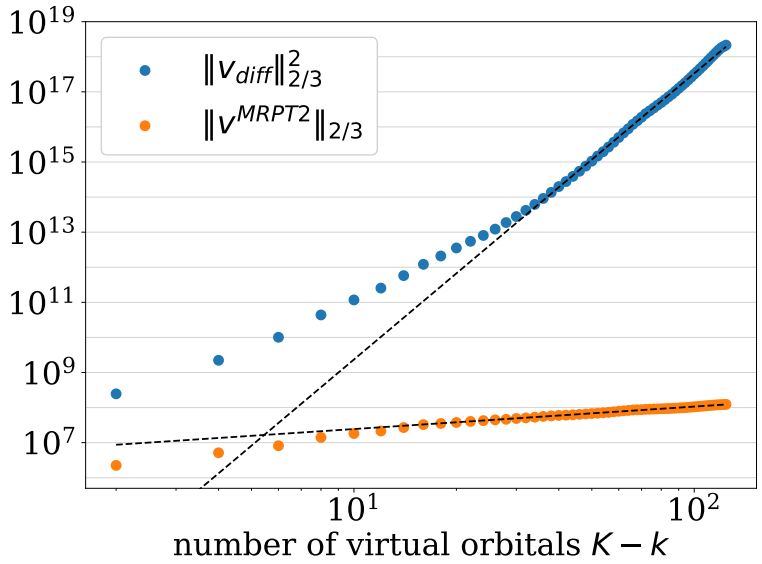


Figure 4.4: Perturbation coefficient norms $\|v_{\text{diff}}\|^2_{2/3}$ and $\left\|v^{\text{MRPT2}}\right\|_{2/3}$ at the H_2O_2 geometry with increasing number of virtual orbitals. The dashed lines are power-law fits $\alpha(K-k)^\beta$ to the data points, the obtained values α and β are given in Table 4.3.

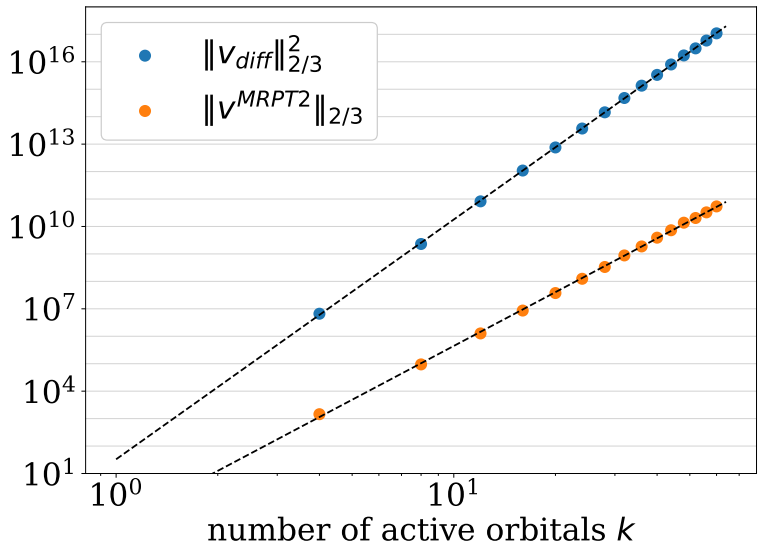


Figure 4.5: Perturbation coefficient norms $\|v_{\text{diff}}\|^2_{2/3}$ and $\left\|v^{\text{MRPT2}}\right\|_{2/3}$ for the hydrogen chain with increasing number of active orbitals while $k/K = const$. The dashed lines are power-law fits αk^β to the data points, the obtained values α and β are given in Table 4.3.

	scaling with $(K-k)$		scaling with k	
	α	β	α	β
$\|v_{\text{diff}}\|_{2/3}^2$	16	8.2	33	8.7
$\left\|v^{\text{MRPT2}}\right\|_{2/3}$	$5.6 \cdot 10^6$	0.64	0.13	6.5

Table 4.3: Fitted power-law coefficients for $\|v_{\text{diff}}\|_{2/3}^2$ and $\left\|v^{\text{MRPT2}}\right\|_{2/3}$ as a function of k and $K-k$ as presented in Figures 4.4 and 4.5.

the $K-k$ inactive orbitals could be taken into account via perturbation theory. In this regard, the reduction in the number of qubits from a priori $\mathcal{O}(K)$ to $\mathcal{O}(k)$ qubits is significant and implies that even more accurate quantum chemical results could be obtained with a limited number of qubits. While FeMoco is just one prominent example, the role of dynamical correlation is, without a doubt, a general problem in computational chemistry for active-space based approaches [248, 249, 250]. Including dynamical correlation effects on top of active-space ground state energies obtained on a quantum computer is necessary in order for quantum computing to be of value for routine applications in quantum chemistry.

Finally, as we assumed $K \gg k$, we have focused on K in our analysis, and our result has not been optimized with respect to the number of active orbitals, k. An improvement in the scaling of $\|v^{\text{MRPT(2)}}\|_1$, both in K and in k, might be achieved by working out factorizations of the coefficient-tensors in equation (4.24), similar to those presented in references [45] and [46]. To account for the overall scaling of our method with k it would not only be necessary to consider $\|\beta_{\Delta,R}\|_1$, but also it would be relevant to inspect the implementation of the ground state preparation unitary U_0 and the time-evolution $U(t)$ under the unperturbed Hamiltonian, which we kept as black-boxes here. Including the explicit implementation of these subroutines and working out additional improvements for the scaling in k is an important step to make multireference perturbation theory on a quantum computer even more practical, however, it is beyond the scope of this work and will be subject of a future study.

4.5 Application to symmetry-adapted perturbation theory

Symmetry-Adapted Perturbation Theory (SAPT) is a method for calculating the interaction energy between two molecules A and B perturbatively [224]. The molecular complex formed by A and B together is called "dimer", while A and B are referred to, individually, as "monomers". The interaction energy E_{int} is defined as the difference between the energy of the dimer, E_{AB}, and the sum of

energies of the monomers, $E_A + E_B$:

$$E_{int} = E_{AB} - E_A - E_B \,. \tag{4.26}$$

This difference in energy is computed in the Born-Oppenheimer approximation, so each of the three energies corresponds to an electronic energy calculated with clamped nuclei. Furthermore, the monomers are treated as rigid, *i.e.* to obtain the dimer configuration, the monomer configurations of the nuclei (used to compute E_A and E_B) are kept frozen and are simply placed in proximity to each other. This way, for given monomer configurations of the nuclei, the interaction energy E_{int} is a function only of the distance between the monomers and a set of angles describing their mutual orientation. Additional energy contributions due to relaxation of the nuclei in the dimer configuration are typically small [251].

When applying SAPT, particular care must be paid to the antisymmetry properties of the electronic wave function. In the standard description of many-electron wave-functions, a state is an element of the antisymmetrized Hilbert space $\Lambda^N(\mathcal{H}_1)$, which is spanned by all N-electron Slater determinants constructed from molecular spin orbitals in the single-particle space $\mathcal{H}_1$. By contrast, SAPT treats the N_A electrons of A and N_B electrons of B as two distinguishable sets of particles, described by two different single-particle Hilbert spaces $\mathcal{H}_{1,A}$ and $\mathcal{H}_{1,B}$. The full many-electron Hilbert space is then constructed as $\mathcal{H}_{AB} = \mathcal{H}_A \otimes \mathcal{H}_B = \Lambda^{N_A}(\mathcal{H}_{1,A}) \otimes \Lambda^{N_B}(\mathcal{H}_{1,B})$ [252]. The corresponding two sets of annihilation and creation operators, $a_\mu, a_\mu^\dagger$ for $\mathcal{H}_A$ and $b_\nu, b_\nu^\dagger$ for $\mathcal{H}_B$, fulfill the usual fermionic commutation relations within each set. However, operators belonging to different sets commute. The two molecular spin orbital basis sets are denoted by $\mathcal{I}_A = \{\phi_1^A, \ldots, \phi_{K_A}^A\}$ and $\mathcal{I}_B = \{\phi_1^B, \ldots, \phi_{K_B}^B\}$. They are usually chosen to be "dimer-centered bases" i.e., both bases describe a set of molecular spin orbitals spanning the whole dimer configuration and typically expressed in terms of the same atom-centered Gaussians [253]. Other basis set choices are viable, for example monomer-centered basis sets with additional basis functions spanning the interaction region [254].

The unperturbed Hamiltonian H is defined as the sum of Hamiltonians H_A and H_B, they describe the monomers individually on some level of electronic structure theory, and their ground state energies are E_A and E_B. Consequently, the ground state $|\Phi_0\rangle$ of H is the product of the monomer ground states, $|\Phi_0^A\rangle$ and $|\Phi_0^B\rangle$, with energy given by

$$\begin{aligned} H\,|\Phi_0\rangle &= (H_A \otimes I + I \otimes H_B)\,|\Phi_0^A\rangle \otimes |\Phi_0^B\rangle \\ &= (E_A + E_B)\,|\Phi_0\rangle \,. \end{aligned} \tag{4.27}$$

The perturbation operator V contains the interaction between the monomers, i.e. the electron-electron, electron-nuclei and nuclei-nuclei interactions between

particles associated with A and B. With this choice of H and the resulting perturbative expansion of E_{AB}, the energies of the monomers cancel in the expression of the interaction energy in (4.26), which is then given by the perturbative energy corrections only. However, the Rayleigh-Schrödinger energy-corrections (see Section 4.2) are modified in order to mitigate the errors caused by the unphysical partitioning into A and B electrons. This is achieved by applying a-posteriori an antisymmetrization operator $\mathcal{A}$ onto the wave function, yielding the so-called Symmetrized Rayleigh-Schrödinger (SRS) corrections $E^{(k)}_{SRS}$, which are summed to yield the interaction energy.

In practice the antisymmetrizer $\mathcal{A}$ is approximated by a one-electron exchange operator [255, 253]

$$\begin{aligned} \mathcal{P}_1 &= -\sum_{ijkl} S_{il} S_{kj} a_i^\dagger a_j b_k^\dagger b_l \\ \text{where} \qquad S_{il} &= \int \overline{\phi_i^A(\mathbf{x})} \phi_l^B(\mathbf{x}) d\mathbf{x} \end{aligned}$$

is the overlap between molecular orbitals $\phi_i^A(\mathbf{x})$ and $\phi_l^B(\mathbf{x})$. This approximation, usually referred to as the S^2-approximation [224], yields the following expression for the first- and second-order corrections:

$$\begin{aligned} E^{(1)}_{SRS}(S^2) &= \langle V \rangle + \langle (\langle V \rangle - V) \mathcal{P}_1 \rangle \\ &=: E^{(1)}_{pol} + E^{(1)}_{exch} \\ E^{(2)}_{SRS}(S^2) &= \langle V R_0 V \rangle + \langle (\langle V \rangle - V)(\langle \mathcal{P}_1 \rangle - \mathcal{P}_1) R_0 V \rangle \\ &=: E^{(2)}_{pol} + E^{(2)}_{exch}\,, \end{aligned} \tag{4.28}$$

where we have used the usual partitioning into the so-called polarization and exchange energies. The inter-monomer interaction is given by [256]

$$\begin{aligned} V &= V_{elA-elB} + V_{elA-nucB} + V_{nucA-elB} - V_0 \\ &= \sum_{pqrs} g^{AB}_{pqrs} a_p^\dagger a_s b_q^\dagger b_r + \sum_{jk} h^B_{jk} a_k^\dagger a_j \\ &\quad + \sum_{lm} h^A_{lm} b_m^\dagger b_l - V_0, \end{aligned} \tag{4.29}$$

where

$$
\begin{aligned}
g^{AB}_{pqrs} &= \int\int \frac{\overline{\phi^A_p}(\mathbf{x}_1)\overline{\phi^B_q}(\mathbf{x}_2)\phi^B_r(\mathbf{x}_2)\phi^A_s(\mathbf{x}_1)}{|\mathbf{x}_1-\mathbf{x}_2|} d\mathbf{x}_1 d\mathbf{x}_2 \\
h^B_{jk} &= -\sum_{i=1}^{N^B_{nuc}} \langle\phi_j| \frac{Z_i}{|\mathbf{r}-\mathbf{R}_{B_i}|} |\phi_k\rangle \\
h^A_{lm} &= -\sum_{i=1}^{N^A_{nuc}} \langle\phi_l| \frac{Z_i}{|\mathbf{r}-\mathbf{R}_{A_i}|} |\phi_m\rangle \\
V_0 &= \sum_{i,j=1}^{N^A_{nuc},N^B_{nuc}} \frac{Z_{A_i} Z_{B_j}}{|\mathbf{R}_{A_i}-\mathbf{R}_{B_j}|} \,.
\end{aligned}
$$

We will now illustrate how our method enables us to compute contributions to first- and second-order corrections while only having access to one of the states. As an example, we consider the contribution $\langle V_{elA-nucB}\mathcal{P}_1 R_0 V_{elA-elB}\rangle$ to the second-order correction:

$$
\begin{aligned}
&\langle V_{elA-nucB}\mathcal{P}_1 R_0 V_{elA-elB}\rangle \\
&= \sum_{\substack{ijkl\\mn\\pqrs}} S_{il}S_{kj}h^B_{mn}g^{AB}_{pqrs} \left\langle a^\dagger_n a_m a^\dagger_i a_j b^\dagger_k b_l R_0 a^\dagger_p a_s b^\dagger_q b_r \right\rangle \\
&= \sum_{n\in\mathbb{Z}} \sum_{\substack{ijkl\\mn\\pqrs}} S_{il}S_{kj}h^B_{mn}g^{AB}_{pqrs}\beta_{\Delta,R,n} e^{iE_0 t_n} \\
&\quad \times \left\langle a^\dagger_n a_m a^\dagger_i a_j b^\dagger_k b_l e^{-iHt_n} a^\dagger_p a_s b^\dagger_q b_r \right\rangle \\
&= \sum_{n\in\mathbb{Z}} \sum_{\substack{ijkl\\mn\\pqrs}} S_{il}S_{kj}h^B_{mn}g^{AB}_{pqrs}\beta_{\Delta,R,n} e^{iE_0 t_n} \\
&\quad \times \left\langle a^\dagger_n a_m a^\dagger_i a_j e^{-iH_A t_n} a^\dagger_p a_s \right\rangle_A \left\langle b^\dagger_k b_l e^{-iH_B t_n} b^\dagger_q b_r \right\rangle_B \,,
\end{aligned}
$$

where $\langle\ldots\rangle_X = \langle\Phi^X_0|\ldots|\Phi^X_0\rangle$ involves only one of the monomer. Consequently, $\langle V_{elA-nucB}\mathcal{P}_1 R_0 V_{elA-elB}\rangle$ can be computed from quantities that are obtained from the monomers' states $|\Phi^A_0\rangle, |\Phi^B_0\rangle$ individually. The procedure is analogous for the other contributions to the energy corrections.

4.6 Conclusions and outlook

In this work, we presented a general quantum algorithm for the calculation of second-order perturbative energy corrections, and specified it for molecular electronic energies.

For the application to multi-reference perturbation theory, we found that no qubits are required to represent the virtual orbitals, and in general the quantum algorithm requires only a single ancilla qubit. We have analyzed our quantum algorithm in terms of calls to U_0 and total Hamiltonian evolution time under H with a focus on minimzing the K dependence since it is the one deemed the most impactful, in contrast to the number of active space orbitals k.

In terms of K, we observed numerically for a chemically realistic system that with our approach for a fixed active space (*i.e.* fixed k), we achieve sublinear scaling in the number of virtual orbitals by taking the structure of the unperturbed Hamiltonian into account explicitly. By contrast, we also showed that naively computing the second-order correction results in a runtime that scales to the eighth power in the number of virtual orbitals, which lead the previous paper on fault-tolerant MRPT to the conclusion that their method was impractical [21]. Consequently, our method opens up the way to treat a significant number of virtual orbitals at low cost, which is a key improvement for multi-reference perturbation theory. Furthermore, we demonstrated that the energy correction computed with our method significantly improves on the result of the active space calculation by including dynamical correlation effects of virtual orbitals.

With this, we have laid the groundwork for multi-reference perturbation theory on a quantum computer. While the dependence on k is polynomial, we have not optimized it. This aspect of the quantum algorithm could be investigated in future work. Furthermore, for the application to symmetry-adapted perturbation theory, we found that it is possible to compute the second-order intermolecular interaction energy while only having access to one of the monomers' states at a time.

Our algorithm can be considered a method to be implemented on a quantum computer to refine the energy (e.g., evaluated by quantum phase estimation) of a discretized representation of a physical model. The discretization may be achieved by activating only a limited number of one-particle functions (i.e. orbitals), the so-called active space. In turn, the reduction in the number of orbitals reduces the accuracy of the energy obtained (as is also the case for state-of-the-art traditional approaches such as CASSCF, DMRG, and FCIQMC) and requires a procedure that allows one to assess the importance of the many neglected orbitals for a quantitatively correct total energy of the physical system. Our approach delivers perturbative energy corrections for this assessment that can be viewed as

an approximation of the error introduced by switching to a reduced-dimensional discretization (the active space) from a more fine-grained discretization with far more orbitals.

In terms of applications, the algorithm can be used to improve on QPE results and hence provides a quantum MRPT version of traditional approaches such as CASPT2 and NEVPT2. As such, it can be a key component of endeavors to elucidate reaction mechanisms as proposed for enzymatic and catalytic reactions in Refs. [40, 45] that would otherwise require switching to a classical computer for this task.

However, our algorithm is also applicable in the field of intermolecular interactions, where the SAPT approach is known to deliver accurate interaction energies for systems that cannot be treated as a whole, but only in terms of its parts (for instance, in host-guest chemistry). As such, our algorithm can become an integral part of drug-discovery and drug-design attempts, where accurate intermolecular interaction energies are key to the free energy of a docking process. Moreover, we note that our setting and algorithm might be of value also to other fields, not only those that rely on electronic or vibrational structure theory applied in chemistry, solid state physics, materials science, and biochemistry.

We have focused our presentation here to second-order perturbation theory as this a relevant and widely used method in practical applications. Still, higher-order perturbative corrections can, in a straightforward fashion, be considered within our framework.

Acknowledgements

We acknowledge financial support from the Novo Nordisk Foundation (Grant No. NNF20OC0059939 'Quantum for Life'). MC and JG also acknowledge support from the European Research Council (ERC Grant Agreement No. 81876) and VILLUM FONDEN via the QMATH Centre of Excellence (Grant No.10059).

4.7 Appendices

4.7.1 Definition of $g_{\Delta,R}(E)$

In this section we provide the definition of the function $g_{\Delta,R}(E)$, which is used to implement the resolvent R_0 in Section 4.3. We start from

$$g'_{\Delta,R}(E) = -\frac{I_{[\frac{\Delta}{4},R+\Delta\frac{3}{4}]}}{E}\,,$$

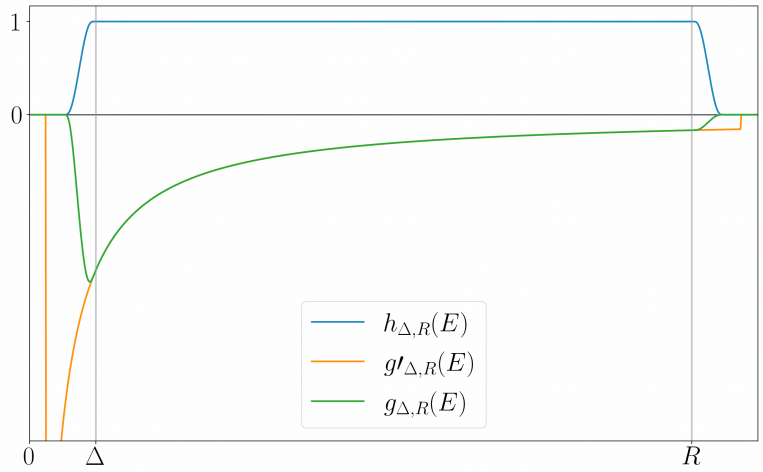


Figure 4.6: Illustration of the functions used in the construction of $g_{\Delta,R}(E)$, the function playing the role of the resolvent.

where $I_{[a,b]}$ is the indicator function of the interval $[a,b] \subset \mathbb{R}$. The function $g_{\Delta,R}(E)$ can be smoothened by multiplying it with another indicator function ($h_{\Delta,R}$ below) that smoothly decays at its edges and whose support is inside the support of $g'_{\Delta,R}$. To construct $h_{\Delta,R}$, we employ a standard bump function

$$\eta(E) = \begin{cases} c_\eta e^{-(1-E)^{-1}} e^{-(1+E)^{-1}}, & |E| < 1 \\ 0, \quad \text{else} \end{cases}$$

where c_η is a normalizing constant such that

$$\int_{-1}^{1} \eta(E) dE = 1\,. \tag{4.30}$$

The function $\eta(E)$ is smooth but not analytic, and the derivatives $\frac{d^n}{dE^n}\eta$ at $E = 1$ and $E = -1$ are zero for all n [257]. We define the narrowed mollifier

$$\eta_\Delta(E) = \frac{4}{\Delta}\eta\Big(\frac{4E}{\Delta}\Big)\,,$$

whose support is $[-\frac{\Delta}{4}, \frac{\Delta}{4}]$ and which is still normalized. The function

$$h_{\Delta,R}(E) = (I_{[\frac{3}{4}\Delta, R+\frac{\Delta}{4}]} * \eta_\Delta)(E) \tag{4.31}$$

evaluates to 1 on the interval $[\Delta, R]$ and goes smoothly to 0 at $E = \frac{\Delta}{2}$ and $E = R + \frac{\Delta}{2}$, see also Figure 4.6.

Consequently, we can define the function $g_{\Delta,R}(E)$ with the desired properties by

$$g_{\Delta,R}(E) = (g'_{\Delta,R} \cdot h_{\Delta,R})(E)\,. \tag{4.32}$$

Bounds on Fourier Coefficients of $g_{\Delta,R}(E)$

The Fourier coefficients of $g_{\Delta,R}(E)$ are defined as

$$\begin{aligned}\beta_{\Delta,R,n} =& \frac{1}{R+\Delta}\int_0^{R+\Delta} g_{\Delta,R}(E)e^{-iEt_n}dE\,,\\ t_n =& \frac{2\pi n}{R+\Delta}\,, \quad n\in\mathbb{Z}\,,\end{aligned}$$

and in this section we prove an upper bound on the Fourier coefficients $\beta_{\Delta,R,n}$ in terms of n, Δ and R. We use a standard approach and relate the decay properties of the Fourier coefficients to the smoothness of $g_{\Delta,R}(E)$. In particular, we prove the following lemma

Lemma 4.7.1. *The Fourier coefficients $\beta_{\Delta,R,n}$ of the function $g_{\Delta,R}(E)$ on the interval $[0, R+\Delta]$ satisfy*

$$|\beta_{\Delta,R,n}| \leq \frac{61R}{\Delta^2}2^{-(\frac{|n|\Delta}{256R})^{1/2}}$$

for all $|n| \geq 0$ and $R \geq \Delta > 0$.

Proof: The Fourier coefficients relate to the Fourier transform $\widehat{g_{\Delta,R}}(t)$ of $g_{\Delta,R}(E)$ by

$$\begin{aligned}\beta_{\Delta,R,n} =& \frac{1}{R+\Delta}\widehat{g_{\Delta,R}}(t_n)\\ \text{where}\quad \widehat{g_{\Delta,R}}(t) =& \int_0^{R+\Delta} g_{\Delta,R}(E)e^{-iEt}dE\,.\end{aligned}$$

Since the Fourier transform of the ℓth derivative (which we denote by D^ℓ) is given by [258]

$$\widehat{D^\ell g_{\Delta,R}}(t) = (-it)^\ell \widehat{g_{\Delta,R}}(t)$$

we can take absolute values to get

$$|\beta_{\Delta,R,n}| = \frac{1}{R+\Delta}|\widehat{g_{\Delta,R}}(t_n)| = \frac{1}{R+\Delta}\frac{|\widehat{D^\ell g_{\Delta,R}}(t_n)|}{|t_n|^\ell}\,.$$

By construction, $g_{\Delta,R}(E)$ itself and with all its derivatives vanishes outside the interval $[\frac{\Delta}{2}, R+\frac{\Delta}{2}]$. Therefore we can employ the bound

$$|\widehat{D^\ell g_{\Delta,R}}(t)| \leq \int_{\frac{\Delta}{2}}^{R+\frac{\Delta}{2}} \left|D^\ell g_{\Delta,R}(E)\right|dE = R\left\|D^\ell g_{\Delta,R}\right\|_\infty\,,$$

which results in

$$|\beta_{\Delta,R,n}| \leq \frac{R}{R+\Delta}\frac{\left\|D^\ell g_{\Delta,R}\right\|_\infty}{|t_n|^\ell} \leq \frac{\left\|D^\ell g_{\Delta,R}\right\|_\infty}{|t_n|^\ell}\,. \tag{4.33}$$

Next, by means of the Leibniz product rule the numerator of inequality Eq. (4.33) is upper bounded by

$$\begin{aligned}\left\|D^\ell g_{\Delta,R}\right\|_\infty &= \sup_{E\in[\frac{\Delta}{2},R+\frac{\Delta}{2}]}\left|D^\ell(g'_{\Delta,R}\cdot h_{\Delta,R})(E)\right| \qquad (4.34)\\ &= \sup_{E\in[\frac{\Delta}{2},R+\frac{\Delta}{2}]}\left|\sum_{\ell'=0}^{\ell}\binom{\ell}{\ell'}\left(D^{\ell-\ell'}g'_{\Delta,R}(E)\right)\left(D^{\ell'}h_{\Delta,R}(E)\right)\right|\\ &\leq \sup_{E\in[\frac{\Delta}{2},R+\frac{\Delta}{2}]}\sum_{\ell'=0}^{\ell}\binom{\ell}{\ell'}|D^{\ell-\ell'}g'_{\Delta,R}(E)|\,|D^{\ell'}h_{\Delta,R}(E)|\,.\end{aligned}$$

The derivatives of $h_{\Delta,R}(E)$ can be bounded by

$$\begin{aligned}|D^{\ell'}h_{\Delta,R}(E)| &= |D^{\ell'}(I_{[\frac{3}{4}\Delta,R+\frac{\Delta}{4}]} * \eta_\Delta)(E)|\\ &= |(I_{[\frac{3}{4}\Delta,R+\frac{\Delta}{4}]} * D^{\ell'}\eta_\Delta)(E)|\\ &= |\int_{\frac{3}{4}\Delta}^{R+\frac{\Delta}{4}} D^{\ell'}\eta_\Delta(E-E')dE'|\\ &\leq R\|D^{\ell'}\eta_\Delta\|_\infty \qquad (4.35)\end{aligned}$$

Lemma 2 in the work of A. Israel [259] provides a bound for the derivatives of the bump function:

$$\|D^{\ell'}\eta\|_\infty \leq c_\eta(16)^{\ell'}\ell'^{2\ell'}\,,$$

where c_η is the normalization constant of $\eta(E)$, see Eq. (4.30). Since the derivatives of $\eta(E)$ and $\eta_\Delta(E)$ are related by $D^{\ell'}\eta_\Delta(E) = \left(\frac{4}{\Delta}\right)^{\ell'+1}D^{\ell'}\eta(\frac{4E}{\Delta})$, we find, via Eq. (4.35), the bound

$$|D^{\ell'}h_{\Delta,R}(E)| \leq \frac{4c_\eta R}{\Delta}\left(\frac{64}{\Delta}\right)^{\ell'}\ell'^{2\ell'}\,. \tag{4.36}$$

On the interval $[\frac{\Delta}{2}, R+\frac{\Delta}{2}]$ the function $g'_{\Delta,R}(E)$ evaluates to $-\frac{1}{E}$, so that we can bound

$$\begin{aligned}&\sup_{E\in[\frac{\Delta}{2},R+\frac{\Delta}{2}]}|D^{\ell-\ell'}g'_{\Delta,R}(E)|\\ &= \sup_{E\in[\frac{\Delta}{2},R+\frac{\Delta}{2}]}|D^{\ell-\ell'}\frac{1}{E}| \qquad (4.37)\\ &= (\ell-\ell')!\left(\frac{2}{\Delta}\right)^{\ell-\ell'+1}\,.\end{aligned}$$

Inserting the bounds Eq. (4.36) and Eq. (4.37) into Eq. (4.34) gives

$$\begin{aligned}
&\left\|D^\ell g_{\Delta,R}\right\|_\infty \\
&\leq \frac{4c_\eta R}{\Delta}\sum_{\ell'=0}^{\ell}\binom{\ell}{\ell'}(\ell-\ell')!\Big(\frac{2}{\Delta}\Big)^{\ell-\ell'+1}\big(\frac{64}{\Delta}\big)^{\ell'}\ell'^{2\ell'} \\
&= \frac{8c_\eta R 2^\ell \ell!}{\Delta^{\ell+2}}\sum_{\ell'=0}^{\ell}(32)^{\ell'}\frac{\ell'^{2\ell'}}{\ell'!} \\
&\leq \frac{8c_\eta R 2^\ell}{\Delta^{\ell+2}}(\ell+1)(32\ell^2)^\ell \\
&\leq \frac{8c_\eta R}{\Delta^2}\Big(\frac{128\ell^2}{\Delta}\Big)^\ell .
\end{aligned}$$

Consequently, by virtue of Eq. (4.33) and the definition of t_n we get

$$\begin{aligned}
|\beta_{\Delta,R,n}| &\leq \frac{8c_\eta R}{\Delta^2}\Big(\frac{128\ell^2}{\Delta|t_n|}\Big)^\ell \\
&\leq \frac{8c_\eta R}{\Delta^2}\Big(\frac{64R\ell^2}{\Delta|n|}\Big)^\ell . \qquad (4.38)
\end{aligned}$$

Now the idea is to choose the integer ℓ in such a way that the right side of this inequality decays exponentially in $|n|^{1/2}$. We do this by first considering large $|n|$ and then small $|n|$. For $|n|$ large enough it is possible to find ℓ such that

$$\Big(\frac{|n|\Delta}{256R}\Big)^{1/2} \leq \ell \leq \Big(\frac{|n|\Delta}{128R}\Big)^{1/2} . \qquad (4.39)$$

Inserting the upper bound of (4.39) into the parenthesis of Eq. (4.38) results in

$$|\beta_{\Delta,R,n}| \leq \frac{8c_\eta R}{\Delta^2}2^{-\ell} ,$$

and by inserting the lower bound of (4.39) into the exponent we obtain

$$|\beta_{\Delta,R,n}| \leq \frac{8c_\eta R}{\Delta^2}2^{-(\frac{|n|\Delta}{256R})^{1/2}} . \qquad (4.40)$$

Condition Eq. (4.39) is fulfilled if $\big(\frac{|n|\Delta}{128R}\big)^{1/2}$ and $\big(\frac{|n|\Delta}{256R}\big)^{1/2}$ are at least an integer apart, i.e. for

$$\Big(\frac{|n|\Delta}{128R}\Big)^{1/2} - \Big(\frac{|n|\Delta}{256R}\Big)^{1/2} \geq 1 , \qquad (4.41)$$

which is equivalent to the condition $|n| \geq \frac{256R}{(\sqrt{2}-1)^2\Delta}$. Now we will show that bound Eq. (4.40) holds also for smaller values of $|n|$, i.e. for $|n| \leq \frac{256R}{(\sqrt{2}-1)^2\Delta}$. From the

definition of $\beta_{\Delta,R,n}$ one obtains the upper bound

$$|\beta_{\Delta,R,n}| \leq \|g_{\Delta,R}\|_\infty = \frac{4}{\Delta}\,. \tag{4.42}$$

Then, for $|n| \leq \frac{256R}{(\sqrt{2}-1)^2\Delta}$ and using $R \geq \Delta$ and $c_\eta \approx 7.514$, we get

$$\begin{aligned} |\beta_{\Delta,R,n}| &\leq \frac{4R}{\Delta^2} \\ &\leq \frac{4c_\eta R}{\Delta^2} 2^{-\frac{1}{\sqrt{2}-1}} \\ &\leq \frac{4c_\eta R}{\Delta^2} 2^{-(\frac{|n|\Delta}{256R})^{1/2}}\,. \end{aligned}$$

This shows that the bound of Eq. (4.40) holds for all n, and by inserting the numerical value for c_η we obtain the bound as stated in the lemma. □

4.7.2 Truncation Error of Second-Order Energy

In this part of the Appendix we show how the error $|E^{(2)} - E_N^{(2)}|$ depends on the order of truncation N of the Fourier series of $g_{\Delta,R}$. We prove the following Lemma:

Lemma 4.7.2. *For R, Δ and $\|v\|_1$ as defined in Section 4.3 and for $\epsilon > 0$, there is an integer $\widetilde{N} = \widetilde{N}(R, \Delta, \epsilon, \|v\|_1)$ such that the truncation error $|E^{(2)} - E_N^{(2)}|$ is bounded by*

$$|E^{(2)} - E_N^{(2)}| \leq \frac{\epsilon}{2} \quad \textit{if} \quad N \geq \widetilde{N}\,,$$

and furthermore, $\widetilde{N} \leq \left\lceil \frac{2132R}{\Delta} \ln^2\left(\frac{5.3\cdot 10^4 R^2 \|v\|_1^2}{\Delta^3 \epsilon}\right)\right\rceil$.

Proof: With Eq. (4.10) and Eq. (4.12) we have

$$|E^{(2)} - E_N^{(2)}| = |\langle V R_0 V\rangle - \langle V R_{0,N} V\rangle|$$

an by inserting Eq. (4.9) and Eq. (4.11) we get

$$\begin{aligned} |E^{(2)} - E_N^{(2)}| &\leq \sum_{|n|>N} |\beta_{\Delta,R,n}|\,|\,\langle V U(t_n) V\rangle| \\ &\leq \sum_{|n|>N} |\beta_{\Delta,R,n}|\,\|V\,|\Phi_0\rangle\|_2 \|U(t_n) V\,|\Phi_0\rangle\|_2 \\ &= \|V\,|\Phi_0\rangle\|_2^2 \sum_{|n|>N} |\beta_{\Delta,R,n}| \\ &= \langle V^2\rangle \sum_{|n|>N} |\beta_{\Delta,R,n}|\,, \end{aligned}$$

where in the second line the Cauchy-Schwarz inequality was used. With the decomposition of V into Pauli operators Eq. (4.4) $\langle V^2 \rangle$ is upper bounded by

$$\langle V^2 \rangle = \sum_{i,j=1}^{L_V} v_i v_j \langle \sigma_i \sigma_j \rangle \leq \|v\|_1^2 ,$$

such that

$$|E^{(2)} - E_N^{(2)}| \leq \|v\|_1^2 \sum_{|n|>N} |\beta_{\Delta,R,n}| .$$

We use constants $M_{\Delta,R} = \frac{61R}{\Delta^2}$ and $q_{\Delta,R} = \ln(2)\left(\frac{\Delta}{256R}\right)^{\frac{1}{2}}$ to write the bound on the Fourier coefficients $\beta_{\Delta,R,n}$ of Lemma 4.7.1 in a more compact form,

$$|\beta_{\Delta,R,n}| \leq M_{\Delta,R} e^{-q_{\Delta,R}|n|^{1/2}} .$$

Now we bound:

$$\begin{aligned}
\sum_{|n|>N} |\beta_{\Delta,R,n}| &\leq M_{\Delta,R} \sum_{|n|>N} e^{-q_{\Delta,R}|n|^{\frac{1}{2}}} \\
&\leq 2M_{\Delta,R} \int_N^\infty e^{-q_{\Delta,R}x^{\frac{1}{2}}} dx \\
&= \frac{4M_{\Delta,R}}{q_{\Delta,R}^2} \int_{q_{\Delta,R}N^{\frac{1}{2}}}^\infty y e^{-y} dy \\
&= \frac{4M_{\Delta,R}}{q_{\Delta,R}^2} (q_{\Delta,R}N^{\frac{1}{2}} + 1) e^{-q_{\Delta,R}N^{\frac{1}{2}}} \\
&\leq \frac{8M_{\Delta,R}}{q_{\Delta,R}^2} e^{-\frac{q_{\Delta,R}}{2}N^{\frac{1}{2}}} ,
\end{aligned}$$

where in the second line $xe^{-x} \leq e^{-\frac{x}{2}}$ was used. As a result, the truncation error is bounded by

$$|E^{(2)} - E_N^{(2)}| \leq \frac{8M_{\Delta,R}\|v\|_1^2}{q_{\Delta,R}^2} e^{-\frac{q_{\Delta,R}}{2}N^{\frac{1}{2}}} . \tag{4.43}$$

Now define

$$\widetilde{N} \equiv \left\lceil \frac{4}{q_{\Delta,R}^2} \ln^2\left(\frac{16M_{\Delta,R}\|v\|_1^2}{q_{\Delta,R}^2\epsilon}\right) \right\rceil ,$$

then, for $N \geq \widetilde{N}$, and via Eq. (4.43), we get

$$|E^{(2)} - E_N^{(2)}| \leq \frac{\epsilon}{2} . \tag{4.44}$$

Lastly, we insert the definitions of $M_{\Delta,R}$ and $g_{\Delta,R}$ to get the final result

$$\widetilde{N} \leq \left\lceil \frac{2132R}{\Delta} \ln^2\left(\frac{5.3 \cdot 10^4 R^2 \|v\|_1^2}{\Delta^3\epsilon}\right) \right\rceil \qquad \square$$

Overlap Estimation with Iterative Quantum Amplitude Estimation

As explained in Section 4.3, a key aspect of our algorithm is to compute overlaps using a quantum computer. The following lemma provides upper bounds on the costs for computing such overlaps:

Lemma 4.7.3. *Given a confidence level $1-\alpha \in (0,1)$, a target accuracy $\epsilon > 0$ and d-qubit unitaries U and U_0 with $U_0|0\rangle = |\Phi_0\rangle$, it is possible to obtain an estimate X' to the overlap $X = \langle\Phi_0|U|\Phi_0\rangle$ such that*

$$\mathbb{P}[|X' - X| < \epsilon] \geq 1 - \alpha$$

with $d+1$ qubits, and at most $\frac{1116}{\epsilon}\ln\frac{63}{\alpha}$ applications of U_0 and at most $\frac{558}{\epsilon}\ln\frac{63}{\alpha}$ applications of controlled-U.

The algorithm behind Lemma 4.7.3 is amplitude estimation, in particular the Iterative Quantum Amplitude Estimation algorithm (IQAE) [236][237] . Before we prove Lemma 4.7.3, we give a brief summary of amplitude estimation and present key aspects of IQAE. With Lemma 4.7.6 we will cast the query-complexity of IQAE as shown in Ref. [237] into a suitable form to prove Lemma 4.7.3.

In the general setting of amplitude estimation a state $|\psi\rangle$ is given, and one is interested in the component of $|\psi\rangle$ along another state $|\psi'\rangle$. Access is given to a unitary U_ψ preparing

$$U_\psi|0\rangle = |\psi\rangle = \sqrt{a}\,|\psi'\rangle + \sqrt{1-a}\,|\psi'^{\perp}\rangle\,, \quad \text{with} \quad \langle\psi'^{\perp}|\psi'\rangle = 0\,,$$

and the challenge is to estimate the amplitude $a = |\langle\psi'|\psi\rangle|^2$ with as few applications of U_ψ as possible. It is convenient to introduce an angle $\theta_a \in [0, \frac{\pi}{2}]$ such that $\sin^2(\theta_a) = a$. A unitary operator $Q = -S_\psi S_{\psi'}$ is defined, where

$$S_\psi = I - 2\,|\psi\rangle\langle\psi|$$
$$S_{\psi'} = I - 2\,|\psi'\rangle\langle\psi'|$$

are reflection operators along $|\psi\rangle$ and $|\psi'\rangle$, respectively. Note that with access to the state-preparation unitary U_ψ one can implement S_ψ as

$$S_\psi = U_\psi Z_0 U_\psi^\dagger \quad \text{where} \quad Z_0 = I - 2\,|0\rangle\langle 0|\,. \tag{4.45}$$

The key insight of amplitude estimation is that the probability of measuring $|\psi'\rangle$ in the state $Q^k|\psi\rangle$ is [260]:

$$\mathbb{P}[|\psi'\rangle] = |\,\langle\psi'|Q^k|\psi\rangle|^2 = \sin^2((2k+1)\theta_a)\,. \tag{4.46}$$

As other amplitude estimation algorithms, IQAE estimates $\sin^2((2k+1)\theta_a)$ for different powers of k and uses that information to infer the value of a. The algorithm iteratively calculates estimates to θ_a and stops once a specified precision is attained. Compared to the textbook amplitude estimation algorithm by Brassard et al. [260], IQAE has the advantage that no quantum Fourier transform is needed. IQAE was first published in 2021 by Grinko and coworkers [236], and this year Fukuzawa et al.[237] presented an updated version of IQEA ('modified IQAE') for which it was shown that optimal query complexity for amplitude estimation is achieved. Their main result is the following theorem:

Theorem 4.7.4 (Theorem 3.1 [237]). *Given a confidence level $1-\alpha \in (0,1)$, a target accuracy $\epsilon > 0$, and a (d+1)-qubit unitary $U_{\psi,flag}$ satisfying*

$$U_{\psi,flag}\left|0\right\rangle\left|0\right\rangle_{flag} = \sqrt{a}\left|\psi'\right\rangle\left|0\right\rangle_{flag} + \sqrt{a-1}\left|\psi'^{\perp}\right\rangle\left|1\right\rangle_{flag}$$

where $\left|\psi'\right\rangle$ and $\left|\psi'^{\perp}\right\rangle$ are arbitrary d-qubit states and $a \in [0,1]$, modified IQAE outputs a confidence interval for a that satisfies $\mathbb{P}[a \notin [a_l, a_u]] \leq \alpha$, where $a_u - a_l < 2\epsilon$, leading to an estimate $\hat{a}$ for a such that $|a-\hat{a}| < \epsilon$ with a confidence of $1-\alpha$, using $\mathcal{O}(\frac{1}{\epsilon}\ln\frac{1}{\alpha})$ applications of $U_{\psi,flag}$.

More specifically, Lemma 3.7. of the same reference gives an explicit bound:

Lemma 4.7.5 (Lemma 3.7 [237]). *Modified IQAE requires at most $\frac{62}{\epsilon}\ln\frac{21}{\alpha}$ applications of $U_{\psi,flag}$, achieving the desired query complexity of $\mathcal{O}(\frac{1}{\epsilon}\ln\frac{1}{\alpha})$*

In our setting, U_ψ does not include the action on an ancillary flag qubit, instead we have access to a unitary $U_{\psi'}$ preparing $\left|\psi'\right\rangle$. Therefore, we adapt Lemma 4.7.5 to our setting and show

Lemma 4.7.6. *Given a confidence level $1-\alpha \in (0,1)$, a target accuracy $\epsilon > 0$, a d-qubit state preparation unitary $U_{\psi'}$ with $U_{\psi'}\left|0\right\rangle = \left|\psi'\right\rangle$ and another unitary V, modified IQAE leads to an estimate $\hat{a}$ for*

$$a = |\left\langle\psi'|V|\psi'\right\rangle|^2$$

such that $|a-\hat{a}| < \epsilon$ with a confidence of $1-\alpha$, using at most $\frac{62}{\epsilon}\ln\frac{21}{\alpha}$ applications of V and at most $\frac{124}{\epsilon}\ln\frac{21}{\alpha}$ applications of $U_{\psi'}$.

Proof: The idea is to define $U_\psi = VU_{\psi'}$ with $U_\psi\left|0\right\rangle = \left|\psi\right\rangle$ such that $a = |\left\langle\psi'|\psi\right\rangle|^2$, and apply the results of modified IQAE. First, we explain how to run IQAE without the ancillary flag qubit. The quantum computing part of IQAE consists of measuring the flag qubit of the circuit $Q^k U_{\psi,flag}\left|0\right\rangle\left|0\right\rangle_{\text{flag}}$ to get

$$\mathbb{P}[\left|0\right\rangle_{\text{flag}}] = \sin^2((2k+1)\theta_a)\,.$$

However, with access to the state-preparation unitary $U_{\psi'}$ one can perform the measurement of Eq. (4.46) directly by preparing the circuit $U_{\psi'}^\dagger Q^k U_\psi |0\rangle$ and measuring in the computational basis,

$$\mathbb{P}[|0\rangle] = |\langle 0|U_{\psi'}^\dagger Q^k U_\psi|0\rangle|^2 = \sin^2((2k+1)\theta_a)\,.$$

Thus, IQAE can readily be applied to the setting where U_ψ does not include the action on a flag qubit. According to Theorem 3.1 of Ref. [237], IQAE will lead to an estimate $\hat{a}$ for $a = |\langle\psi'|V|\psi'\rangle|^2 = |\langle\psi'|\psi\rangle|^2$ such that $|\hat{a}-a| < \epsilon$ with confidence $1-\alpha$. It remains now to prove the upper bounds on the number of applications of $U_{\psi'}$ and V.

As stated earlier in Eq. (4.45), the reflection operators, and therefore the operator Q, can be implemented through the state preparation unitaries U_ψ and $U_{\psi'}$

$$Q = -S_\psi S_{\psi'} = -U_\psi Z_0 U_\psi^\dagger U_{\psi'} Z_0 U_{\psi'}^\dagger\,.$$

IQAE runs the circuits $U_{\psi'}^\dagger Q^k U_\psi |0\rangle$ for different values of k. In each circuit, the number of applications of $U_{\psi'}$ equals the number of applications of U_ψ, $2k+1$ applications each. Thus, we can use the upper bound $\frac{62}{\epsilon}\ln\frac{21}{\alpha}$ on the number of applications of U_ψ given in Lemma 4.7.5 both for U_ψ and $U_{\psi'}$. Lastly, with the definition $U_\psi = VU_{\psi'}$, the upper bounds $\frac{124}{\epsilon}\ln\frac{21}{\alpha}$ for $U_{\psi'}$ and $\frac{62}{\epsilon}\ln\frac{21}{\alpha}$ for V follow. □

We now have all ingredients to prove Lemma 4.7.3.

Proof Lemma 4.7.3: First, the overlap is written in terms of amplitude and phase, and in terms of real and imaginary part:

$$X = re^{i\theta} = r\cos(\theta) + ir\sin(\theta), \quad r \in \mathbb{R}_{\geq \not<}, \quad \theta \in [0, 2\pi)\,.$$

We consider the following amplitudes

$$x_0 = |\langle\Phi_0|U|\Phi_0\rangle|^2 = r^2, \tag{4.47}$$

$$x_1 = |\langle\Phi_0|\langle+|\,cU\,|\Phi_0\rangle|+\rangle|^2 = \frac{1}{2}\left|1+re^{i\theta}\right|^2, \tag{4.48}$$

$$x_2 = |\langle\Phi_0|\langle+|\,cU(I\otimes e^{iZ\frac{\pi}{4}})\,|\Phi_0\rangle|+\rangle|^2 = \frac{1}{2}\left|1-ire^{i\theta}\right|^2, \tag{4.49}$$

where $cU = I \otimes |0\rangle\langle 0| + U \otimes |1\rangle\langle 1|$. As noted in Ref. [261], from x_0, x_1 and x_2 we can extract

$$r = \sqrt{x_0}, \tag{4.50}$$

$$\cos\theta = \frac{4x_1 - 1 - x_0}{2\sqrt{x_0}}, \tag{4.51}$$

$$\sin\theta = -\frac{4x_2 - 1 - x_0}{2\sqrt{x_0}}, \tag{4.52}$$

so the amplitudes x_0, x_1 and x_2, lead to the value of X. The strategy is to compute estimates $\hat{x}_0, \hat{x}_1$ and $\hat{x}_2$ to x_0, x_1 and x_2 via IQAE, in particular the IQAE version laid out in Lemma 3. For the estimation of x_0, x_1 and x_2, the unitaries V and $U_{\psi'}$ of Lemma 4.7.6 correspond to:

$$\begin{aligned} x_0: &\quad V = U && U_{\psi'} = U_0 \\ x_1: &\quad V = cU && U_{\psi'} = U_0 \otimes \mathrm{H} \\ x_2: &\quad V = cU(I \otimes e^{iZ\frac{\pi}{4}}) && U_{\psi'} = U_0 \otimes \mathrm{H}\,. \end{aligned} \tag{4.53}$$

The accuracy and confidence parameters for the estimation of $x_i\,(i = 0, 1, 2)$ will be denoted by ϵ_i and α_i respectively, i.e. with IQAE we get x_i' such that $|x_i' - x_i| < \epsilon_i$ with confidence $1 - \alpha_i$. The estimate to X we get with our strategy is denoted by $X' = r'e^{i\theta'}$, where r' and θ' are related to x_0', x_1' and x_2' analogous to Eqs. (4.50)-(4.52). The estimation error is bounded by

$$\begin{aligned} |X' - X| &= |r'e^{i\theta'} - re^{i\theta}| \\ &\leq |r'\cos\theta' - r\cos\theta| + |r'\sin\theta' - r\sin\theta| \\ &\leq |x_0' - x_0| + 2|x_1' - x_1| + 2|x_2' - x_2| \\ &\leq \epsilon_0 + 2\epsilon_1 + 2\epsilon_2\,. \end{aligned}$$

We choose $\epsilon_0 = \epsilon_1 = \epsilon_2 = \epsilon/5$, resulting in

$$|X' - X| \leq \epsilon\,.$$

Moreover, since the computations for x_0', x_1' and x_2' are independent, the probability of obtaining all three estimates with specified accuracies ϵ_i is given by

$$\begin{aligned} &\mathbb{P}[|x_i' - x_i| < \epsilon_i\ \forall i \in \{0, 1, 2\}] \\ &\geq (1 - \alpha_0)(1 - \alpha_1)(1 - \alpha_2) \\ &\geq 1 - (\alpha_0 + \alpha_1 + \alpha_2)\,. \end{aligned}$$

We choose $\alpha_0 = \alpha_1 = \alpha_2 = \alpha/3$ to obtain the desired confidence

$$\begin{aligned} \mathbb{P}[|X' - X| \leq \epsilon] &= \mathbb{P}[|x_i' - x_i| < \epsilon_i\ \forall i \in \{0, 1, 2\}] \\ &\geq 1 - \alpha\,. \end{aligned}$$

Lastly, we show the upper bounds on the number of applications of cU and U_0 by employing the upper bounds on the number of calls to V and $U_{\psi'}$ of Lemma 4.7.6. For simplicity, we count an application of U as an application of cU. From Eq. (4.53) we see that for each case $i = 0, 1, 2$, the unitaries $U_{\psi'}$ and V contain U_0 and cU once, respectively. Consequently, the total number of applications of U_0 is at most $\frac{186}{\epsilon_i} \ln \frac{21}{\alpha_i} = \frac{558}{\epsilon} \ln \frac{63}{\alpha}$, and the total number of applications of cU is at most $\frac{378}{\epsilon_i} \ln \frac{21}{\alpha_i} = \frac{1116}{\epsilon} \ln \frac{63}{\alpha}$. □

The next lemma applies the overlap estimation to linear combinations of unitaries:

Lemma 4.7.7. *Given a confidence level $1 - \alpha \in (0, 1)$, a target accuracy $\epsilon > 0$, a linear combination of d-qubit unitaries $V_U = \sum_{i=1}^{L} u_i U_i$ and a d-qubit state-preparation unitary U_0, then it is possible to obtain an estimate X' to the overlap $X = \langle \Phi_0 | V_U | \Phi_0 \rangle$ such that*

$$\mathbb{P}[|X' - X| < \epsilon] \geq 1 - \alpha$$

with 1 ancillary qubit, a total $\mathcal{O}\left(\epsilon^{-1} \|u\|_{2/3} \ln \frac{L}{\alpha} \sqrt{\ln \frac{1}{\alpha}}\right)$ applications of U_0 and $\mathcal{O}\left(\epsilon^{-1} |u_i|^{2/3} \|u\|_{2/3}^{1/2} \ln \frac{L}{\alpha} \sqrt{\ln \frac{1}{\alpha}}\right)$ applications of controlled-U_i for each i=1,…,L, where $\|u\|_{2/3} = \left(\sum_{i=1}^{L} |u_i|^{2/3}\right)^{3/2}$.

Proof: The quantum algorithm of Lemma 4.7.3 is applied to each U_i and the results are summed to get

$$\sum_{i=1}^{L} u_i X_i' = X' .$$

Each term $X_i = \langle \Phi_0 | U_i | \Phi_0 \rangle$ is associated with a confidence level $1 - \alpha_i \in (0, 1)$ and an accuracy $\epsilon_i > 0$, i.e. $\mathbb{P}[|X_i' - X_i| < \epsilon_i] \geq 1 - \alpha'$. We choose to have the same confidence level $1 - \alpha'$ for each term, but we will optimize the individual accuracies ϵ_i to reduce the overall runtime. The probability of all L terms X_i' to lie in their respective interval $(X_i - \epsilon_i, X_i + \epsilon_i)$ is lower bounded by

$$\mathbb{P}\big[|X_i' - X_i| < \epsilon_i, \quad \forall 1 \leq i \leq L\big] = (1 - \alpha')^L \geq 1 - L\alpha' ,$$

where Bernoulli's inequality, $(1 + x)^r \geq 1 + xr$ for $x \geq -1$ and integer $r \geq 1$, was used in the last line. By choosing $\alpha' = \frac{\alpha}{2L}$ we know that with probability $1 - \frac{\alpha}{2}$ all random variables $u_i(X_i' - X_i)$ are bounded by $2|u_i|\epsilon_i$. Under the assumption

that $|X_i' - X_i| \leq \epsilon_i$ for all i we can upper bound the probability that the global error $|X' - X|$ is large:

$$\begin{aligned}
&\mathbb{P}[|X' - X| \geq \epsilon \text{ AND } |X_i' - X_i| \leq \epsilon_i \,\forall i] \\
=&\mathbb{P}\left(\left|\sum_{i=1}^{L} u_i(X_i' - X_i) \geq \epsilon\right| \text{ AND } |X_i' - X_i| \leq \epsilon_i \,\forall i\right) \\
\leq&\mathbb{P}\left(\sum_{i=1}^{L} |u_i||X_i' - X_i| \geq \epsilon \text{ AND } |X_i' - X_i| \leq \epsilon_i \,\forall i\right) \\
\leq&\exp\left\{-\frac{\epsilon^2}{2\sum_i |u_i|^2 \epsilon_i^2}\right\} \leq \frac{\alpha}{2}\,,
\end{aligned} \tag{4.54}$$

where Hoeffding's inequality was employed from third to fourth line for the random variables $|u_i||X_i' - X_i|$. Together with the probability that there is at least one i for which $|X_i' - X_i| > \epsilon_i$ we can upper bound

$$\begin{aligned}
\mathbb{P}[|X' - X| \geq \epsilon] \leq\ &\mathbb{P}[|X' - X| \geq \epsilon \text{ AND } |X_i' - X_i| \leq \epsilon_i \,\forall i] \\
&+ \mathbb{P}[|X_i' - X_i| > \epsilon_i \text{ for some } i] \\
&\leq \alpha,
\end{aligned}$$

resulting in the global confidence $1 - \alpha$.

Next, Hoeffding's inequality in Eq. (4.54) from above is used to relate global accuracy ϵ, individual accuracies ϵ_i and α, so we rewrite the inequality to

$$\epsilon^2 \geq 2\ln\left(\frac{2}{\alpha}\right) \sum_{i=1}^{L} |u_i|^2 \epsilon_i^2\,.$$

Lemma 4.7.3 shows that the cost of computing X_i with accuracy ϵ_i is proportional to $\frac{c}{\epsilon_i}$ for some constant c, so we set up a Lagrangian minimization problem in the variables ϵ_i:

$$\mathcal{L} = \sum_{i=1}^{L} \frac{c}{\epsilon_i} + \lambda\left(2\ln\left(\frac{2}{\alpha}\right) \sum_{i=1}^{L} |u_i|^2 \epsilon_i^2 - \epsilon^2\right)\,.$$

Solving $\frac{\partial \mathcal{L}}{\partial \epsilon_i} = 0$ and $\frac{\partial \mathcal{L}}{\partial \lambda} = 0$ results in the optimized values

$$\epsilon_i = \frac{\epsilon}{\sqrt{2\ln(2/\alpha)}|u_i|^{2/3}\|u\|_{2/3}^{1/2}}\,,$$

where $\|u\|_{2/3} = (\sum_j |u_j|^{2/3})^{3/2}$. Therefore, by Lemma 4.7.3, for each i the number of applications of U_0 and controlled-U_i is

$$\mathcal{O}\left(\epsilon^{-1}|u_i|^{2/3}\|u\|_{2/3}^{1/2} \ln\frac{L}{\alpha}\sqrt{\ln\frac{1}{\alpha}}\right).$$

Consequently, the cumulative number of applications of U_0 and controlled-U_i for all i is

$$\mathcal{O}\left(\epsilon^{-1}\|u\|_{2/3}\ln\frac{L}{\alpha}\sqrt{\ln\frac{1}{\alpha}}\right). \qquad \square$$

4.7.3 Complexity Analysis of Quantum Algorithm

In this part of the Appendix we discuss the complexity analysis of the quantum algorithm presented in Section 4.3 for calculating the second-order perturbation energy $E^{(2)}$. Lemma 4.7.1 and Lemma 4.7.7 are the main ingredients of the proof, allowing us put bounds on the truncation error $|E^{(2)} - E_N^{(2)}|$ and on the error due to finite precision in estimating overlap terms.

Lemma 4.7.8. *Given a Hamiltonian $H = \sum_{i=1}^{L_H} h_i\sigma_i$ with spectral gap Δ and a perturbation $V = \sum_{i=1}^{L_V} v_i\sigma_i$, both acting on d qubits, and given access to a unitary U_0 preparing the ground state $|\Phi_0\rangle$ of H, it is possible to compute an estimate to the second-order perturbation energy $E^{(2)} = \langle\Phi_0|VR_0V|\Phi_0\rangle$ using $d+1$ qubits and with precision ϵ and confidence $1-\alpha \in (0,1)$. Computing the estimate to $E^{(2)}$ requires implementing the Hamiltonian evolution under H for a total time upper bounded by*

$$\mathcal{O}\left(\frac{\|v\|_{2/3}^2\|h\|_1^{3/2}}{\Delta^{5/2}\epsilon}\ln^3\left(\frac{\|h\|_1^2\|v\|_1^2}{\Delta^3\epsilon}\right)\right.$$
$$\left.\times\ln\left(\frac{L_V^2\|h\|_1}{\alpha\Delta}\ln^2\left(\frac{\|h\|_1^2\|v\|_1^2}{\Delta^3\epsilon}\right)\right)\sqrt{\ln\frac{1}{\alpha}}\right).$$

Furthermore, the total number of calls to U_0 is upper bounded by

$$\mathcal{O}\left(\frac{\|v\|_{2/3}^2\|h\|_1^{3/2}}{\Delta^{7/2}\epsilon}\ln^5\left(\frac{\|h\|_1^2\|v\|_1^2}{\Delta^3\epsilon}\right)\right.$$
$$\left.\times\ln\left(\frac{L_V^2\|h\|_1}{\alpha\Delta}\ln^2\left(\frac{\|h\|_1^2\|v\|_1^2}{\Delta^3\epsilon}\right)\right)\sqrt{\ln\frac{1}{\alpha}}\right).$$

Proof: We explain how the errors that are involved are bounded, and afterwards work out the resulting cost of the quantum algorithm. Two types of errors need to be controlled: The truncation error $|E^{(2)} - E_N^{(2)}|$ in the Fourier series and the error due to the finite precision in the estimation of the individual overlap

terms. The full second-order correction, $E^{(2)}$, and the estimate obtained by the algorithm, denoted by $X_N^{(2)}$, are given by (compare also to $E_N^{(2)}$ Eq. 4.13)

$$E^{(2)} = \sum_{n\in\mathbb{Z}} \beta_{\Delta,R,n} e^{iE_0 t_n} \sum_{i,j=1}^{L_V} v_i v_j \langle \sigma_i U(t_n) \sigma_j \rangle$$
$$X_N^{(2)} = \sum_{|n|\leq N} \beta_{\Delta,R,n} e^{iE_0 t_n} \sum_{i,j=1}^{L_V} v_i v_j X_{nij} \,,$$

where X_{nij} is the finite-sample estimate to $\langle \sigma_i U(t_n) \sigma_j \rangle$. For the truncation error, Lemma 4.7.2 provides the bound

$$|E^{(2)} - E_N^{(2)}| \leq \epsilon \quad \text{if} \quad N \geq \widetilde{N} \,,$$

where $\widetilde{N}$ satisfies

$$\widetilde{N} \leq \mathcal{O}\left(\frac{R}{\Delta} \ln^2\left(\frac{R^2 \|v\|_1^2}{\Delta^3 \epsilon}\right)\right) . \tag{4.55}$$

Next, we discuss the error

$$|X_{\widetilde{N}}^{(2)} - E_{\widetilde{N}}^{(2)}| \leq \sum_{|n|\leq \widetilde{N}} \sum_{i,j=1}^{L_V} |\beta_{\Delta,R,n} v_i v_j| \times |X_{nij} - \langle \sigma_i U(t_n) \sigma_j \rangle|$$

due to finite precision in the estimation of the overlap. This will result in an upper bound on the total number of calls to U_0 and an upper bound on the total evolution time. With Lemma 4.7.7 we know that with confidence $1-\alpha$ an ϵ-approximation $X_{\widetilde{N}}^{(2)}$ to $E_{\widetilde{N}}^{(2)}$ can be obtained, such that

$$|E^{(2)} - X_{\widetilde{N}}^{(2)}| \leq |E^{(2)} - E_{\widetilde{N}}^{(2)}| + |E_{\widetilde{N}}^{(2)} - X_{\widetilde{N}}^{(2)}| \leq 2\epsilon \,,$$

as desired. The cost of computing $X_{\widetilde{N}}^{(2)}$ depends on the total number of overlap

terms in $E_N^{(2)}$, which is $L_V^2 \widetilde{N}$, and the coefficients,

$$\begin{aligned}\left\|v^2\beta_{\Delta,R}\right\|_{2/3} &\equiv \left(\sum_{|n|\leq\widetilde{N}}\sum_{i,j=1}^{L_V}\left|\beta_{\Delta,R,n}v_iv_j\right|^{2/3}\right)^{3/2} \\ &= \|v\|_{2/3}^2\left(\sum_{|n|\leq\widetilde{N}}\left|\beta_{\Delta,R,n}\right|^{2/3}\right)^{3/2} \\ &= \mathcal{O}\left(\frac{\|v\|_{2/3}^2 R^{3/2}}{\Delta^{5/2}}\ln^3\left(\frac{R^2\|v\|_1^2}{\Delta^3\epsilon}\right)\right),\end{aligned}$$

where equations (4.42) and (4.55) were used. Therefore, by Lemma 4.7.7, the total number of calls to U_0 is upper bounded by

$$\mathcal{O}\left(\frac{\left\|v^2\beta_{\Delta,R}\right\|_{2/3}}{\epsilon}\ln\left(\frac{L_V^2\widetilde{N}}{\alpha}\right)\sqrt{\ln\frac{1}{\alpha}}\right) \tag{4.56}$$

$$= \mathcal{O}\left(\frac{\|v\|_{2/3}^2 R^{3/2}}{\Delta^{5/2}\epsilon}\ln^3\left(\frac{R^2\|v\|_1^2}{\Delta^3\epsilon}\right)\right. \tag{4.57}$$

$$\left.\times\ln\left(\frac{L_V^2 R}{\alpha\Delta}\ln^2\left(\frac{R^2\|v\|_1^2}{\Delta^3\epsilon}\right)\right)\sqrt{\ln\frac{1}{\alpha}}\right). \tag{4.58}$$

Next, we will upper bound the total evolution time. If we upper bound the evolution time t_n of each term by $t_{\widetilde{N}}$, then we can multiply $t_{\widetilde{N}} = \frac{2\pi\widetilde{N}}{R+\Delta}$ by the overall number of applications of all terms to obtain an upper bound on the total evolution time (see Lemma 4.7.7):

$$\mathcal{O}\left(\frac{\widetilde{N}\left\|v^2\beta_{\Delta,R}\right\|_{2/3}}{R\epsilon}\ln\left(\frac{L_V^2\widetilde{N}}{\alpha}\right)\sqrt{\ln\frac{1}{\alpha}}\right) \tag{4.59}$$

$$= \mathcal{O}\left(\frac{\|v\|_{2/3}^2 R^{3/2}}{\Delta^{7/2}\epsilon}\ln^5\left(\frac{R^2\|v\|_1^2}{\Delta^3\epsilon}\right)\right. \tag{4.60}$$

$$\left.\times\ln\left(\frac{L_V^2 R}{\alpha\Delta}\ln^2\left(\frac{R^2\|v\|_1^2}{\Delta^3\epsilon}\right)\right)\sqrt{\ln\frac{1}{\alpha}}\right) \tag{4.61}$$

Lastly, the range $R = E_{D-1} - E_0$ of the eigenvalues of H is upper bounded

by

$$\begin{aligned} R &\leq 2 \max_{i=0,\ldots,D-1} |E_i| \\ &= 2 \sup_{\||\psi\rangle\|=1} \|H|\psi\rangle\|_2 \\ &\leq 2 \sup_{\||\psi\rangle\|=1} \sum_{i=1}^{L_H} |h_i| \|\sigma_i |\psi\rangle\|_2 \\ &\leq 2\|h\|_1 . \end{aligned} \tag{4.62}$$

Inserting equation (4.62) into equations (4.58) and (4.61) results in the complexities stated in the Lemma. □

4.7.4 Simplifications for MRPT

This part of the Appendix completes the derivation of all four terms in Eq. (4.23), starting from Eq. (4.22). As explained in the main text, the creation and annhiliation operators of virtual orbitals must appear in pairs, otherwise a term

vanishes. Therefore we can write out

$$
\begin{aligned}
T_1(t) &= \sum_{\substack{v\in\mathcal{I}_{virt}\\ a,b\in\mathcal{I}_{act}}} h'_{va}h'_{vb}\,\langle a_b^\dagger a_v U(t) a_v^\dagger a_a\rangle \\
T_2(t) &= \tfrac{1}{2}\sum_{\substack{v\in\mathcal{I}_{virt}\\ a,b,c,d\in\mathcal{I}_{act}}} h'_{vd}g_{vabc}\,\langle a_d^\dagger a_v U(t) a_v^\dagger a_a^\dagger a_b a_c\rangle \\
&\quad + \tfrac{1}{2}\sum_{\substack{v\in\mathcal{I}_{virt}\\ a,b,c,d\in\mathcal{I}_{act}}} h'_{vd}\underbrace{g_{avbc}}_{g_{vacb}}\,\langle a_d^\dagger a_v U(t)\underbrace{a_a^\dagger a_v^\dagger a_b a_c}_{a_v^\dagger a_a^\dagger a_c a_b}\rangle \\
T_3(t) &= \tfrac{1}{2}\sum_{\substack{v\in\mathcal{I}_{virt}\\ a,b,c,d\in\mathcal{I}_{act}}} h'_{va}g_{vbcd}\,\langle a_d^\dagger a_c^\dagger a_b a_v U(t) a_v^\dagger a_a\rangle \\
&\quad + \tfrac{1}{2}\sum_{\substack{v\in\mathcal{I}_{virt}\\ a,b,c,d\in\mathcal{I}_{act}}} h'_{va}\underbrace{g_{bvcd}}_{g_{vbdc}}\,\langle\underbrace{a_d^\dagger a_c^\dagger a_v a_b}_{a_c^\dagger a_d^\dagger a_b a_v} U(t) a_v^\dagger a_a\rangle \\
T_4(t) &= \tfrac{1}{4}\sum_{\substack{v\in\mathcal{I}_{virt}\\ a,b,c,d,e,f\in\mathcal{I}_{act}}} g_{vabc}g_{vdef}\,\langle a_f^\dagger a_e^\dagger a_d a_v U(t) a_v^\dagger a_a^\dagger a_b a_c\rangle \\
&\quad + \tfrac{1}{4}\sum_{\substack{v\in\mathcal{I}_{virt}\\ a,b,c,d,e,f\in\mathcal{I}_{act}}} g_{vabc}\underbrace{g_{dvef}}_{g_{vdfe}}\,\langle\underbrace{a_f^\dagger a_e^\dagger a_v a_d}_{a_e^\dagger a_f^\dagger a_d a_v} U(t) a_v^\dagger a_a^\dagger a_b a_c\rangle \\
&\quad + \tfrac{1}{4}\sum_{\substack{v\in\mathcal{I}_{virt}\\ a,b,c,d,e,f\in\mathcal{I}_{act}}} g_{vdef}\underbrace{g_{avbc}}_{g_{vacb}}\,\langle a_f^\dagger a_e^\dagger a_d a_v U(t)\underbrace{a_a^\dagger a_v^\dagger a_b a_c}_{a_v^\dagger a_a^\dagger a_c a_b}\rangle \\
&\quad + \tfrac{1}{4}\sum_{\substack{v\in\mathcal{I}_{virt}\\ a,b,c,d,e,f\in\mathcal{I}_{act}}} \overbrace{g_{avbc}}^{g_{vacb}}\underbrace{g_{dvef}}_{g_{vdfe}}\,\langle\underbrace{a_f^\dagger a_e^\dagger a_v a_d}_{a_e^\dagger a_f^\dagger a_d a_v} U(t)\underbrace{a_a^\dagger a_v^\dagger a_b a_c}_{a_v^\dagger a_a^\dagger a_c a_b}\rangle \\
&\quad + \tfrac{1}{4}\sum_{\substack{v,w\in\mathcal{I}_{virt}\\ a,b,c,d\in\mathcal{I}_{act}}} g_{vwab}g_{vwcd}\,\langle a_d^\dagger a_c^\dagger a_w a_v U(t) a_v^\dagger a_w^\dagger a_a a_b\rangle \\
&\quad + \tfrac{1}{4}\sum_{\substack{v,w\in\mathcal{I}_{virt}\\ a,b,c,d\in\mathcal{I}_{act}}} g_{vwab}\underbrace{g_{wvcd}}_{g_{vwdc}}\,\langle\underbrace{a_d^\dagger a_c^\dagger a_v a_w}_{a_c^\dagger a_d^\dagger a_w a_v} U(t) a_v^\dagger a_w^\dagger a_a a_b\rangle
\end{aligned}
$$

By anti-commuting the operators as indicated (and swapping indices $c \leftrightarrow d$ in the last sum) the expressions can be combined:

$$
\begin{aligned}
T_1(t) &= \sum_{\substack{v\in\mathcal{I}_{virt}\\ a,b\in\mathcal{I}_{act}}} h'_{va}h'_{vb}\,\langle a_b^\dagger a_v U(t) a_v^\dagger a_a\rangle \\
T_2(t) &= \sum_{\substack{v\in\mathcal{I}_{virt}\\ a,b,c,d\in\mathcal{I}_{act}}} h'_{vd}g_{vabc}\,\langle a_d^\dagger a_v U(t) a_v^\dagger a_a^\dagger a_b a_c\rangle \\
T_3(t) &= \sum_{\substack{v\in\mathcal{I}_{virt}\\ a,b,c,d\in\mathcal{I}_{act}}} h'_{va}g_{vbcd}\,\langle a_d^\dagger a_c^\dagger a_b a_v U(t) a_v^\dagger a_a\rangle \\
T_4(t) &= \sum_{\substack{v\in\mathcal{I}_{virt}\\ a,b,c,d,e,f\in\mathcal{I}_{act}}} g_{vabc}g_{vdef}\,\langle a_f^\dagger a_e^\dagger a_d a_v U(t) a_v^\dagger a_a^\dagger a_b a_c\rangle \\
&\quad + \tfrac{1}{2}\sum_{\substack{v,w\in\mathcal{I}_{virt}\\ a,b,c,d\in\mathcal{I}_{act}}} g_{vwab}g_{vwcd}\,\langle a_d^\dagger a_c^\dagger a_w a_v U(t) a_v^\dagger a_w^\dagger a_a a_b\rangle\,.
\end{aligned}
$$

Next, as explained in the main text, the time-evolution of the virtual operators under H is simple, $a_v(t) \equiv e^{iHt}a_v e^{-iHt} = a_v e^{-i\epsilon_v t}$, so we get

$$
\begin{aligned}
T_1(t) &= \sum_{\substack{v\in\mathcal{I}_{virt}\\ a,b\in\mathcal{I}_{act}}} h'_{va}h'_{vb}e^{-i(E_{core}+\epsilon_v)t}\,\langle a_b^\dagger e^{-iH_{CAS}t} a_a\rangle \\
T_2(t) &= \sum_{\substack{v\in\mathcal{I}_{virt}\\ a,b,c,d\in\mathcal{I}_{act}}} g_{vabc}h'_{vd}e^{-i(E_{core}+\epsilon_v)t}\,\langle a_d^\dagger e^{-iH_{CAS}t} a_a^\dagger a_b a_c\rangle \\
T_3(t) &= \sum_{\substack{v\in\mathcal{I}_{virt}\\ a,b,c,d\in\mathcal{I}_{act}}} h'_{va}g_{vbcd}e^{-i(E_{core}+\epsilon_v)t}\,\langle a_d^\dagger a_c^\dagger a_b e^{-iH_{CAS}t} a_a\rangle \\
T_4(t) &= \sum_{\substack{v\in\mathcal{I}_{virt}\\ a,b,c,d,e,f\in\mathcal{I}_{act}}} g_{vabc}g_{vdef}e^{-i(E_{core}+\epsilon_v)t}\,\langle a_f^\dagger a_e^\dagger a_d e^{-iH_{CAS}t} a_a^\dagger a_b a_c\rangle \\
&\quad + \tfrac{1}{2}\sum_{\substack{v,w\in\mathcal{I}_{virt}\\ a,b,c,d\in\mathcal{I}_{act}}} g_{vwab}g_{vwcd}e^{-i(E_{core}+\epsilon_v+\epsilon_w)t} \\
&\qquad \times\,\langle a_d^\dagger a_c^\dagger e^{-iH_{CAS}t} a_a a_b\rangle\,.
\end{aligned}
$$

Lastly, by defining the complex coefficients

$$\begin{aligned} v_{a,b}(t) &= \sum_{v\in\mathcal{I}_{virt}} h'_{va}h'_{vb}e^{-i(\epsilon_v+E_{core})t} \\ v_{abc,d}(t) &= \sum_{v\in\mathcal{I}_{virt}} h'_{vd}g_{vabc}e^{-i(\epsilon_v+E_{core})t} \\ v_{abc,def}(t) &= \sum_{v\in\mathcal{I}_{virt}} g_{vabc}g_{vdef}e^{-i(\epsilon_v+E_{core})t} \\ v_{ab,cd}(t) &= \tfrac{1}{2}\sum_{v,w\in\mathcal{I}_{virt}} g_{vwab}g_{vwcd}e^{-i(\epsilon_v+\epsilon_w+E_{core})t}\,, \end{aligned}$$

we arrive at the expressions in Equation Eq. (4.23).

MRPT2 norm bounds

First, we discuss the expressions for the 2/3-norms $\|h\|_1$ and $\|v\|_{2/3}$ corresponding to the Dyall Hamiltonian H (eq. (4.20)) and to the full difference $V = H'_{el} - H$.

A fermion-to-qubit mapping will map each creation or annihilation operator to the sum of two Pauli strings with prefactor 1/2. Therefore, the fermion-to-qubit mapping of the Dyall Hamiltonian is

$$\begin{aligned} H =& \sum_{a,b\in\mathcal{I}_{act}} h'_{ab}a_a^\dagger a_b + \frac{1}{2}\sum_{a,b,c,d\in\mathcal{I}_{act}} g_{abcd}a_a^\dagger a_b^\dagger a_c a_d \\ &+ \sum_{v\in\mathcal{I}_{virt}} \epsilon_v a_v^\dagger a_v + E_{core} \\ &\longrightarrow \\ &\sum_{a,b\in\mathcal{I}_{act}}\sum_{i=1}^{4} \frac{h'_{ab}}{4}\sigma_{abi} + \sum_{a,b,c,d\in\mathcal{I}_{act}}\sum_{i=1}^{16} \frac{g_{abcd}}{32}\sigma_{abcdi} \\ &+ \sum_{v\in\mathcal{I}_{virt}}\sum_{i=1}^{4} \frac{\epsilon_v}{4}\sigma_{vi} + E_{core}\,, \end{aligned}$$

where $\sigma_{abi}, \sigma_{abcdi}$ and σ_{vi} are Pauli strings. The Pauli strings are not necessarily all distinct in the expression above, however, we still obtain an upper bound on $\|h\|_1$ by calculating

$$\|h\|_1 \leq \sum_{a,b\in\mathcal{I}_{act}} |h'_{ab}| + \frac{1}{2}\sum_{a,b,c,d\in\mathcal{I}_{act}} |g_{abcd}| + \sum_{v\in\mathcal{I}_{virt}} |\epsilon_v| + |E_{core}|\,.$$

Since two strings of creation and annihilation operators with distinct sets of indices are mapped to distinct sets of Paulistrings, this upper bound is tight up

to taking permutations of indices into account and collecting identity terms. In a similar way, the perturbation V is mapped to

$$
\begin{aligned}
V = & \sum_{\substack{p,q \in \mathcal{I}_{act} \cup \mathcal{I}_{virt} \\ \{p,q\} \not\subset \mathcal{I}_{act}}} h'_{pq} a_p^\dagger a_q + \frac{1}{2} \sum_{\substack{p,q,r,s \in \mathcal{I}_{virt} \cup \mathcal{I}_{act} \\ \{p,q,r,s\} \not\subset \mathcal{I}_{act}}} g_{pqrs} a_p^\dagger a_q^\dagger a_r a_s - F_{virt} \\
& \longrightarrow \\
& \sum_{\substack{p,q \in \mathcal{I}_{act} \cup \mathcal{I}_{virt} \\ \{p,q\} \not\subset \mathcal{I}_{act}}} \sum_{i=1}^{4} \frac{h'_{pq}}{4} \sigma_{pqi} - \sum_{v \in \mathcal{I}_{virt}} \sum_{i=1}^{4} \frac{\epsilon_v}{4} \sigma_{vi} \\
& + \sum_{\substack{p,q,r,s \in \mathcal{I}_{virt} \cup \mathcal{I}_{act} \\ \{p,q,r,s\} \not\subset \mathcal{I}_{act}}} \sum_{i=1}^{16} \frac{g_{pqrs}}{32} \sigma_{pqrsi}
\end{aligned}
$$

and $\|v\|_{2/3}$ can be upper bound by

$$
\begin{aligned}
\|v\|_{2/3} \leq 2 \Bigg(& \sum_{\substack{p,q \in \mathcal{I}_{act} \cup \mathcal{I}_{virt} \\ \{p,q\} \not\subset \mathcal{I}_{act}}} \left| h'_{pq} \right|^{2/3} + \sum_{v \in \mathcal{I}_{virt}} |\epsilon_v|^{2/3} \\
& + \sum_{\substack{p,q,r,s \in \mathcal{I}_{virt} \cup \mathcal{I}_{act} \\ \{p,q,r,s\} \not\subset \mathcal{I}_{act}}} |g_{pqrs}|^{2/3} \Bigg)^{3/2} .
\end{aligned}
$$

Next, we discuss upper bounds on $\left\|h^{CAS}\right\|_1$ and $\left\|v^{\mathrm{MRPT(2)}}\right\|_{2/3}$. Analogous to the preceding derivation of the upper bound on $\|h\|_1$, we see that

$$
\left\|h^{CAS}\right\|_1 \leq \sum_{a,b \in \mathcal{I}_{act}} |h'_{ab}| + \frac{1}{2} \sum_{a,b,c,d \in \mathcal{I}_{act}} |g_{abcd}| .
$$

For $\left\|v^{\mathrm{MRPT(2)}}\right\|_{2/3}$, we need to consider how the creation and annihilation opera-

tors are mapped to Pauli operators in equation (4.23):

$$
\begin{aligned}
&\langle VU(t_n)V\rangle = T_1(t_n) + T_2(t_n) + T_3(t_n) + T_4(t_n) \\
&\longrightarrow \\
&\frac{1}{4} \sum_{a,b\in\mathcal{I}_{act}} \sum_{ij=1}^{2} v_{a,b}(t_n) \left\langle \sigma_{bj} e^{-iH_{CAS}t_n} \sigma_{ai} \right\rangle \\
+&\frac{1}{16} \sum_{a,b,c,d\in\mathcal{I}_{act}} \sum_{i=1}^{8}\sum_{j=1}^{2} v_{abc,d}(t_n) \left\langle \sigma_{dj} e^{-iH_{CAS}t_n} \sigma_{abci} \right\rangle \\
+&\frac{1}{16} \sum_{a,b,c,d\in\mathcal{I}_{act}} \sum_{i=1}^{2}\sum_{j=1}^{8} v_{bcd,a}(t_n) \left\langle \sigma_{bcdj} e^{-iH_{CAS}t_n} \sigma_{ai} \right\rangle \\
+&\frac{1}{32} \sum_{a,b,c,d\in\mathcal{I}_{act}} \sum_{ij=1}^{4} v_{ab,cd}(t_n) \left\langle \sigma_{cdj} e^{-iH_{CAS}t_n} \sigma_{abi} \right\rangle \\
+&\frac{1}{64} \sum_{a,b,c,d,e,f\in\mathcal{I}_{act}} \sum_{ij=1}^{8} v_{abc,def}(t_n) \left\langle \sigma_{defj} e^{-iH_{CAS}t_n} \sigma_{abci} \right\rangle \ ,
\end{aligned}
$$

which is of the desired form

$$
\sum_{i=1}^{L^{\mathrm{MRPT(2)}}} v_{ni}^{\mathrm{MRPT(2)}} \left\langle \sigma_{i,1} e^{-iH_{CAS}t_n} \sigma_{i,2} \right\rangle
$$

see also equation (4.25).

Consequently, we get

$$
\begin{aligned}
&\left(\sum_i \left|v_{ni}^{\mathrm{MRPT(2)}}\right|^{2/3}\right)^{3/2} \\
&\leq \Bigg(2^{2/3}\sum_{a,b\in\mathcal{I}_{act}} |v_{a,b}(t_n)|^{2/3} + 2^{7/3}\sum_{a,b,c,d\in\mathcal{I}_{act}} |v_{abc,d}(t_n)|^{2/3} \\
&+ 2^{2/3}\sum_{a,b,c,d\in\mathcal{I}_{act}} |v_{ab,cd}(t_n)|^{2/3} + 4\sum_{a,b,c,d,e,f\in\mathcal{I}_{act}} |v_{abc,def}(t_n)|^{2/3}\Bigg)^{3/2} \\
&\leq \Bigg(2^{2/3}\sum_{a,b\in\mathcal{I}_{act}}\left(\sum_{v\in\mathcal{I}_{virt}} |h'_{va}h'_{vb}|\right)^{2/3} \\
&+ 2^{7/3}\sum_{a,b,c,d\in\mathcal{I}_{act}}\left(\sum_{v\in\mathcal{I}_{virt}} |h'_{va}g_{vbcd}|\right)^{2/3} \\
&+ 2^{2/3}\sum_{a,b,c,d\in\mathcal{I}_{act}}\left(\sum_{vw\in\mathcal{I}_{virt}} |g_{vwab}g_{vwcd}|\right)^{2/3} \\
&+ 4\sum_{a,b,c,d,e,f\in\mathcal{I}_{act}}\left(\sum_{v\in\mathcal{I}_{virt}} |g_{vabc}g_{vdef}|\right)^{2/3}\Bigg)^{3/2} \\
&\equiv \left\|v^{\mathrm{MRPT(2)}}\right\|_{2/3}.
\end{aligned}
$$

End of Phys. Rev. Research 6, 043021 - More Quantum Chemistry with Fewer Qubits

4.8 Results beyond the publication

In this section we will discuss improvements to the method that were recognised and developed, together with Freek Witteveen and Matthias Christandl, after the manuscript was submitted. On a high level, the sample complexity of the method is driven by the normalization factors of the probability distributions that is sampled from. Concretely, from Eq. (4.58) the number of calls to U_0 is upper bounded by

$$\tilde{\mathcal{O}}\left(\frac{\|v\|_{2/3}^2 \|\beta_{\Delta,R}\|_{2/3}}{\varepsilon}\right),$$

where $\|v\|_{2/3}$ is determined by the decomposition of V into a weighted sum of Pauli strings, and $\|\beta_{\Delta,R}\|_{2/3}$ is determined by the approximation of $1/x$ by a Fourier series on the interval $[\Delta, R]$. The total time evolution cost behaves the same, up to an additional factor given by the maximal depth $t_{\tilde{N}} = \mathcal{O}(\tilde{N}/R)$. In the following we will propose ways to reduce the cost of the method by reducing the size of $\|\beta_{\Delta,R}\|_{2/3}$ and of $\|v\|_{2/3}$.

4.8.1 Improved representation of Fourier series for $1/x$

The construction of the Fourier series by convolving $1/x$ with a smoothened window-function on the interval $[\Delta, R]$ resulted in the bound

$$\|\beta_{\Delta,R}\|_{2/3} = \left(\sum_{|n|\leq\widetilde{N}} |\beta_{\Delta,R,n}|^{2/3}\right)^{3/2} \leq \frac{4\widetilde{N}}{\Delta} = \mathcal{O}\left(\frac{\|h\|_1^{3/2}}{\Delta^{5/2}}\right),$$

see Eqs. (4.42), (4.55) and (4.62). In terms of scaling with Δ and $\|h\|_1$, this is not optimal. A better representation of $1/x$ was introduced by Childs, Kothari and Somma in [262]:

$$\frac{1}{x} = \frac{i}{\sqrt{2\pi}} \int_0^\infty dy \int_{-\infty}^\infty dz z e^{-z^2/2} e^{-ixyz}$$

In the same reference a bound on the truncation error due to finite integration range is shown:

Lemma 4.8.1 (Lemma 12, [262])**.** *The function*

$$\frac{i}{\sqrt{2\pi}} \int_0^{y_J} dy \int_{-z_K}^{z_K} dz z e^{-z^2/2} e^{-ixyz}$$

is ϵ-close to $1/x$ on the domain $(-1,-1/\kappa]\cup[1/\kappa,1)$ for some $y_J = \Theta\left(\kappa\sqrt{\log(\kappa/\epsilon)}\right)$ and $z_K = \Theta\left(\sqrt{\log\kappa/\epsilon}\right)$.

As noted in [263], this result straightforwardly extends to the domain $(-\infty,-1/\kappa]\cup[1/\kappa,\infty)$. We will use $\Delta = 1/\kappa$ and perform a substitution of variables $zy \to t$:

$$\begin{aligned}
&\frac{i}{\sqrt{2\pi}}\int_0^{y_J} dy \int_{-z_K}^{z_K} dzze^{-z^2/2}e^{-ixyz} \\
&= \sqrt{\frac{2}{\pi}}\int_0^{z_K} dzze^{-z^2/2}\int_0^{y_J} dy\sin(xyz) \\
&= \sqrt{\frac{2}{\pi}}\int_0^{z_K} dze^{-z^2/2}\int_0^{y_Jz} dt\sin(xt) \\
&= \sqrt{\frac{2}{\pi}}\int_0^{y_Jz_K} dt\sin(xt)\int_{t/y_J}^{z_K} dze^{-z^2/2} \\
&\equiv \int_0^T dt\hat{g}(t)\sin(xt)
\end{aligned}$$

where in the last line $T = y_Jz_K = \Theta\left(\Delta^{-1}\log(\epsilon^{-1}\Delta^{-1})\right)$ and

$$\hat{g}(t) := \sqrt{\frac{2}{\pi}}\int_{t/y_J}^{z_K} dze^{-z^2/2} = \operatorname{erf}\left(\frac{z_K}{\sqrt{2}}\right) - \operatorname{erf}\left(\frac{t}{\sqrt{2}y_J}\right).$$

was defined. Here, $\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}}\int_0^x dte^{-t^2}$ is the error function. Note that $\hat{g}(t) \geq 0$ for $t \in [0,T]$. The sampling complexity corresponding to this representation of $1/x$ is determined by

$$\begin{aligned}
\left(\int_0^T dt|\hat{g}(t)|^{2/3}\right)^{3/2} &\leq \left(\int_0^\infty dt\left(1-\operatorname{erf}\left(\frac{t}{\sqrt{2}y_J}\right)\right)^{2/3}\right)^{3/2} \\
&= y_J^{3/2}\left(\sqrt{2}\int_0^\infty dt'\left(1-\operatorname{erf}(t')\right)^{2/3}\right)^{2/3} \\
&= \Theta\left(y_J^{3/2}\right)
\end{aligned}$$

Lastly, the integral is discretized into n steps $t_1, t_2, \ldots, t_n$ with $t_\ell = \ell\delta$ and step size $\delta = T/n$,

$$\int_0^T dt\hat{g}(t)\sin(xt) \to \sum_{\ell=1}^n \delta\hat{g}(t_\ell)\sin(t_\ell).$$

To assess the discretization error, we first note that the derivative of $\hat{g}(t)$ is bounded by

$$\left|\frac{d\hat{g}}{dt}\right| = \frac{2}{\sqrt{\pi}} e^{-t^2/(2y_J)} \leq \frac{2}{\sqrt{\pi}},$$

such that $|\hat{g}(t+\delta) - \hat{g}(t)| \leq 2\delta/\sqrt{\pi}$. The discretization error is therefore bounded by

$$\begin{aligned}
\left|\int_0^T dt \hat{g}(t)\sin(xt) - \sum_{\ell=1}^{n} \delta \hat{g}(t_\ell)\sin(xt_\ell)\right| &\leq \sum_{\ell=1}^{n} \int_{t_{\ell-1}}^{t_\ell} |\hat{g}(t)\sin(xt) - \hat{g}(t_\ell)\sin(xt_\ell)| dt \\
&\leq 2\delta \sum_{\ell=1}^{n} \sup_{t\in[t_{\ell-1},t_\ell]} |\hat{g}(t) - \hat{g}(t_\ell)| \\
&\leq \frac{4n\delta^2}{\sqrt{\pi}} \\
&= \frac{4T^2}{n\sqrt{\pi}}.
\end{aligned}$$

Thus, by choosing n appropriately, the discretization error can be controlled. Since our algorithms do not depend on the n, the discretization error can be suppressed to be arbitrarily small and will be neglected.

All in all, this proves the following lemma:

Lemma 4.8.2 (Fourier representation of $1/x$)**.** *The function*

$$g(x) = \sum_{\ell=1}^{n} \delta \hat{g}(t_\ell)\sin(xt_\ell) \quad \textit{with} \quad \hat{g}(t) = \operatorname{erf}\left(\frac{z_K}{\sqrt{2}}\right) - \operatorname{erf}\left(\frac{t}{\sqrt{2}y_J}\right)$$

is ϵ-close to $1/x$ on the domain $(-\infty,-\Delta]\cup[\Delta,\infty)$ for some $y_J = \Theta\left(\Delta^{-1}\sqrt{\log(\Delta^{-1}\epsilon^{-1})}\right)$, $z_K = \Theta\left(\sqrt{\log\Delta^{-1}\epsilon^{-1}}\right)$, $n = \mathcal{O}(y_J^2 z_K^2/\epsilon)$ and $\delta = y_J z_K/n$. Furthermore, $\left(\sum_{\ell=1}^{n} |\delta\hat{g}(t_\ell)|^{2/3}\right)^{3/2} = \mathcal{O}\left(y_J^{3/2}\right)$.

Compared to the method in the paper, this representation of $1/x$ reduces the associated sampling cost from $\|\beta_{\Delta,R}\|_{2/3} = \mathcal{O}(\|h\|_1^{3/2}\Delta^{-5/2})$ to

$$\left(\sum_{\ell=1}^{n} |\delta\hat{g}(t_\ell)|^{2/3}\right)^{3/2} = \tilde{\mathcal{O}}\left(\Delta^{-3/2}\right).$$

Moreover, instead of performing importance sampling for the different terms t_ℓ and running an individual circuit for each sample, one could employ a LCU-based approach, and implement all terms coherently. In this case the ℓ_1-norm of

the coefficients enters the sampling complexity, which would then be reduced to $\sum_{\ell=1}^{n} |\delta\hat{g}(t_\ell)| = \Theta(\Delta^{-1})$.

Conclusion and open problems

One hundred years after the foundations of modern quantum mechanics were laid, we understand that quantum mechanics is not only the source of difficult problems, but perhaps also the key ingredient in their solution. With this thesis we hope to contribute to the efforts of a quantum mechanics-based understanding of chemistry. We investigated the quantum many-body problem underlying the theoretical study of molecules from a quantum computing perspective.

In chapter 2 we presented a new approach to phase estimation that combines deterministic Trotter product formulas with randomized time evolution. The method fares particularly well if an approximation to the full Hamiltonian is available for which the time evolution is much cheaper to implement. In our results we used a simple truncation approach to obtain such approximations. However, the design space for approximations is huge, and it remains an open problem to find more effective ones.

In chapter 3 we performed classical simulations of product formulas, and found that rigorous bounds significantly overestimated algorithmic errors in the partially randomized method. It would be interesting to numerically investigate how sensitive these algorithmic errors are to the choice of decomposition strategies into deterministic and randomized part.

In chapter 4 we proposed a quantum algorithm that perturbatively improves on standard active space-based methods. Key advantages of the method are the very low qubit overhead and the low scaling with the number of virtual orbitals (for a fixed active space). However, to arrive at a practically useful method, more work is needed to reduce the number of circuit samples.

As a final remark, we note that essentially all quantum computing approaches to chemistry - including ours - mirror the traditional strategy of separating the dynamics of electrons and nuclei. Given that the molecular Hamiltonian provides a compact description of the full molecule, one might wonder whether a unified treatment of electrons and nuclei is not a more natural path towards accurate and efficient quantum computations of molecular properties.

Bibliography

[1] Jakob Günther, Alberto Baiardi, Markus Reiher, and Matthias Christandl. "More Quantum Chemistry with Fewer Qubits". *Physical Review Research* 6.4 (2024), p. 043021.

[2] Mihael Erakovic, Freek Witteveen, Dylan Harley, Jakob Günther, Moritz Bensberg, Oinam Romesh Meitei, Minsik Cho, Troy Van Voorhis, Markus Reiher, and Matthias Christandl. "High Ground State Overlap via Quantum Embedding Methods". *PRX Life* 3.1 (2025), p. 013003.

[3] Jakob Günther, Freek Witteveen, Alexander Schmidhuber, Marek Miller, Matthias Christandl, and Aram Harrow. "Phase estimation with partially randomized time evolution". *arXiv:2503.05647* (2025).

[4] Marek Miller, Jakob Günther, Freek Witteveen, Matthew S. Teynor, Mihael Erakovic, Markus Reiher, Gemma C. Solomon, and Matthias Christandl. "Phase2: Full-State Vector Simulation of Quantum Time Evolution at Scale". *arXiv:2504.17881* (2025).

[5] Jakob Günther et al. "How to Use Quantum Computers for Biomolecular Free Energies". *arXiv:2506.20587* (2025).

[6] Paul Adrien Maurice Dirac and Ralph Howard Fowler. "Quantum Mechanics of Many-Electron Systems". *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character* 123.792 (1929), pp. 714–733.

[7] Trygve Helgaker, Poul Jørgensen, and Jeppe Olsen. *Molecular Electronic-Structure Theory*. 1st ed. John Wiley & Sons, Ltd, 2014.

[8] D. J. Thouless. "Stability Conditions and Nuclear Rotations in the Hartree-Fock Theory". *Nuclear Physics* 21 (1960), pp. 225–232.

[9] Emiel Koridon, Saad Yalouz, Bruno Senjean, Francesco Buda, Thomas E. O'Brien, and Lucas Visscher. "Orbital Transformations to Reduce the 1-Norm of the Electronic Structure Hamiltonian for Quantum Computing Applications". *Physical Review Research* 3.3 (2021), p. 033127.

[10] Ignacio Loaiza, Alireza Marefat Khah, Nathan Wiebe, and Artur F Izmaylov. "Reducing molecular electronic hamiltonian simulation cost for linear combination of unitaries approaches". *Quantum Science and Technology* 8.3 (2023), p. 035019.

[11] Ignacio Loaiza and Artur F Izmaylov. "Block-Invariant Symmetry Shift: Preprocessing technique for second-quantized Hamiltonians to improve their decompositions to Linear Combination of Unitaries". *Journal of Chemical Theory and Computation* 19.22 (2023), pp. 8201–8209.

[12] Athena Caesura, Cristian L. Cortes, William Pol, Sukin Sim, Mark Steudtner, Gian-Luca R. Anselmetti, Matthias Degroote, Nikolaj Moll, Raffaele Santagati, Michael Streif, and Christofer S. Tautermann. "Faster Quantum Chemistry Simulations on a Quantum Computer with Improved Tensor Factorization and Active Volume Compilation". *arXiv:2501.06165* (2025).

[13] David P. DiVincenzo. "Topics in Quantum Computers". *arXiv:cond-mat/9612126* (1996).

[14] Daniel Gottesman. "The Heisenberg Representation of Quantum Computers". arXiv:quant-ph/9807006 (1998).

[15] A. Yu Kitaev. "Quantum Computations: Algorithms and Error Correction". *Russian Mathematical Surveys* 52.6 (1997), pp. 1191–1249.

[16] Dolev Bluvstein et al. "Logical Quantum Processor Based on Reconfigurable Atom Arrays". *Nature* 626.7997 (2024), pp. 58–65.

[17] Rajeev Acharya et al. "Quantum Error Correction below the Surface Code Threshold". *Nature* 638.8052 (2025), pp. 920–926.

[18] Reza Haghshenas et al. "Digital Quantum Magnetism at the Frontier of Classical Simulations". *arXiv:2503.20870* (2025).

[19] Dorit Aharonov and Michael Ben-Or. "Fault-Tolerant Quantum Computation With Constant Error Rate". *arXiv:quant-ph/9906129* (1999).

[20] Daniel Litinski. "A Game of Surface Codes: Large-Scale Quantum Computing with Lattice Surgery". *Quantum* 3 (2019), p. 128.

[21] Kosuke Mitarai, Kiichiro Toyoizumi, and Wataru Mizukami. "Perturbation theory with quantum signal processing". *Quantum* 7 (2023), p. 1000.

[22] Zhang Jiang, Amir Kalev, Wojciech Mruczkiewicz, and Hartmut Neven. "Optimal Fermion-to-Qubit Mapping via Ternary Trees with Applications to Reduced Quantum States Learning". *Quantum* 4 (2020), p. 276.

[23] Alexander Yurievich Vlasov. "Clifford Algebras, Spin Groups and Qubit Trees". *Quanta* 11 (2022), pp. 97–114.

[24] Aaron Miller, Zoltán Zimborás, Stefan Knecht, Sabrina Maniscalco, and Guillermo García-Pérez. "Bonsai Algorithm: Grow Your Own Fermion-to-Qubit Mappings". *PRX Quantum* 4.3 (2023), p. 030314.

[25] Sergey Bravyi and Alexei Kitaev. "Fermionic Quantum Computation". *Annals of Physics* 298.1 (2002), pp. 210–226.

[26] Federico Belliardo and Vittorio Giovannetti. "Achieving Heisenberg scaling with maximally entangled states: An analytic upper bound for the attainable root-mean-square error". *Physical Review A* 102.4 (2020), p. 042613.

[27] Shouzhen Gu, Rolando D. Somma, and Burak Şahinoğlu. "Fast-Forwarding Quantum Evolution". *Quantum* 5 (2021), p. 577.

[28] Yuan Su. "Fast-Forwardable Quantum Evolution and Where to Find Them". *Quantum Views* 5 (2021), p. 62.

[29] Dominic W. Berry, Graeme Ahokas, Richard Cleve, and Barry C. Sanders. "Efficient Quantum Algorithms for Simulating Sparse Hamiltonians". *Communications in Mathematical Physics* 270.2 (2007), pp. 359–371.

[30] Jeongwan Haah, Matthew Hastings, Robin Kothari, and Guang Hao Low. "Quantum Algorithm for Simulating Real Time Evolution of Lattice Hamiltonians". *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*. 2018, pp. 350–360.

[31] Daniel S. Abrams and Seth Lloyd. "Quantum Algorithm Providing Exponential Speed Increase for Finding Eigenvalues and Eigenvectors". *Physical Review Letters* 83.24 (1999), pp. 5162–5165.

[32] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Publisher: Cambridge University Press, 2010.

[33] Lin Lin and Yu Tong. "Heisenberg-limited ground-state energy estimation for early fault-tolerant quantum computers". *PRX Quantum* 3.1 (2022), p. 010318.

[34] Zhiyan Ding, Yulong Dong, Yu Tong, and Lin Lin. "Robust ground-state energy estimation under depolarizing noise". *arXiv:2307.11257* (2023).

[35] Yulong Dong, Lin Lin, and Yu Tong. "Ground-State Preparation and Energy Estimation on Early Fault-Tolerant Quantum Computers via Quantum Eigenvalue Transformation of Unitary Matrices". *PRX Quantum* 3.4 (2022), p. 040305.

[36] Guoming Wang, Daniel Stilck França, Ruizhe Zhang, Shuchen Zhu, and Peter D Johnson. "Quantum algorithm for ground state energy estimation using circuit depth with exponentially improved dependence on precision". *Quantum* 7 (2023), p. 1167.

[37] Guoming Wang, Daniel Stilck França, Gumaro Rendon, and Peter D. Johnson. "Faster Ground State Energy Estimation on Early Fault-Tolerant Quantum Computers via Rejection Sampling". *arXiv:2304.09827* (2023).

[38] Hongkang Ni, Haoya Li, and Lexing Ying. "On low-depth algorithms for quantum phase estimation". *Quantum* 7 (2023), p. 1165.

[39] Zhiyan Ding and Lin Lin. "Even shorter quantum circuit for phase estimation on early fault-tolerant quantum computers with applications to ground-state energy estimation". *PRX Quantum* 4.2 (2023), p. 020331.

[40] Markus Reiher, Nathan Wiebe, Krysta M Svore, Dave Wecker, and Matthias Troyer. "Elucidating reaction mechanisms on quantum computers". *Proceedings of the national academy of sciences* 114.29 (2017), pp. 7555–7560.

[41] Alberto Baiardi and Markus Reiher. "The density matrix renormalization group in chemistry and molecular physics: Recent developments and new challenges". *J. Chem. Phys.* 152.4 (2020), p. 040903.

[42] Zhendong Li, Junhao Li, Nikesh S Dattani, CJ Umrigar, and Garnet Kin Chan. "The electronic complexity of the ground-state of the FeMo cofactor of nitrogenase as relevant to quantum simulations". *The Journal of chemical physics* 150.2 (2019).

[43] Andor Menczer, Maarten van Damme, Alan Rask, Lee Huntington, Jeff Hammond, Sotiris S. Xantheas, Martin Ganahl, and Örs Legeza. "Parallel Implementation of the Density Matrix Renormalization Group Method Achieving a Quarter petaFLOPS Performance on a Single DGX-H100 GPU Node". *Journal of Chemical Theory and Computation* 20.19 (2024), pp. 8397–8404.

[44] Maximilian Mörchen, Guang Hao Low, Thomas Weymuth, Hongbin Liu, Matthias Troyer, and Markus Reiher. "Classification of electronic structures and state preparation for quantum computation of reaction chemistry". *arXiv:2409.08910* (2024).

[45] Vera von Burg, Guang Hao Low, Thomas Häner, Damian S. Steiger, Markus Reiher, Martin Roetteler, and Matthias Troyer. "Quantum computing enhanced computational catalysis". *Phys. Rev. Res.* 3 (2021), p. 033055.

[46] Joonho Lee, Dominic W Berry, Craig Gidney, William J Huggins, Jarrod R McClean, Nathan Wiebe, and Ryan Babbush. "Even more efficient quantum computations of chemistry through tensor hypercontraction". *PRX Quantum* 2.3 (2021), p. 030305.

[47] Stepan Fomichev, Kasra Hejazi, Modjtaba Shokrian Zini, Matthew Kiser, Joana Fraxanet Morales, Pablo Antonio Moreno Casares, Alain Delgado, Joonsuk Huh, Arne-Christian Voigt, Jonathan E Mueller, et al. "Initial state preparation for quantum chemistry on quantum computers". *arXiv:2310.18410* (2023).

[48] Konrad Heinrich Marti and Markus Reiher. "The density matrix renormalization group algorithm in quantum chemistry". *Z. Phys. Chem.* 224.3-4 (2010), pp. 583–599.

[49] Garnet Kin-Lic Chan and Martin Head-Gordon. "Highly Correlated Calculations with a Polynomial Cost Algorithm: A Study of the Density Matrix Renormalization Group". *The Journal of Chemical Physics* 116.11 (2002), pp. 4462–4476.

[50] Masuo Suzuki. "Fractal Decomposition of Exponential Operators with Applications to Many-Body Theories and Monte Carlo Simulations". *Physics Letters A* 146.6 (1990), pp. 319–323.

[51] Andrew M. Childs, Yuan Su, Minh C. Tran, Nathan Wiebe, and Shuchen Zhu. "Theory of Trotter Error with Commutator Scaling". *Physical Review X* 11.1 (2021), p. 011020.

[52] Ryan Babbush, Jarrod McClean, Dave Wecker, Alán Aspuru-Guzik, and Nathan Wiebe. "Chemical Basis of Trotter-Suzuki Errors in Quantum Chemistry Simulation". *Physical Review A* 91.2 (2015), p. 022311.

[53] Luis A. Martínez-Martínez, Prathami Divakar Kamath, and Artur F. Izmaylov. "Estimating Trotter Approximation Errors to Optimize Hamiltonian Partitioning for Lower Eigenvalue Errors". *arXiv:2312.13282* (2024).

[54] David Poulin, Matthew B. Hastings, Dave Wecker, Nathan Wiebe, Andrew C. Doberty, and Matthias Troyer. "The Trotter Step Size Required for Accurate Quantum Simulation of Quantum Chemistry". *Quantum Info. Comput.* 15.5-6 (2015), pp. 361–384.

[55] Earl Campbell. "Random compiler for fast Hamiltonian simulation". *Physical review letters* 123.7 (2019), p. 070503.

[56] Dominic W. Berry, Mária Kieferová, Artur Scherer, Yuval R. Sanders, Guang Hao Low, Nathan Wiebe, Craig Gidney, and Ryan Babbush. "Improved Techniques for Preparing Eigenstates of Fermionic Hamiltonians". *npj Quantum Information* 4.1 (2018), pp. 1–7.

[57] Ryan Babbush, Craig Gidney, Dominic W Berry, Nathan Wiebe, Jarrod McClean, Alexandru Paler, Austin Fowler, and Hartmut Neven. "Encoding electronic spectra in quantum circuits with linear T complexity". *Physical Review X* 8.4 (2018), p. 041015.

[58] Guang Hao Low and Isaac L Chuang. "Hamiltonian simulation by qubitization". *Quantum* 3 (2019), p. 163.

[59] Guang Hao Low and Isaac L Chuang. "Optimal Hamiltonian simulation by quantum signal processing". *Physical review letters* 118.1 (2017), p. 010501.

[60] Dominic W Berry, Craig Gidney, Mario Motta, Jarrod R McClean, and Ryan Babbush. "Qubitization of arbitrary basis quantum chemistry leveraging sparsity and low rank factorization". *Quantum* 3 (2019), p. 208.

[61] Ethan Bernstein and Umesh Vazirani. "Quantum Complexity Theory". *Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing*. STOC '93. New York, NY, USA: Association for Computing Machinery, 1993, pp. 11–20.

[62] J. Watrous. "Succinct Quantum Proofs for Properties of Finite Groups". *Proceedings 41st Annual Symposium on Foundations of Computer Science*. 2000, pp. 537–546.

[63] Peter W. Shor. "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer". *SIAM Journal on Computing* 26.5 (1997), pp. 1484–1509.

[64] Adam D. Bookatz. "QMA-complete Problems". *arXiv:1212.6312* (2013).

[65] A. Yu. Kitaev, A. Shen, and M. N. Vyalyi. *Classical and Quantum Computation*. USA: American Mathematical Society, 2002.

[66] Dorit Aharonov, Itai Arad, and Thomas Vidick. "The Quantum PCP Conjecture". *arXiv:1309.7495* (2013).

[67] Julia Kempe, Alexei Kitaev, and Oded Regev. "The Complexity of the Local Hamiltonian Problem". *SIAM Journal on Computing* (2006).

[68] Roberto Oliveira and Barbara M. Terhal. "The Complexity of Quantum Spin Systems on a Two-Dimensional Square Lattice". *Quantum Info. Comput.* 8.10 (2008), pp. 900–924.

[69] Andrew M. Childs, David Gosset, and Zak Webb. "The Bose-Hubbard Model Is QMA-complete". *Automata, Languages, and Programming*. Ed. by Javier Esparza, Pierre Fraigniaud, Thore Husfeldt, and Elias Koutsoupias. Berlin, Heidelberg: Springer, 2014, pp. 308–319.

[70] Norbert Schuch and Frank Verstraete. "Computational Complexity of Interacting Electrons and Fundamental Limitations of Density Functional Theory". *Nature Physics* 5.10 (2009), pp. 732–735.

[71] Bryan O'Gorman, Sandy Irani, James Whitfield, and Bill Fefferman. "Intractability of Electronic Structure in a Fixed Basis". *PRX Quantum* 3.2 (2022), p. 020322.

[72] Jacob D. Biamonte and Peter J. Love. "Realizable Hamiltonians for Universal Adiabatic Quantum Computers". *Physical Review A* 78.1 (2008), p. 012352.

[73] Yichen Huang. "A Polynomial-Time Algorithm for the Ground State of One-Dimensional Gapped Hamiltonians". *arXiv:1406.6355* (2015).

[74] Zeph Landau, Umesh Vazirani, and Thomas Vidick. "A Polynomial Time Algorithm for the Ground State of One-Dimensional Gapped Local Hamiltonians". *Nature Physics* 11.7 (2015), pp. 566–569.

[75] Abhinav Deshpande, Alexey V. Gorshkov, and Bill Fefferman. "Importance of the Spectral Gap in Estimating Ground-State Energies". *PRX Quantum* 3.4 (2022), p. 040327.

[76] Carlos E. González-Guillén and Toby S. Cubitt. "History-State Hamiltonians Are Critical". *arXiv:1810.06528* (2018).

[77] Roman Abrosimov and Bernd Moosmann. "The HOMO-LUMO Gap as Discriminator of Biotic from Abiotic Chemistries". *Life (Basel, Switzerland)* 14.10 (2024), p. 1330.

[78] Sevag Gharibian and François Le Gall. "Dequantizing the Quantum Singular Value Transformation: Hardness and Applications to Quantum Chemistry and the Quantum PCP Conjecture". *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*. STOC 2022. New York, NY, USA: Association for Computing Machinery, 2022, pp. 19–32.

[79] Chris Cade, Marten Folkertsma, Sevag Gharibian, Ryu Hayakawa, François Le Gall, Tomoyuki Morimae, and Jordi Weggemans. "Improved Hardness Results for the Guided Local Hamiltonian Problem". *50th International Colloquium on Automata, Languages, and Programming (ICALP 2023)*. Ed. by Kousha Etessami, Uriel Feige, and Gabriele Puppis. Vol. 261. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023, 32:1–32:19.

[80] Jiaqing Jiang. "Local Hamiltonian Problem with Succinct Ground State Is MA-Complete". *PRX Quantum* 6.2 (2025), p. 020312.

[81] Andrew M Childs, Dmitri Maslov, Yunseong Nam, Neil J Ross, and Yuan Su. "Toward the first quantum simulation with quantum speedup". *Proceedings of the National Academy of Sciences* 115.38 (2018), pp. 9456–9461.

[82] Yuan Su, Dominic W Berry, Nathan Wiebe, Nicholas Rubin, and Ryan Babbush. "Fault-tolerant quantum simulations of chemistry in first quantization". *PRX Quantum* 2.4 (2021), p. 040332.

[83] Bela Bauer, Sergey Bravyi, Mario Motta, and Garnet Kin-Lic Chan. "Quantum algorithms for quantum chemistry and quantum materials science". *Chemical Reviews* 120.22 (2020), pp. 12685–12717.

[84] Guang Hao Low, Yuan Su, Yu Tong, and Minh C Tran. "Complexity of implementing Trotter steps". *PRX Quantum* 4.2 (2023), p. 020323.

[85] Yuan Su, Hsin-Yuan Huang, and Earl T Campbell. "Nearly tight Trotterization of interacting electrons". *Quantum* 5 (2021), p. 495.

[86] Kianna Wan, Mario Berta, and Earl T Campbell. "Randomized quantum algorithm for statistical phase estimation". *Physical Review Letters* 129.3 (2022), p. 030503.

[87] Yingkai Ouyang, David R White, and Earl T Campbell. "Compilation by stochastic Hamiltonian sparsification". *Quantum* 4 (2020), p. 235.

[88] Shi Jin and Xiantao Li. "A partially random Trotter algorithm for quantum Hamiltonian simulations". *Communications on Applied Mathematics and Computation* (2023), pp. 1–28.

[89] Matthew Hagan and Nathan Wiebe. "Composite quantum simulations". *Quantum* 7 (2023), p. 1181.

[90] Dario Rocca, Cristian L Cortes, Jerome Gonthier, Pauline J Ollitrault, Robert M Parrish, Gian-Luca Anselmetti, Matthias Degroote, Nikolaj Moll, Raffaele Santagati, and Michael Streif. "Reducing the runtime of fault-tolerant quantum simulations in chemistry through symmetry-compressed double factorization". *arXiv:2403.03502* (2024).

[91] Guang Hao Low, Robbie King, Dominic W Berry, Qiushi Han, A Eugene DePrince III, Alec White, Ryan Babbush, Rolando D Somma, and Nicholas C Rubin. "Fast quantum simulation of electronic structure by spectrum amplification". *arXiv:2502.15882* (2025).

[92] Craig Gidney. "Halving the cost of quantum addition". *Quantum* 2 (2018), p. 74.

[93] Yuval R Sanders, Dominic W Berry, Pedro CS Costa, Louis W Tessler, Nathan Wiebe, Craig Gidney, Hartmut Neven, and Ryan Babbush. "Compilation of fault-tolerant quantum heuristics for combinatorial optimization". *PRX quantum* 1.2 (2020), p. 020312.

[94] Dominic W Berry, Yu Tong, Tanuj Khattar, Alec White, Tae In Kim, Sergio Boixo, Lin Lin, Seunghoon Lee, Garnet Kin Chan, Ryan Babbush, et al. "Rapid initial state preparation for the quantum simulation of strongly correlated molecules". *arXiv:2409.11748* (2024).

[95] Mario Motta, Erika Ye, Jarrod R McClean, Zhendong Li, Austin J Minnich, Ryan Babbush, and Garnet Kin-Lic Chan. "Low rank representations for quantum simulation of electronic structure". *npj Quantum Information* 7.1 (2021), p. 83.

[96] Ian D Kivlichan, Christopher E Granade, and Nathan Wiebe. "Phase estimation with randomized Hamiltonians". *arXiv:1907.10070* (2019).

[97] Abhishek Rajput, Alessandro Roggero, and Nathan Wiebe. "Hybridized methods for quantum simulation in the interaction picture". *Quantum* 6 (2022), p. 780.

[98] Oriel Kiss, Michele Grossi, and Alessandro Roggero. "Importance sampling for stochastic quantum simulations". *Quantum* 7 (2023), p. 977.

[99] Norm M Tubman, Carlos Mejuto-Zaera, Jeffrey M Epstein, Diptarka Hait, Daniel S Levine, William Huggins, Zhang Jiang, Jarrod R McClean, Ryan Babbush, Martin Head-Gordon, et al. "Postponing the orthogonality catastrophe: efficient state preparation for electronic structure simulations on quantum devices". *arXiv:1809.05523* (2018).

[100] Seunghoon Lee, Joonho Lee, Huanchen Zhai, Yu Tong, Alexander M Dalzell, Ashutosh Kumar, Phillip Helms, Johnnie Gray, Zhi-Hao Cui, Wenyuan Liu, et al. "Evaluating the evidence for exponential quantum advantage in ground-state quantum chemistry". *Nature communications* 14.1 (2023), p. 1952.

[101] A Yu Kitaev. "Quantum measurements and the Abelian stabilizer problem". *arXiv:quant-ph/9511026* (1995).

[102] Miroslav Dobšíček, Göran Johansson, Vitaly Shumeiko, and Göran Wendin. "Arbitrary accuracy iterative quantum phase estimation algorithm using a single ancillary qubit: A two-qubit benchmark". *Physical Review A* 76.3 (2007), p. 030306.

[103] Shelby Kimmel, Guang Hao Low, and Theodore J Yoder. "Robust calibration of a universal single-qubit gate set via robust phase estimation". *Physical Review A* 92.6 (2015), p. 062315.

[104] Nathan Wiebe and Chris Granade. "Efficient Bayesian phase estimation". *Physical review letters* 117.1 (2016), p. 010503.

[105] Rolando D Somma. "Quantum eigenvalue estimation via time series analysis". *New Journal of Physics* 21.12 (2019), p. 123025.

[106] Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. "Quantum Metrology". *Physical Review Letters* 96.1 (2006), p. 010401.

[107] BL Higgins, DW Berry, SD Bartlett, MW Mitchell, HM Wiseman, and GJ Pryde. "Demonstrating Heisenberg-limited unambiguous phase estimation without adaptive measurements". *New Journal of Physics* 11.7 (2009), p. 073023.

[108] Thomas E O'Brien, Brian Tarasinski, and Barbara M Terhal. "Quantum phase estimation of multiple eigenvalues for small-scale (noisy) experiments". *New Journal of Physics* 21.2 (2019), p. 023022.

[109] Dave Wecker, Bela Bauer, Bryan K Clark, Matthew B Hastings, and Matthias Troyer. "Gate-count estimates for performing quantum chemistry on small quantum computers". *Physical Review A* 90.2 (2014), p. 022305.

[110] Ian D Kivlichan, Craig Gidney, Dominic W Berry, Nathan Wiebe, Jarrod McClean, Wei Sun, Zhang Jiang, Nicholas Rubin, Austin Fowler, Alán Aspuru-Guzik, et al. "Improved fault-tolerant quantum simulation of condensed-phase correlated electrons via Trotterization". *Quantum* 4 (2020), p. 296.

[111] Burak Şahinoğlu and Rolando D Somma. "Hamiltonian simulation in the low-energy subspace". *npj Quantum Information* 7.1 (2021), p. 119.

[112] Chi-Fang Chen and Fernando GSL Brandão. "Average-case speedup for product formulas". *Communications in Mathematical Physics* 405.2 (2024), p. 32.

[113] Qi Zhao, You Zhou, Alexander F Shaw, Tongyang Li, and Andrew M Childs. "Hamiltonian simulation with random inputs". *Physical Review Letters* 129.27 (2022), p. 270502.

[114] Chi-Fang Chen, Hsin-Yuan Huang, Richard Kueng, and Joel A Tropp. "Concentration for random product formulas". *PRX Quantum* 2.4 (2021), p. 040305.

[115] Etienne Granet and Henrik Dreyer. "Hamiltonian dynamics on digital quantum computers without discretization error". *npj Quantum Information* 10.1 (2024), p. 82.

[116] Chusei Kiumi and Bálint Koczor. "TE-PAI: Exact Time Evolution by Sampling Random Circuits". *arXiv:2410.16850* (2024).

[117] Sam McArdle, Earl Campbell, and Yuan Su. "Exploiting fermion number in factorized decompositions of the electronic structure Hamiltonian". *Physical Review A* 105.1 (2022), p. 012403.

[118] Daniel Burgarth, Paolo Facchi, Alexander Hahn, Mattias Johnsson, and Kazuya Yuasa. "Strong Error Bounds for Trotter & Strang-Splittings and Their Implications for Quantum Chemistry". *arXiv:2312.08044* (2023).

[119] Craig Gidney, Noah Shutty, and Cody Jones. "Magic state cultivation: growing T states as cheap as CNOT gates". *arXiv:2409.17595* (2024).

[120] Earl Campbell. "Shorter gate sequences for quantum computing by mixing unitaries". *Physical Review A* 95.4 (2017), p. 042306.

[121] Vadym Kliuchnikov, Kristin Lauter, Romy Minko, Adam Paetznick, and Christophe Petit. "Shorter quantum circuits via single-qubit gate approximation". *Quantum* 7 (2023), p. 1208.

[122] Bálint Koczor, John JL Morton, and Simon C Benjamin. "Probabilistic interpolation of quantum rotation angles". *Physical Review Letters* 132.13 (2024), p. 130602.

[123] Ian D. Kivlichan, Jarrod McClean, Nathan Wiebe, Craig Gidney, Alán Aspuru-Guzik, Garnet Kin-Lic Chan, and Ryan Babbush. "Quantum Simulation of Electronic Structure with Linear Depth and Connectivity". *Physical Review Letters* 120.11 (2018), p. 110501.

[124] Mario Motta, James Shee, Shiwei Zhang, and Garnet Kin-Lic Chan. "Efficient ab initio auxiliary-field quantum Monte Carlo calculations in Gaussian bases via low-rank tensor decomposition". *Journal of chemical theory and computation* 15.6 (2019), pp. 3510–3521.

[125] Bo Peng and Karol Kowalski. "Highly efficient and scalable compound decomposition of two-electron integral tensor and its application in coupled cluster calculations". *Journal of chemical theory and computation* 13.9 (2017), pp. 4179–4192.

[126] Daniel Litinski. "Magic state distillation: Not as costly as you think". *Quantum* 3 (2019), p. 205.

[127] Yutaro Akahoshi, Kazunori Maruyama, Hirotaka Oshima, Shintaro Sato, and Keisuke Fujii. "Partially fault-tolerant quantum computing architecture with error-corrected Clifford gates and space-time efficient analog rotations". *PRX Quantum* 5.1 (2024), p. 010337.

[128] Guang Hao Low, Vadym Kliuchnikov, and Nathan Wiebe. "Well-conditioned multiproduct Hamiltonian simulation". *arXiv:1907.11679* (2019).

[129] Almudena Carrera Vazquez, Daniel J Egger, David Ochsner, and Stefan Woerner. "Well-conditioned multi-product formulas for hardware-friendly Hamiltonian simulation". *Quantum* 7 (2023), p. 1067.

[130] Gumaro Rendon, Jacob Watkins, and Nathan Wiebe. "Improved Accuracy for Trotter Simulations Using Chebyshev Interpolation". *Quantum* 8 (2024), p. 1266.

[131] Junaid Aftab, Dong An, and Konstantina Trivisa. "Multi-product Hamiltonian simulation with explicit commutator scaling". *arXiv:2403.08922* (2024).

[132] Maxine Luo and J Ignacio Cirac. "Efficient simulation of quantum chemistry problems in an enlarged basis set". *arXiv:2407.04432* (2024).

[133] Matthew Pocrnic, Matthew Hagan, Juan Carrasquilla, Dvira Segal, and Nathan Wiebe. "Composite qDRIFT-product formulas for quantum and classical simulations in real and imaginary time". *Physical Review Research* 6.1 (2024), p. 013224.

[134] Dominic W Berry and Howard M Wiseman. "Optimal states and almost optimal adaptive measurements for quantum interferometry". *Physical review letters* 85.24 (2000), p. 5098.

[135] Alicja Dutkiewicz, Stefano Polla, Maximilian Scheurer, Christian Gogolin, William J Huggins, and Thomas E O'Brien. "Error mitigation and circuit division for early fault-tolerant quantum phase estimation". *arXiv:2410.05369* (2024).

[136] William J Huggins, Jarrod R McClean, Nicholas C Rubin, Zhang Jiang, Nathan Wiebe, K Birgitta Whaley, and Ryan Babbush. "Efficient and noise resilient measurements for quantum chemistry on near-term quantum computers". *npj Quantum Information* 7.1 (2021), p. 23.

[137] Jeffrey Cohn, Mario Motta, and Robert M Parrish. "Quantum filter diagonalization with compressed double-factorized Hamiltonians". *PRX Quantum* 2.4 (2021), p. 040352.

[138] Isaac H Kim, Ye-Hua Liu, Sam Pallister, William Pol, Sam Roberts, and Eunseok Lee. "Fault-tolerant resource estimate for quantum chemical simulations: Case study on Li-ion battery electrolyte molecules". *Physical Review Research* 4.2 (2022), p. 023019.

[139] Oumarou Oumarou, Maximilian Scheurer, Robert M Parrish, Edward G Hohenstein, and Christian Gogolin. "Accelerating quantum computations of chemistry through regularized compressed double factorization". *arXiv:2212.07957* (2022).

[140] Sergey Bravyi, Jay M Gambetta, Antonio Mezzacapo, and Kristan Temme. "Tapering off qubits to simulate fermionic Hamiltonians". *arXiv:1701.08213* (2017).

[141] Andrew Tranter, Peter J Love, Florian Mintert, and Peter V Coveney. "A comparison of the Bravyi–Kitaev and Jordan–Wigner transformations for the quantum simulation of quantum chemistry". *Journal of chemical theory and computation* 14.11 (2018), pp. 5617–5630.

[142] Andrew Tranter, Peter J Love, Florian Mintert, Nathan Wiebe, and Peter V Coveney. "Ordering of trotterization: Impact on errors in quantum simulation of electronic structure". *Entropy* 21.12 (2019), p. 1218.

[143] Changhao Yi and Elizabeth Crosson. "Spectral analysis of product formulas for quantum simulation". *npj Quantum Information* 8.1 (2022), p. 37.

[144] Alex Bocharov, Martin Roetteler, and Krysta M Svore. "Efficient synthesis of probabilistic quantum circuits with fallback". *Physical Review A* 91.5 (2015), p. 052317.

[145] Craig Gidney and Austin G Fowler. "Efficient magic state factories with a catalyzed $|CCZ\rangle \to 2|T\rangle$ transformation". *Quantum* 3 (2019), p. 135.

[146] M. B. Hastings, D. Wecker, B. Bauer, and M. Troyer. "Improving Quantum Algorithms for Quantum Chemistry". *arXiv:1403.1539* (2014).

[147] Earl T Campbell. "Early fault-tolerant simulations of the Hubbard model". *Quantum Science and Technology* 7.1 (2021), p. 015007.

[148] Tzu-Ching Yen, Vladyslav Verteletskyi, and Artur F Izmaylov. "Measuring all compatible operators in one series of single-qubit measurements using unitary transformations". *Journal of chemical theory and computation* 16.4 (2020), pp. 2400–2409.

[149] Tzu-Ching Yen, Aadithya Ganeshram, and Artur F Izmaylov. "Deterministic improvements of quantum measurements with grouping of compatible operators, non-local transformations, and covariance estimates". *npj Quantum Information* 9.1 (2023), p. 14.

[150] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J Love, Alán Aspuru-Guzik, and Jeremy L O'brien. "A variational eigenvalue solver on a photonic quantum processor". *Nature Communications* 5.1 (2014), p. 4213.

[151] Jules Tilly, Hongxiang Chen, Shuxiang Cao, Dario Picozzi, Kanav Setia, Ying Li, Edward Grant, Leonard Wossnig, Ivan Rungger, George H. Booth, and Jonathan Tennyson. "The Variational Quantum Eigensolver: A Review of Methods and Best Practices". *Physics Reports*. The Variational Quantum Eigensolver: A Review of Methods and Best Practices 986 (2022), pp. 1–128.

[152] Guifré Vidal. "Efficient classical simulation of slightly entangled quantum computations". *Physical review letters* 91.14 (2003), p. 147902.

[153] Igor L Markov and Yaoyun Shi. "Simulating quantum computation by contracting tensor networks". *SIAM Journal on Computing* 38.3 (2008), pp. 963–981.

[154] Feng Pan and Pan Zhang. "Simulation of quantum circuits using the big-batch tensor network method". *Physical Review Letters* 128.3 (2022), p. 030501.

[155] Thomas Ayral, Thibaud Louvet, Yiqing Zhou, Cyprien Lambert, E Miles Stoudenmire, and Xavier Waintal. "Density-matrix renormalization group algorithm for simulating quantum circuits with a finite fidelity". *PRX Quantum* 4.2 (2023), p. 020304.

[156] Joseph Tindall, Matthew Fishman, E Miles Stoudenmire, and Dries Sels. "Efficient tensor network simulation of IBM's eagle kicked Ising experiment". *PRX quantum* 5.1 (2024), p. 010308.

[157] Sergey Bravyi, Dan Browne, Padraic Calpin, Earl Campbell, David Gosset, and Mark Howard. "Simulation of quantum circuits by low-rank stabilizer decompositions". *Quantum* 3 (2019), p. 181.

[158] Tomislav Begušić, Johnnie Gray, and Garnet Kin-Lic Chan. "Fast and converged classical simulations of evidence for the utility of quantum computing before fault tolerance". *Science Advances* 10.3 (2024), eadk4321.

[159] Tomislav Begušić and Garnet Kin-Lic Chan. "Real-time operator evolution in two and three dimensions via sparse Pauli dynamics". *PRX Quantum* 6.2 (2025), p. 020302.

[160] Xiaosi Xu, Simon Benjamin, Jinzhao Sun, Xiao Yuan, and Pan Zhang. "A Herculean task: Classical simulation of quantum computers". *arXiv:2302.08880* (2023).

[161] Ali Javadi-Abhari, Matthew Treinish, Kevin Krsulich, Christopher J. Wood, Jake Lishman, Julien Gacon, Simon Martiel, Paul D. Nation, Lev S. Bishop, Andrew W. Cross, Blake R. Johnson, and Jay M. Gambetta. "Quantum computing with Qiskit" (2024).

[162] Jones, Tyson and Brown, Anna and Bush, Ian and Benjamin, Simon C. "QuEST and High Performance Simulation of Quantum Computers". *Scientific Reports* 9.1 (2019), p. 10736.

[163] Harun Bayraktar et al. "cuQuantum SDK: A High-Performance Library for Accelerating Quantum Science". *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)*. Vol. 01. 2023, pp. 1050–1061.

[164] Amit Jamadagni Gangapuram, Andreas M. Läuchli, and Cornelius Hempel. "Benchmarking quantum computer simulation software packages: State vector simulators". *SciPost Phys. Core* 7 (2024), p. 075.

[165] Roeland Wiersema, Cunlu Zhou, Yvette de Sereville, Juan Felipe Carrasquilla, Yong Baek Kim, and Henry Yuen. "Exploring entanglement and optimization within the hamiltonian variational ansatz". *PRX Quantum* 1.2 (2020), p. 020319.

[166] Harper R Grimsley, Sophia E Economou, Edwin Barnes, and Nicholas J Mayhall. "An adaptive variational algorithm for exact molecular simulations on a quantum computer". *Nature Communications* 10.1 (2019), p. 3007.

[167] Kostas Blekos, Dean Brand, Andrea Ceschini, Chiao-Hui Chou, Rui-Hao Li, Komal Pandya, and Alessandro Summer. "A review on quantum approximate optimization algorithm and its variants". *Physics Reports* 1068 (2024), pp. 1–66.

[168] Barratt, F, Bal, Matthias, Dborin, James, Stojevic, Vid, Pollmann, Frank, Green, A. G. "Parallel quantum simulation of large systems on small NISQ computers". *npj Quantum Information* 7.1 (2021), p. 79.

[169] Hans De Raedt, Fengping Jin, Dennis Willsch, Madita Willsch, Naoki Yoshioka, Nobuyasu Ito, Shengjun Yuan, and Kristel Michielsen. "Massively parallel quantum computer simulator, eleven years later". *Computer Physics Communications* 237 (2019).

[170] Xin-Chuan Wu, Sheng Di, Emma Maitreyee Dasgupta, Franck Cappello, Hal Finkel, Yuri Alexeev, and Frederic T. Chong. "Full-state quantum circuit simulation by using data compression". *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. SC '19. Denver, Colorado: Association for Computing Machinery, 2019.

[171] Jones, Tyson and Koczor, Bálint and Benjamin, Simon C. "Distributed Simulation of Statevectors and Density Matrices". *arXiv:2311.01512* (2023).

[172] Message Passing Interface Forum. *MPI: A Message-Passing Interface Standard Version 4.1*. 2023.

[173] Quincey Koziol, Russ Rew, Mark Howison, et al. "HDF5". *High Performance Parallel I/O*. Chapman and Hall/CRC, 2014, pp. 227–244.

[174] *CUDA C++ Programming Guide*. `https://docs.nvidia.com/cuda/pdf/CUDA_C_Programming_Guide.pdf`. NVIDIA Corporation. 2025.

[175] Gene M Amdahl. "Validity of the single processor approach to achieving large scale computing capabilities, reprinted from the AFIPS conference proceedings, vol. 30 (atlantic city, nj, apr. 18–20)". *IEEE Solid-State Circuits Society Newsletter* 12.3 (2007), pp. 19–20.

[176] John L Gustafson. "Reevaluating Amdahl's law". *Communications of the ACM* 31.5 (1988), pp. 532–533.

[177] Robert Trondl, Petra Heffeter, Christian R. Kowol, Michael A. Jakupec, Walter Berger, and Bernhard K. Keppler. "NKP-1339, the first ruthenium-based anticancer drug on the edge to clinical application". *Chem. Sci.* 5 (2014), pp. 2925–2932.

[178] Wolfgang Peti, Thomas Pieper, Martina Sommer, Bernhard K. Keppler, and Gerald Giester. "Synthesis of Tumor-Inhibiting Complex Salts Containing the Aniontrans-Tetrachlorobis(indazole)ruthenate(III) and Crystal Structure of the Tetraphenylphosphonium Salt". *Eur. J. Inorg. Chem.* 1999 (1999), pp. 1551–1555.

[179] C. V. Ramamoorthy and H. F. Li. "Pipeline Architecture". *ACM Comput. Surv.* 9.1 (1977), pp. 61–102.

[180] Alba T. Macias, Douglas S. Williamson, Nicola Allen, Jenifer Borgognoni, Alexandra Clay, Zoe Daniels, Pawel Dokurno, Martin J. Drysdale, Geraint L. Francis, Christopher J. Graham, Rob Howes, Natalia Matassova, James B. Murray, Rachel Parsons, Terry Shaw, Allan E. Surgenor, Lindsey Terry, Yikang Wang, Mike Wood, and Andrew J. Massey. "Adenosine-Derived Inhibitors of 78 kDa Glucose Regulated Protein (Grp78) ATPase: Insights into Isoform Selectivity". *J. Med. Chem.* 54 (2011), pp. 4034–4041.

[181] Moritz Bensberg, Marco Eckhoff, F. Emil Thomasen, William Bro-Jørgensen, Matthew S. Teynor, Valentina Sora, Thomas Weymuth, Raphael T. Husistein, Frederik E. Knudsen, Anders Krogh, Kresten Lindorff-Larsen, Markus Reiher, and Gemma C. Solomon. "Machine Learning Enhanced Calculation of Quantum-Classical Binding Free Energies". *arXiv:2503.03955* (2025).

[182] Moritz Bensberg, Marco Eckhoff, Raphael T. Husistein, Matthew S. Teynor, Valentina Sora, William Bro-Jørgensen, F. Emil Thomasen, Anders Krogh, Kresten Lindorff-Larsen, Gemma C. Solomon, Thomas Weymuth, and Markus Reiher. "Hierarchical quantum embedding by machine learning for large molecular assemblies". *arXiv:2503.03928* (2025).

[183] Robert B Best and Gerhard Hummer. "Optimized molecular dynamics force fields applied to the helix- coil transition of polypeptides". *The Journal of Physical Chemistry B* 113.26 (2009), pp. 9004–9015.

[184] Kresten Lindorff-Larsen, Stefano Piana, Kim Palmo, Paul Maragakis, John L. Klepeis, Ron O. Dror, and David E. Shaw. "Improved side-chain torsion potentials for the Amber ff99SB protein force field". *Proteins: Structure, Function, and Bioinformatics* 78 (2010), pp. 1950–1958.

[185] J. P. Perdew, K. Burke, and M. Ernzerhof. "Generalized Gradient Approximation Made Simple". *Phys. Rev. Lett.* 77 (1996), p. 3865.

[186] Stefan Grimme, Jens Antony, Stephan Ehrlich, and Helge Krieg. "A consistent and accurate ab initio parametrization of density functional dispersion correction (DFT-D) for the 94 elements H-Pu". *J. Chem. Phys.* 132 (2010), p. 154104.

[187] S. Grimme, S. Ehrlich, and L. Goerigk. "Effect of the damping function in dispersion corrected density functional theory". *J. Comput. Chem.* 32 (2011), pp. 1456–1465.

[188] F. Weigend and Reinhart Ahlrichs. "Balanced basis sets of split valence, triple zeta valence and quadruple zeta valence quality for H to Rn: Design and assessment of accuracy". *Phys. Chem. Chem. Phys.* 7 (2005), pp. 3297–3305.

[189] Jan P. Unsleber, Thomas Dresselhaus, Kevin Klahr, David Schnieders, Michael Böckers, Dennis Barton, and Johannes Neugebauer. "Serenity : A Subsystem Quantum Chemistry Program". *J. Comput. Chem.* 39 (2018), pp. 788–798.

[190] Niklas Niemeyer, Patrick Eschenbach, Moritz Bensberg, Johannes Tölle, Lars Hellmann, Lukas Lampe, Anja Massolle, Anton Rikus, David Schnieders, Jan P. Unsleber, and Johannes Neugebauer. "The subsystem quantum chemistry program Serenity". *WIREs Comput. Mol. Sci.* 13 (2022), e1647.

[191] Christopher J. Stein and Markus Reiher. "Automated Selection of Active Orbital Spaces". *Journal of Chemical Theory and Computation* 12 (2016), pp. 1760–1771.

[192] Christopher J. Stein and Markus Reiher. "autoCAS: A Program for Fully Automated Multiconfigurational Calculations". *J. Comput. Chem.* 40 (2019), pp. 2216–2226.

[193] Moritz Bensberg, Maximilian Mörchen, Christopher Johannes Stein, Jan Patrick Unsleber, Thomas Weymuth, and Markus Reiher. "qcscine/autocas: Release 2.3.0" (2024). DOI: 10.5281/ZENODO.7179859.

[194] Steven R. White. "Density Matrix Formulation for Quantum Renormalization Groups". *Phys. Rev. Lett.* 69 (1992), pp. 2863–2866.

[195] Sebastian Keller, Michele Dolfi, Matthias Troyer, and Markus Reiher. "An efficient matrix product operator representation of the quantum chemical Hamiltonian". *J. Chem. Phys.* 143 (2015), p. 244118.

[196] Qiming Sun, Timothy C. Berkelbach, Nick S. Blunt, George H. Booth, Sheng Guo, Zhendong Li, Junzi Liu, James D. McClain, Elvira R. Sayfutyarova, Sandeep Sharma, Sebastian Wouters, and Garnet Kin-Lic Chan. "PySCF: the Python-based simulations of chemistry framework". *WIREs Comput. Mol. Sci.* 8 (2018), e1340.

[197] G. Barcza, Ö. Legeza, K. H. Marti, and M. Reiher. "Quantum-information analysis of electronic states of different molecular structures". *Phys. Rev. A* 83 (2011), p. 012508.

[198] Katharina Boguslawski, Konrad H. Marti, and Markus Reiher. "Construction of CASCI-type wave functions for very large active spaces". *Journal of Chemical Physics* 134.22 (2011), p. 224101.

[199] Ivan Kassal, James D. Whitfield, Alejandro Perdomo-Ortiz, Man-Hong Yung, and Alán Aspuru-Guzik. "Simulating Chemistry Using Quantum Computers". *Ann. Rev. Phys. Chem.* 62.1 (2011), pp. 185–207.

[200] Yudong Cao, Jonathan Romero, Jonathan P. Olson, Matthias Degroote, Peter D. Johnson, Mária Kieferová, Ian D. Kivlichan, Tim Menke, Borja Peropadre, Nicolas P. D. Sawaya, Sukin Sim, Libor Veis, and Alán Aspuru-Guzik. "Quantum Chemistry in the Age of Quantum Computing". *Chem. Rev.* 119.19 (2019), pp. 10856–10915.

[201] Sam McArdle, Suguru Endo, Alán Aspuru-Guzik, Simon C. Benjamin, and Xiao Yuan. "Quantum computational chemistry". *Rev. Mod. Phys.* 92 (2020), p. 015003.

[202] Bela Bauer, Sergey Bravyi, Mario Motta, and Garnet Kin-Lic Chan. "Quantum algorithms for quantum chemistry and quantum materials science". *Chem. Rev.* 120 (2020), pp. 12685–12717.

[203] Alberto Baiardi, Matthias Christandl, and Markus Reiher. "Quantum Computing for Molecular Biology". *ChemBioChem* 24.13 (2023), e202300120.

[204] Joshua J Goings, Alec White, Joonho Lee, Christofer S Tautermann, Matthias Degroote, Craig Gidney, Toru Shiozaki, Ryan Babbush, and Nicholas C Rubin. "Reliably assessing the electronic structure of cytochrome P450 on today's classical computers and tomorrow's quantum computers". *Proc. Natl. Acad. Sci.* 119.38 (2022), e2203533119.

[205] Hans Hon Sang Chan, Richard Meister, Tyson Jones, David P. Tew, and Simon C. Benjamin. "Grid-based methods for chemistry simulations on a quantum computer". *Science Advances* 9.9 (2023).

[206] Björn O. Roos. "Chapter 25 - Multiconfigurational quantum chemistry". *Theory and Applications of Computational Chemistry*. Ed. by Clifford E. Dykstra, Gernot Frenking, Kwang S. Kim, and Gustavo E. Scuseria. Amsterdam: Elsevier, 2005, pp. 725–764.

[207] Jeppe Olsen. "The CASSCF method: A perspective and commentary". *Int. J. Quantum Chem.* 111 (2011), pp. 3267–3272.

[208] Hans Lischka, Dana Nachtigallová, Adélia J. A. Aquino, Péter G. Szalay, Felix Plasser, Francisco B. C. Machado, and Mario Barbatti. "Multireference Approaches for Excited States of Molecules". *Chem. Rev.* 118 (2018), pp. 7293–7361.

[209] L Gonzalez. *Quantum chemistry and dynamics of excited states*. Ed. by Roland Lindh and Leticia Gonzalez. Nashville, TN: John Wiley & Sons, 2020.

[210] Garnet Kin-Lic Chan, Jonathan J. Dorando, Debashree Ghosh, Johannes Hachmann, Eric Neuscamman, Haitao Wang, and Takeshi Yanai. "An Introduction to the Density Matrix Renormalization Group Ansatz in Quantum Chemistry". *Frontiers in Quantum Systems in Chemistry and Physics*. Springer-Verlag, 2008, pp. 49–65.

[211] Garnet Kin Lic Chan and Dominika Zgid. "The Density Matrix Renormalization Group in Quantum Chemistry". *Annu. Rep. Comput. Chem.* 5 (2009), pp. 149–162.

[212] Ulrich Schollwöck. "The density-matrix renormalization group in the age of matrix product states". *Ann. Phys.* 326.1 (2011), pp. 96–192.

[213] Garnet Kin-Lic Chan and Sandeep Sharma. "The density matrix renormalization group in quantum chemistry". *Annu. Rev. Phys. Chem.* 62 (2011), pp. 465–481.

[214] Sebastian Wouters and Dimitri Van Neck. "The density matrix renormalization group for ab initio quantum chemistry". *Eur. Phys. J. D* 31.2 (2013), p. 272.

[215] Yuki Kurashige. "Multireference electron correlation methods with density matrix renormalisation group reference functions". *Mol. Phys.* 112.11 (2014), pp. 1485–1494.

[216] Roberto Olivares-Amaya, Weifeng Hu, Naoki Nakatani, Sandeep Sharma, Jun Yang, and Garnet Kin-Lic Chan. "The ab-initio density matrix renormalization group in practice". *J. Chem. Phys.* 142.3 (2015), p. 34102.

[217] Szilárd Szalay, Max Pfeffer, Valentin Murg, Gergely Barcza, Frank Verstraete, Reinhold Schneider, and Örs Legeza. "Tensor product methods and entanglement optimization for ab initio quantum chemistry". *Int. J. Quantum Chem.* 115.19 (2015), pp. 1342–1391.

[218] Takeshi Yanai, Yuki Kurashige, Wataru Mizukami, Jakub Chalupský, Tran Nguyen Lan, and Masaaki Saitow. "Density matrix renormalization group for ab initio calculations and associated dynamic correlation methods: A review of theory and applications". *Int. J. Quantum Chem.* 115.5 (2015), pp. 283–299.

[219] George H. Booth, Alex J. W. Thom, and Ali Alavi. "Fermion Monte Carlo without fixed nodes: A game of life, death, and annihilation in Slater determinant space". *J. Chem. Phys.* 131.5 (2009), p. 054106.

[220] D. M. Cleland, George H. Booth, and Ali Alavi. "A study of electron affinities using the initiator approach to full configuration interaction quantum Monte Carlo". *J. Chem. Phys.* 134 (2011).

[221] Kerstin. Andersson, Per Aake. Malmqvist, Bjoern O. Roos, Andrzej J. Sadlej, and Krzysztof. Wolinski. "Second-order perturbation theory with a CASSCF reference function". *J. Phys. Chem.* 94.14 (1990), pp. 5483–5488.

[222] Kerstin Andersson, Per-Åke Malmqvist, and Björn O. Roos. "Second-order perturbation theory with a complete active space self-consistent field reference function". *J. Chem. Phys.* 96 (1992), pp. 1218–1226.

[223] C. Angeli, R. Cimiraglia, S. Evangelisti, T. Leininger, and J.-P. Malrieu. "Introduction of n-electron valence states for multireference perturbation theory". *J. Chem. Phys.* 114 (2001), pp. 10252–10264.

[224] Bogumil Jeziorski, Robert Moszynski, and Krzysztof Szalewicz. "Perturbation Theory Approach to Intermolecular Potential Energy Surfaces of van der Waals Complexes". *Chem. Rev.* 94.7 (1994), pp. 1887–1930.

[225] Celestino Angeli, Renzo Cimiraglia, and Jean-Paul Malrieu. "N-Electron Valence State Perturbation Theory: A Spinless Formulation and an Efficient Implementation of the Strongly Contracted and of the Partially Contracted Variants". *The Journal of Chemical Physics* 117.20 (2002), pp. 9138–9153.

[226] Fionn D. Malone, Robert M. Parrish, Alicia R. Welden, Thomas Fox, Matthias Degroote, Elica Kyoseva, Nikolaj Moll, Raffaele Santagati, and Michael Streif. "Towards the simulation of large scale protein–ligand interactions on NISQ-era quantum computers". *Chem. Sci.* 13.11 (2022), pp. 3094–3108.

[227] Matthias Loipersberger, Fionn D. Malone, Alicia R. Welden, Robert M. Parrish, Thomas Fox, Matthias Degroote, Elica Kyoseva, Nikolaj Moll, Raffaele Santagati, and Michael Streif. "Accurate non-covalent interaction energies on noisy intermediate-scale quantum computers via second-order symmetry-adapted perturbation theory". *Chem. Sci.* 14.13 (2023), pp. 3587–3599.

[228] Alessandro Tammaro, Davide E. Galli, Julia E. Rice, and Mario Motta. "N-Electron Valence Perturbation Theory with Reference Wave Functions from Quantum Computing: Application to the Relative Stability of Hydroxide Anion and Hydroxyl Radical". *J. Phys. Chem. A* 127 (2023), pp. 817–827.

[229] Michal Krompiec and David Muñoz Ramo. "Strongly Contracted N-Electron Valence State Perturbation Theory Using Reduced Density Matrices from a Quantum Computer". *arXiv:2210.05702* (2022).

[230] Cristian L. Cortes, Matthias Loipersberger, Robert M. Parrish, Sam Morley-Short, William Pol, Sukin Sim, Mark Steudtner, Christofer S. Tautermann, Matthias Degroote, Nikolaj Moll, Raffaele Santagati, and Michael Streif. "Fault-tolerant quantum algorithm for symmetry-adapted perturbation theory". *arXiv:2305.07009* (2023).

[231] Tosio Kato. *Perturbation Theory for Linear Operators*. 2nd edition. Berlin: Springer, 1995.

[232] Isaiah Shavitt and Rodney J. Bartlett. *Many-Body Methods in Chemistry and Physics: MBPT and Coupled-Cluster Theory*. Cambridge Molecular Science. Cambridge: Cambridge University Press, 2009.

[233] Antoine Marie, Hugh G. A. Burton, and Pierre-François Loos. "Perturbation theory in the complex plane: exceptional points and where to find them". *Journal of Physics: Condensed Matter* 33 (2021), p. 283001.

[234] K. A. Brueckner. "Many-Body Problem for Strongly Interacting Particles. II. Linked Cluster Expansion". *Phys. Rev.* 100 (1955), pp. 36–45.

[235] A. Zygmund. *Trigonometric Series*. 3rd ed. Cambridge Mathematical Library. Cambridge: Cambridge University Press, 2003.

[236] Dmitry Grinko, Julien Gacon, Christa Zoufal, and Stefan Woerner. "Iterative quantum amplitude estimation". *npj Quantum Information* 7.1 (2021), pp. 1–6.

[237] Shion Fukuzawa, Christopher Ho, Sandy Irani, and Jasen Zion. "Modified Iterative Quantum Amplitude Estimation is Asymptotically Optimal". *SIAM 2023 Proceedings of the Symposium on Algorithm Engineering and Experiments (ALENEX)*. Proceedings (2023), pp. 135–147.

[238] Lin Lin and Yu Tong. "Near-optimal ground state preparation". *Quantum* 4 (2020), p. 372.

[239] David Poulin and Pawel Wocjan. "Preparing Ground States of Quantum Many-Body Systems on a Quantum Computer". *Phys. Rev. Lett.* 102.13 (2009), p. 130503.

[240] Yimin Ge, Jordi Tura, and J. Ignacio Cirac. "Faster ground state preparation and high-precision ground energy estimation with fewer qubits". *J. Math. Phys.* 60.2 (2019), p. 022202.

[241] Kenneth G. Dyall. "The choice of a zeroth-order Hamiltonian for second-order perturbation theory with a complete active space self-consistent-field reference function". *J. Chem. Phys.* 102 (1995), pp. 4909–4918.

[242] Marco Häser. "Møller-Plesset (MP2) Perturbation Theory for Large Molecules". *Theoretica chimica acta* 87.1 (1993), pp. 147–173.

[243] Philippe Y. Ayala and Gustavo E. Scuseria. "Linear Scaling Second-Order Moller–Plesset Theory in the Atomic Orbital Basis for Large Molecular Systems". *The Journal of Chemical Physics* 110.8 (1999), pp. 3660–3671.

[244] Alexander Yu. Sokolov and Garnet Kin-Lic Chan. "A Time-Dependent Formulation of Multi-Reference Perturbation Theory". *The Journal of Chemical Physics* 144.6 (2016), p. 064102.

[245] Qiming Sun et al. "Recent developments in the PySCF program package". *The Journal of Chemical Physics* 153.2 (2020), p. 024109.

[246] Cynthia Meredith, Tracy P. Hamilton, and Henry F. III Schaefer. "Oxywater (Water Oxide): New Evidence for the Existence of a Structural Isomer of Hydrogen Peroxide". *The Journal of Physical Chemistry* 96.23 (1992), pp. 9250–9254.

[247] Hsing Hua Huang, Yaoming Xie, and Henry F. Schaefer. "Can Oxywater Be Made?" *The Journal of Physical Chemistry* 100.15 (1996), pp. 6076–6080.

[248] Andreas Fiedler, Jan Hrušák, Wolfram Koch, and Helmut Schwarz. "The Energetical and Structural Properties of FeO+. An Application of Multireference Perturbation Theory". *Chemical Physics Letters* 211.2 (1993), pp. 242–248.

[249] Kristine Pierloot. "The CASPT2 Method in Inorganic Electronic Spectroscopy: From Ionic Transition Metal to Covalent Actinide Complexes∗". *Molecular Physics* 101.13 (2003), pp. 2083–2094.

[250] Christopher J. Stein, Vera von Burg, and Markus Reiher. "The Delicate Balance of Static and Dynamic Electron Correlation". *Journal of Chemical Theory and Computation* 12.8 (2016), pp. 3764–3773.

[251] Krzysztof Szalewicz and Bogumil Jeziorski. "Comment on "On the importance of the fragment relaxation energy terms in the estimation of the basis set superposition error correction to the intermolecular interaction energy" [J. Chem. Phys. 104, 8821 (1996)]". *J. Chem. Phys.* 109.3 (1998), pp. 1198–1200.

[252] Stanislaw Rybak, Bogumil Jeziorski, and Krzysztof Szalewicz. "Many-body symmetry-adapted perturbation theory of intermolecular interactions. H2O and HF dimers". *J. Chem. Phys.* 95 (1991), pp. 6576–6601.

[253] Konrad Patkowski. "Recent developments in symmetry-adapted perturbation theory". *WIREs Comput. Mol. Sci.* 10 (2020).

[254] Hayes L. Williams, Eric M. Mas, Krzysztof Szalewicz, and Bogumil Jeziorski. "On the effectiveness of monomer-, dimer-, and bond-centered basis functions in calculations of intermolecular interaction energies". *J. Chem. Phys.* 103 (1995), pp. 7374–7391.

[255] Robert Moszyński, Sławomir M. Cybulski, and Grzegorz Chałasiński. "Many-body theory of intermolecular induction interactions". *J. Chem. Phys.* 100.7 (1994), pp. 4998–5010.

[256] Robert Moszynski, Bogumił Jeziorski, and Krzysztof Szalewicz. "Møller–Plesset expansion of the dispersion energy in the ring approximation". *Int. J. Quantum Chem.* 45.5 (1993), pp. 409–431.

[257] Jet Nestruev. *Smooth Manifolds and Observables*. Springer Graduate Texts in Mathematics. Springer, 2003.

[258] Gerd Grubb. *Distributions and Operators*. Graduate Texts in Mathematics. Springer New York, NY, 2009.

[259] Arie Israel. "The Eigenvalue Distribution of Time-Frequency Localization Operators". *arXiv:1502.04404* (2015).

[260] Gilles Brassard, Peter Hoyer, Michele Mosca, and Alain Tapp. "Quantum Amplitude Amplification and Estimation". *arXive:quant-ph/0005055* 305 (2002), pp. 53–74.

[261] Emanuel Knill, Gerardo Ortiz, and Rolando D. Somma. "Optimal quantum measurements of expectation values of observables". *Phys. Rev. A* 75 (2007), p. 012328.

[262] Andrew M. Childs, Robin Kothari, and Rolando D. Somma. "Quantum Algorithm for Systems of Linear Equations with Exponentially Improved Dependence on Precision". *SIAM Journal on Computing* 46.6 (2017), pp. 1920–1950.

[263] Samson Wang, Sam McArdle, and Mario Berta. "Qubit-Efficient Randomized Quantum Algorithms for Linear Algebra". arXiv:2302.01873 (2023).

Acknowledgements

This PhD thesis may be written by me, but I would not have come this far alone.

I am very grateful to have had Matthias Christandl as my supervisor. In the three years of my PhD he supported and challenged me, he instilled some rigour into a chemistry-minded brain, and he enthusiastically shared his insights and ideas. The opportunities he provided continue to shape my path in research.

Many thanks to Freek Witteveen for being a fantastic collaborator, and for having endless motivation to mentor and share knowledge with me.

Thanks to Dylan Harley, Marek Miller, Tommaso Aschieri, Daniel Malz, Harold Nieuwboer, Li Liu, Alexander Schmidhuber, Thomas Fraser, Sigurd Storgaard and many more people from QMATH, the Quantum for Life Centre and the Q4Bio collaboration for answering many of my questions and raising new ones in countless entertaining discussions. I want to thank all current and former members of QMATH for our silly discussions over lunch, for coffee breaks and tea hours, and for all the social events that made my PhD life so enjoyable. To Lise Steen Nielsen: thank you for always having an open ear and being a cheerful soul at the office.

I want to thank Aram Harrow for many insightful and stimulating discussions, for being a great host during my stay in Cambridge, and for letting me borrow his violin. Thanks to all members of the CTP at MIT who engaged with me in interesting research discussions and explored the Boston area with me.

Many thanks to Markus Reiher for introducing me to quantum chemistry during my Bachelor studies at ETH, for pointing me to the PhD position in Copenhagen, and for all the fruitful collaborations that followed. Thank you to Mihael Erakovic and Alberto Baiardi for all the chemistry advice you shared.

A special thanks to Dylan Harley, Frederik Ravn Klausen, Freek Witteveen, Marek Miller and Emil Mærsk Pedersen for helping me with useful comments and corrections when writing this thesis.

Thanks to my friends — in Copenhagen and abroad — and to all my fellow musicians: you never failed to keep my spirits up.

And of course I want to thank my family; Nina, Uwe, Marlene, Lasse — you always trusted and supported me in my decisions, and have always had my back. Thank you.