# Quantum Chemistry and Quantum Computers

## Testing the Bravyi-Kitaev mapping and Trotter order optimisations

Andrew Tranter

A thesis submitted for the award of Doctor of Philosophy

Centre for Doctoral Training in Controlled Quantum Dynamics
Department of Physics
Imperial College London

22nd November 2018

The copyright of this thesis rests with the author. Unless otherwise indicated, its contents are licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International Licence (CC BY-NC-SA).

Under this licence, you may copy and redistribute the material in any medium or format. You may also create and distribute modified versions of the work. This is on the condition that; you credit the author, do not use it for commercial purposes and share any derivative works under the same licence.

When reusing or sharing this work, ensure you make the licence terms clear to others by naming the licence and linking to the licence text. Where a work has been adapted, you should indicate that the work has been changed and describe those changes.

Please seek permission from the copyright holder for uses of this work that are not included in this licence or permitted under UK Copyright Law.

The author declares that the work contained herein is their own, and that all else is appropriately referenced.

# Abstract

The quantum simulation of quantum chemistry is often cited as one of the most important potential uses for emerging quantum technology. Quantum computing methods are expected to be able to perform full configuration interaction level quantum chemistry calculations with tractable computational resource requirements. However, early quantum computers are likely to be constrained in both the number of qubits available, and the number of gates that can be performed. Implementing chemical algorithms on such quantum devices will require algorithmic improvements.

In this thesis, two avenues for optimisation of the canonical method for the simulation of quantum chemistry on quantum computers are discussed. The *Bravyi-Kitaev transformation* is an alternative to the Jordan-Wigner transformation, the canonical method of mapping electronic states and operators to states and operators of qubits. The resource implications of this transformation scheme are assessed using a variety of chemical examples. Known techniques for optimising quantum circuits for quantum chemistry are applied, and the implications for the use of the Bravyi-Kitaev transformation are considered.

Trotterization – the use of Trotter-Suzuki formulae to approximate the evol-

ution operator for the molecular Hamiltonian – is discussed. In particular, ordering schemes designed to minimise the Trotter error are described. Finally, the importance of Trotter ordering when using the Bravyi-Kitaev transformation is considered.

The performance of either transformation scheme is found to be highly dependent on the Trotter ordering and level of optimisation chosen. As such, the relative merits of either transformation scheme are found to be not as simple as previously thought, although in most cases the Bravyi-Kitaev transformation is found to be advantageous.

The relative merits of both mapping and ordering schemes are considered in the context of performing quantum simulation of quantum chemistry on real quantum devices. Particular focus is paid to molecular systems which are intractable to classical computational simulation. The results emphasise the need for thorough analysis of optimisation schemes, such that quantum chemistry may indeed be a principal use of early quantum computation.

# Contents


1. Introduction 19
   - 1.1. Background . . . 19
     - 1.1.1. Electronic structure theory . . . 20
     - 1.1.2. Quantum computing . . . 26
   - 1.2. Quantum computing and chemistry . . . 30
     - 1.2.1. Experimental demonstrations . . . 32
     - 1.2.2. Alternative schemes . . . 34
     - 1.2.3. Resource estimates and algorithmic improvements 36

2. Theoretical Overview 39
   - 2.1. Basis sets . . . 43
     - 2.1.1. Hartree-Fock process . . . 48
   - 2.2. Integrals . . . 52
   - 2.3. Mapping electrons to qubits . . . 54
     - 2.3.1. Jordan-Wigner operators . . . 56
   - 2.4. Trotterization . . . 60
     - 2.4.1. Circuit description . . . 64
   - 2.5. Phase estimation . . . 67
     - 2.5.1. State preparation . . . 71
     - 2.5.2. Iterative phase estimation . . . 73

3. General Implementations 77
3.1. Overlap integrals . . . 80
3.1.1. Data sets . . . 83
3.2. Hamiltonians, unitaries and ordering . . . 86
3.3. Circuit simulation . . . 95

4. Theory of the Bravyi-Kitaev Transformation 101
4.1. The parity basis . . . 102
4.2. The Bravyi-Kitaev transformation . . . 106
4.2.1. Sets of qubits in the Bravyi-Kitaev mapping . . . 110
4.2.2. Bravyi-Kitaev operators . . . 116

5. Assessing the Bravyi-Kitaev Transformation 121
5.1. Methane – a case study . . . 122
5.2. Preliminary bulk studies . . . 126
5.2.1. Optimised circuits . . . 128
5.2.2. Impact of Trotter ordering . . . 133
5.3. Larger Bulk Studies . . . 136
5.3.1. Magnitude ordering . . . 139
5.3.2. Impact of Trotter ordering . . . 144

6. Trotterization Ordering Schemes 151
6.1. Molecular hydrogen . . . 155
6.2. Generalised ordering strategies . . . 163
6.2.1. Other ordering schemes . . . 167
6.2.2. Testing – random Hamiltonians . . . 169
6.2.3. Testing – molecular Hamiltonians . . . 174
6.3. Trotter error and the Bravyi-Kitaev mapping . . . 179
6.3.1. Commutator based orderings . . . 185

7. Conclusions and Outlook 191

Appendix A. Data sets 199

Appendix B. Gate.act() 213

Appendix C. Hydrogen Hamiltonian orderings 219

# List of Figures


1.1. The Bloch sphere . . . . . 27

2.1. Procedures involved in performing an electronic structure calculation . . . . . 41
2.2. A circuit to exponentiate a string of Pauli $Z$ operators . . . 65
2.3. A circuit to exponentiate a string of general Pauli operators 66
2.4. The phase estimation circuit . . . . . 68

3.1. An overview of the code . . . . . 79

5.1. Trotter error of methane . . . . . 125
5.2. Comparison of simple circuits for the SMALL_DATA set . . 127
5.3. Gate count estimates of the SMALL_DATA set with a magnitude ordering . . . . . 130
5.4. Gate count estimates of the SMALL_DATA set with a lexicographic ordering . . . . . 135
5.5. Unoptimised gate counts of the LARGE_DATA set . . . . . 140
5.6. Optimised gate counts of the LARGE_DATA set with a magnitude ordering . . . . . 142
5.7. Optimised gate counts of the LARGE_DATA set with a lexicographic ordering . . . . . 145
5.8. Gate savings of the Bravyi-Kitaev mapping in various orderings . . . . . 147

6.1. Commutativity graph of the hydrogen Hamiltonian . . . 156
6.2. Cumulative density plot of hydrogen ordering errors . . . 159
6.3. Further representations of the commutativity graph of the hydrogen Hamiltonian . . . 161
6.4. Flowcharts representing the commutator and reverseCommutator ordering schemes . . . 166
6.5. Ordering performance for random Hamiltonians . . . 172
6.6. Performance of ordering schemes for the SMALL_DATA set 176
6.7. Trotter error of ordering schemes using the LARGE_DATA set. . . . 178
6.8. Trotter error using the Bravyi-Kitaev mapping for different ordering schemes . . . 181
6.9. Relative error performance of Jordan-Wigner and Bravyi-Kitaev mappings . . . 183
6.10. Trotter error of Bravyi-Kitaev mapping using commutator and reverseCommutator orderings . . . 186
6.11. Comparison of Bravyi-Kitaev and Jordan-Wigner mappings using the commutator and reverseCommutator orderings 187

# List of Tables


3.1. An overview of the SMALL_DATA set . . . . . 84
3.2. An overview of the LARGE_DATA set . . . . . 86
3.3. An overview of the `Circuit` and `Gate` classes . . . . . 96

4.1. Sets of qubits for the occupation and parity bases . . . . . 105

6.1. Summary of random Hamiltonians . . . . . 170

A.1. A full breakdown of the SMALL_DATA set . . . . . 199
A.2. A full breakdown of the LARGE_DATA set . . . . . 208

# Listings

3.1. The hydrogen molecular Hamiltonian in oplist format . . 87
3.2. A function to implement one X gate on a state vector . . . 92

B.1. The `act` method of `Gate` . . . . . 213

C.1. Hydrogen oplist in a magnitude ordering . . . . . 219
C.2. Hydrogen oplist in a lexicographic ordering . . . . . 220
C.3. Hydrogen oplist in a lexoMag ordering . . . . . 220
C.4. Hydrogen oplist in a commutator ordering . . . . . 221
C.5. Hydrogen oplist in a reverseCommutator ordering . . . . 222

# Acknowledgements

I am phenomenally grateful to Peter V. Coveney of University College London, Peter J. Love of Tufts University, and Florian Mintert of Imperial College London, for their supervision of this project. It was an immense stroke of luck to have three remarkable supervisors, each of whom brought their own unique and insightful perspective on the work. Their support was both invaluable and infallible, even at the times when I caused more than my fair share of headache.

Over this project, I've had the fortune to meet, learn from and work with a great many colleagues, to whom I am indebted. To name only a few, I'd like to thank the participants of the 2015 Benasque Quantum Simulations conference, the participants of the 2015 UCLQ Quantum Information for Quantum Chemistry workshop (in particular, Borzu Toloui for organising the workshop and for discussions at UCL on sparse methods, and Nathan Wiebe for insightful comments on Trotter orderings). Similarly, the participants of the 2015 Telluride Novel Computing Approaches to Quantum Chemistry workshop (particularly James Whitfield and Zoltán Zimborás for organising the workshop), the participants of the 2016 Delft ScaleQIT conference, and the participants of the 2017 Sherbrooke Summer School on Quantum Information.

I'd also like to thank Frank Wilhelm-Mauch and the past and present members of his group (in particular, Michael Kaicher for many discussions on the Bravyi-Kitaev mapping, amongst other matters). Similar thanks go to Lucas Kocia and Yifei Huang at Tufts, and to Alán Aspuru-Guzik and his group (particularly Jhonathan Romero Fontalvo and Ian Kivlichan), for making

me feel welcome on various trips to the United States. I am also extremely grateful to Jarrod McClean and Ryan Babbush for (in addition to useful discussions) providing integral data that was used as one of our input data sets. Last but not least, my thanks go to my PhD examiners, Michael Bearpark and Seth Lloyd, for their helpful comments on this thesis and in my viva.

In London, my appreciation goes to the staff and students of the Centre for Doctoral Training in Controlled Quantum Dynamics, both present and former. In particular, Miranda Smith and Veena Dhulipala – the CDT's admin team for most of my time here – for putting up with my frequent causation of logistical nightmares. At least in my opinion, the special prize for Best Cohort goes to Cohort 5 of the aforementioned CDT. Without the mutual support, friendship and melancholic poetry, my time at Imperial would've been deeply worsened. I'd also like to thank Richard Thompson for his consistently understanding and warm support as our cohort mentor.

Looking back further, I'd also like to express my thanks to the many teachers in school and university who lit the path here. Few will be surprised by my decision to work on theory, many will probably be surprised that I didn't find a way to seriously injure myself anyway. On such a note, thanks also go to the staff of the NHS who have treated my family, friends and self. Without some of them, I likely wouldn't be here to write these words.

As few of them are likely to read this, I think I can safely also thank my friends, for sharing the best of times. I can't imagine a better rogues' gallery to spend my days with. Special shout-out goes to Greg and Tom for their typographical wisdom.

So much more than this thesis would not exist, were it not for my late grandparents: Olive, Ted and Elsie, the latter two of whom passed away during the course of this project. I can only hope that I follow even a fraction of their examples.

The final word, of course, must go to my parents and brother, who I must reluctantly admit shaped me into the person I am today.

The actual final word goes to Patch and Susie.

# 1 Introduction

## 1.1 Background

Computational chemistry is the use of computers to solve chemical problems. Numerical methods have long been used in chemistry to predict properties that may not be experimentally measurable, or validate theories which aim to describe chemical systems. However, after the Second World War, the rapid growth in the power of computing devices enabled such analysis on a previously unprecedented scale. Modern computational chemistry is a vast field, with myriad applications in both pure science and industry. To name only a few: in pharmaceutical science, computational chemical methods are used for the screening and optimisation of potential drugs.[1] In petrochemistry and the chemical industry, computational methods are used for process design.[2] In environmental science, computational methods are used to develop models of ecotoxicity and environmental processes.[3] Several Nobel prizes have been awarded in the discipline; of recent note is that of Karplus, Levitt and Warshell in 2013, and that of Kohn and Pople in 1998.

Computational chemistry consists of a broad spectrum of methods. They include the calculation of molecular geometric structure[4,5], electronic structure and energetics[6,7], spectroscopic data[8–10], reactivity[11,12], and chemical reaction dynamics and kinetics[13,14]. As discussed below, there are a range of approaches in the field, employing various degrees of abstraction away from the quantum mechanical nature of chemistry. The focus here is on methods that do not involve the use of classical mechanics or empirical data. At their core, these methods attempt to solve the Schrödinger equation for a Hamiltonian describing a chemical system. In some methods, this Hamiltonian only describes the state of the electrons in a molecule or system – these methods fall under the scope of electronic structure theory. In other methods, further degrees of freedom may be considered. For instance, molecular dynamical methods study the motion of atoms within molecules, typically in order to simulate the dynamics of a chemical reaction.[15]

In all quantum chemical methods considered here, however, the initial starting point is some form of electronic structure calculation.

### 1.1.1 *Electronic structure theory*

With the birth of quantum mechanics in the early 20th century, its application to the study of chemistry was inevitable. In 1929, Paul Dirac wrote:

> 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. It therefore becomes desirable that approximate practical methods of applying quantum mechanics should be developed, which can lead to an explanation of the main features of complex atomic systems without too much computation. (Dirac 1929[16])

While this neglects a great deal of the complexity of applying quantum mechanical methods to chemistry, it astutely states that the details of all chemical systems and reactions can – at least, provided one has access to a sufficiently powerful computer – be determined from first principles. In other words, such details can be ascertained through the solution of the Schrödinger equation, given a Hamiltonian of interest. The class of methods which aim to directly solve this equation are therefore referred to as *ab initio* methods.

Provided sufficient computing power, these methods could theoretically be used to study chemical systems of any size. However, in practice, the scaling properties of most *ab initio* methods – the functional form of the increase in computational difficulty associated with an increase in system size – can make their application difficult for large systems. For example, the full configuration interaction[17] method – discussed below – incurs a computational cost that scales exponentially with respect to the number of molecular orbitals considered, limiting its application to extremely small molecules. To compensate for this, theoretical chemists have developed a vast array of al-

ternative methods.

Density functional theory (DFT) is often used to study larger systems[18], due to its comparatively low cost. Here, the expected energy of the system is described using functionals of the electronic density, expressed as a function of position. However, as there is no known analytic functional form for the exchange interaction in terms of electronic density, approximation must be involved, often using some form of semi-empirical approach.[19,20]

Similarly, other methods can be used to study even larger chemical systems. Molecular mechanics can be used to simulate even proteins and other macromolecules, consisting of potentially millions of atoms. In order to achieve this, individual atoms or groups of atoms are treated as primitives. Quantum mechanics is often neglected entirely, with interactions being described according to purely Newtonian physics.

These methods have achieved a great deal of success in the simulation of large chemical systems. However, they all involve some level of approximation away from the underlying quantum mechanical nature of the electronic structure problem, and often require the specification of some form of non-fundamental empirical constants. Often, this makes *ab initio* methods better suited to the study of systems where quantum mechanical phenomena are important. Due to these constraints, there is no immediately intuitive application of quantum computing to these classical or semiclassical techniques.* Accordingly, this thesis instead focuses on *ab initio* methods.

*Nonetheless, useful applications have been found, such as the use of quantum annealing

*Ab initio* methods aim to ascertain the electronic wavefunctions of the molecular system, and their corresponding energies. Typically the ground state is of most interest, but excited states can also be calculated. The properties of the determined energetics and wavefunctions may provide useful chemical information, and are often the first step in a more involved calculation. For example, by varying molecular geometries and comparing ground state energies, geometry optimisation can be performed to elucidate the structure of molecules.

The most basic technique for tackling this problem is the *Hartree-Fock* method. As this is used in the work described in this thesis, the theory underpinning it is discussed in Section 2.1.1. The original procedure was developed in 1927 by Hartree.[22] This procedure was later adapted by Fock[23], and Slater[24] to account for antisymmetrisation of electronic wavefunctions. In essence, the algorithm uses an Ansatz molecular electronic wavefunction, represented by a Slater determinant of single-particle wavefunctions, themselves represented by a linear combination of basis functions parametrised by real space.[24] Electron-electron interactions are treated as interactions with a mean field, and the variational principle is invoked. This allows the generation of a new optimal set of coefficients for the basis functions constituting the single-particle wavefunctions, which in turn allows generation of new single-particle wavefunctions. This process is repeated until self-consistent.

This approach, whilst computationally inexpensive, is rarely considered to

to aid in the study of protein folding.[21]

yield quantitatively accurate results in contrast to other methods of quantum chemistry. The use of a single Slater determinant excludes a vast swathe of the true state space, and electronic correlation is only taken into consideration through antisymmetrisation - i.e. electrons are considered uncorrelated other than to the extent required by the Pauli principle. This leads to consistent overestimates of the ground state energy. The approach is particularly weak when multiple Slater determinants are degenerate, or nearly degenerate. This is particularly troublesome when studying strongly correlated systems.

A variety of *post Hartree-Fock* methods exist in quantum chemistry to deal with this problem. Perturbation theory, first applied in this context in 1934 by Møller and Plesset is often used[25–27]. For more accurate results, the whole electronic Hamiltonian is considered; but truncated so as to eliminate terms which involve multiple levels of excitation away from the Hartree-Fock ground state. This approach is the *configuration interaction* method. The inclusion of higher levels of excitation terms (singles, doubles, triples and so on) deterministically results in better energy estimates, although with an increase in computational cost that grows exponentially with the level of excitation considered.

A key problem with this approach is that truncation of the configuration interaction Hamiltonian results in a loss of size consistency. For size consistency to hold, the energy of particles at sufficiently large distance to be treated as non-interacting should be the sum of the independent energies of the constituent particles. As truncated configuration interaction inherently permits

terms which include the excitation of such individual particles but neglects the excitation of *all* individual particles to the same level, this property is not obeyed for truncated configuration interaction.[28] This has obvious consequences when studying systems of multiple molecules (for instance, when generating the potential energy surface of a reaction), but also in the study of large molecules, as coupling between highly spatially separated components may be overestimated.

To compensate for this, a typical approach is the coupled cluster method, wherein the initial wavefunction is acted upon by a truncation of the cluster operator, being a weighted sum of excitation operators. This maintains size consistency while still enabling truncation based on level of excitation. The CCSD(T) level[29] – including single and double excitations, with triples being treated peturbatively – is often considered to be the "gold standard" of quantum chemistry.[30]

These methods have achieved ubiquity through their ability to generate accurate results using existing classical computational resources. However, strong limitations still exist, particularly when using electronic structure theory as starting point for further calculations. Small relative errors in predicted ground state energy obtained by the above techniques can easily become large relative errors when considering energy *differences* – for instance, if calculating the barrier height of a reaction. When predicting rate constants for reactions, such barrier heights are usually exponentiated, potentially resulting in catastrophic error effects.[15]

The full configuration interaction[17] method is the configuration interaction method, but with no truncation applied. Conceptually, this technique is extremely simple. One generates the full electronic Hamiltonian, and diagonalises it. As no terms are dropped from the Hamiltonian (and the space on which it acts is not reduced) *, this technique is exact to within Born-Oppenheimer and non-relativistic approximations, and to within the constraint of using a finite basis set to express electronic wavefunctions. However, the dimension of the Fock space on which the Hamiltonian acts grows exponentially with respect to the number of orbitals considered. To describe the full electronic Hamiltonian using this method requires storing matrices of a size that scales exponentially. While this storage requirement can be circumvented – for instance, by generating matrix elements "on the fly" – there is no way to avoid the exponential computational cost of this algorithm. This makes the procedure intractable for all but the smallest systems, limiting its main application to benchmarking other methods.

### 1.1.2 *Quantum computing*

While the use of computers to perform simulation of quantum mechanical systems stretches back to the 1950s[31], the idea of using quantum mechanical systems to perform computation has a more recent history. Feynman, in his 1959 speech *There's Plenty of Room at the Bottom*, referred (in a context of

* A simple optimisation is to reduce the Fock space to the subspace with the correct number of electrons. While this substantially reduces the overhead, it does not eliminate the problematic scaling of the algorithm.

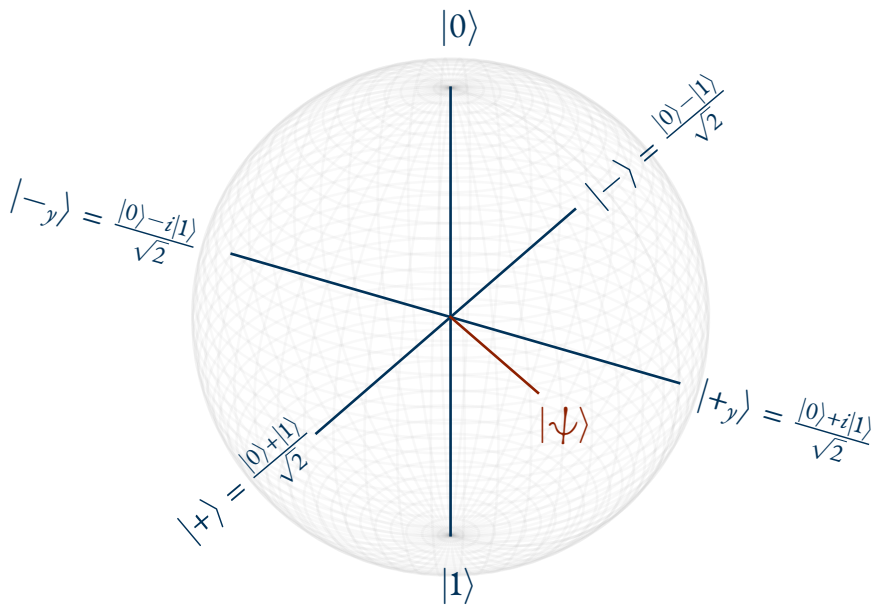



Figure 1.1:
The Bloch sphere, describing the state of a single qubit. State $|\psi\rangle$ is defined as $\cos\left(\frac{\vartheta}{2}\right)|0\rangle + e^{i\phi}\sin\left(\frac{\vartheta}{2}\right)|1\rangle$, where $\vartheta$ and $\phi$ are the polar and azimuthal angles of the point representing the state on this sphere.

miniaturisation) to the possibilities that the quantum mechanical nature of small systems could present.[32] However, it was not until 1979 that this was made concrete by Beinoff, who developed a quantum mechanical model of a Turing machine.[33] In 1985, Deutsch described a framework for a universal quantum computer[34], and demonstrated that such a device could solve a toy classical problem *faster than a classical computer*. The demonstration of quantum speed-up for useful tasks – prime factorisation of large numbers, for example[35], soon followed.

A quantum computer is a device which performs computations using *qubits* – quantum mechanical analogues of classical bits.[36] Whereas classical bits

can take state values of either 0 or 1, qubits can exist as superpositions of $|0\rangle$ and $|1\rangle$ states, i.e. $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$. These states can be represented as points on the *Bloch sphere*, shown in Figure 1.1. A system of $N$ qubits can then exist in a superposition of $2^N$ basis states, $|n\rangle$. Similar to classical computation, operations which manipulate the state of qubits are described as quantum gates and are analogous to classical logic gates. A sequence of quantum gates, intended to perform a computational task, is referred to as a quantum circuit.

Classical bits can be stored in a variety of ways – for instance, as two levels of current in a circuit, or two different charge levels of a capacitor. Similarly, in principle, any two-level quantum system can be used to store a qubit. For example, the two spin states of a fermion or the horizontal and vertical polarisation states of a single photon could be used. However, a number of experimental difficulties make finding an appropriate system difficult. The DiVincenzo criteria[37] describe the properties which a quantum computing architecture must obtain in order to be experimentally practical. Among these are the requirement that the qubits must have only two easily-accessible states, and that the decoherence of the system – the tendency of an arbitrary superposition state to transform into a mixed state – must be appropriately slow as to allow computation.

A plethora of experimental approaches have been proposed and implemented to this end. To name only a few, these include devices using trapped ions[38], photonics[39], nuclear spin[40], quantum dots[41], and NV centres[42]. Twenty years of experimental progress has conclusively demonstrated that the de-

velopment of a device which scales to large numbers of qubits is a difficult task. However, recently, it has been suggested that the creation of superconducting devices manipulating up to 50 qubits is plausible within even five years.[43,44] Such devices are rapidly approaching the realm where it may be possible to solve problems that are classically intractable.[44,45]

With regards to establishing the value of quantum computing, two major breakthroughs are necessary. The first, clearly, is the demonstration of a quantum device performing some computational task of which a classical computer is incapable. It has been argued[46] that devices will demonstrate this capacity in the near future. However, while such a demonstration would show that quantum devices are capable of performing otherwise impossible feats, it would not demonstrate that they could perform *useful* procedures.*

For this task, it will be necessary to perform a useful algorithm in a way which is faster than any conceivable classical computer. This is inherently a more difficult task, as it precludes the construction of algorithms specifically designed to demonstrate a quantum improvement. In the near future, any such demonstration will be constrained by extremely limited resources. As such, any algorithm intended to serve such a task will require as low an overhead as possible, thus necessitating the optimisation of algorithms and simulation procedures, such that the available resources can be used with maximal efficiency in the near future. This fact motivates the work described in this thesis.

*It could, of course, be argued that such a demonstration is useful in and of itself, but for the sake of brevity we will ignore these philosophical quandaries here.

Quantum computing has garnered a great deal of media attention due to Shor's algorithm – that is, the ability to efficiently factorise, and the consequent breaking of RSA cryptography. However, it is well known that the quantum computational resources required for this task are colossal in comparison to other expected benefits.[47] A perhaps less nihilistic example of the use of quantum computing is in the simulation of quantum systems.

## 1.2 Quantum computing and chemistry

As quantum devices fundamentally rely on quantum mechanics, it is intuitive that they may not be constrained by difficulties that classical computers face when simulating quantum systems. In a 1981 lecture, Richard Feynman stated:

'[...] nature isn't classical, dammit, and if you want to make a simulation of nature, you'd better make it quantum mechanical, and by golly it's a wonderful problem, because it doesn't look so easy.' (Feynman 1981[48])

This argument would later be proven correct by Lloyd[49], who demonstrated that quantum systems can be efficiently simulated by quantum computers through what would later become known as the *phase estimation algorithm*. In 1999, Abrams and Lloyd[50] developed a scheme for the simulation of fermions in a first-quantised representation, and generated a scheme to simulate the Hubbard model in a second-quantised representation. This led

to the development of a quantum algorithm for the explicit simulation of quantum chemistry in a second-quantised representation by Aspuru-Guzik, Dutoi, Love and Head-Gordon.[51]

In the full configuration interaction limit, these algorithms allow for an exponential speed-up over the simulation of quantum chemistry on classical computers. Obtaining exact full configuration interaction electronic ground state energies on a classical computer is intractable for large systems – on the order of hundreds of molecular orbitals, the resources required exceed the number of atoms in the observable universe. However, on a quantum device this can be done at polynomial cost, provided a suitable approximation to the ground state can be prepared on the quantum device.

The implications of this could have a dramatic impact on how quantum chemistry is performed.[52] It is extremely unlikely that classical computers will become redundant for quantum chemistry in anything but the far future, as classical computing methods will remain more practical in many circumstances. However, quantum devices are rapidly becoming larger and more stable.[43,44] Their use in performing extremely accurate electronic structure calculations could become pivotal in providing the last few digits of precision required to enable computational studies of reaction properties, in systems that are otherwise difficult to simulate. Reiher, Wiebe, Svore, Wecker and Troyer[45] estimate that a 142 logical qubit device performing one gate every 10 ns could determine the full configuration interaction energy of a FeMoco

cofactor *in less than a month of runtime at a precision of $\pm 0.1mH$. This may initially appear unexciting; however, to perform a full configuration interaction calculation using direct diagonalisation on a classical computer, would require the storage of matrices with a number of elements only a few orders of magnitude shy of the number of atoms in the milky way. While this example is somewhat contrived, and more advanced alternatives to direct diagonalisation approaches exist[17,54], these inherently retain the same problematic scaling or introduce additional approximation. It is for these reasons that the simulation of quantum chemistry is sometimes (perhaps overexuberantly) referred to as one of quantum computing's "killer apps".[55]

Although we do not dwell on them here, a cornucopia of quantum algorithms to address various chemical tasks have been developed, including the determination of energy spectra[51], reaction rates[45,56,57], and reaction dynamics[58].

### 1.2.1 *Experimental demonstrations*

While theoretical predictions regarding the capabilities of quantum computing for quantum chemistry are useful to assess the practicality of the task, real experimental demonstrations would be more conclusive. Unfortunately, this

*Being the primary cofactor of the nitrogenaeses, the enzymes which are biologically responsible for the conversion of nitrogen into ammonia. A great deal of debate and uncertainty exists as to much of the details regarding the mechanism of this process. Particularly, it would be desirable to understand how this process occurs at atmospheric temperature and pressure, in contrast to the requirements of the Haber process. For context, the Haber process consumes approximately 1-2% of the entirety of the world's energy production.[53]

task requires the (extremely difficult, as discussed above) development of reliable quantum devices. To date, the experimental demonstration of a simulation that could not be performed on a classical device has not been conducted. However, quantum chemistry is rapidly becoming an important choice of example calculation for the testing of new quantum devices.[59–63]

The first experimental demonstration of the quantum simulation of quantum chemistry was performed by Lanyon et al.[59] in 2010. This used an optical device to simulate the energy of the hydrogen molecule in a minimal basis, achieving a precision of 20 bits of the full configuration interaction energy. This was shortly followed by similar demonstrations using NMR[60] and diamond NV centres[61]. These simulations represent important milestones of experimental progress; however, they relied on the block diagonal nature of the hydrogen Hamiltonian – thus assuming that the Hamiltonian in full had already been described outside the quantum device. As this cannot be assumed for large systems, these approaches were not *algorithmically* scalable.

The first algorithmically scalable demonstration of quantum simulation of quantum chemistry – that is, involving no classical computational steps which scale intractably – was reported by O'Malley, Babbush et al. in 2016[62], to which some of the work described in Chapter 6 contributed. This used a superconducting device to again simulate the energetics of the hydrogen molecule in a minimal basis, and was the first algorithmically scalable demonstration of the entire procedure. Unfortunately, due to the difficulty of the task, their attempts were limited to a relatively short circuit, and thus their

implementation of the phase estimation algorithm did not achieve chemical accuracy (±1 kcal/mol). Their implementation of the variational quantum eigensolver (discussed in Section 1.2.2) did, however, achieve this, and in combination these efforts represented another important step in the experimental demonstration of the quantum simulation of quantum chemistry.

In light of the high computational requirements of this method, experimental demonstrations have largely moved in favour of the use of hybrid quantum-classical schemes such as the variational quantum eigensolver[64], briefly discussed below. This was first experimentally implemented in a photonic system by Peruzzo et al. in 2014.[65] This algorithm, which is substantially cheaper than the canonical scheme in terms of quantum resources, allowed for the recent calculation of the ground state energy of beryllium hydride by Kandala et al.[63], the largest quantum simulation of quantum chemistry performed to date. While these systems are very small in comparison to those typically studied by quantum chemists on classical computers, the rapid rate of progress combined with the large-scale chips discussed above are extremely encouraging for the prospect of experimental quantum quantum chemistry.

### 1.2.2 *Alternative schemes*

The work presented here focussed on assessing improvements to the second-quantised, Trotterization based scheme of Aspuru-Guzik, Dutoi, Love and Head-Gordon[51], the theory of which is discussed in Chapter 2. However,

other schemes for the quantum simulation of quantum chemistry have been developed. We discuss some of these here for context.

As mentioned above, the variational quantum eigensolver algorithm is an attractive alternative to full phase-estimation based schemes.[64–66] As in many classical computational schemes, this involves the specification of a parametrised Ansatz state. The expectation value of the Hamiltonian is measured, and the state re-parametrised. This process is optimised *variationally* until self-consistency is achieved. The final result is a variational estimate of the true eigenvalue.

This procedure is one adopted by many schemes when using classical computers. However, on a quantum device, it is possible to use a unitary form of the coupled-cluster ansatz, as this can be prepared by standard unitary evolution. This is known to yield strictly superior results to canonical coupled-cluster theory, but is classically intractable, as unlike typical coupled-cluster schemes it cannot be truncated based on level of excitation.[67]

In contrast to the gate-model paradigm discussed within this thesis, other approaches to quantum computing have been applied to chemistry. A native algorithm for the electronic structure problem on adiabatic quantum devices has been developed[68] and improved upon[69]. While these two paradigms are mutually emulable in polynomial time[70], it is nonetheless useful to have a native algorithm so as to reduce overheads. Approaches removing the necessity for Trotterization (the use of Trotter-Suzuki approximations to approximate evolution operators, discussed in Chapter 2) have been proposed, for

instance by using a truncated Taylor expansion.[71] Similarly, methods which exploit the sparsity of the configuration interaction matrix (discussed in Section 1.1.1) have been developed.[72,73]

Another alternative is the use of first-quantised schemes. Here, physical position space is discretised as a grid. The electronic wavefunction in the position representation is then stored in the state of the qubits. The effect of this Hamiltonian can be simulated by unitarily evolving the qubits through the propagator corresponding to the molecular Hamiltonian, approximated using the split operator method of Zalka[74]. This splits the first quantised Hamiltonian into components diagonal in the position and momentum bases. Each is simulated, and the quantum Fourier transform appropriately performed such that the qubits hold the correct representation. This approach has been extensively theoretically studied, although experimental demonstrations have not taken this approach.[58,75] In contrast to the canonical algorithm discussed in Chapter 2, an attractive aspect of this technique is its superior asymptotic scaling with regards to the number of orbitals considered, which is encouraging in the context of large quantum devices. However, the qubit cost of discretising real space is extremely large, possibly limiting the use of this technique in the immediate future.

### 1.2.3 *Resource estimates and algorithmic improvements*

Although the use of quantum computing for quantum chemistry shows a great deal of promise, it is clear that algorithmic advances are necessary to per-

form real simulations in the future. In 2013, Wecker, Bauer, Clark, Hastings and Troyer[76] concluded that despite the superior scaling properties, the large overheads of the standard algorithms of the time would result in prohibitively high runtimes for simulations involving approximately 100 qubits.

Later estimates by Poulin, Hastings, Wecker, Wiebe, Doherty and Troyer were more optimistic. Here, real molecular data was used in comparison to the random Hamiltonians generated in previous attempts. When using real molecular Hamiltonians, the scaling of the resources required for a chemically accurate simulation was substantially more optimistic than when using random Hamiltonians.[77] This emphasised the importance of using real molecular data to characterise the efficiency of algorithms. Nonetheless, they still concluded that algorithmic improvements will be necessary to perform quantum chemistry algorithms on real devices.

Restricting attention to Trotterized phase estimation circuits, many schemes for optimising circuits have been proposed. The *Bravyi-Kitaev mapping* is a method of mapping electronic states and operations to states and operations of and upon qubits, which scales asymptotically superiorly to canonical techniques.[78–80] This was first studied in a chemical context by Seeley, Richard and Love, who examined its resource implications for the simulation of the hydrogen molecule in a minimal basis.[79] However, prior to the work discussed in this thesis, no large-scale assessment of its resource implication on optimised circuits had been performed. Such an assessment is reported in Chapter 5.

Hastings, Wecker, Bauer and Troyer report a scheme involving the cancellation of duplicate gate sequences, and the use of ancillas to more easily determine parities. We consider these schemes in interaction with the Bravyi-Kitaev mapping in Chapter 5. Schemes have been developed for the use of multi-resolution Trotterization, wherein terms are simulated with differing numbers of Trotter steps, depending on the importance of the term.[77,81] More recently, the Bravyi-Kitaev superfast mapping[82] (not to be confused with the ordinarily fast Bravyi-Kitaev mapping discussed in Chapter 4) has obtained low Trotter error estimates in the simulation of molecular hydrogen.

Quantum simulation of quantum chemistry is a rapidly developing field, with a great deal of both theoretical and experimental progress made in the last decade. New quantum devices are rapidly approaching the realm where classically intractable calculations may be possible. To reach these auspicious heights, highly optimised algorithms will be necessary. In this thesis, two avenues for optimisation are discussed. Chapters 2 and 3 discuss the theory underpinning the quantum simulation of quantum chemistry and the technical implementation of our code respectively. Chapter 4 discusses the theory of the Bravyi-Kitaev mapping, with our work on quantifying its resource implications when studying a variety of molecular systems discussed in Chapter 5. Finally, in Chapter 6, we discuss ordering schemes for Trotter-Suzuki approximants, and the search for such a scheme that minimises the error incurred in their use.

# 2 Theoretical Overview

The ultimate aim is to solve the electronic structure theory problem for a specified molecule or chemical system – that is, to find the eigenstates and energies of the electronic Hamiltonian. As discussed above, we take a second quantised approach, to avoid the large qubit requirements of maintaining a real space representation of the wavefunction. In order to achieve this, we require a set of basis single-particle spin-orbitals. The states in which these are occupied according to a set of occupation numbers $o_i$ then form a Fock space, which represents the space of possible electronic wavefunctions.

The electronic Hamiltonian can then be described using creation and annihilation operators acting upon this space. Specifically, provided such a basis, the electronic Hamiltonian is given by:

$$\hat{H} = \sum_{i,j} h_{ij} a_i^\dagger a_j + \frac{1}{2} \sum_{i,j,k,l} h_{ijkl} a_i^\dagger a_j^\dagger a_k a_l \,. \tag{2.0.0.1}$$

Here, the first summation corresponds to one electron terms, with the second summation corresponding to the two electron interaction terms. $a_i^\dagger$ and $a_i$

are creation and annihilation operators acting on the basis one particle spin-orbitals. $h_{ij}$ and $h_{ijlk}$ are integrals over real space, discussed in Section 2.2.

The problem is thus to find the eigenstates and eigenvalues of this Hamiltonian. This task involves several steps, summarised by Figure 2.1. In both quantum and classical computational methodologies, an appropriate basis set must be chosen to describe the electronic states. We discuss this task in Section 2.1.

Following this, the $h_{ij}$ and $h_{ijkl}$ integrals of eqn. (2.0.0.1) must be generated. In principle, eqn. (2.0.0.1) suggests that the number of these scales quartically with the number of spin-orbitals. Although potentially expensive, this scaling can be reduced by consideration of symmetry. Regardless, as it is polynomial, this step is considered classically tractable and, in the current scheme, treated as a classically computable preprocessing step. We remark on a number of subtleties which must be considered in Section 2.2. Having generated the required integrals, a full configuration interaction calculation on a classical computer is a matter of forming a matrix representation of the Hamiltonian and diagonalising it to find eigenstates and eigenvalues. While this is conceptually simple, the exponential growth of the size of the Fock space makes this intractable for systems of any reasonable size.

On a quantum device, as a $2^N$ dimensional Fock space can be represented using $N$ qubits, we do not encounter this problem. Here, we must first find a representation of the states upon which the Hamiltonian acts in terms of

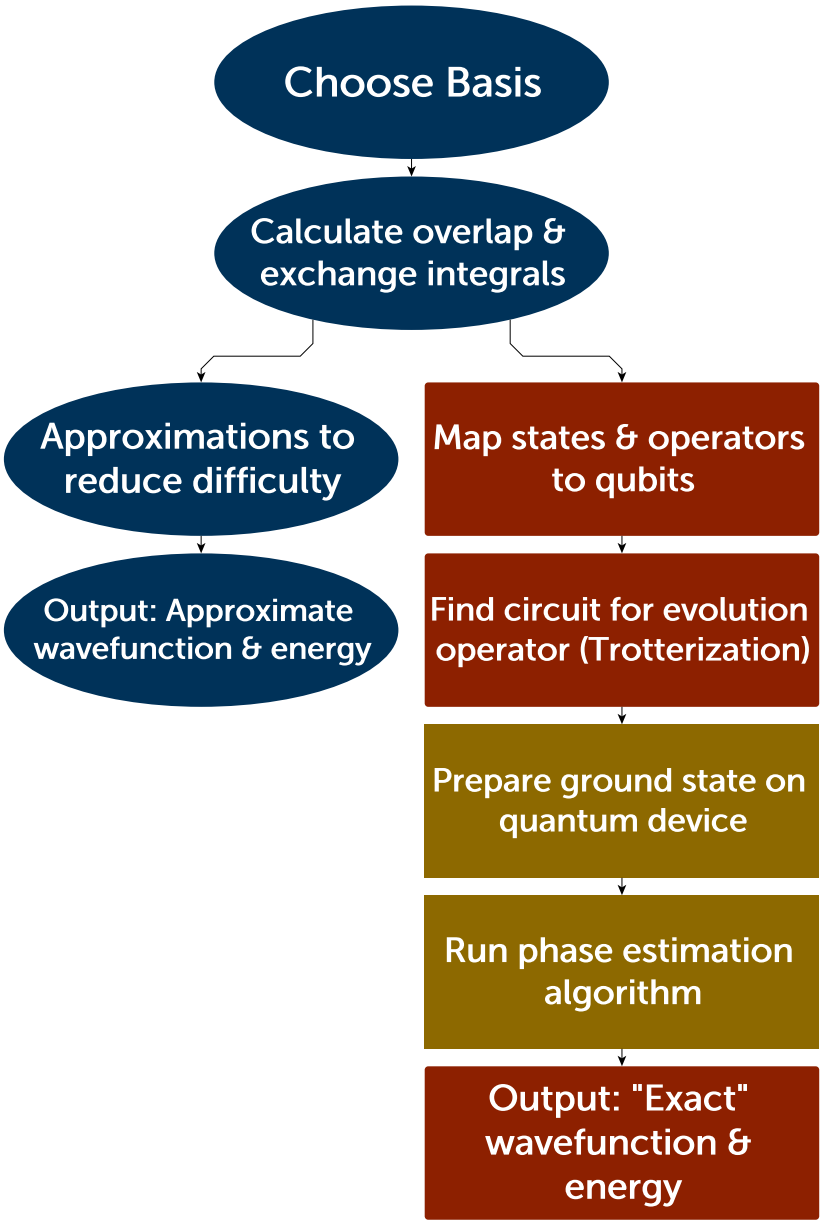



FIGURE 2.1.:
Procedures involved in a electronic structure calculation. The left hand side depicts a calculation on a classical computer, the right hand side depicts a calculation on a quantum device. Blue ovals denote steps performed on a classical computer. Red rounded rectangles denote preprocessing steps in the quantum algorithm performed on a classical computer. Yellow rectangles denote steps performed on the quantum device.

states of qubits. Similarly, we must find a representation of the electronic creation and annihilation operators, in terms of operators acting on qubits. The canonical scheme for this mapping procedure is the *Jordan-Wigner transformation*, which we review in Section 2.3. An alternative is the *Bravyi-Kitaev transformation*, which we discuss in Chapter 4. Having obtained a representation of electronic states and the component operators of the electronic Hamiltonian in terms of states of and circuits acting upon qubits, a circuit which simulates the action of the Hamiltonian can be generated.

In order to obtain ground state energies from this qubit Hamiltonian, a phase estimation algorithm can be performed in order to determine the eigenvalue and thus the electronic energy. However, in order to do this, a circuit describing the unitary evolution operator of the Hamiltonian must be found. In the canonical scheme, this involves the use of a Trotter-Suzuki approximation. We discuss this in Section 2.4.

Thus far, the procedure has been entirely classical, intended to describe the steps which must be taken on a quantum device. The ground state of the system would now be prepared on a quantum computer, as described in Section 2.5.1. While on a classical computer this would require knowledge of the ground state itself (and thus be self-defeating), a number of methods are available to circumvent this requirement on a quantum computer.

Finally, with a circuit that describes the Trotterized evolution operator of the system, and the ground state prepared on the quantum device, we can implement a phase estimation algorithm (considered in Section 2.5) to obtain

the ground state energy of the system.

Each of these algorithmic stages contains potential for optimisation in terms of the quantum resources required. Although this thesis focusses on mapping schemes and Trotterization, it is important to consider these stages within the context of the entire algorithm. As such, we discuss each stage in this process sequentially.

## 2.1 Basis sets

The initial process involved in the algorithm is the choice of basis of spin-orbitals to describe the system. This is a problem very familiar to quantum chemistry performed on classical devices, where the choice of basis set is crucial to appropriately describe the system while minimising computational cost. While the use of plane wave basis functions is common in condensed matter physics and when using a density functional approach[83], for *ab initio* methods in chemistry these most commonly take the form of approximations to atomic orbitals, as molecular orbitals can be well described by a linear combination of such.

However, while an exact analytic form for such atomic orbitals can be obtained for hydrogen-like atoms, finding a description of them for other atoms is non-trivial. One approach would be to numerically diagonalise the lone atomic electronic Hamiltonian over a real space grid, and use a subset of the

resulting numerical atomic orbitals to serve as a basis for further calculations. However, while this approach is taken in some cases[84], an approach which maintains an analytic description is typically preferred, as this substantially reduces computational cost.

Physically, it is intuitive to choose a basis set which replicates the structure of analytic hydrogen-like atomic orbitals. The class of orbitals known as *Slater-type orbitals* can be chosen for this task. The 1*s* orbital in this scheme is given by

$$\chi_{1s}\left(\zeta, \mathrm{r} - \mathrm{R}_A\right) = \left(\frac{\zeta^3}{\pi}\right)^{1/2} \exp\left(-\zeta\left|\mathrm{r} - \mathrm{R}_A\right|\right) , \tag{2.1.0.1}$$

where $\mathrm{R}_A$ is the nuclear position and $\zeta$ is the *Slater orbital exponent*, related to the effective post-screening charge of the nucleus. Functionally, this replicates the form of the hydrogenic 1*s* orbital. Higher orbitals are a product of radial functions of this form and the appropriate spherical harmonic. While we do not consider examples of higher orbitals here, it should be noted that these orbitals exhibit exponential decay at large distances from the nucleus, as required. Being intended to replicate the form of the (analytically derivable) hydrogenic orbitals, these orbitals arguably provide the most physical analytic choice of basis, and in an ideal world would be preferred in most cases. However, the integrals that give the coefficients of the electronic Hamiltonian cannot be computed analytically for these hydrogenic orbitals.

As such, the most commonly encountered form of atomic orbitals are the *Gaussian-type orbitals*. These use Gaussian functions as their primary form.

Specifically, in contrast to eqn. (2.1.0.1), the Gaussian-type 1*s* orbital is given by

$$\chi_{1s}(\alpha, \mathrm{r} - \mathrm{R}_A) = \left(\frac{2\alpha}{\pi}\right)^{3/4} \exp\left(-\alpha \left|\mathrm{r} - \mathrm{R}_A\right|^2\right) , \qquad (2.1.0.2)$$

where $\alpha$ is the *Gaussian orbital exponent*, analogous to the Slater orbital exponent. Similar to Slater-type orbitals, angular components simply consist of spherical harmonics. These differ from the Slater-type orbitals in two important ways. Firstly, they have zero slope at the nuclear centre, leading to a poor description at low radial distances. Secondly, they decay substantially more rapidly at large distance from the nuclear centre. These two factors result in a weaker description of the electronic wavefunction than through the use of Slater-type orbitals. However, as they consist of Gaussian functions, a variety of useful identities exist to simplify the calculation of the necessary integrals. This dramatically reduces the computational cost of simulation.

Rather than using these orbitals directly, it is instead typical that a linear combination of Gaussian-type orbitals are used. By tweaking the parameters and coefficients of each orbital, a functional form can be generated which better approximates the physical reality. Crucially, these parameters are fixed in advance of the desired electronic structure calculation. This allows for physically grounded atomic orbitals to be used without sacrificing the computational advantage of using Gaussian functions.

Assigning a certain number of these composite Gaussian functions to each atom generates a set of basis functions, a linear combination of which will describe the molecular orbitals of the system. Generally, the more basis func-

tions that are used, the more accurate a simulation will be, as there is greater flexibility for the optimisation procedure. However, this comes at a commensurate cost that increases with the number of functions.

Many types of basis sets that include various types of contracted Gaussian-type orbitals have been developed. A minimal example is the STO-nG family, whereby n Gaussian primitives are used to approximate a single Slater-type orbital. Each atom is assigned a number of these pseudo-Slater-type orbitals corresponding to the number of occupied orbitals in the noble gas atom corresponding to the period of the atom in question – one for H and He, five for Li to Ne, nine for Na to Ar, and so on. These orbitals are purely spatial, with no spin component, and thus can be doubly occupied. As an example, the study of $CH_4$ using an STO-3G basis involves nine pseudo-Slater-type orbitals (five from the carbon atom and one from each hydrogen atom), each comprised of a linear combination of three Gaussian primitives, with fixed coefficients. Although these minimal basis sets are typically too small to yield numerical results sufficiently accurate for chemical purposes, we use them extensively in the course of this project. This is to minimise the computational cost of the procedures, which would otherwise rapidly become infeasible. As the focus here is on characterising the cost of quantum algorithms, rather than obtaining novel quantitative computational chemistry predictions, this trade-off is not considered to be a substantial factor.

In molecular systems, core orbitals are typically expected to closely resemble atomic orbitals – more so than valence orbitals. As such, when using larger basis sets, it is common to assign more emphasis to valence orbitals than

to core orbitals. For example, Pople basis sets[85] are designated according to the code $i - jkG$. Each integer corresponds to one pseudo-Slater-type orbital, with the number of integers before the hyphen denoting the number representing the core orbitals, with the number of integers after the hyphen denoting the number representing the valence orbitals. The value of the integer denotes the number of Gaussian primitives that comprise each pseudo-Slater-type orbital. For example, the 3-21G basis assigns one pseudo-Slater-type orbital to each core orbital, comprised of three Gaussian primitives, and two pseudo-Slater-type orbitals to each valence orbital, with two Gaussian primitives for the first, and only one Gaussian primitive for the second.

These classes of basis sets comprise the entirety of those used in this project. Computational chemistry studies typically use larger basis sets such as the correlation-consistent sets of Dunning[86], wherein the orbital parameters are optimised at a post-Hartree-Fock level of theory. However, due to their greater computational expense of use we do not consider these here.

It should be noted that as the formed orbitals are atomic-centred, they are non-orthogonal. This is not a problem for many quantum chemistry methods performed on classical computers, but it precludes the use of quantum simulation. A variety of strategies can be used here. Löwdin symmetric orthogonalization[87] yields the orthonormal set of functions that are closest to (i.e. maximise the overlap with) the original set of basis atomic orbitals. These have been considered within the context of quantum simulation with impressive results, demonstrating a reduction in the number of significant integrals necessary.[88] Although one of the data sets used in this thesis in-

cludes integrals calculated in this basis; as the exploration of basis set choices was not the focus in this project, we do not consider this in detail here.

On a classical device, at this stage the $h_{ij}$ and $h_{ijkl}$ integrals of the Hamiltonian (eqn. (2.0.0.1)) could be calculated. Subsequently, the Hamiltonian could be formed and diagonalised. However, it is typically more efficient to first perform a calculation at an inexpensive level of theory. This has several advantages. Firstly, the set of molecular orbitals – or a manipulation of them – can serve as a new orthonormal basis to describe the electronic wavefunction. For systems where the true ground state is expected to have substantial overlap with the Hartree-Fock ground state, this provides a simple initial "guess" for the eigenstate. Additionally, performing preliminary calculations at a low level of theory allows for granular control of where computational resources are spent. For example, it is common to optimise molecular geometries – a relatively expensive process – at a Hartree-Fock level before using more expensive levels of theory for further calculations. Due to its relative computational ease, when performing a preliminary calculation, the Hartree-Fock process is typically used. For context we briefly review this process in more detail here.

### 2.1.1 *Hartree-Fock process*

The Hartree-Fock method is one of the least computationally expensive *ab initio* quantum chemical methods available.[89,90] Having specified a molecular geometry, a basis of atomic orbitals is generated to describe the system

as discussed above. The overall electronic wavefunction is treated as a Slater determinant of molecular spin-orbitals, themselves a linear combination

$$\psi_i(\mathrm{r}_i) = \sum_j c_{ij}\chi_J(\mathrm{r}_i) \tag{2.1.1.1}$$

of the atomic basis orbitals $\chi_j$ with expansion coefficients $c_{ij}$. Invoking the variational principle, the (Hartree-Fock) molecular orbitals will be the linear combination that minimises the expectation value of the electronic Hamiltonian. To solve this computationally, an initial Ansatz set of coefficients $c_{ij}$ are generated. These can be chosen through a wide range of strategies, but are constrained such that the molecular orbitals are orthonormal.

Following this, the mean-field approximation is invoked, with the two-electron interactions treated as the interaction of each electron with a static electric field. This allows for an effective Hamiltonian to be described for the system, and thus the energy of the Ansatz state obtained. By variationally minimising this energy, we obtain the Hartree-Fock equations

$$\hat{F}\phi_i = \varepsilon_i\phi_i\,, \tag{2.1.1.2}$$

where $\phi_i$ are a set of new linear combinations of the original atomic basis orbitals (which form a set of one-electron molecular orbitals) and $F$ is the Fock operator, the matrix representation of which is efficiently generatable and diagonalizable. Consequentially, diagonalizing the Fock matrix yields a new set of molecular orbitals for the system.

The initial Fock matrix depends on the initial Ansatz state chosen. As such, a new Fock matrix can be generated from the new molecular orbitals, and the process repeated. This entire sequence is repeated until self-consistency is achieved. For this reason, the method is often referred to as the *self-consistent field* method.

Once self-consistency is achieved, the final molecular orbitals – which diagonalise the final Fock matrix – are the Hartree-Fock molecular orbitals, often referred to as the canonical molecular orbitals. Chemically the eigenvalues can be related to (for example) ionisation energies through Koopman's theorem[91], which states that the first ionisation energy of a system is (approximately) equal to the eigenvalue of the highest occupied molecular orbital.

As this technique neglects electron correlation other than to the extent required by the Pauli principle, it is rarely considered to yield quantitatively accurate results. However, it is relatively computationally cheap, scaling somewhere between $O(N^3)$ and $O(N^4)$[92]. As such, it is typically used as a starting point for further, more accurate methods. As discussed above, the quantum algorithm – along with many classical computational methods – cannot use a standard basis set of atomic orbitals to describe space, as these are typically non-orthogonal. As such, the Hartree-Fock molecular orbitals are often used as a basis in which to perform further calculations, as in the case with the majority of the analysis described in this thesis. In addition to their orthonormality, the true electronic ground state often displays significant overlap with the Hartree-Fock ground state. The advantage in this is twofold: firstly, in these cases, there is an obvious physically relevant "first guess" for more ad-

vanced approaches – the Hartree-Fock ground state. Additionally, in this basis, said first guess will be a Fock state given by the occupation numbers of the $N_e$ lowest energy orbitals being 1, where $N_e$ is the number of electrons. This aids in state preparation, as discussed in Section 2.5.1.

These are not the only possible choice of orthonormal basis set, although they do form the principal choice in this thesis. As discussed above, Löwdin orthogonalised atomic orbitals are an orthonormalised set of localised orbitals which maximise overlap with the original atomic orbitals. The SMALL_DATA set (discussed in Chapter 3) also contains calculations performed in a basis of natural orbitals. These are the orbitals which diagonalise the one-electron reduced density matrices calculated from a post-Hartree-Fock (although still tractable) level of theory. Although other work has considered in detail the use of these types of basis for the quantum simulation of quantum chemistry[88,93],this was not the focus of this project, and thus we do not perform such an analysis here.

It should be noted that many of the typical use cases for a quantum approach to quantum chemistry will be in systems which are not adequately described by the Hartree-Fock process, or by more advanced classical computational techniques. Much work has been performed to assess whether an alternative basis would be desirable in these cases.[88,93] Regardless, many systems exist for which classical computational methods yield a reasonable accuracy, but still with a clear benefit to be gained from the use of a full configuration interaction level approach. It is in these cases that a quantum approach would see the most value.

## 2.2 Integrals

In order to represent the Hamiltonian, the $h_{ij}$ and $h_{ijkl}$ integrals present in the molecular Hamiltonian (eqn. (2.0.0.1)) must first be determined. In this thesis, we use the notation of Whitfield, Biamonte and Aspuru-Guzik[94]. The one electron integrals are given by

$$h_{ij} = \int d\mathrm{x} \chi_i^*(\mathrm{x}) \left( -\frac{1}{2}\nabla^2 - \sum_\alpha \frac{Z_\alpha}{r_{\alpha,\mathrm{x}}} \right) \chi_j(\mathrm{x}) \,, \tag{2.2.0.1}$$

with the two-electron integrals being given by

$$h_{ijkl} = \int d\mathrm{x}_1 d\mathrm{x}_2 \frac{\chi_i^*(\mathrm{x}_1)\chi_j^*(\mathrm{x}_2)\chi_k(\mathrm{x}_2)\chi_l(\mathrm{x}_1)}{r_{1,2}} \,. \tag{2.2.0.2}$$

Here, $\alpha$ indexes the nuclei, $Z_\alpha$ is the nuclear charge, $r_{\alpha,x}$ is the distance between the electron and nucleus, $r_{1,2}$ is the distance between the two electrons and $\chi(\mathrm{x})$ is the spin-orbit position representation wavefunction for a given basis state.

Crucially, each integral can be classically computed efficiently, particularly when using a spatial basis derived from Gaussian functions. They are, however, numerous. At first inspection, the number of terms in the electronic Hamiltonian – and thus, the number of integrals – would appear to scale quartically with the number of spin-orbitals considered. To an extent, this is not a significant problem for a quantum algorithm. While a quartic scaling is

significant in comparison to linear cost methods, it is small in comparison to high accuracy classical computational methods. As the goal of the quantum algorithm is to obtain highly accurate (i.e. to the level of full configuration interaction) results, it is presumed that sufficient classical computing capacity would be available for this task.

Furthermore, much symmetry is notable in the structure of these integrals. If we assume that the basis orbitals are real (as is typically constructed for ease of calculation), from inspection we have that

$$\begin{aligned} h_{ijkl} &= h_{jilk}\,, \\ h_{ijkl} &= h_{klji}\,, \end{aligned} \tag{2.2.0.3}$$

and thus we have an eightfold symmetry

$$h_{ijkl} = h_{jilk} = h_{lkji} = h_{klij} = h_{klji} = h_{lkij} = h_{ijlk} = h_{jikl}\,, \tag{2.2.0.4}$$

which cuts down the number of unique integrals required by a factor of 8.

Similarly, symmetries of the molecular geometry can be invoked here in order to reduce the number of integrals required. These methods are well-studied and used in many traditional quantum chemistry packages to reduce computational cost. The crucial point here is that there are no inherent scaling difficulties with respect to the generation of integrals, and that they may essentially be treated as classical input into the quantum algorithm.

## 2.3 Mapping electrons to qubits

In generating a quantum circuit to simulate the electronic Hamiltonian, the first task is to translate the electronic basis states into states of qubits. In doing so, we must also find an appropriate representation of the operators that constitute the electronic Hamiltonian - i.e. the creation and annihilation operators acting on the molecular orbital spaces. The canonical method of performing this task is the *Jordan-Wigner transformation*. Other constructions have been examined.[78–80,82,95] In particular, we consider the *Bravyi-Kitaev transformation* in Chapter 4 and 5.

The initial premise of the Jordan-Wigner transformation is simple. For an analysis of $N$ spin-orbitals, we use a system of $N$ qubits. The qubits are used to store the occupation number of each spin-orbital. As electrons are fermions, each spin-orbital can only be singly occupied, thus occupation numbers can only be zero or one. In this scheme, if spin-orbital $j$ is unoccupied (i.e. has an occupation number of 0), then its corresponding qubit is in the $|0\rangle$ state. If spin-orbital $j$ is occupied (i.e. has an occupation number of 1), then its corresponding qubit is in the $|1\rangle$ state. Collectively, this generates a basis that spans the combined space of all qubits. For example, in a simulation consisting of four spin-orbitals, the state where the first and third orbitals are occupied is represented by the $|1010\rangle$ state. This basis is called the *occupation basis*.[96]

From a chemistry perspective, it is important to note that the qubit Fock

space represents all possible electronic configurations, with no consideration given to antisymmetrisation or even fixed electron number. This is in contrast to classical computational approaches in quantum chemistry, where a reduction in computational cost is often achieved by restricting the space to only that with the physically appropriate number of electrons, and antisymmetrisation is enforced through the use of Slater determinants. Here, antisymmetrisation is instead enforced through the anti-commutation relations of the creation and annihilation operations, discussed below. Quantum approaches involving restricting the subspace of study have been examined, although we do not consider these here.[72,73] Note that as all terms in the electronic Hamiltonian (eqn. (2.0.0.1)) commute with the number operator (and thus the total electronic Hamiltonian commutes with the number operator), conservation of electron number is implied, and thus eigenstates cannot be combinations of states with variable particle number. As the Hamiltonian represents the actual physics of the molecule, the ground state will correspond to the correct particle number, without the need for its explicit specification.*

For reasons that will become apparent in comparison to other mapping schemes (in Chapter 4), it is instructive to consider the matrix that transforms a vector of occupation numbers f to a corresponding vector of states of qubits.

*It should, however, be noted that with small basis sets, the basis set error can lead to unexpected results. For example, using an STO-3G basis, the aluminium ion is predicted to be *more stable than the aluminium atom*, and thus the ground state of the aluminium atom is predicted to have 12 electrons. For the purposes of morale, it is important to bear this in mind *prior* to testing numerical results against reference full configuration interaction calculations.

In the case of eight qubits this matrix is given by

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \end{bmatrix} = \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \end{bmatrix} . \tag{2.3.0.1}$$

Here, the vector on the left hand side is a vector representing the occupation numbers of spin-orbitals ($f_j$ is the occupation number of the $j$th spin-orbital), and the vector on the right hand side is a vector representing the states of the qubits. The matrix that transforms the former to the latter is then the identity matrix of dimension $N$.

### 2.3.1 *Jordan-Wigner operators*

Having found a description of electronic states in terms of states of qubits, the next task is to represent the electronic creation and annihilation operators as operators on the qubit space - i.e. the $a_j^\dagger$ and $a_j$ of eqn. (2.0.0.1). As stated above, in contrast to traditional quantum chemistry approaches where antisymmetrisation of the final electronic states is enforced through restricting the available state space, here we instead enforce this requirement

through the anti-commutation relations of the fermionic operators. Where $i$ and $j$ are different, we note that $a_i a_j = -a_j a_i$, $a_i^\dagger a_j = -a_j a_i^\dagger$, and so on, as necessary to introduce the required sign flip for anti-symmetrisation. Setting $i = j$ results in $a_i a_i = -a_i a_i$ and thus $a_i a_i = a_i^\dagger a_i^\dagger = 0$, recovering the Pauli exclusion principle, requiring that the spin-orbitals can only be singly occupied. It then follows that $\left(a_i a_i^\dagger + a_i^\dagger a_i\right)$ acting on a state will resolve to the identity, as one term will be immediately resolve to 0 (due to either attempting to create an already existing state or annihilate the vacuum), and the other creates and immediately annihilates a state (or vice versa). These relations can be more elegantly written as

$$\begin{aligned} \left\{a_i, a_j\right\} &= \left\{a_i^\dagger, a_j^\dagger\right\} = 0\,, \\ \left\{a_i, a_j^\dagger\right\} &= \delta_{ij} I\,. \end{aligned} \tag{2.3.1.1}$$

For the quantum simulation to accurately represent the physics of the fermionic system, any representation of the fermionic creation and annihilation operations must maintain these relations. In order to represent their actual creation and annihilation operation, they must clearly also implement the appropriate action

$$\begin{aligned} \hat{Q}^+ |0\rangle &= |1\rangle & \hat{Q}^+ |1\rangle &= 0 \\ \hat{Q}^- |0\rangle &= 0 & \hat{Q}^- |1\rangle &= |0\rangle \end{aligned}\,. \tag{2.3.1.2}$$

At first inspection, the standard Pauli plus and minus operators

$$\hat{Q}^{+} = |1\rangle\langle 0| = \frac{1}{2}(X - iY) ,$$
$$\hat{Q}^{-} = |0\rangle\langle 1| = \frac{1}{2}(X + iY) \tag{2.3.1.3}$$

should suffice. However, these only act on the specific qubit which is being created or annihilated. As such, these operators corresponding to different qubits must commute. As a consequence of this they do not fulfil the anti-commutation relations, eqn. (2.3.1.1).

To resolve this, an example is instructive. In a six spin-orbital system, consider the action of $a_3$ on the state with orbitals 0,1, 3 and 4 occupied. The operator $a_3$ can be sequentially commuted through, introducing a sign flip at each step according to the anti-commutation relations

$$\begin{aligned} a_3\left(a_0^\dagger a_1^\dagger a_3^\dagger a_4^\dagger |vac\rangle\right) &= -a_0^\dagger a_3 a_1^\dagger a_3^\dagger a_4^\dagger |vac\rangle \\ &= a_0^\dagger a_1^\dagger a_3 a_3^\dagger a_4^\dagger |vac\rangle = a_0^\dagger a_1^\dagger a_4^\dagger |vac\rangle , \end{aligned} \tag{2.3.1.4}$$

where the final line has implemented the required annihilation operation. It is thus apparent that the practical impact of the imposition of the anti-commutation relations is to impose a sign flip depending on whether the sum of the occupation numbers of the orbitals with lower index is even (no sign flip) or odd (sign flip). On a quantum device, this behaviour can be emulated by adding a string of Pauli $Z$ operations to all qubits with index

lower than $j$. As such, we have

$$a_j^\dagger \to \frac{1}{2}\left(X_j - iY_j\right) \otimes Z^{\otimes j}\,,$$
$$a_j \to \frac{1}{2}\left(X_j + iY_j\right) \otimes Z^{\otimes j}\,, \tag{2.3.1.5}$$

the Jordan-Wigner transformation.[97] By taking products of these, a Hamiltonian acting on qubits can now be generated, which is equivalent to the electronic Hamiltonian. After some algebra, the form of this qubit Hamiltonian can be reduced to a weighted sum of strings of Pauli operations. It is useful here to consider an example. For molecular Hydrogen using an STO-3G basis set, the above process yields the Hamiltonian

$$\begin{aligned}\hat{H} = &- 0.81261I + 0.17120\sigma_0^z + 0.17120\sigma_1^z - 0.22280\sigma_2^z \\ &- 0.22280\sigma_3^z + 0.16862\sigma_1^z\sigma_0^z + 0.12055\sigma_2^z\sigma_0^z + 0.16587\sigma_3^z\sigma_0^z \\ &+ 0.16587\sigma_2^z\sigma_1^z + 0.12055\sigma_3^z\sigma_1^z + 0.17435\sigma_3^z\sigma_2^z \\ &- 0.04532\sigma_3^y\sigma_2^y\sigma_1^x\sigma_0^x + 0.04532\sigma_3^x\sigma_2^y\sigma_1^y\sigma_0^x \\ &+ 0.04532\sigma_3^y\sigma_2^x\sigma_1^x\sigma_0^y - 0.04532\sigma_3^x\sigma_2^x\sigma_1^y\sigma_0^y\end{aligned} \tag{2.3.1.6}$$

with the exact values of the weighting constants being dependent on the bond length chosen. Here, the superscript on each $\sigma$ operator determines the type of Pauli operation to be applied, and the numerical subscript indicates which qubit the operation is applied to. The crucial fact here is that the individual terms in this Hamiltonian are guaranteed to be strings of Pauli operations.

While this scheme allows for the simulation of an $N$ spin-orbital system using only $N$ qubits (before consideration of utility qubits for phase estimation, error correction and so on), each annihilation or creation operator applied to orbital $i$ requires a gate being performed on all qubits of index $< i$. As such, $O(n)$ gates are required to simulate an operation on a single orbital. While this linear scaling does not appear problematic, it must be remembered that this is true of every electronic operation in the Hamiltonian – of which there are (at most) $O(N^4)$. As such, it is desirable to reduce this. We consider schemes aimed at this task in Chapter 4.

## 2.4 Trotterization

The above process has allowed us to generate a representation of the molecular Hamiltonian in terms of Pauli operators. As indicated by Figure 2.4, the next stage is to find a quantum circuit that simulates evolution under the Hamiltonian. In other words, we require a circuit that approximates

$$\hat{U} = e^{-i\hat{H}t}, \tag{2.4.0.1}$$

the evolution operator for a given propagation time $t$. Recall that the qubit Hamiltonian consists of a weighted sum of strings of Pauli operations. Standard quantum circuits exist for the exponentiated form for an individual string, as considered in Section 2.4.1. However, it is clear that in general, the strings do not mutually commute. As such, exponentiating the total Hamiltonian

is non-trivial. The canonical method for this is through the use of a Trotter-Suzuki approximation to find a *Trotterized* unitary. Intuitively, this method can be thought of as applying the unitary corresponding to each term in sequence, rapidly switching between the terms. The overall evolution time is broken into *Trotter steps*, thereby controlling the number of times each term is simulated. With a sufficiently large number of Trotter steps (that is, sufficiently rapid alternation between the simulation of terms), an approximation to the true evolution operator of the system can be obtained.

In order to formalise this, we first consider the Lie-Trotter formula

$$e^{A+B} = \lim_{N_{TS}\to\infty} \left( e^{A/N_{TS}} e^{B/N_{TS}} \right)^{N_{TS}}. \tag{2.4.0.2}$$

By sequentially breaking down the Hamiltonian and applying this formula, we can generalise to more than two terms

$$e^{-i\hat{H}t} = \lim_{N_{TS}\to\infty} \left( \prod_{k=1}^{m} e^{-i\hat{H}_k t/N_{TS}} \right)^{N_{TS}}, \tag{2.4.0.3}$$

where $m$ is the number of terms in the Hamiltonian. Generalising this to more than two terms and by sequentially breaking down the total Hamiltonian into doublets, for a high enough number of time steps $n$ the following approximation holds[98]:

$$e^{-i\hat{H}t} \approx \left( \prod_{k=1}^{m} e^{-i\hat{H}_k t/N_{TS}} \right)^{N_{TS}}. \tag{2.4.0.4}$$

Where the Hamiltonian $\hat{H}$ is a sum of $m$ terms $\hat{H}_k$. Intuitively, this states that the Hamiltonian can be approximated by rapidly switching between each term, across the desired evolution time. The exponential of a single Hamiltonian term can be described easily, as discussed below. Increasing the number of Trotter steps ($N_{TS}$) deterministically reduces the error incurred by this approximation, but with commensurate increase in circuit length.

A second order Trotter-Suzuki approximant

$$e^{-i\hat{H}t} \approx \left( \prod_{k=1}^{m} e^{-i\hat{H}_k t/2N_{TS}} \prod_{k=m}^{1} e^{-i\hat{H}_k t/2N_{TS}} \right)^{N_{TS}} \tag{2.4.0.5}$$

can be obtained by symmetrising the first order approximant. Higher order terms can be built recursively[98]. By collecting constants into a generic constant $x$, eqn. (2.4.0.5) can be written as

$$e^{-i\hat{H}t} \approx S_2(x)^{N_{TS}}\,, \tag{2.4.0.6}$$

where

$$S_2(x) = \left( \prod_{k=1}^{m} e^{\hat{H}_k x} \right) \left( \prod_{k=m}^{1} e^{\hat{H}_k x} \right)\,. \tag{2.4.0.7}$$

Higher order approximants are then given by[99]

$$\begin{aligned} S_4(x) &= S_2(z_2 x)^2 S_2((1-4z_2)x) S_2(z_2 x)^2\,, \\ S_{2k}(x) &= S_k(z_k x)^2 S_k((1-4z_k)x) S_2(z_k x)^2\,. \end{aligned} \tag{2.4.0.8}$$

Increasing the Trotter order ($2k$) will yield reductions in error. However, the overall number of gates needed to simulate higher order Trotter approximants increases substantially. The gate cost of the second order approximant is double that of the first order approximant. An inspection of eqn. (2.4.0.8) shows that the fourth order formula increases it tenfold. Choosing an appropriate strategy for simulation is then a three-way trade-off between the order of Trotter-Suzuki approximant, the number of time steps, and the need to minimise the overall circuit length.

Although equivalent to the solution of the original eigenvalue problem (and thus intractable for large system sizes), for our analysis, we can gauge the quality of a given Trotter-Suzuki approximation by comparing the effect on an eigenstate. If a true evolution operator $\hat{U}$ has the eigenvalue relationship

$$\hat{U} |\psi\rangle = e^{-iE_\psi t} |\psi\rangle \,, \tag{2.4.0.9}$$

then an approximate operator $\hat{U}'$ will cause the evolution

$$\hat{U}' |\psi\rangle = e^{-iE'_\psi t} |\psi'\rangle \,, \tag{2.4.0.10}$$

and thus:

$$\langle\psi| \hat{U}' |\psi\rangle = e^{-iE'_\psi t} \langle\psi|\psi'\rangle \,, \tag{2.4.0.11}$$

$$\arg\left(\frac{\langle\psi| \hat{U}' |\psi\rangle}{|\langle\psi|\psi'\rangle|}\right) = -E'_\psi t \,. \tag{2.4.0.12}$$

This gives us a Trotterized estimate of the eigenvalue, $E'_\psi$. Comparing this

against the true eigenvalue gives the *Trotter error*. Analysis of the factors influencing this Trotter error forms a substantial part of the discussion of Chapter 6.

The eigenvalue estimate is always yielded as a phase. Consequentially, to avoid aliasing, care must be taken to ensure that $E'_{\downarrow} t$ is within the interval of 0 and $2\pi$. The overall evolution time must be tailored such that this is the case. This requires a rough estimate of the eigenvalue, for which the Hartree-Fock energy is typically sufficient.

The error introduced by the use of Trotter-Suzuki formulae is dependent on the number of Trotter steps used, but also (perhaps unexpectedly) the order in which terms are simulated. This fact motivates the search for ordering schemes that reduce Trotter error, and is the subject of Chapter 6.

### 2.4.1 *Circuit description*

Having obtained an analytic form of an approximate evolution operator, the next step is to describe it in terms of a quantum circuit. Fortunately, circuits that perform this task are well known.[36] Again, note that each individual term in eqn. (2.4.0.5) consists of an exponentiated weighted string of Pauli operations. First, consider the instance where the Pauli string consists of one Pauli Z operation, with the rest of the string consistent of identity operators.

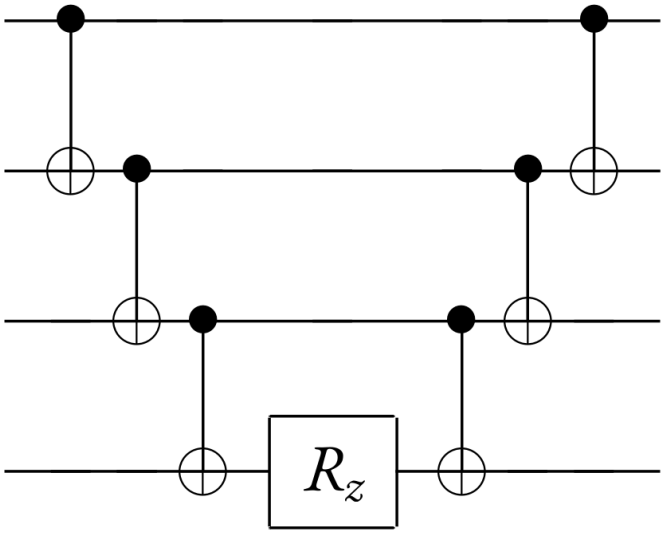



Figure 2.2.:
A circuit to exponentiate a string of Pauli $Z$ operators.

Here, the exponentiated form of this

$$R_z(\vartheta) = e^{\frac{-i\vartheta Z}{2}} = \begin{bmatrix} e^{\frac{-i\vartheta}{2}} & 0 \\ 0 & e^{\frac{i\vartheta}{2}} \end{bmatrix} \tag{2.4.1.1}$$

is a rotation of an angle given by the weighting constant. As such, we can effect such a term with one $R_z$ gate applied to the qubit to which the Pauli Z is applied to. This can be extended to a string of Pauli Z operations by observing that in this case, there are only two possible outcomes. Where the parity of states of the relevant qubits is even, the individual operations will cancel out, whereas if the parity is odd, they will not.

The overall action can be performed in three steps: the overall parity is com-

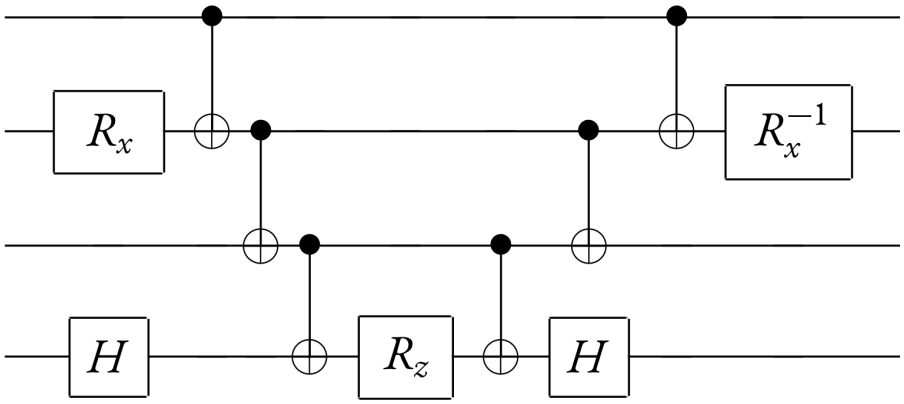



Figure 2.3.:
A circuit to exponentiate a string of general Pauli operators – specifically, in this case, $Z_0 Y_1 Z_2 X_3$

puted using a chain of CNOT gates, the appropriate rotation gate is applied to one qubit, and then the parity is decomputed. Figure 2.2 demonstrates this overall circuit.

To extend this to an exponentiated string of arbitrary Pauli operators is relatively simple. The overall procedure is the same, but each qubit is first transformed into the appropriate basis, by the means of $H$ and $R_X$ gates. After performing the decomputation of the CNOT chain, the transformed qubits must similarly be rotated back to the computational basis. A circuit to perform this is shown in Figure 2.3.

As the Trotterized unitary is a sequence of individual $\exp\left(\bigotimes_{n=0}^{N} P_n\right)$ operations, generating a circuit for the simulation of the whole unitary is simply a matter of chaining these individual circuits together. Computationally, determining the overall circuit length is then a matter of counting the number of gates in this composite circuit. Overall, this procedure has allowed the

generation of a circuit which simulates the action of the total unitary evolution operator of the electronic Hamiltonian, as required by quantum phase estimation.

## 2.5 Phase estimation

The above theory has allowed for a description of the unitary evolution operator of the electronic Hamiltonian in terms of a quantum circuit, i.e. $\hat{U}(t)$. From this, we require an algorithm which obtains eigenvalues of the electronic Hamiltonian.

The *quantum phase estimation* algorithm, constructed independently by Kitaev[100] and Abrams and Lloyd[50] is a method of achieving this task. Most relevant for the present task, the latter of these was in the context of simulating a quantum system. Although consideration of the phase estimation algorithm in and of itself was not a focus during this project, we briefly review it here for context.

The quantum phase estimation algorithm is in essence a method for estimating the eigenphase of a unitary operator. In turn, this method obtains the eigenvalues of Hermitian operators. The algorithm requires a controllable black box circuit that implements the correct unitary operator, a register to encode the eigenstate $|\psi\rangle$, and an extra register of $N$ qubits, where $N$ is the number of desired bits of precision. In the present case, the controllable

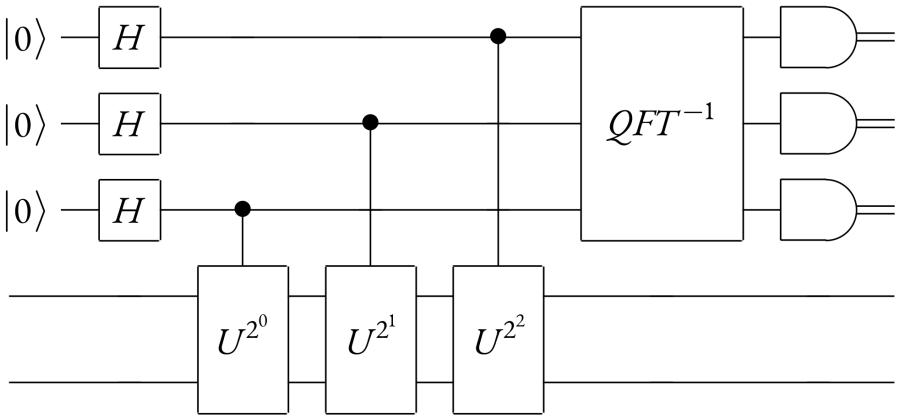



Figure 2.4.:
The phase estimation circuit. The upper register is the phase estimation register, initialised in the $|0\rangle$ state. The lower register stores the qubit representation of the electronic ground state.

black box is provided by the procedures above. The register containing the eigenstate (or a close approximation thereof) must be explicitly prepared. We discuss this task in Section 2.5.1. Figure 2.4 illustrates the entire circuit for a precision of three bits.* In addition to the eigenstate register, the phase estimation register is initially prepared in an equal superposition of all states

$$\frac{1}{\sqrt{2^N}} \sum_{x=0}^{2^N-1} |x\rangle \otimes |\psi\rangle \,, \tag{2.5.0.1}$$

by applying Hadamard gates to all $N$ qubits. Following this, the successive powers of two of the controlled unitary black box are applied to the eigenstate register, according to Figure 2.4. This applies the black box the number of times given by the binary representation of the index of the qubit in the es-

*In an actual experiment, a larger register – and thus a higher precision – would be used. Three bits of precision are used here for the sake of clarity.

timation register for each term. In other words, where we have an eigenvalue relationship

$$\hat{H} |\psi\rangle = E_\psi |\psi\rangle \tag{2.5.0.2}$$

for a Hamiltonian $H$, the unitary black box causes corresponding evolution

$$\hat{U} |\psi\rangle = e^{-iE_\psi t} |\psi\rangle \ , \tag{2.5.0.3}$$

and thus we have the overall post black box state given by

$$\frac{1}{\sqrt{2^N}} \sum_{x=0}^{2^N-1} |x\rangle \otimes \hat{U}^x |\psi\rangle = \frac{1}{\sqrt{2^N}} \sum_{x=0}^{2^N-1} e^{-iE_\psi xt} |x\rangle \otimes |\psi\rangle \, . \tag{2.5.0.4}$$

Finally, the inverse quantum Fourier transform

$$\begin{aligned} &QFT^\dagger \left( \frac{1}{\sqrt{2^N}} \sum_{x=0}^{2^N-1} e^{-iE_\psi xt} |x\rangle \otimes |\psi\rangle \right) \\ &= \frac{1}{2^N} \sum_{x=0}^{2^N-1} \sum_{y=0}^{2^N-1} e^{-\frac{2\pi i}{2^N} x.y} e^{-iE_\psi xt} |y\rangle \otimes |\psi\rangle \end{aligned} \tag{2.5.0.5}$$

is applied. We note that:

$$\sum_{x=0}^{2^N-1} e^{-\frac{2\pi i}{2^N} x.y} e^{-iE_\psi xt} = \sum_{x=0}^{2^N-1} \left( e^{-\frac{2\pi i}{2^N} y - iE_\psi t} \right)^x \, . \tag{2.5.0.6}$$

This can be summed as a geometric series

$$\sum_{x=0}^{2^N-1}\left(e^{-\frac{2\pi i}{2^N}y-iE_\psi t}\right)^x = \frac{1-e^{-2\pi iy-2^N iE_\psi t}}{1-e^{-\frac{2\pi i}{2^N}y-iE_\psi t}} = \frac{1-e^{-2\pi iy}e^{-2^N iE_\psi t}}{1-e^{-\frac{2\pi i}{2^N}y-iE_\psi t}} \quad (2.5.0.7)$$

noting that $y$ is an integer and thus $e^{-2\pi iy}=1$. Combining with eqn. (2.5.0.5), the post-algorithm state is

$$\frac{1}{2^N}\sum_{y=0}^{2^N-1}\frac{1-e^{-2^N iE_\psi t}}{1-e^{-\frac{2\pi i}{2^N}y-iE_\psi t}}|y\rangle\otimes|\psi\rangle\,. \quad (2.5.0.8)$$

When the exponential in the denominator of eqn. (2.5.0.8) approaches 1

$$e^{-\frac{2\pi i}{2^N}y-iE_\psi t}\approx 1\,, \quad (2.5.0.9)$$

the amplitude of the corresponding state is maximised. For this approximation to become an exact equality (and thus yield a somewhat problematic infinite amplitude for a given state), the size of the phase estimation register $N$ would have to be infinite. Thus, we have a maximised amplitude where:

$$E_\psi \approx -\frac{2\pi}{2^N t}y\,. \quad (2.5.0.10)$$

Finally, as indicated in Figure 2.4, the phase estimation register is measured. The most likely outcome of this process yields a binary representation of $E_\psi$. The precision of this representation is limited only by the size of the phase estimation register, although this itself is constrained by experimental diffi-

culty. Due to the increasing powers of the controlled unitary black box, the procedure scales exponentially in gate count with respect to the desired precision. This would be problematic if the required precision scaled with the number of qubits. However, in chemical problems, the required accuracy is fixed, removing this as an impediment.

### 2.5.1 *State preparation*

While the above discussion assumed that the exact ground state $|\psi\rangle$ is prepared, it is easy to show that this is not a strict requirement. Expanding the prepared state $|\Psi\rangle$ in the true eigenbasis

$$|\Psi\rangle = \sum_{i}^{2N-1} c_i \psi_i \tag{2.5.1.1}$$

and performing the above analysis, we note that the same result is probabilistically obtained, with the probability of success being given by $|c_i|^2$. Consequentially, the desired eigenvalue estimate can be efficiently obtained provided that the prepared state $|\Psi\rangle$ has polynomial overlap with the true eigenstate.

Unfortunately, it is not necessarily a given that such a state is preparable. Efficient methods for state preparation are an ongoing subject of research. One approach is to use the Hartree-Fock ground state as the guiding state. In a basis of canonical molecular orbitals, the Hartree-Fock ground state will be a

Fock state. This makes it simple to prepare on most quantum architectures. However, while this is likely to be an acceptable choice in many chemical scenarios, it is known that in some cases (particularly in weakly bound systems where the quantum approach will likely be most useful), the overlap of the Hartree-Fock ground state and the true ground state is exponentially small.

One alternative is to prepare the state adiabatically, as initially proposed by Aspuru-Guzik, Dutoi, Love and Head-Gordon.[51,68,101] Here, the final self-consistent Hartree-Fock Hamiltonian is mapped to quantum operations, and the qubits are prepared in the Hartree-Fock ground state. Initially, the Hartree-Fock Hamiltonian is applied. By linear interpolation, the applied Hamiltonian is gradually adjusted into the full electronic Hamiltonian given by eqn. (2.0.0.1). This is performed sufficiently slowly so as to obey the adiabatic theorem. This ensures that the qubits stay in the ground state of the interpolated Hamiltonian at all times, such that the final state is given by the ground state of the full Hamiltonian as required.

Other approaches have been developed to deal with state preparation. These include those based on truncated configuration interaction performed on a classical computer.[93] For brevity, the details of state preparation is not dwelled upon here.

### 2.5.2 *Iterative phase estimation*

The canonical phase estimation procedure requires a register of $N$ extra qubits in addition to those used to store the electronic state, where the desired level of precision is $1/2^N$. While this would likely not be problematic for large-scale quantum devices, it presents a difficulty for experimental demonstrations of these techniques on currently available small devices. As such, most experimental processes have used the *iterative phase estimation* algorithm.[51,94,102] Although was not the topic of the present investigation, we (briefly and non-rigorously) review this here for context.

The iterative phase estimation algorithm is a modification of the canonical phase estimation algorithm. If we expand the desired phase as a sum of individual bits

$$\phi = \frac{b_0}{2} + \frac{b_1}{4} + \frac{b_2}{8} + \ldots \tag{2.5.2.1}$$

then each $U^{2^i}$ subcircuit in Figure 2.4 is considered to set register qubit $|e_i\rangle$ into the appropriate value of $b_i$ for each bit in the expansion. Thus, whereas the canonical phase estimation procedure stores all bits in their corresponding qubits prior to readout, it would be desirable to instead *sequentially* readout bits, thus requiring only one qubit in the phase estimation register.

A scheme could be envisaged whereby the size of the phase estimation register is set to one, and the first controlled $U^1$ gate applied according to Figure 2.4. After readout, this would yield the first bit of the true phase, $b_0$. Following this, the same procedure could be applied using the $U^2$ gate to achieve

the second bit of precision, and so on. However, the terms in eqn. (2.5.2.1) of less significance than the bit being studied ($b_{>i}$ for the $i$th bit) would result in this process succeeding only probabilistically. As bits are sequentially read out, the overall probability of success thus decays with the desired precision, which is clearly undesirable.

Instead, one can start from the least significant bit. Unless the phase is exactly represented by a binary expansion of up to this level of precision, this will succeed only probabilistically, as before. However, on the assumption that this procedure succeeds (which can be guaranteed by repeated trials), the least significant bit of the phase is now known. Instead of performing canonical phase estimation for the bits of higher significance, a phase shift can be introduced so as to remove this bit from the binary expansion of the true phase, before performing phase estimation to find second least significant bit. As the least significant bit has been removed from eqn. (2.5.2.1), the probability of failure is now vanishingly small.

This sequence can then be repeated until the most significant bit of the phase has been determined, yielding the phase accurate to the desired precision. Crucially, the chance of failure is now independent of the desired precision, thus not requiring an intractable number of repetitions of the procedure.

In this chapter, the theoretical methodology for performing the quantum simulation of quantum chemistry has been discussed. Although all of the steps necessary for this have been discussed, the focus has been on the aspects most relevant for the optimisation schemes analysed in this thesis. For the

sake of clarity, some detail regarding the specifics of these schemes will be covered in Chapters 4 and 6.

# 3 General Implementations

The analysis presented in this thesis required the development of a not insubstantial library of code. This chapter attempts to catalogue the relevant sections of this library. Here, the broad approaches and structures used are discussed. The code is written to allow interface with relevant available packages (in particular, electronic structure packages and the FermiLib package[103]). The bulk of the library was written specifically for this project, with pre-existing code noted below. Similarly, the overwhelming majority of the code was written in Python v2.7.8, with exceptions discussed in their relevant sections.

There are a variety of ultimate intended outcomes when using the library. For example, in many cases the intended goal of a simulation is to determine the length of the quantum circuit necessary to simulate the energetics of a particular molecule. Alternatively, the Trotter error incurred in simulation may be required. Due to the wide range of possible endpoints for a process, the library should be considered as a catalogue of modules, rather than as a single linear process. Nonetheless, as the aim is typically to simulate or characterise a quantum simulation, there is an inherent sequential nature to

how each module is invoked. It is thus helpful to consider each stage of the code in turn.

The modules roughly correlate to the theoretical picture of how quantum simulation of quantum chemistry is performed. An overview of the code is included as Figure 3.1. Firstly, a separate quantum chemistry package is used to generate and store the integrals present in the molecular Hamiltonian. In principle, any quantum chemistry package may be used for this step. Following this, the code translates these integrals into the form used internally, before generating a symbolic representation of a fermionic Hamiltonian (using either the Jordan-Wigner or Bravyi-Kitaev transformations). This can be ordered according to various ordering algorithms. This symbolic representation can then be transformed into a sparse or dense matrix representation and diagonalised to find eigenvectors and eigenvalues, using the `sparseFermions` and `denseFermions` modules respectively. Alternatively, the `directFermions` module can be used to generate a symbolic Trotterized Hamiltonian, and the evolution operator corresponding to such. The `sparseFermions` and `denseFermions` modules can also be used for this task, generating the Trotterized Hamiltonian and unitary in matrix form; however, this is typically expensive and unnecessary. From a symbolic Trotterized Hamiltonian, an object of the `Circuit` class (containing a set of `Gate` objects) can be constructed. These objects have a variety of useful methods associated with them. These include self-optimisation techniques and readout of various important properties, most notably the length of the circuit (i.e. the amount of gates that it contains). Provided

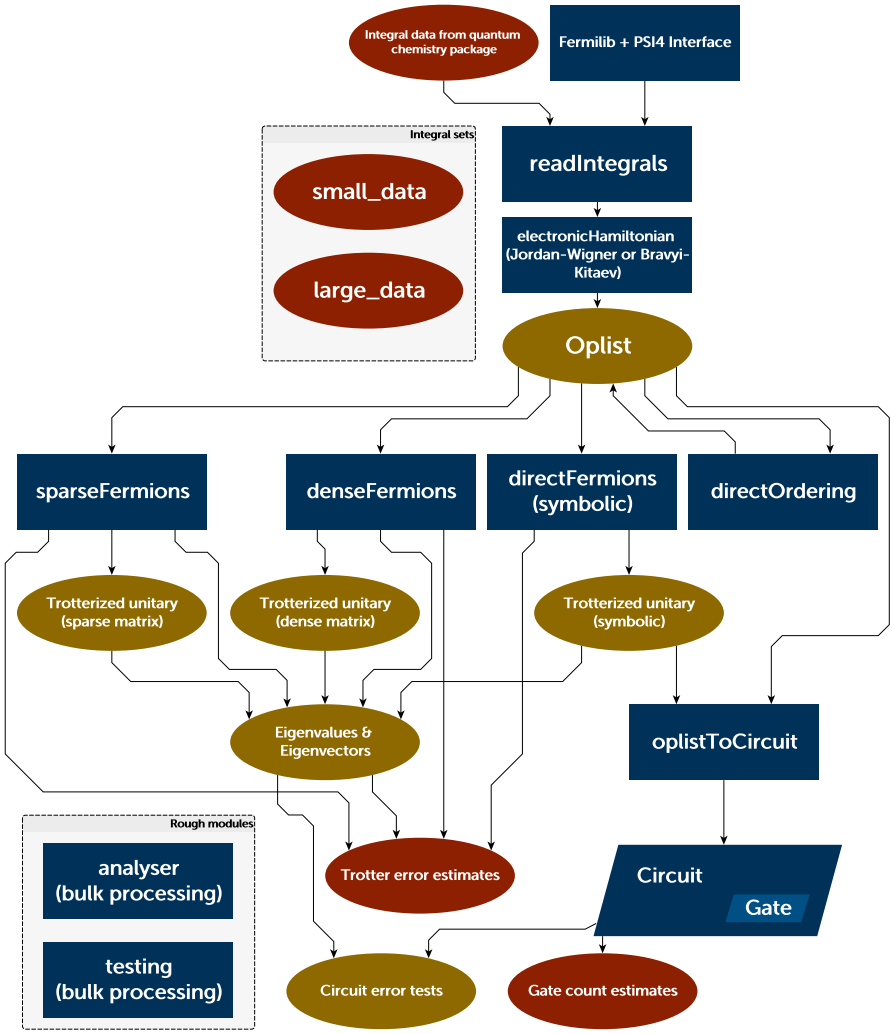



FIGURE 3.1.:
An overview of the library. Red ovals denote input and output data, whereas yellow ovals denote intermediary data that can be used for testing or referencing. Blue rectangles denote code modules, with blue trapeziums denoting objects that contain both code and data.

an input state as a vector (either in sparse or dense form), both `Gate` and `Circuit` objects can act on it, returning an output state. This effectively allows for the simulation of generated quantum circuits, which is helpful both when determining error implications of circuits, and for testing the validity of circuits. Additionally, a variety of analysis modules provide rough code to assist in performing batch trials, and testing modules provide debugging functionality. We discuss these modules in sequence.

## 3.1 Overlap integrals

The first modules to be considered are `readIntegrals` and `fermilib-Interface`, which deal with the input of molecular orbital integrals necessary for circuit generation and analysis. The form of these integrals is discussed in Section 2.2.

Although other choices of basis have been considered in other works,[88,93] this project primarily involved the use of canonical molecular spin-orbitals. These are the eigenvectors of the final Fock matrix obtained self-consistently through the Hartree-Fock process, as discussed above. Generation of these integrals first requires generation of these eigenvectors, and thus implementation of the Hartree-Fock process. Following this, the integrals themselves must be generated.

While the Hartree-Fock process is conceptually relatively simple in comparison to other quantum chemical methods, there are a variety of subtleties to consider in its implementation. For example, a method of obtaining an initial guess state must be considered, and symmetry can be invoked to reduce the computational cost. As such, the implementation of even the Hartree-Fock process in an optimally efficient and reliable way is non-trivial. Furthermore, a smorgasbord of well-tested quantum chemistry packages are available to implement this process. It was therefore desirable to avoid replicating these efforts, instead focusing on interfacing the code with existing packages.

Although the scripts to generate input for, run, and parse output from chemistry packages are unremarkable, care must be taken with regards to a number of subtleties. Firstly, standard quantum chemistry packages typically provide any integral data in a basis of atomic orbitals (for example, a STO-3G or 6-31G basis set) as used in the initial stage of a Hartree-Fock calculation. While these could be manually transformed into the final molecular orbital basis, it is preferable to ensure that the chemistry package outputs integrals in the molecular orbital basis, and that these are read in as appropriate.

To save on disk space and CPU time, these packages typically only provide permutationally unique integrals. Our procedures typically involve small basis sets and thus significantly lower numbers of integrals than a typical quantum chemistry calculation. As such, in the `readIntegrals` module, it is more convenient to store all integrals as a two dimensional (for the one electron integrals) or a four dimensional (for the two electron integrals)

array. For simplicity, the functions in `readIntegrals` do not take advantage of the symmetry, such that the arrays can be easily queried.

Quantum chemistry packages typically output integrals in "chemist's notation", a differing convention for the definition of the integrals wherein two indices are switched. Specifically, the four-electron integrals in this notation are given by

$$[pq|rs] = \int d\mathrm{x}_1 d\mathrm{x}_2 \frac{\chi_p^*(\mathrm{x}_1)\chi_q(\mathrm{x}_1)\chi_r^*(\mathrm{x}_2)\chi_s(\mathrm{x}_2)}{r_{1,2}}, \tag{3.1.0.1}$$

whereas the "physicist's notation" of the integrals is given by:

$$\langle pq|rs\rangle = \int d\mathrm{x}_1 d\mathrm{x}_2 \frac{\chi_p^*(\mathrm{x}_1)\chi_q^*(\mathrm{x}_2)\chi_r(\mathrm{x}_1)\chi_s(\mathrm{x}_2)}{r_{1,2}}. \tag{3.1.0.2}$$

Furthermore, an inspection of eqns. (2.2.0.1), (2.2.0.2) and (3.1.0.2) reveals that the convention used in the former, and widely in quantum simulation of quantum chemistry literature,[51] is additionally differing from the "physicist's notation" in that the $r$ and $s$ indices are flipped. Consequently, the three conventions can be converted between according to the relationships

$$h_{pqrs} = \langle pq|sr\rangle = \big[ps|qr\big] \ . \tag{3.1.0.3}$$

When writing scripts to parse quantum chemistry package output, this must be taken into account.

Finally, integrals are typically yielded from quantum chemistry packages in a purely spatial basis (wherein occupation numbers can be 0, 1 or 2). The `readIntegrals` module takes this into account by doubling the length of each array, taking into consideration the orthogonality of the spin states.

While initial stages of this project used the presently discussed code for the above tasks (in the `readIntegrals` module), in later sections it was desirable to develop an interface to the FermiLib package and its FermiLibPsi4 plugin, due to its greater functionality and reliability.[103] As such, the small `fermilibInterface` module was developed. This consisted of a FermiLibMolecule class, which served as a wrapper to FermiLib MolecularData objects. Various methods are present allowing extraction of the data used in the code, most notably the molecular orbital integrals. Although in our case this was used solely to extract integral data in a reliable manner, this interface code could be useful in future work, particularly with regards to verification and testing.

### 3.1.1 *Data sets*

As the tests aimed to assess techniques when applied across a variety of molecular systems, sets of molecular orbital integrals were required as input to the code. Our initial approaches used integrals provided by collaborators.[93] In this thesis this set is referred to as the small_data set. This included systems from $2 - 28$ spin-orbitals. Unlike later work, in addition to canonical

Table 3.1.: A breakdown of the small_data set. A further breakdown is available as Appendix A. For most of the systems counted here, circuits were generated using canonical Hartree-Fock molecular orbitals, Löwdin orthogonalised atomic orbitals and natural molecular orbitals. Numbers in parenthesis indicate the number of systems where the impact on error of Trotter ordering was considered with regards to at least one choice of molecular orbital basis, as discussed in Chapter 6. The polyatomic category includes molecules, ions and radicals.

| Qubits | 1-5 | 6-10 | 11-15 | 16-20 | 21-25 | 26-30 | Total |
|---|---|---|---|---|---|---|---|
| Polyatomic | 2 (1) | 0 (0) | 5 (4) | 0 (0) | 0 (0) | 0 (0) | 7 (5) |
| Atoms | 2 (2) | 8 (6) | 0 (0) | 8 (4) | 0 (0) | 0 (0) | 18 (12) |
| ECP bases | 2 (1) | 4 (2) | 0 (0) | 9 (1) | 1 (1) | 0 (0) | 16 (5) |
| Other bases | 2 (2) | 7 (5) | 4 (3) | 4 (0) | 3 (0) | 3 (0) | 23 (10) |
| Total | 8 (6) | 19 (13) | 9 (7) | 21 (5) | 4 (1) | 3 (0) | 64 (32) |

Hartree-Fock molecular orbitals, it included a choice of Löwdin orthogonalised, and natural molecular orbitals. A breakdown of the systems used in this section is included as Table 3.1 and Appendix A.

After performing analyses using the small_data set, it became apparent that a more comprehensive source of molecular orbital integrals would be useful for comparing schemes in the classically intractable regime. For this purpose, Hartree-Fock molecular orbitals and their one and two electron integrals ($h_{ij}$ and $h_{ijkl}$ of eqns. (2.2.0.1) and (2.2.0.2)) were obtained using the PSI4 electronic structure theory package[104], the FermiLib PSI4 Plugin[103], and the `fermilibInterface` module. These form what is referred

to in this thesis as the LARGE_DATA set. Integral data was generated for 86 small molecules and atoms. Molecular structures were gathered from the NIST CCCBDB database[105] geometrically optimised at the Hartree-Fock level. Most systems used a STO-3G basis; however, larger Pople basis sets were used in 18 trials. Of these, four systems ($CH_2$, $^2\bullet$, HF, LiH, $H_2O$) using a 3-21G basis set were examined, with the remainder studying $H_2$ and $HeH^+$. Clearly, this choice of basis is insufficient for an accurate solution of the electronic structure problem. When performing a simulation on a real quantum device, a larger basis set would be chosen as in conventional quantum chemistry methods. Fortunately, the error introduced by the choice of basis is fixed and independent of the choice of quantum methodology. The benchmarking procedure described in this thesis is not directly concerned with the exact energies predicted. The choice of basis set was therefore chosen to allow for the simulation of a variety of systems with relatively low computational overhead.

Details of the systems studied can be found in Table 3.2 and in Appendix A. They contain a variety of different types of chemical systems, including both organic and inorganic compounds, and several ions. The systems ranged in size from 2 (the hydrogen atom) to 56 (the iodine atom) spin-orbitals. Crucially, the larger end of this range exceeds the largest systems that can be simulated on a classical computer using direct diagonalisation of the full configuration interaction Hamiltonian.

This dataset is still relatively modest in contrast to simulations that would be performed upon a real quantum device[45]; however, it allowed us to main-

Table 3.2.: A breakdown of the large_data set. Note that most of the systems involving a non-minimal basis set were $H_2$ and $HeH^+$ systems, as specified in the appendices. Numbers in parenthesis indicate the number of systems where Trotter error was considered, as discussed in Chapter 6. The polyatomic category includes molecules, ions and radicals.

| Qubits | 1-10 | 11-20 | 21-30 | 31-40 | 41-50 | 51-60 | Total |
|---|---|---|---|---|---|---|---|
| Polyatomic | 1 (1) | 13 (13) | 12 (3) | 4 (0) | 8 (0) | 3 (0) | 41 (17) |
| Atoms | 10 (4) | 7 (6) | 1 (0) | 1 (0) | 0 (0) | 1 (0) | 20 (10) |
| Ions | 2 (2) | 4 (3) | 0 (0) | 2 (0) | 1 (0) | 1 (0) | 10 (5) |
| Other bases | 4 (4) | 6 (5) | 5 (2) | 0 (0) | 0 (0) | 0 (0) | 15 (11) |
| Total | 17 (11) | 30 (27) | 18 (5) | 7 (0) | 9 (0) | 5 (0) | 86 (43) |

tain relatively low computational expense – approximately one week of CPU time for the longest calculations on the largest examples.

## 3.2 Hamiltonians, unitaries and ordering

Having read in the integrals, the code requires some way of representing qubit Hamiltonians, using either a Jordan-Wigner or Bravyi-Kitaev mapping scheme. As discussed in Chapter 2, the qubit Hamiltonian will be a weighted sum of strings of Pauli X, Y, and Z operators (along with the identity) acting on each qubit. A logical structure here is to store a list of terms, with each term storing both the coefficient of the term (as a complex num-

ber) and a further list denoting which Pauli operator is applied to each qubit. In keeping with the code of Seeley, Richard and Love[79], therefore, Hamiltonians are stored as "oplists". This format is as described above, with each type of Pauli represented by a base 4 integer – 0 for the identity, 1 for an X, 2 for a Y and 3 for a Z. Listing 3.1 shows an example of this format.

Listing 3.1: The oplist describing the hydrogen molecular Hamiltonian using the Jordan-Wigner transformation. The first item in each element is the coefficient of the term (stored as a standard python complex128, with the `j` denoting the imaginary part), and the second being a list encoding which Pauli operations act on which qubits.

```
[[(-0.81260998249999994+0j), [0, 0, 0, 0]],
[(0.17120109750000007+0j), [0, 0, 0, 3]],
[(0.17120109750000007+0j), [0, 0, 3, 0]],
[(0.1686232725+0j), [0, 0, 3, 3]],
[(-0.222796615+0j), [0, 3, 0, 0]],
[(0.12054612249999999+0j), [0, 3, 0, 3]],
[(0.1658680075+0j), [0, 3, 3, 0]],
[(-0.045321884999999999+0j), [1, 1, 2, 2]],
[(0.045321884999999999+0j), [1, 2, 2, 1]],
[(0.045321884999999999+0j), [2, 1, 1, 2]],
[(-0.045321884999999999+0j), [2, 2, 1, 1]],
[(-0.22279661499999995+0j), [3, 0, 0, 0]],
[(0.1658680075+0j), [3, 0, 0, 3]],
[(0.12054612249999999+0j), [3, 0, 3, 0]],
[(0.174349485+0j), [3, 3, 0, 0]]]
```

An advantage of the use of this format is that code was developed by Seeley, Richard and Love[79] for the generation of Jordan-Wigner and Bravyi-Kitaev representations of two-fold and four-fold products of electronic creation and annihilation operators in this format. They similarly developed code for

the basic manipulation of oplists, particularly with regard to combination and simplification of several.* Oplists are generated by the `electronic-Hamiltonian` function. This takes five parameters: the number of orbitals, the one and two electron integrals (as two and four dimensional nested lists), a bool indicating whether the Jordan-Wigner or Bravyi-Kitaev mapping is to be used, and a bool indicating whether the two electron integrals require a sign flip. To prevent rounding errors, a level of precision must be specified under which terms are neglected. For the bulk of this project, this level was set to $10^{-12}$, although this likely could be increased without substantial impact on error rates.

Our strategy here does not retain knowledge of which individual *i, j, k, l* term in the electronic Hamiltonian (eqn. (2.0.0.1)) generates each term in the oplist. Instead, all terms are reduced to a list of weighted Pauli strings in the form of an oplist. This is in contrast to other works, which have maintained the *i, j, k, l* subterms and generated sequential quantum circuits simulating each of these.[106] Our approach somewhat abstracts away the physical significance of individual terms, and may introduce difficulty if a slightly modified Hamiltonian was required (for instance, if studying the importance of individual interaction terms). However, performing this abstraction allows us to perform optimisation and ordering at the level of individual Pauli strings. This allows greater flexibility when designing circuits used in simulation. Terms in an

*During later stages of this project, when considering large system sizes of upwards 80 spin-orbitals, the computational effort involved in manipulating these oplists became non-trivial. As such, an attempt was made to develop a new, dictionary-based structure, but yielded only small efficiency gains. This could be considered in future work.

oplist with non-unique Pauli strings are automatically combined.

From here, three modules can be used to generate further data. The simplest of these is `denseFermions`. This contains functions that generate matrix representations of the Hamiltonians stored as oplists. This module makes extensive use of the NumPy and SciPy[107] libraries for efficiency. Additionally, the unitary evolution operator corresponding to a Hamiltonian can be generated in matrix form. Trotterization can be performed prior to the generation of a matrix representation, with the optimal total evolution time automatically generated provided a reference energy (typically the Hartree-Fock energy).

Most importantly for this project, the `getTrueEigensystem` function (essentially a wrapper for SciPy's Hermitian matrix diagonalisation function) yields ground state eigenvalues and eigenvectors when provided with an oplist. This technique is clearly not scalable to large systems, as it is equivalent to the (factorially difficult) full configuration interaction problem. However, it is a requirement for the analysis of Trotter errors, as discussed in Chapter 6.

In addition to diagonalisation for the elucidation of eigenvalues and eigenvectors, basic manipulation using this approach is trivial. For example, provided states as vectors, the determination of expectation values is a simple matter of multiplying the yielded matrices. Specific functions exist to determine Trotter errors, by comparing the expectation value of a generated Trotterized evolution operator with the eigenstate against the calculated exact eigenvalue.

While this approach is convenient for small systems, these matrix representations rapidly grow in memory for larger than approximately 15 spin-orbitals, prohibiting their use. For example, the Hamiltonian for the Methane molecule in an STO-3G basis would have contained $2^{36}$ 128 bit complex numbers – requiring approximately one terabyte of RAM to store the Hamiltonian alone. While computational resources of this scale are available, analysis of substantially larger systems would have been impossible, and a bulk analysis of many molecules of this size infeasible. Some attempts were made to mitigate this by using memory mapped files; however, these resulted in critical performance reductions.

For systems of intermediate size (approximately 15 to 30 spin-orbitals), the `sparseFermions` module is preferable. While largely similar in terms of functionality, this instead utilises SciPy's sparse matrix tools to store the Hamiltonians and evolution operators as Compressed Sparse Column matrices. As the electronic Hamiltonian is sparse, this greatly reduces the memory costs incurred. With regards to the determination of energies and corresponding eigenvectors, this is the most efficient option available.*

While this is undoubtedly more efficient than the `denseFermions` module, it should be noted that generating the evolution operator results in a loss of sparsity. Consequentially, little benefit is observed here when attempting

*A technical aside: care must be taken to use the correct version of the SciPy libraries. Prior to version 0.14.0, the variable containing the number of non-zero elements was stored in a 32 bit integer, limiting the number of non-zero elements in a sparse matrix to $2^{31}$. This is problematic for larger systems.

to consider evolution under the generated Hamiltonians.

The third option – the `directFermions` module – operates in a different manner. Here, the symbolic oplist representation is maintained at all stages. The action of an oplist (i.e. a Hamiltonian) on a state vector can be simulated in a multi-stage process. The oplist is first broken into individual terms. For each term, the impact of every Pauli operation on the given state is simulated in sequence, with the coefficient applied as a final scalar multiplication. The functions which implement an individual Pauli operation – `oneXOnState` and `oneZOnState` – are compiled into C using Cython. This reduces runtime by several orders of magnitude. While the state vector does lose sparsity, the overall reduction in the number of stored values required (a factor of $2^N$) reduces the importance of this fact. These are the only compiled or platform dependent components of the code, and this mode can be disabled and (dramatically slower) standard Python functions used. Y operations are performed by running X and Z sequentially, with the appropriate phase shift. The resultant states are summed to yield a final state acted upon by the oplist.

The result of this procedure is code which allows for the simulation of the action of an oplist upon a state which circumvents the generation of a full matrix representation of the oplist. This avoids the dramatic memory costs associated with this process. Runtimes are reduced commensurately. However, this does not allow for the determination of eigenvalues and eigenvectors – a state which is to be acted upon must be provided, for instance from diagonalisation using the `sparseFermions` or `denseFermions` module. In

Listing 3.2: A function to implement one X gate on a state vector

```
def oneXOnState(numpy.ndarray[INDEXTYPE_t, ndim=1]
    indices,xIndex):
     cdef unsigned int xIndexStatic = xIndex
     cdef unsigned int qubitMask = 1<<xIndexStatic
     cdef unsigned int maxIndex = len(indices)
     cdef unsigned int i
     for i in range(maxIndex):
         indices[i] = indices[i] ^ qubitMask
     return(indices)
```

a sense, this can be considered to be a more faithful simulation of how an actual quantum simulation would be performed, in that the initial state to be studied must be explicitly prepared.

Trotterization here requires three steps: firstly, the terms are ordered and multiplied by factors so as to yield the appropriate Trotter-Suzuki approximant, given by eqn. (2.4.0.8). Following this, each term is exponentiated. Finally, terms are multiplied together to give the overall Trotterized evolution operator. The code allows for calculation of Trotter approximants of any order of form $2^N$, where $N$ is any positive integer. As the cost of the use of Trotter-Suzuki approximants scales exponentially with the Trotter order, this remains untested for Trotter orders greater than 4. Prior work has suggested that there is little to be gained from extending analysis beyond the fourth order Trotter-Suzuki approximant.[79] It should be noted that an oplist generated through this technique corresponds not to a direct Pauli string, but rather to the exponent of each individual unitary within the Trotter se-

quence.

Some attempts at determining the action of Trotterized unitaries – and thus Trotter errors, when compared to exact eigenvalues – resorted at this stage to the use of matrix methods. However, unlike prior attempts, if a state vector to be acted upon is specified, there is no need to generate the entire unitary in this form. Each individual Trotter term can be generated in matrix form, exponentiated, and allowed to act upon the state vector specified. This procedure can be iterated through the entire unitary, discarding each term after it is applied. This cuts down storage requirements, as only one exponentiated term and one state vector must be stored at any time. Each term is subject to the loss of relatively little sparsity. It should be stressed that this technique requires the explicit specification of the starting state, thereby limiting any Hamiltonians to be studied to those where eigenvectors can feasibly be found through diagonalisation. This is to be expected, as tractably avoiding this problem would be tantamount to solving the electronic structure problem itself, thereby entirely nullifying the advantage of using an actual quantum device.

However, later efforts revealed that even this method required an infeasible amount of computational resources. Even the simulation of methane (18 spin-orbitals) was impossible when including all terms in the Hamiltonian, provided $16GB$ of RAM and several days of CPU time. Instead, it was desirable to remove any necessity of the determination of matrix representations for anything other than the state vector itself. For this, note that the oplist can be further manipulated to provide an oplist corresponding to the evol-

ution operator in terms of direct strings of Pauli operations. If we define a Pauli string operator

$$\hat{P} = \bigotimes_{j=0}^{N} \sigma_j^{p(j)}, \tag{3.2.0.1}$$

where $N$ is the number of qubits and $p(j)$ is some map which shows which type of Pauli operation (or identity) is to be applied to qubit $j$, and $\sigma_j^{p(j)}$ is the Pauli operation of type $p(j)$ applied to qubit $j$, then by simple Taylor expansion we have:

$$e^{i\alpha\hat{P}} = \cos(\alpha)I + i\sin(\alpha)\hat{P}. \tag{3.2.0.2}$$

Performing this yields an oplist corresponding to the overall Trotterized unitary in terms of weighted Pauli strings. This can be manipulated in the same way as oplist-form Hamiltonians, as discussed above. Maintaining this symbolic representation relieves us of the extreme computational cost of exponentiating large matrices. In addition to the ability to generate quantum circuit representations (discussed below), this is the principal advantage to this symbolic approach. While eigenstates must be found through direct diagonalisation of the matrix representation of the Hamiltonian, the action of a Trotterized unitary on such an eigenstate can be determined with relative efficiency. This is useful for testing, and for the determination of Trotter errors.

At the symbolic oplist representation level, Hamiltonians can be ordered

prior to Trotterization, for the analysis of the ordering dependence of Trotter error. This is performed using the `directOrdering` module. The module contains helper functions required to determine useful properties while ordering. For example, the `generateAdjacencyMatrix()` function generates the commutativity adjacency matrix, which is useful when employing ordering strategies based on the commutativity of terms. Here, each ordering strategy has its own function (for example, `lexiographic` or `magnitude`) taking an unordered oplist, and returning an ordered oplist. These can then be fed to the Trotterization code in order to generate ordered Trotterized unitaries.

## 3.3 Circuit simulation

The code also allows for the generation of circuit objects from a specified symbolic Jordan-Wigner or Bravyi-Kitaev Hamiltonian. The `oplistToCircuit(oplist)` function takes an oplist as input, returning a `Circuit` object. This object contains a list of `Gate` objects. These classes form the backbone of the circuit simulation code. The `Gate` class describes a quantum gate. Internally, this is accomplished by storing the type of gate, the index or a list of indices of the qubit that it is to be applied to, and the angle of rotation in the case of single qubit rotations. For our purposes, it suffices to restrict our consideration to CNOT, Hadamard, Y-Hadamard (and inverse), identity, controlled-Z and arbitrary Z rotation gates. Gate objects can gen-

Table 3.3.: A simplified overview of the `Circuit` and `Gate` classes.

| `Circuit` | |
|---|---|
| Methods | Properties |
| `__init__(listGates,numQubits)` | `listGates` |
| `update()` | `numQubits` |
| `addRight(gate)` | |
| `addLeft(gate)` | |
| `getReverse()` | |
| `getInverse()` | |
| `readable()` | |
| `act()` | |
| `expectationValue(state)` | |
| `parallelDepth()` | |
| `qasm()` | |
| `removeGate(index)` | |
| `numGates()` | |
| `cancelDuplicates()` | |
| `fullCancelDuplicates()` | |
| `find(gate)` | |
| `dump(filePath)` | |

| `Gate` | |
|---|---|
| Methods | Properties |
| `__init__(gateType,qubits)` | `qubits` |
| `readable()` | `angle` |
| `qasm()` | `type` |
| `getInverse()` | |
| `act(state)` | |
| `sparseMatrixForm()` | |
| `sparseUnitary(circuitDepth)` | |
| `checkCommute(otherGate)` | |
| `canCancel(otherGate)` | |

erate inverses and matrix representations of themselves, with the latter case requiring specification of the total number of qubits within the overall circuit. Importantly for optimisation considerations, when provided with another gate, they can check for commutativity, combinability and cancellability. Finally – and importantly for verification purposes – if a state vector is provided, a gate can simulate its action on that state, returning a new vector. This action is performed directly, without recourse to the generation of a matrix representation of the gate, cutting on computational cost. The code to perform this is included in Appendix B.

The Circuit class is somewhat more complex, incorporating a variety of attributes and methods. Table 3.3 summarises the most important of these. The most important element here is the `listGates` attribute, which consists of a list of `Gate` objects which constitute the circuit. Consequentially, the number of elements in this list is equivalent to the circuit length. This structure allows for easy iteration over all gates in the circuit, to provide functionality that is not otherwise baked into the `Circuit` code. Some relatively self-explanatory functionality exists. For example, the `addRight` and `addLeft` add `Gates` to the right and left hand sides of the circuit, and `getReverse` yields the reversed circuit. The `act` method loops through the entire circuit, performing each `Gate.act` on a provided state for every gate in the circuit, thereby simulating the action of the circuit on such a state. Methods relating to the parallel depth of the circuit exist, although these are somewhat untested, and were not used to generate real data in this project.

Most importantly for the purposes of Chapter 5, the `cancelDuplica-`

`tes` method iterates through the circuit, searching for gate pairs that resolve to unity and removes them. In this process, individual gates can be checked against gates that follow it for commutativity. If a gate does indeed commute with its successor, it is checked against gates further in sequence until it hits a gate that it does not commute with. In other words, the code commutes each gate as far through the circuit as is possible, searching for and removing pairs that resolve to the identity (typically self-inverse duplicate pairs). This commutation is performed according to a set of rules – gates acting on different qubits always commute, CNOT gates commute unless one targets the other's control, and so on – thus avoiding the generation of matrix representations of the gates.

The somewhat cryptically named `fullCancelDuplicates` method performs the `cancelDuplicates` method iteratively, repeating the process until self-consistency is reached, thereby ensuring that all possible cancellations are performed. This procedure is not optimised and relatively slow, in contrast to other codes intended to develop circuits that would run on actual quantum devices. However, for our purposes, this stage was not considered performance critical, as other procedures (particularly generating eigenvectors and eigenvalues of the electronic Hamiltonians) dominated computational cost.

`Circuit` and `Gate` objects can provide several alternative representations of themselves. As discussed above, matrix representations of individual gates can be determined. More usefully, a human-readable representation is available for debugging purposes. Finally, a QASM[108] representation of gates

and circuits can be generated.* While this can be used to "translate" circuits generated by the code into other packages, some gates are defined as simple text strings, with no reference to their true action. As such, the primary use of this functionality is for visualisation of circuits using the qasm2circ package.[109] This is mostly useful for debugging, and demonstration purposes.

When performing a full analysis using the code (generating a quantum circuit for the simulation of a molecular system at a specified geometry), it is necessary to convert the obtained Trotterized electronic Hamiltonian in oplist form to a `Circuit` object. This is performed through the `oplistToCircuit` function, which loops through the oplist, generating `Circuit` objects corresponding to the simulation of each term and combining them. Subcircuits, corresponding to individual terms in the oplist, can be tracked; however, as this structure is inherently destroyed when performing cancellation of duplicate gates, this functionality is rarely of major use.

This chapter has been intended as a reference for the functionality of the code used to generate the results of Chapters 5 and 6. Although to aid reproduction we have discussed a variety of technical details, the main focus here is on the output data that can be generated – most importantly, Trotter error estimates and circuit gate counts.

*This implementation of QASM is relatively simple in contrast to more sophisticated treatments, such as Open QASM[108].

# 4 Theory of the Bravyi-Kitaev Transformation

In Chapter 2, the discussion of the Jordan-Wigner transformation (which maps fermionic states and operators to states of and operations upon qubits) noted that it requires a number of gates that scales linearly with the number of electronic operations. Ideally – as the number of electronic operations scales in the worst case quartically with the number of spin-orbitals considered – this would be reduced. In this chapter, the *Bravyi-Kitaev transformation*, an alternative mapping scheme with asymptotically superior scaling properties, is discussed. Firstly, the theory underpinning the mapping is described, so as to contextualise the discussion of Chapters 5 and 6, where a numerical assessment of the efficacy of the mapping is reported.

Firstly, we consider a simple alternative to the occupation basis discussed in Chapter 2. This is the *parity basis*. Limitations of this basis are considered, before the Bravyi-Kitaev transformation itself is discussed in Section 4.2.

## 4.1 The parity basis

As discussed in Chapter 2, the Jordan-Wigner transformation relies on using states of qubits to represent the occupation numbers of spin-orbitals. If a spin-orbital is unoccupied, its corresponding qubit is in the $|0\rangle$ state, whereas if it is occupied, its corresponding qubit is in the $|1\rangle$ state. The main cost of the use of the Jordan-Wigner transformation is the required chain of Pauli Z operations which apply the correct phase shift depending on the parity of preceding qubits. To avoid this difficulty, a scheme was envisaged by Bravyi and Kitaev[78] whereby the parity information – rather than occupation numbers – is encoded within the qubits. In this scheme, the matrix transforming orbital occupation numbers to qubit states (in contrast to eqn. (2.3.0.1) for the occupation basis) is given by:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \end{bmatrix} = \begin{bmatrix} f_0 \\ f_0 + f_1 \\ f_0 + f_1 + f_2 \\ f_0 + f_1 + f_2 + f_3 \\ f_0 + f_1 + f_2 + f_3 + f_4 \\ f_0 + f_1 + f_2 + f_3 + f_4 + f_5 \\ f_0 + f_1 + f_2 + f_3 + f_4 + f_5 + f_6 \\ f_0 + f_1 + f_2 + f_3 + f_4 + f_5 + f_6 + f_7 \end{bmatrix}. \tag{4.1.0.1}$$

In other words, the state of qubit $j$ is given by the sum $\sum_{k \leq j} o_k$, where $o_j$ is the occupation number of the $j$th spin-orbital. An initial assessment proves promising. If a parity-dependent phase shift is required, this can be introduced in one $Z$ gate. However, updating the qubit states is no longer as simple as applying a $\sigma^+$ or $\sigma^-$ gate. Here, there are two cases. If the preceding qubit is in the $|0\rangle$ state (indicating that the sum of the occupation numbers of orbitals $< j$ is 0), then the $j$th qubit stores the occupation number of the $j$th orbital. If the preceding qubit is in the $|1\rangle$ state (indicating that the sum of the occupation numbers of orbitals $< j$ is 1), then the *jth* qubit stores the complement of occupation number of the $j$th orbital. Thus, in the latter case, the opposite $\sigma^\pm$ operation must be applied. We also apply a $Z_{j-1}$ gate to introduce the necessary phase shift. In other words,

$$\hat{P}_j^\pm = \left( |0\rangle_{j-1} \langle 0|_{j-1} \sigma_j^\pm + |1\rangle_{j-1} \langle 1|_{j-1} \sigma_j^\mp \right) Z_{j-1} \,, \tag{4.1.0.2}$$

and thus:

$$\begin{aligned} \hat{P}_j^\pm &= \left( |0\rangle_{j-1} \langle 0|_{j-1} \sigma_j^\pm - |1\rangle_{j-1} \langle 1|_{j-1} \sigma_j^\mp \right) \\ &= \frac{1}{2} \left( |0\rangle_{j-1} \langle 0|_{j-1} \left( X_j \mp iY_j \right) - |1\rangle_{j-1} \langle 1|_{j-1} \left( X_j \pm iY_j \right) \right) \\ &= \frac{1}{2} \left( \left( |0\rangle_{j-1} \langle 0|_{j-1} - |1\rangle_{j-1} \langle 1|_{j-1} \right) X_j \mp iY_j \right) \\ &= \frac{1}{2} \left( Z_{j-1} X_j \mp iY_j \right) \,. \end{aligned} \tag{4.1.0.3}$$

This updates qubit $j$ and introduces the necessary phase shift. However,

in this basis, information about the occupation number of spin-orbital $j$ is stored in all qubits with index $\geq j$. For example, note that $f_2$ appears in six rows of the right hand vector of eqn. (2.3.0.1), as opposed to only one in eqn. (4.1.0.1). As such, any one fermionic creation or annihilation operation requires the update of the states of all qubits with index $\geq j$, and thus:

$$a_j^\dagger \rightarrow \frac{1}{2}\left(\bigotimes_{k>j} X_k\right)\left(Z_{j-1}X_j - iY_j\right) ,$$
$$a_j \rightarrow \frac{1}{2}\left(\bigotimes_{k>j} X_k\right)\left(Z_{j-1}X_j + iY_j\right) . \tag{4.1.0.4}$$

Unfortunately, it is clear from this that the use of the parity basis also results in a number of gates that is linear with respect to $j$. Non-locality in the calculation and update of parity information has been reduced; however, non-locality in the calculation and update of occupation information has been increased. However, this approach does yield benefit when combined with the occupation basis, as discussed below. Prior to discussion of the Bravyi-Kitaev transformation, it is useful to consider the sets of qubits that are used when employing the parity basis.

Firstly, the *update set* of qubits is, eponymously, the set of qubits whose state must be updated when an operation is performed, other than the qubit corresponding to the spin-orbital acted upon. For example, using the parity basis, the update set of qubit $j$ consists of all qubits with index $> j$. In other

Table 4.1.: The update, parity and flip sets for the occupation and parity bases.

| | Basis | |
|---|---|---|
| Set | Occupation | Parity |
| $U(j)$ | $\emptyset$ | $\{k \mid k > j\}$ |
| $P(j)$ | $\{k \mid k < j\}$ | $\{j-1\}$ |
| $F(j)$ | $\emptyset$ | $\{j-1\}$ |

words, for the parity basis, we have that $U(j) = \{k \mid k > j\}$.

The *parity set* is the set of qubits whose sum determines the parity such that the correct phase shift is implemented. Using the parity basis, this simply consists of the qubit with index $j-1$, i.e. $P(j) = \{j-1\}$.

Finally, as discussed above, in the case where qubit $j-1$ was in the $|1\rangle$ state, the creation or annihilation operation to be performed had to be flipped. We can thus define a *flip set*, being the set of qubits whose parity determines whether or not this flipping is to be performed. In the case of the parity basis, this set is identical to the parity set ($F(j) = \{j-1\}$).

Table 4.1 summarises these sets for both the occupation and parity bases. Note that these sets do not include qubit $j$, as this is handled independently, as above.

## 4.2 The Bravyi-Kitaev transformation

The Bravyi-Kitaev basis[78] is, in essence, a combination of both the occupation basis and the parity basis. An analogy can be made with the Fenwick trees of classical computer science[95,110], which are an efficient way to balance the update of elements and the determination of prefix sums of a table. As with the other constructions, we initially describe the mapping of electronic states to qubit states.

In this scheme, even indexed qubits are used to store the occupation of the electronic states, as in the Jordan-Wigner mapping. Similarly to the parity basis, qubits with indices that are powers of 2 are used to store the parity of qubits with index $\leq j$. However, qubits with odd indices that are not powers of two are used to store *partial* sums of sets of qubits. This is in contrast to the straightforward parity basis, where qubits were used to store the sum of all preceding qubits.

The qubit with the highest index is used to store the parity of all electronic occupations. The middle qubit stores that sum of the first half of the electronic occupations. This procedure is repeated, splitting the qubits in binary fashion until all qubits are allocated states. To illustrate this process, it is instructive to consider the transformation matrix taking a vector of spin-orbital occupation numbers to states of qubits, in comparison to eqns. (2.3.0.1) and (4.1.0.1). Here, for clarity, we use a basis of 16 spin-orbitals, such that the

transformation matrix is given by

$$\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1
\end{bmatrix}
\begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \\ f_8 \\ f_9 \\ f_{10} \\ f_{11} \\ f_{12} \\ f_{13} \\ f_{14} \\ f_{15} \end{bmatrix}
=
\begin{bmatrix} f_0 \\ f_0 + f_1 \\ f_2 \\ \sum_{n=0}^{3} f_n \\ f_4 \\ f_4 + f_5 \\ f_6 \\ \sum_{n=0}^{7} f_n \\ f_8 \\ f_8 + f_9 \\ f_{10} \\ \sum_{n=8}^{11} f_n \\ f_{12} \\ f_{12} + f_{13} \\ f_{14} \\ \sum_{n=0}^{15} f_n \end{bmatrix}. \tag{4.2.0.1}$$

The binary branching structure of this matrix is notable from inspection. Consider, for example, the calculation of the occupation number of the final electronic basis state – that is, $f_{15}$. Here, we require the state of the final Bravyi-Kitaev qubit - containing the sum of all occupation numbers. We also require the state of qubit 7, which stores the occupation numbers of the first

half of the orbitals. This allows us to calculate the sum of the occupation numbers of the second half of the orbitals. Then, we examine the state of qubit 11, which stores $f_8 + f_9 + f_{10} + f_{11}$. This allows us to determine the sum $f_{12} + f_{13} + f_{14} + f_{15}$. We take the state of qubit 13 to determine $f_{14} + f_{15}$ and finally this can be compared against the state of qubit 14 to yield $f_{15}$, as required. In total, we have required the manipulation of five qubits.

Assuming that a similar number of queries is required for the even indexed qubits (to determine the parity of the appropriate subset of orbitals), a similar analysis shows that the manipulation of five qubits is required for the determination of the occupation of the first eight states, four qubits for the preceding four states, three qubits for the preceding two states, two qubits for the second state and only one qubit for the first state.

This suggests that the number of qubits required increases by one every time the number of electronic states is doubled, implying that $O(\log N)$ queries are required to determine the occupation of a single state. Similarly, only $O(\log N)$ qubits - the qubits containing the partial sums of relevance - must be updated for any creation or annihilation operation.

This perhaps provides an intuitive explanation as to the key result of Bravyi and Kitaev – that this mapping results in a quantum simulative cost that scales logarithmically with the number of electronic basis states. This is an improvement on the linear scaling of both the occupation (as used in the Jordan-Wigner transformation) and parity bases.[78–80]

In order to put this on firmer footing, we must consider explicitly the transformation of electronic creation and annihilation operators when using the Bravyi-Kitaev transformation. In order to achieve this (and to demonstrate the process used to determine these operators computationally), we note that the matrix which converts Jordan-Wigner states to Bravyi-Kitaev states can be generated recursively. As the matrix taking electronic basis occupation numbers to Jordan-Wigner qubit states is just the identity matrix, this is equivalent to the development of matrices that convert electronic basis occupation numbers to Bravyi-Kitaev states, as in eqn. (4.2.0.1). We first define a matrix $\beta_x$ with $\beta_1 = [1]$. We then have, in block matrix format,

$$\beta_{2^{x+1}} = \begin{bmatrix} \beta_{2^x} & 0 \\ T_{2^x} & \beta_{2^x} \end{bmatrix} \tag{4.2.0.2}$$

with $T_y$ simply being the $y \times y$ zero matrix, but with the bottom row consisting entirely of ones. For example:

$$T_2 = \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix} . \tag{4.2.0.3}$$

This can be observed to generate the transformation in eqn. (4.2.0.1) as $T_{16}$. It should be noted that this process will only generate matrices for systems with numbers of orbitals that are powers of two. Fortuitously, to generate transformation matrices for other system sizes is a simple matter of cropping a larger matrix to the appropriate size.

In the following discussion, it will also be useful to define the inverse relations to eqns. (4.2.0.2) and (4.2.0.3) – i.e. the relations that generate the matrices that transform vectors of Bravyi-Kitaev states into spin-orbital occupation numbers. With $\beta_1^{-1} = [1]$, these are given by

$$\beta_{2^{x+1}}^{-1} = \begin{bmatrix} \beta_{2^x}^{-1} & 0 \\ O_{2^x} & \beta_{2^x}^{-1} \end{bmatrix}, \tag{4.2.0.4}$$

with $O_y$ being the $y \times y$ zero matrix with a one in the bottom right hand corner, such as:

$$O_y = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}. \tag{4.2.0.5}$$

It should be remembered that all operations here are performed modulo 2. These relations allow the generation of transformation matrices from the occupation basis to the Bravyi-Kitaev basis.

### 4.2.1 *Sets of qubits in the Bravyi-Kitaev mapping*

In order to generate the mapping of electronic operations to operations upon qubits, it is instructive to consider the equivalent sets of qubits to those derived for the occupation and parity bases. Before constructing explicit representations of operators, we describe these in sequence. In order to determine these sets, we shall in general transform the Bravyi-Kitaev vectors into the basis most appropriate where the set is known. We first consider the eight qubit case, where $N = 8$, before considering the general case.

Firstly, the parity set, being the set of qubits whose states determine whether a phase shift is necessary. This will consist of the qubits that contain the partial sums which, when summed, yield the parity of the electronic states with index less than $j$. For this we consider the mapping which takes Bravyi-Kitaev vectors to the parity basis. The matrix converting Jordan-Wigner states to the parity basis for eight qubits is given by

$$\pi_8 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}. \tag{4.2.1.1}$$

Similarly, the inverse Bravyi-Kitaev transformation matrix (which takes Bravyi-

Kitaev vectors to Jordan-Wigner vectors)

$$\beta_8^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix} \tag{4.2.1.2}$$

is obtained through eqns. (4.2.0.4) and (4.2.0.5). Consequentially, multiplying these matrices yields a matrix

$$\pi_8\beta_8^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.2.1.3}$$

which converts Bravyi-Kitaev vectors to parity basis vectors.

When using the parity basis, each qubit $k$ stores the partial sum of the occu-

pation numbers of all orbitals with index less than or equal to $k$. As such, in order to determine the partial sum of the occupation numbers of the orbitals with index less than $j$, we examine row $j-1$ of eqn. (4.2.1.3). Where the entries in this row are nonzero, the Bravyi-Kitaev state with corresponding index will contribute to the partial sum of the occupation numbers of orbitals with index less than $j$. Consequentially, we can ascertain the parity set of qubit $j$ by taking the indices of the nonzero values of row $j-1$ of eqn. (4.2.1.3). As such:

$$\left\{\begin{array}{c} P(0) = \emptyset \\ P(1) = \{\,0\,\} \\ P(2) = \{\,1\,\} \\ P(3) = \{\,1, 2\,\} \\ P(4) = \{\,3\,\} \\ P(5) = \{\,3, 4\,\} \\ P(6) = \{\,3, 5\,\} \\ P(7) = \{\,3, 5, 6\,\} \end{array}\right\}. \tag{4.2.1.4}$$

We next consider the update set. To recap, this is the set of qubits which must be updated upon a change in the occupation of the molecular state indexed $j$. These will be any qubits that store the occupation number $f_j$. We can obtain this by examining the matrix which transforms from the occupation basis to the Bravyi-Kitaev basis, given by eqns. (4.2.0.2) and (4.2.0.3). The update set for qubit $j$ will be any nonzero entries in column $j$, excluding row $j$. For

example, the transformation matrix for eight qubits

$$\beta_8 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{4.2.1.5}$$

suggests the following set of update sets:

$$\begin{Bmatrix} U(0) = \{1, 3, 7\} \\ U(1) = \{3, 7\} \\ U(2) = \{3, 7\} \\ U(3) = \{7\} \\ U(4) = \{5, 7\} \\ U(5) = \{7\} \\ U(6) = \{7\} \\ U(7) = \emptyset \end{Bmatrix}. \tag{4.2.1.6}$$

The final set to be found is the flip set, which determines whether qubit $j$ stores the occupation number $f_j$ or its complement. For even indexed qubits, this set is empty, as these store only the occupation number $f_j$ and thus will never need to be flipped. For odd indices, the set will contain the indices that

are in the partial sum of occupation numbers stored in qubit $j$, excluding $j$ itself. To ascertain this, we take the inverse transformation matrix generated by eqns. (4.2.0.4) and (4.2.0.5)

$$\beta_8^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix} \tag{4.2.1.7}$$

and examine the nonzero entries in the row corresponding to each qubit, yielding:

$$\begin{Bmatrix} F(0) = \emptyset \\ F(1) = \{\, 0 \,\} \\ F(2) = \emptyset \\ F(3) = \{\, 1, 2 \,\} \\ F(4) = \emptyset \\ F(5) = \{\, 4 \,\} \\ F(6) = \emptyset \\ F(7) = \{\, 3, 5, 6 \,\} \end{Bmatrix}. \tag{4.2.1.8}$$

Eqn. (4.2.1.8) shows the flip set for the Bravyi-Kitaev mapping, yielding all of the sets that were generated for both occupation and parity bases.

### 4.2.2 *Bravyi-Kitaev operators*

We have now determined all the sets necessary for the determination of Bravyi-Kitaev transformed electronic creation and annihilation operators. For even indexed qubits, this is a relatively simple task. As noted at the start of this section, even indexed qubits are used to store only the occupation numbers of their corresponding orbitals. As such, their flip set is empty, and thus there is no need to introduce variable behaviour depending on the state of the flip set qubits. Thus, all that is required is the actual creation or annihilation operation ($\hat{Q}^{\pm}$), the updating of the update set of qubits with a bit flip (via a Pauli X gate) and the introduction of a phase shift depending on the parity of the parity set (by applying a Pauli Z gate to each member). For even indexed qubits, the mapped creation and annihilation operators are then given by

$$\begin{aligned} a_j^\dagger &= X_{U(j)} \otimes \hat{Q}_j^+ \otimes Z_{P(j)} = \frac{1}{2}\left(X_{U(j)} \otimes X_j \otimes Z_{P(j)} - iX_{U(j)} \otimes Y_j \otimes Z_{P(j)}\right) , \\ a_j &= X_{U(j)} \otimes \hat{Q}_j^- \otimes Z_{P(j)} = \frac{1}{2}\left(X_{U(j)} \otimes X_j \otimes Z_{P(j)} + iX_{U(j)} \otimes Y_j \otimes Z_{P(j)}\right) . \end{aligned} \tag{4.2.2.1}$$

Note that here we have introduced the notation $O_{S(j)}$, being the operation $O$ applied to each qubit with index in the set $S(j)$, i.e. $O_{S(j)} = \bigotimes_{k \in S(j)} O_k$.

The operators for qubits with odd index are dependent on the parity of the flip set. When the flip set has odd parity, the occupation of the qubit is flipped from that of the electronic state. Thus, when the flip set has odd parity, the creation operator must be applied to the qubit when the anni-

hilation operator is applied to the electronic state, and vice versa. Defining projectors onto the even and odd states of a set, $S$, of qubits:

$$\hat{E}_S = \frac{1}{2}\left(I + Z_S\right) , \hat{O}_S = \frac{1}{2}\left(I - Z_S\right) , \tag{4.2.2.2}$$

we then have new creation and annihilation operators

$$\hat{\Pi}_j^{\pm} = \hat{Q}_j^{\pm} \otimes \hat{E}_{F(j)} - \hat{Q}_j^{\mp} \otimes \hat{O}_{F(j)} = \frac{1}{2}\left(X_j \otimes Z_{F(j)} \mp iY_j\right) \tag{4.2.2.3}$$

which express this behaviour. While in the derivation of the parity basis operators, the parity dependent phase shift was yielded by applying a Pauli Z to every member of the parity set, here a further correction must be made. The correct Z gates have already been applied to the flip set. As such, we must only apply Z gates to the members of the parity set that are not also members of the flip set. We define the *remainder set*

$$R(j) = P(j) \setminus F(j) \tag{4.2.2.4}$$

to address this. Combining these, we have a mapping for the creation and annihilation operators for odd indexed qubits

$$\begin{aligned} a_j^{\dagger} &= X_{U(j)} \otimes \hat{\Pi}_j^{+} \otimes Z_{R(j)} = \frac{1}{2}\left(X_{U(j)} \otimes X_j \otimes Z_{P(j)} - iX_{U(j)} \otimes Y_j \otimes Z_{R(j)}\right) , \\ a_j &= X_{U(j)} \otimes \hat{\Pi}_j^{-} \otimes Z_{P(j)} = \frac{1}{2}\left(X_{U(j)} \otimes X_j \otimes Z_{P(j)} + iX_{U(j)} \otimes Y_j \otimes Z_{R(j)}\right) . \end{aligned} \tag{4.2.2.5}$$

The only difference between eqns. (4.2.2.5) and (4.2.2.1) is the application of the final Z operation to the remainder set, instead of the parity set. We define a final set

$$\varrho(j) = \begin{cases} P(j), & j \text{ even} \\ R(j), & j \text{ odd} \end{cases} \tag{4.2.2.6}$$

such that the Bravyi-Kitaev mapped electronic creation and annihilation operators are given by:

$$\begin{aligned} a_j^\dagger &= \frac{1}{2}\left(X_{U(j)} \otimes X_j \otimes Z_{P(j)} - iX_{U(j)} \otimes Y_j \otimes Z_{\varrho(j)}\right) , \\ a_j &= \frac{1}{2}\left(X_{U(j)} \otimes X_j \otimes Z_{P(j)} + iX_{U(j)} \otimes Y_j \otimes Z_{\varrho(j)}\right) . \end{aligned} \tag{4.2.2.7}$$

As for the occupation and parity bases, general operators can be constructed through taking products of these qubit operators, allowing for the construction of the mapped electronic Hamiltonian. Specific terms within the electronic Hamiltonian can be explicitly mapped[79], however these yield no obvious physical insight, and as such are not considered here.

The Bravyi-Kitaev operators (eqn. (4.2.2.7)) consist of operations on the update, parity and remainder sets. An examination of eqns. (4.2.1.4), (4.2.1.6) and (4.2.1.8) shows that these sets contain at most $2^3$ qubits for a given index. A similar construction can be performed for larger systems[78,79]. Recursion relations can be generated to simplify the determination of these sets for arbitrary numbers of qubits.[80] Either approach shows that the size of the update, parity and flip sets (and thus, the remainder set) grows logarithmic-

ally with the number of qubits in the overall system. As the size of these sets determines the number of operations in the Bravyi-Kitaev mapped operators, it is thus apparent that the number of gates required to simulate one electronic operation grows logarithmically with the number of spin-orbitals. This is a significant improvement on the linear scaling of the Jordan-Wigner transformation, strongly suggesting that the use of the Bravyi-Kitaev mapping would result in shorter circuits when performing a quantum simulation. This fact motivates the work discussed in Chapter 5, which aimed to quantify this advantage for real chemical systems.

In this chapter, the established theory of the Bravyi-Kitaev transformation has been discussed. The use of the Bravyi-Kitaev transformation results in a number of gates required to simulate one electronic operation that scales logarithmically with the number of qubits, in contrast to the linear scaling of the Jordan-Wigner transformation. However, this asymptotic superior scaling says little about the computational overhead, which could harm the potential of the scheme when performing real quantum simulations. For an assessment of this, numerical analysis is required.

# 5 Assessing the Bravyi-Kitaev Transformation

In Chapter 4, the established theory of the Bravyi-Kitaev mapping was discussed. However, it was noted that it would be desirable to numerically assess the efficacy of the scheme for practical quantum computation. Such an assessment was the goal of this section of this project. The results of this assessment are reported in this chapter.

We begin with consideration of a case study – the methane molecule in an STO-3G basis. The analysis of this section has previously been published as reference [80]. We then examine the transformation using the SMALL_DATA set described in Chapter 3. Finally, we consider the LARGE_DATA set in the context of the Bravyi-Kitaev mapping, an analysis which is also reported in reference [111].

## 5.1 Methane – a case study

The above section described the established theory regarding the Bravyi-Kitaev mapping. The principle goal of this branch of the present project was the determination of the efficacy of this scheme for practical quantum computation.

The first examination of the use of the Bravyi-Kitaev mapping in a chemical context was performed by Seeley, Richard and Love.[79] In this, the mapping's impact on the determination of the ground state of molecular hydrogen in a minimal basis was examined. A reduction of approximately 32% of the total gate count was observed. Notably, this system is the smallest chemically interesting example. It was thus argued that this serves as a limiting case, and that the Bravyi-Kitaev mapping is likely to be similarly efficient in other systems. Greater increases in efficiency were expected to be observed in larger systems, due to the asymptotically superior scaling properties. The aim of this branch of the present project was to validate and characterise this efficiency gain.

In order to achieve this goal, our initial investigations focussed on a case study: the methane molecule in a minimal basis. This process catalysed the development of much of the code described in Chapter 3, which was later expanded for the analysis presented in the rest of this chapter. Our initial attempts used the `sparseFermions` module to generate a matrix representation of the electronic Hamiltonian, Trotterizing and determining the

Trotter error. However, the computational cost of this process required the neglect of a large number of terms in the electronic Hamiltonian. Initial attempts[112] took this approach, studying the length of Jordan-Wigner and Bravyi-Kitaev circuits for a Hamiltonian comprising of only the 100 terms in the methane Hamiltonian with the highest magnitude.

In order to expand upon this, later efforts involved the development of the `directFermions` module, which allowed for the simulation of the full Hamiltonian in acceptable time. The results of this were first reported in reference [80]. Here, we discuss data yielded from the methane molecule in the LARGE_DATA set, in contrast to that of reference [80], where integral data from collaborators was used. Other than the integral data (derived from a different quantum chemistry package, using an STO-3G rather than STO-5G basis and a marginally different bond length), our procedures (and, mercifully, our conclusions) remain the same.

The analysis of methane was conducted using the methane data in the LARGE_DATA set. Integrals were generated using the PSI4 quantum chemistry package, using a STO-3G basis and a bond length of 1.08301Å. Jordan-Wigner and Bravyi-Kitaev Hamiltonians were generated according to the procedure discussed in Chapter 3, and were ordered lexicographically (i.e. in ascending order in terms of the base 4 integer lists in the oplists). The `sparseFermions` module was used to determine the ground eigenstate and exact eigenvalue. A Trotterized oplist corresponding to the evolution operator was generated, and then its action on the ground state assessed through the `directFermions` module, allowing calculation of the Trotter error. As extensive

analysis of the performance of the mapping under circuit optimisation was not required in this analysis, such optimisations were not considered, and instead the raw gate counts using each transformation were found.

These simulations were performed repeatedly, varying the number of Trotter steps used in the approximation. The results in terms of total gates are shown as Figure 5.1. Curiously, in terms of total gates, the Bravyi-Kitaev circuits yield only a very small reduction in overall gate count. This was in contrast to the argument of Seely, Richard and Love.[79] 1.4% of the gates in the first order Jordan-Wigner circuit with one Trotter step were saved when using the Bravyi-Kitaev mapping. Our later work – discussed in Section 5.3 – considers this in more detail.

Observing Figure 5.1, it is notable that the error incurred through the use of a Trotter-Suzuki approximation is extremely low, even with the use of only one Trotter step. Indeed, all of the simulations performed resulted in an error less than chemical accuracy (1 kcal/mol). This is encouraging for the prospects of quantum simulation, as it suggests – at least in this instance – that the number of Trotter steps (which increase the gate count linearly) will be relatively low compared to previous estimates. It should be noted that in a quantum simulation, this error would only be obtained when determining the first bit of the phase estimated energy.

Of interest to the discussion of the Bravyi-Kitaev mapping is the fact that the use of this scheme consistently results in lower errors than the Jordan-Wigner mapping. To an extent, it is impossible to discuss this without con-

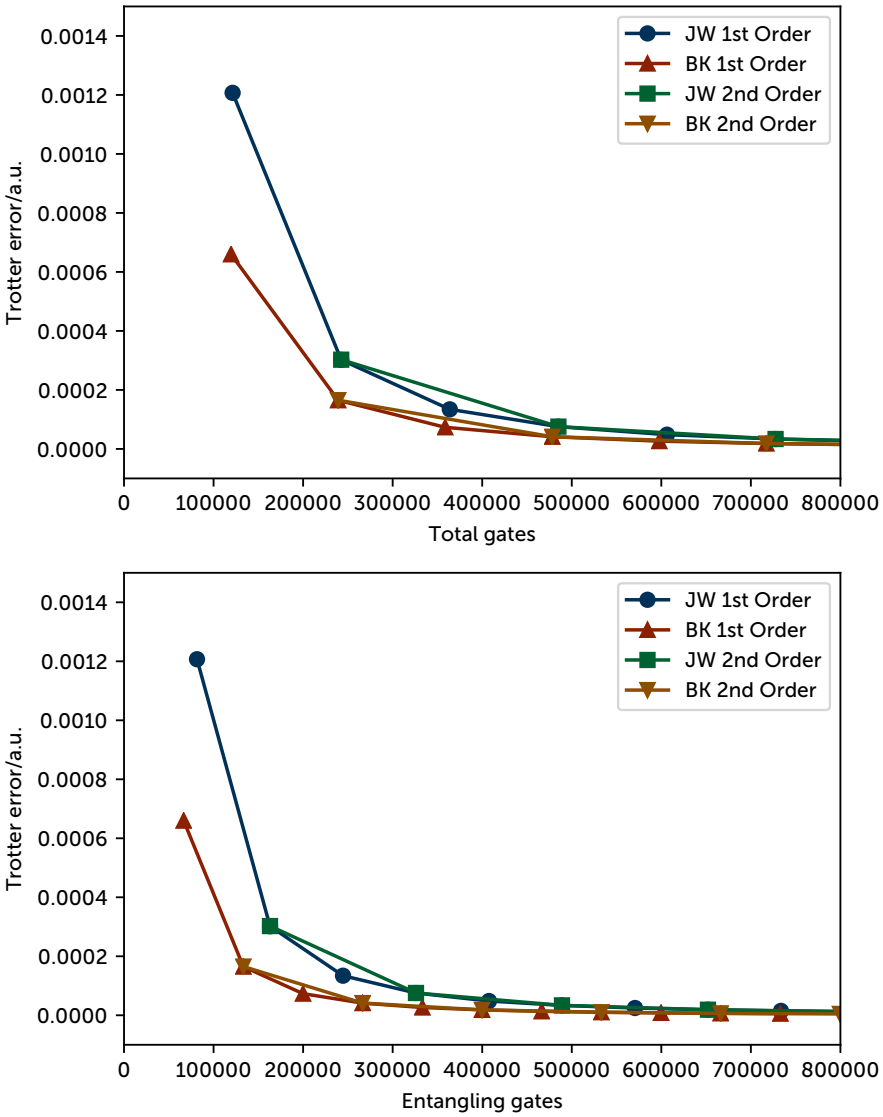



FIGURE 5.1.:
Trotterization error as a function of total gate count for methane using Bravyi-Kitaev and Jordan-Wigner circuits, with a first and second order Trotter-Suzuki approximant. Trotter error is calculated against a reference value of -53.32722732 a.u. The Bravyi-Kitaev mapping has a marginally lower gate count, and results in an approximately halved Trotter error. For entangling gates, the difference in gate count is more substantial.

sideration of the ordering scheme employed when performing Trotterization. Finally, note that the second order Trotter-Suzuki appears to provide almost identical results to the first order approximant. This fact informed our focus on the use of first-order Trotter approximants in later sections of this project.

The lower plot of Figure 5.1 demonstrates the same plot, but only considering CNOT gates. Here, we observe a more pronounced difference. The Bravyi-Kitaev mapping is markedly more efficient in terms of these gates, reducing the amount by 14% for one Trotter step. The effect is more pronounced at higher numbers of Trotter steps. This is important, as it is expected that on most architectures, CNOT gates will be markedly less reliable than single qubit gates. This result is somewhat to be expected, as the Bravyi-Kitaev transformation distributes the locality of both occupation and parity information. The increase in total gate count is due to the large number of necessary basis change operations required to incorporate the increased complexity of the Bravyi-Kitaev mapping. We return to this fact the following section.

## 5.2 Preliminary bulk studies

The analysis of methane highlighted the need for a large-scale assessment of the merits of the Bravyi-Kitaev mapping. In order to assess the validity of this, the code used to count gates for the methane analysis was used to

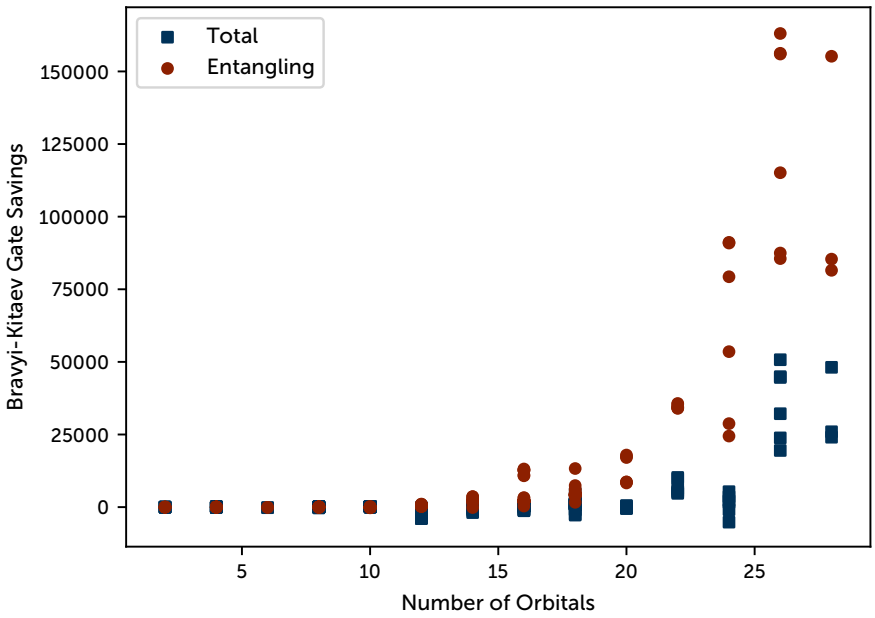



Figure 5.2.:
Preliminary analysis of the gate savings associated with the Bravyi-Kitaev mapping over the Jordan-Wigner mapping using the small_data set. The superior scaling of the Bravyi-Kitaev mapping does seem apparent at beyond approximately 20 spin-orbitals, although not before. The entangling gate savings are more substantial.

roughly count gates required to simulate one Trotter step for a large number of molecular systems. The initial analysis was performed for the small_data data set described in Chapter 3.

Our initial assessment was limited, using the process described above to generate gate counts without generating entire circuit objects. It furthermore did not consider the error incurred in the use of Trotter-Suzuki formulae, nor did it consider the impact of ordering choice. Nonetheless, this approach gave us a simple and indicative comparison of the two strategies.

Figure 5.2 shows the number of gate savings associated with the Bravyi-Kitaev

mapping in contrast to the Jordan-Wigner mapping, as a function of the number of orbitals involved in the chemical system (i.e. the number of qubits). Note that there appears to be a clear gate saving associated with the Bravyi-Kitaev mapping, particularly with larger systems. In light of the analysis on methane, this was reassuring. This saving is only consistent beyond approximately $N = 20$ qubits, which explains the lack of improvement demonstrated for methane, where $N = 18$.

Recall that the number of gates scales linearly with respect to the size of the molecular system in the Jordan-Wigner mapping, but logarithmically for the Bravyi-Kitaev mapping. Consequentially, we expect to see a sub-exponential scaling in the Bravyi-Kitaev gate count improvement. This roughly appears to be obtained. An even more positive picture is painted when examining the estimates for CNOT count.

### 5.2.1 *Optimised circuits*

In light of the previous results, it should be emphasised that this approach was extremely basic. For a full analysis, it was necessary to consider *optimised* circuits. This necessitated the development of the `Circuit` and `Gate` code described in Chapter 3.

In addition to optimisation performed through combination of duplicate Pauli strings, the code implements the gate cancellation scheme of Hastings, Wecker, Bauer and Troyer.[77,106] Examining two circuits for the simulation

of a Pauli string, it is notable that the total sequence may involve chains of duplicate self-inverse gates. These can be removed from the circuit with no impact upon the overall action of the circuit itself. Savings from this procedure arise from two factors. Firstly, redundancy in parity determination is eliminated, as this information is not decomputed after every term in the Hamiltonian. This results in savings in the CNOT string used to determine parities. Additionally, basis changes are not decomputed when unnecessary. This saves on the single qubit $H$ and $Y$ gates necessary to set these bases.[106]

As discussed above, the `fullCancelDuplicates` method allowed us perform this cancellation. This allowed us to yield a more nuanced understanding of the relationship between the Bravyi-Kitaev mapping and circuit optimisations. A reasonable argument would be that such gate cancellation *absorbs* the benefits yielded by the use of the Bravyi-Kitaev transformation. Our analysis enabled us to test this argument.

We first consider the Jordan-Wigner and Bravyi-Kitaev circuits both with and without cancellation of duplicate self-inverse gates. Initially, the Hamiltonians were ordered by the magnitude of their coefficients. This somewhat arbitrary ordering was chosen in order to assess the pre-optimisation efficacy of the Bravyi-Kitaev mapping, and is in contrast to optimised ordering schemes used in other work and discussed both later, and in Chapter 6.[79,106]

Figure 5.3 demonstrates gate counts for a single Trotter step of the determined circuits with respect to the length of the corresponding unoptimised circuit. As with the preliminary analysis, for the bulk of the systems studied

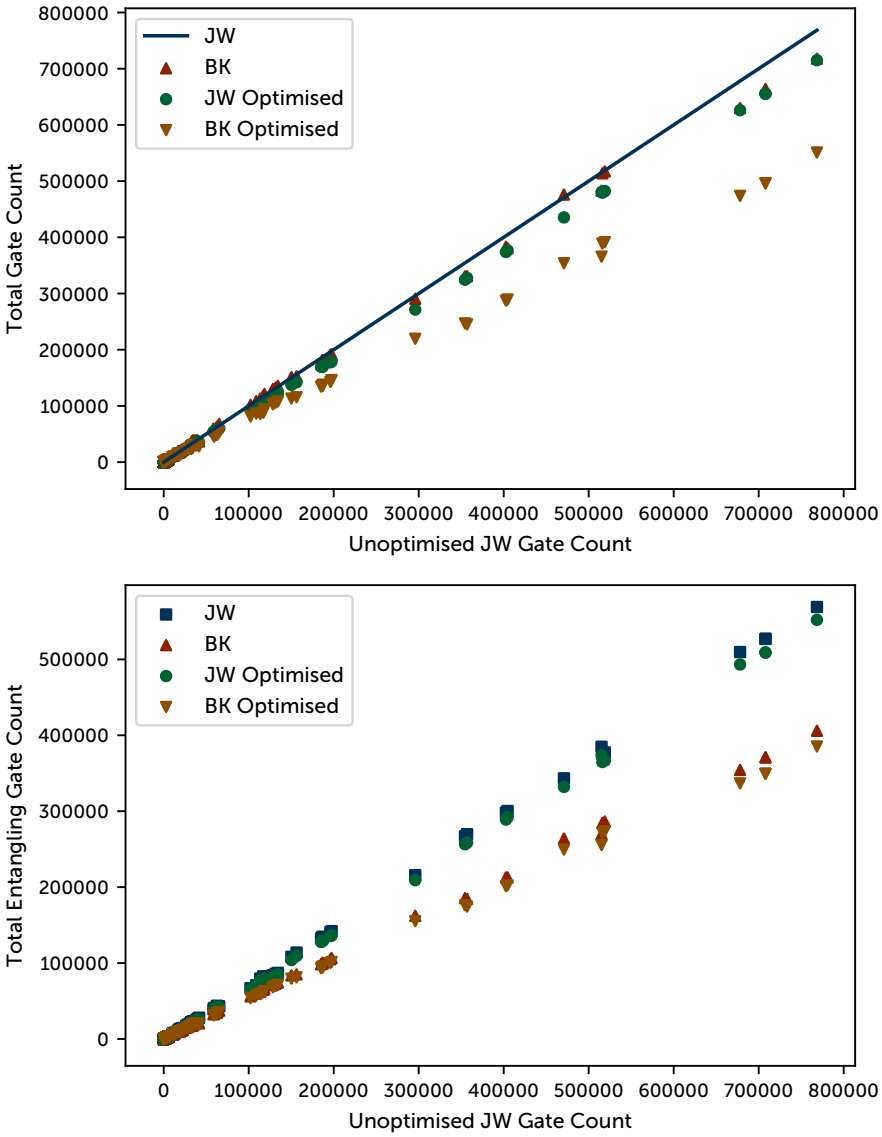



Figure 5.3.:
Gate count estimates of optimised circuits using the Jordan-Wigner and Bravyi-Kitaev mappings using the small_data set with a magnitude ordering. Upper: total gates. The optimised Bravyi-Kitaev mapping substantially outperforms the optimised Jordan-Wigner mapping and the unoptimised Bravyi-Kitaev mapping. Lower: entangling gates. The Bravyi-Kitaev mapping displays significant improvement, although little is gained from gate cancellation, showing that the majority of gates cancelled are single qubit gates.

we do see a small benefit from the use of the Bravyi-Kitaev mapping with unoptimised circuits, which is non-trivial at larger system sizes. With optimised circuits, a dramatically different picture arises. Here, although very short circuits show no noticeable difference, the Bravyi-Kitaev mapping demonstrates a clear superiority from even circuits of only 10000 gates in the unoptimised Jordan-Wigner circuit. With the largest circuits, the Bravyi-Kitaev mapping results in savings of about 25% of the unoptimised total. These savings scale up with increasing circuit length, as is to be expected. In the longest circuits, the gate savings obtained by the use of the Bravyi-Kitaev mapping substantially exceed that of the use of gate cancellation alone. This is attributable to the fact that gate cancellation results in dramatically less savings than should be expected. In turn, this is due to the choice of a magnitude ordering scheme, discussed below.

One curiosity of the data here is that in larger circuits, the savings obtained by performing gate cancellation on Jordan-Wigner circuits are almost identical to those obtained by using the unoptimised Bravyi-Kitaev mapping. Although it could be speculated that the gates cancelled when using the Jordan-Wigner mapping are same gates as the CNOTs saved by the use of the Bravyi-Kitaev mapping, caution should be used here, as our later results contradict this observation.

Figure 5.3 also shows the number of entangling gates saved by each optimisation scheme. The first important observation here is that very little entangling gate savings are obtained through the use of gate cancellation while using the Jordan-Wigner transformation. This suggests that the bulk of the

gate savings obtained using this scheme are single-qubit basis change operations. Given that the Bravyi-Kitaev mapping results in an increased number of these (in favour of reduced CNOT count), this provides an explanation for the increased level of gate cancellation associated with the Bravyi-Kitaev mapping.

More generally, the entangling gate savings observed are dramatic when using the Bravyi-Kitaev mapping, using both unoptimised and optimised circuits. In the unoptimised case, this is to be expected, according to the discussion above. As entangling gates are typically considered more difficult to implement, this could be a principle advantage of the Bravyi-Kitaev mapping.

From a fault-tolerance perspective, it is important to note that the majority of the circuit consists of Clifford gates.[36] The central arbitrary rotation gates in Figure 2.3 are the only non-Clifford gates present. The number of these remains the same when using either mapping, as this is equal to the number of terms in the qubit Hamiltonian. Consequentially, if error correction is to be considered, the T count (an important measure of difficulty for error correction) remains constant for both Bravyi-Kitaev and Jordan-Wigner circuits. This implies that the use of the Bravyi-Kitaev mapping or the Jordan-Wigner mapping does not have substantial implications for the cost of error correcting the quantum circuits. The observed CNOT savings are yielded from an increase in single qubit Clifford gates. In most cases, this is likely to be a worthwhile trade-off.

Examining the impact of gate cancellation on entangling gate count in the Bravyi-Kitaev circuits, we observe the same lack of savings obtained for Jordan-Wigner. This is in keeping with the above discussion.

### 5.2.2 *Impact of Trotter ordering*

The observed lack of gate cancellation on the CNOT chains is troubling, given that gate cancellation is known to reduce gate counts substantially.[106] It is likely that many CNOT strings are being trapped behind non-commuting gates earlier in their respective CNOT strings. This contradiction is attributable to the choice of ordering scheme.

As discussed above, the overall goal is to find a minimal length circuit that can implement the unitary evolution operator of the quantum Hamiltonian. As no standard circuit for the simulation of the exponentiated total Hamiltonian exists, a Trotter-Suzuki approximation (introduced in eqn. (2.4.0.8)) must be used.

The ordering of terms in this approximation is important. It has been demonstrated[93] that the error due to the utilisation of Trotter-Suzuki formulae strongly depends on the term ordering chosen. Additionally, the number of duplicate gates depends strongly on the ordering chosen. Both of these factors influence the length of the overall quantum circuit.

The previous sections utilised a magnitude ordering of Trotter terms. This ordering is significantly physically meaningful, as terms with higher mag-

nitude are likely to correspond to stronger physical interactions. However, it is also known to be suboptimal in terms of gate cancellation procedures.[106]

As the number of potential orderings grows factorially, the problem of finding an optimal ordering scheme is a difficult one. However, ordering schemes that are superior for the process of gate cancellation can be found, as the similarity of sequential Pauli strings - and thus the savings from cancellation - can be determined when specifying the Hamiltonian.

A lexicographic ordering is expected to maximise the gate savings obtained through cancellation, as the similarity of adjacent terms is maximised.[106] As the code optimises on the level of Pauli strings rather than fermionic operators, we also order on this level, with no ordering performed on the fermionic level. As discussed in Chapter 3, the code stores Pauli strings as lists of base 4 integers. A lexicographic ordering in this scheme is then essentially a bitwise numerical ordering.

The results using the small_data set and a lexicographic ordering are included as Figure 5.4. We observe a markedly different trend to that observed using a magnitude ordering. Firstly, note that gate cancellation has now resulted in a dramatically reduced overall gate count. Whereas using a magnitude ordering the longest optimised circuits involved around 500000 gates, they now involve around 200000. This is a reassuring endorsement of gate cancellation.

Here, in terms of total gate count, the optimised Bravyi-Kitaev and Jordan-Wigner circuits are roughly equivalent. Indeed, in larger circuits the Bravyi-

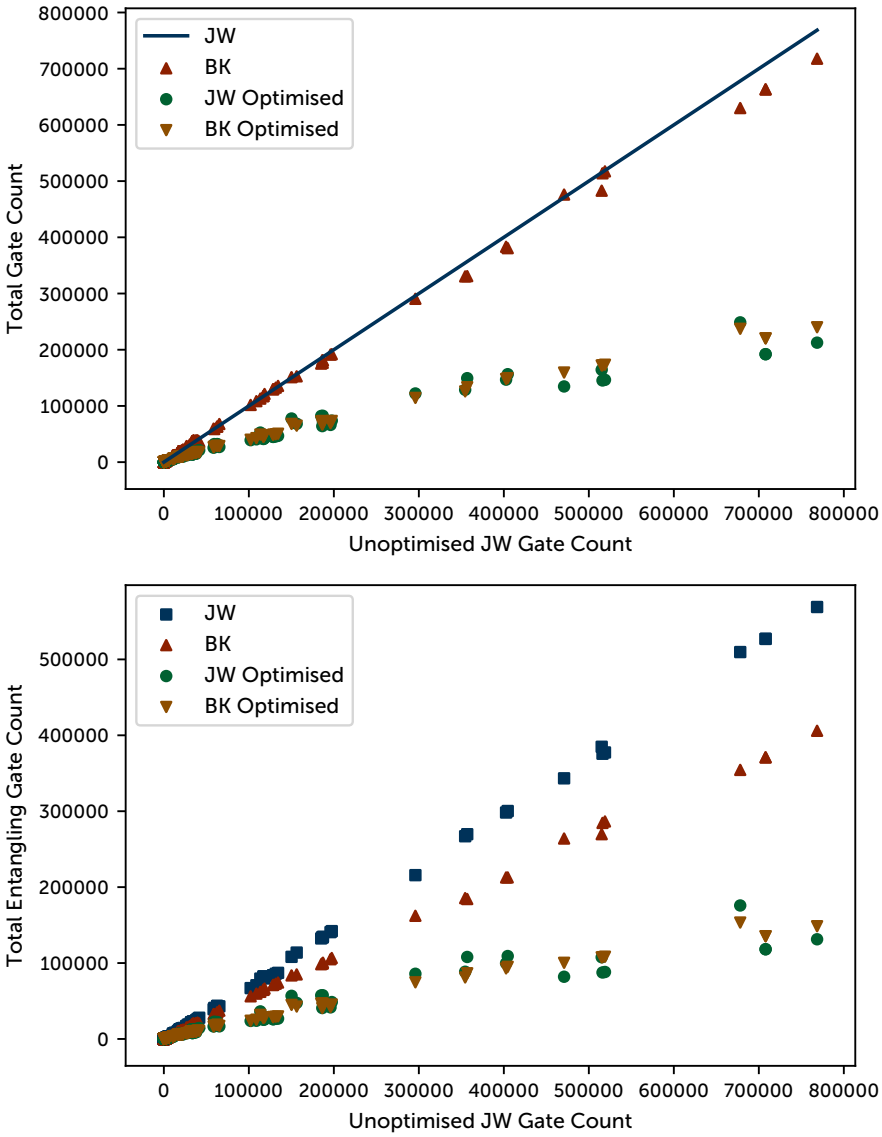



Figure 5.4.:
Gate count estimates of optimised circuits using the Jordan-Wigner and Bravyi-Kitaev mappings using the small_data set with a lexicographic ordering. Upper: total gates. The trend is dramatically reversed from the magnitude ordering, with optimised Jordan-Wigner circuits marginally outperforming the optimised Bravyi-Kitaev circuits. Lower: entangling gates. The Jordan-Wigner mapping is marginally superior. In contrast to the magnitude ordering, large entangling gate savings are observed from gate cancellation.

Kitaev transformation is marginally outperformed by the Jordan-Wigner transformation. This is a remarkable inversion of the previous situation. As this ordering scheme is optimal for gate cancellation, we can assume that this additional cost is due to the increased complexity of the Bravyi-Kitaev mapping preventing the cancellation of gates.

Inspection of the entangling gate count verifies that the bulk of the newly cancelled gates are CNOTs. This is in contrast to the magnitude ordering, where very few CNOT gates could be cancelled.

At this stage, it is tempting to conclude that the Jordan-Wigner transformation in a lexicographic ordering should be preferred in all cases. However, this conclusion cannot be drawn without consideration of the impact of ordering choice on the Trotter error obtained.

## 5.3 Larger Bulk Studies

While the analysis in Section 5.2 was useful for determining the direction of our future efforts, it suffered from several limitations. While some of the systems studied (included in Appendix A) involved several atoms, the overwhelming bulk of the data was from lone atoms. This is arguably not representative of examples that are *chemically* interesting. It was thus desirable to use a set of molecular systems involving a wider range of types of molecules.

More worryingly, the largest system size considered involved 28 spin-orbitals. This is within the range of systems that can be studied using conventional full configuration interaction approaches. As such, the analysis arguably could not claim to represent systems which would be studied in a meaningful way on actual quantum devices. A further problem of this lies in the expected crossover point of the Bravyi-Kitaev's superior scaling properties.

If we examine the Bravyi-Kitaev creation and annihilation operators (for even $j$, although an analysis of eqn. (4.2.2.7) reveals the following to also hold in the case of odd $j$):

$$a_j^\dagger = \frac{1}{2}\left(X_{U(i)} \otimes X_i \otimes Z_{P(i)} - iX_{U(i)} \otimes X_i \otimes Z_{P(j)}\right) , \tag{5.3.0.1}$$

$$a_j^\dagger = \frac{1}{2}\left(X_{U(i)} \otimes X_i \otimes Z_{P(i)} + iX_{U(i)} \otimes X_i \otimes Z_{P(j)}\right) . \tag{5.3.0.2}$$

Here, it is evident that at most $4\log_2 j+2$ gates are required for the simulation of one fermionic operation. This quantity is smaller than the simple $j$ gates of the Jordan-Wigner mapping when $j \geq 19$. Thus, noting that the Bravyi-Kitaev mapping is most efficient when $N$ approaches a power of 2 (as it can take increased advantage of its binary tree structure), we conservatively estimate that this point should be at $N \approx 32$. We thus would expect quantum computational cost to be reduced when using the Bravyi-Kitaev transformation for systems with more than $N \approx 32$ spin-orbitals. This analysis perhaps yields some insight as to the limited improvement observed with methane

and some of the molecules in the small_data set, as they involved fewer spin-orbitals than 32.

Consequentially, it was useful to generate a wealth of larger and more diverse examples. The large_data set was used for this purpose, as described in Chapter 3. The `directFermions` was then used to generate Jordan-Wigner and Bravyi-Kitaev Hamiltonians as oplists. From here, `Circuit` objects can be generated, optimised and their length determined.

Here, we restricted our attention to circuits simulating a Trotterized unitary with a first-order Trotter-Suzuki approximation and only one Trotter step.* Neglecting any benefits from optimisation at the interfaces of Trotter steps (which would save at most $O(N_{trotter})$ gates), the number of gates necessary for larger numbers of Trotter steps is a simple multiple of the number of gates necessary for one Trotter step. Consequently, extending the analysis to higher Trotter numbers was not considered necessary.

A full treatment of the entire phase estimation algorithm (including error correction) was not performed. This is in contrast to other work that has attempted to characterise the resource implications of performing the full procedure.[45,77] For the analysis of the raw gate counts of Bravyi-Kitaev circuits, this was not considered important for the above reasons. Note that at present, the largest simulations conducted have been of around 45 qubits.[43,113,114]

*Data was also generated for second-order approximants and with higher numbers of Trotter steps. However, the analysis of Trotter error in Chapter 6 shows that this was largely unnecessary for our purposes. For clarity, we do not consider these additional fifth and sixth degrees of freedom here.

These required extensive specialised computing resources. Performing simulations at this level would have made it impossible to perform a large, multi-system survey. In particular, assessing the point at which Bravyi-Kitaev scaling is expected to dominate (around 32 qubits, as discussed above) would have been problematic. Despite this, a full consideration of the phase estimation algorithm could yield useful results in regard to the Trotter error of simulation.

### 5.3.1 *Magnitude ordering*

As before, we first consider circuits without any form of gate cancellation using a magnitude ordering. As is expected,[77] the circuit length increases for larger systems, requiring of order $10^7$ gates for the simulation of systems involving around 50 spin-orbitals. While not as ruinous as the exponential difficulty of full configuration interaction on a classical device, this large circuit length illustrates the need for compiler optimisations. An initial assessment of circuits for the implementation of Jordan-Wigner and Bravyi-Kitaev Hamiltonians suggests that the use of the Bravyi-Kitaev mapping is associated with a substantial improvement for the larger systems. Figure 5.5 demonstrates this. From roughly 30 spin-orbitals, this improvement is consistent, and constitutes approximately 25% of the overall circuit length for the largest of the systems examined. However, many systems smaller than this see no improvement, or even demonstrate larger circuit lengths. This is in line with the earlier rough estimate that the superior scaling of the Bravyi-

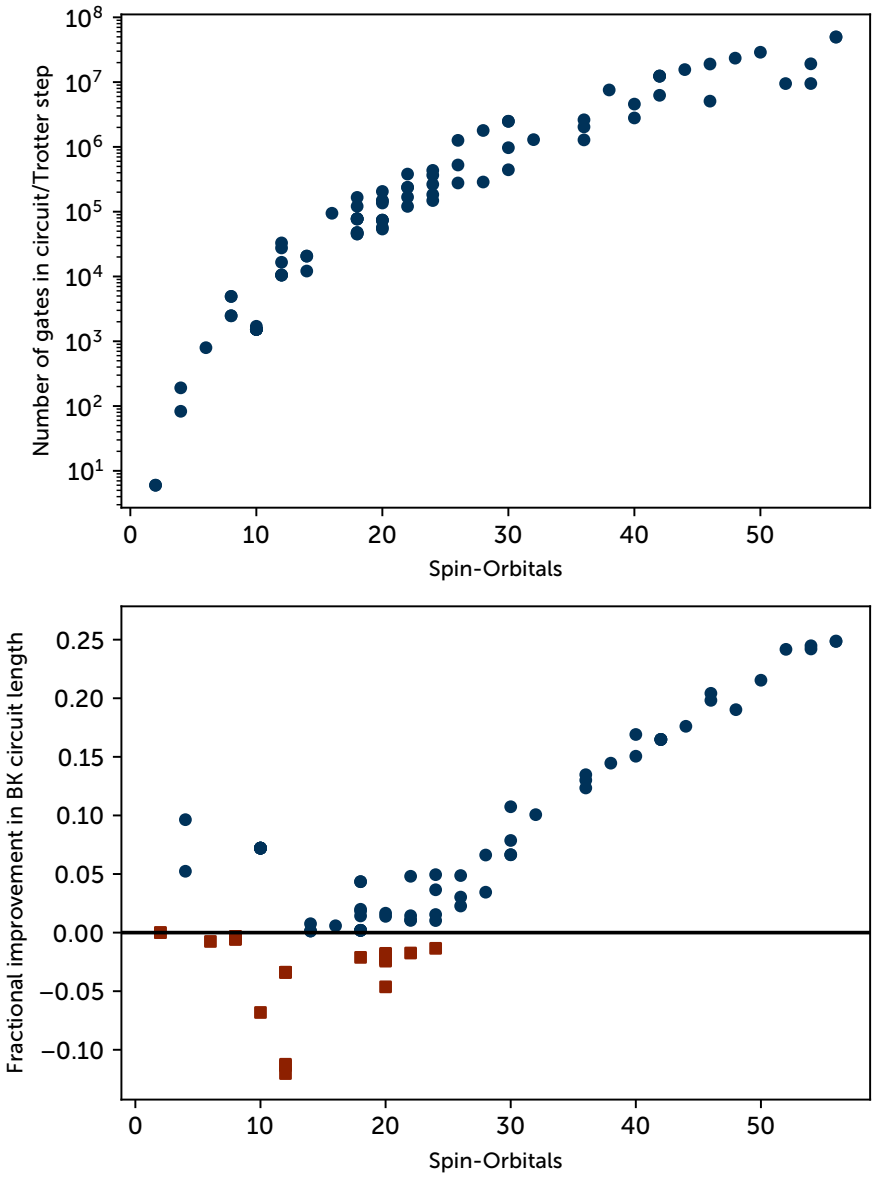



Figure 5.5.: Unoptimised gate counts. Upper: Total number of gates in Jordan-Wigner circuits, before optimisation. Lower: Gate savings through use of Bravyi-Kitaev mapping as a fraction of Jordan-Wigner gate count. Squares indicate instances where the Jordan-Wigner transformation outperformed the Bravyi-Kitaev transformation. In this scheme, the Bravyi-Kitaev transformation reliably results in shorter circuits from around 30 spin-orbitals, with up to around 25% gate savings in the examples with 50 spin-orbitals.

Kitaev mapping dominates the increased overhead at around 32 spin-orbitals. Full configuration interaction calculations have been performed on classical computers marginally larger than this (36 molecular orbitals).[115] Consequently, for simulations aiming to achieve results which are classically intractable, a naive approach involving no circuit optimisations would be substantially eased through the use of the Bravyi-Kitaev mapping.

When duplicate gates in the circuit are removed, the superiority of the Bravyi-Kitaev mapping is even more pronounced while still using a magnitude ordering, in agreement with the previous results. This relative improvement appears to increase with larger circuits, as demonstrated by Figure 5.6. In circuits involving more than about $10^7$ gates, the reduction in gate count associated with the use of the Bravyi-Kitaev mapping is larger than that of gate cancellation. In these cases, the use of the Bravyi-Kitaev mapping results in circuits that are approximately 30-40% shorter. Additionally, the number of gates cancelled using the Bravyi-Kitaev mapping is several times greater than the number of gates cancelled using Jordan-Wigner mapping. In some cases the advantage associated with the Bravyi-Kitaev mapping reduces the circuit length to that observed for systems involving fewer orbitals. For example, the Bravyi-Kitaev circuit for the simulation of the iodine atom (54 spin-orbitals) requires 5204912 gates per Trotter step, whereas the Jordan-Wigner circuit for the simulation of acetone (52 spin-orbitals) requires 8954933 gates per Trotter step.

With systems requiring more than $10^6$ gates to simulate, there are no examples where the optimised Jordan-Wigner technique outperforms the op-

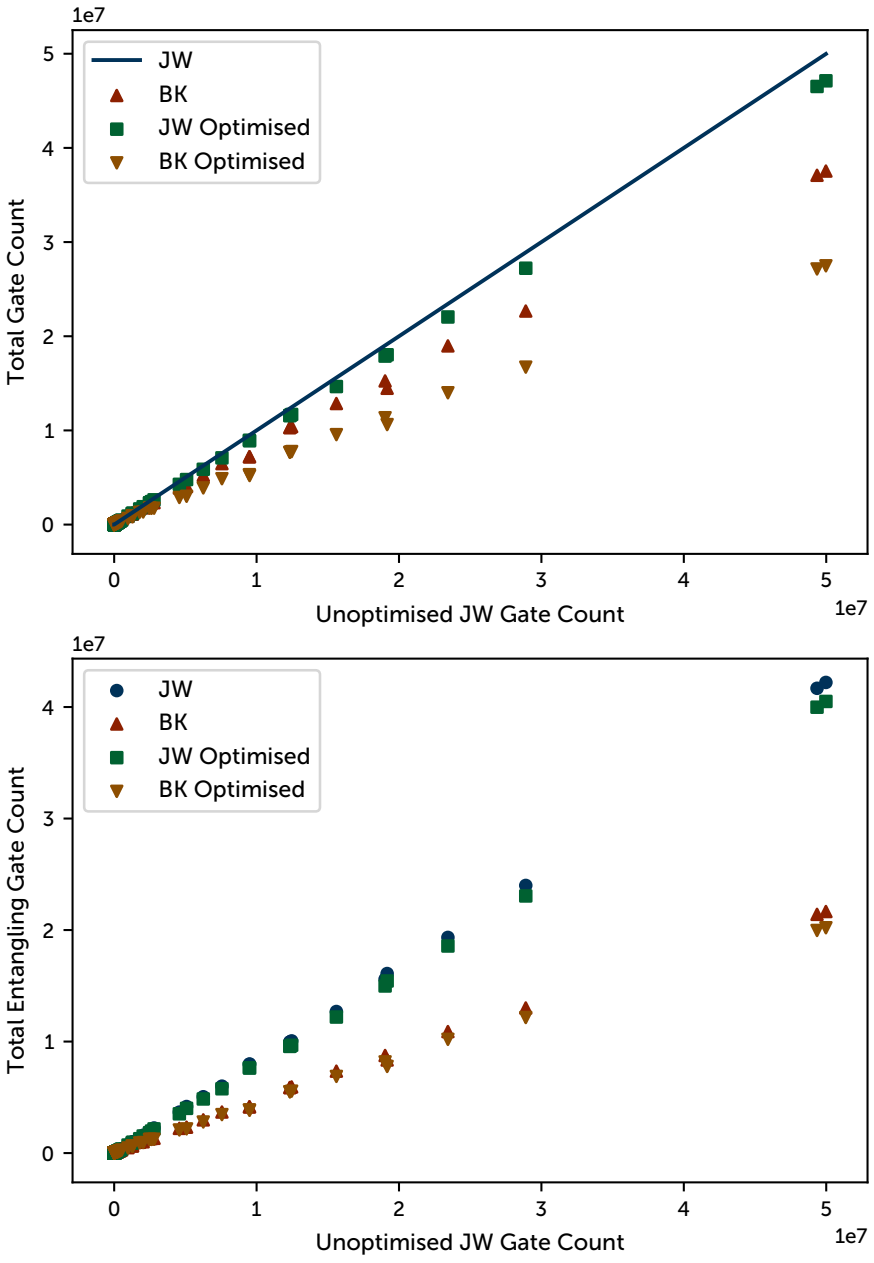



Figure 5.6.:
Number of gates in Bravyi-Kitaev and Jordan-Wigner circuits, before and after gate cancellation, versus the number of gates in Jordan-Wigner circuit before optimisation, using a magnitude ordering. Upper: Total gate count. The same pattern as Figure 5.3 is observed, however here the use of unoptimised Bravyi-Kitaev substantially outperforms even optimised Jordan-Wigner. Lower: entangling gate count. The same dramatic saving associated with the Bravyi-Kitaev mapping as for the SMALL_DATA set is observed.

timised Bravyi-Kitaev technique. Thus, using this magnitude ordering, it is clear that the Bravyi-Kitaev mapping should be preferred to the Jordan-Wigner mapping in the general case.

The above discussion on the methane molecule and with smaller molecules indicated that the Bravyi-Kitaev mapping may be advantageous with particular regard to the number of entangling gates required. The findings here confirm that this advantage holds in general, as shown in Figure 5.6. In addition to the general gate savings associated with the Bravyi-Kitaev mapping, we observe a substantial decrease in the number of CNOT gates required. As before, this advantage is typically offset by a small increase in the number of single qubit gates required (as the total savings are smaller than the entangling gate savings).

As before, the results of this initial section use a magnitude ordering for both the Jordan-Wigner mapping and the Bravyi-Kitaev mapping. Further analysis of the performance of the Bravyi-Kitaev mapping is impossible without consideration of the Trotterization ordering chosen. As discussed above, manipulation of the Trotter ordering can also cause variation in the Trotter error, which could result in an increased number of Trotter steps necessary for constant precision. This increased difficulty could undermine the savings gained from the use of a particular ordering in terms of cancellation, and is examined in Chapter 6. The fact that the Bravyi-Kitaev mapping only saw significant benefit beyond approximately 30 spin-orbitals confirms the necessity of analysis beyond this region, instead of focussing on only small examples.

### 5.3.2 *Impact of Trotter ordering*

As before, the results based on a lexicographic ordering are presented in Figure 5.7. Despite the use of the larger integral set, the conclusions to be drawn from this are strikingly similar to those of Section 5.2. Whereas using a magnitude ordering, the Bravyi-Kitaev mapping provided the bulk of the gate savings once gate cancellation had been performed, now the impact of the Bravyi-Kitaev mapping is relatively minor. The savings associated with the use of a lexicographic ordering far outweigh the savings associated with the use of the Bravyi-Kitaev mapping with a magnitude ordering: in the longer circuits included in the analysis, the former are approximately thrice that of the latter.

In these circumstances, the Jordan-Wigner and Bravyi-Kitaev mappings appear roughly equivalent for the smaller circuits. The Jordan-Wigner mapping outperforms the Bravyi-Kitaev mapping in some of the longer circuits, considering both total and entangling gate counts. In the longest circuits considered (propanol in a STO-3G basis), the use of the Jordan-Wigner mapping resulted in circuits that were approximately 25% shorter. We again attribute this to the relative complexity of the Bravyi-Kitaev mapping resulting in a reduction of linearity in the CNOT chains, which hampers gate cancellation.

Before considering other ordering schemes, it should be emphasised here that their error implications are difficult to predict. Bounds exist on the error yielded from Trotterization[77], although these are often loose[93]. A qualitative

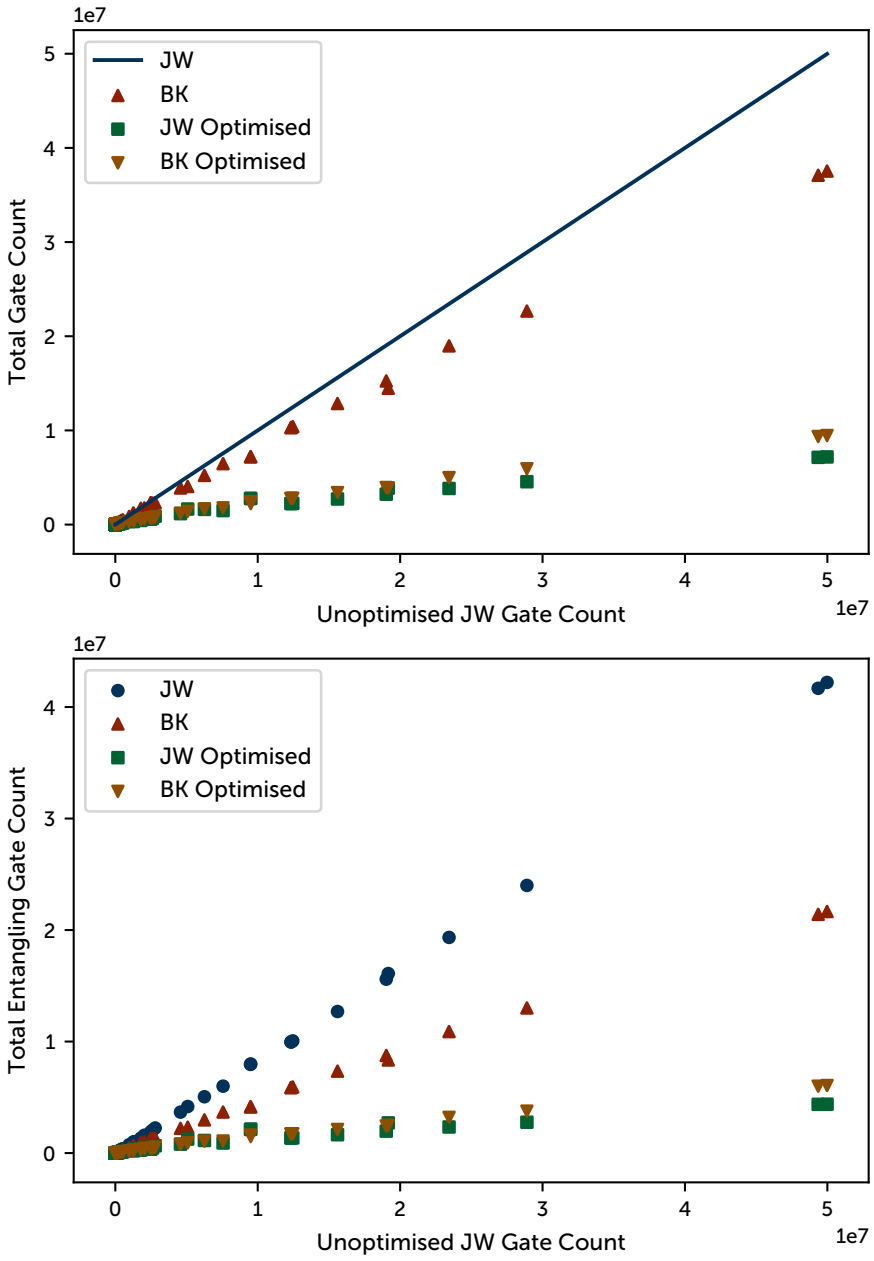



Figure 5.7.:
Number of gates in Bravyi-Kitaev and Jordan-Wigner circuits, before and after gate cancellation, versus the number of gates in Jordan-Wigner circuit before optimisation, using a lexicographic ordering. Upper: Total gate count. As with the small_data set, the Jordan-Wigner mapping is marginally superior for long circuits. Lower: entangling gates. Again, the Jordan-Wigner mapping is marginally superior.

estimate can be obtained through determination of the norm of the Trotter error operator; however, the quantitative behaviour of this still often overestimates the error in real chemical examples.[93]

Before considering the Trotter error itself (necessitating the study of only a subset of the data), it is useful to compare the implications of the Bravyi-Kitaev mapping when using several ordering schemes. To this end, the calculations were repeated using four ordering schemes. In addition to a single randomised ordering, a lexicographic ordering and an ordering of terms by magnitude, an additional ordering generated by regularly interspersing terms from the lexicographic and magnitude orderings is included. This ordering is relatively arbitrary and is intended for comparison purposes.

Across these orderings, the findings – shown in Figure 5.8 – are remarkably consistent, with the Bravyi-Kitaev mapping outperforming the Jordan-Wigner mapping in all cases apart from the lexicographic ordering. This advantage increases with the number of spin-orbitals used. Beyond $N = 10$ in all non-lexicographic cases, the advantage associated with the Bravyi-Kitaev mapping exceeds that of just using gate cancellation. This advantage is increased when considering CNOT count, and can be dramatic: when using a magnitude ordering, the savings associated with the Bravyi-Kitaev mapping are approximately an order of magnitude greater than those obtained by using gate cancellation alone.

For large systems, searching even a statistically significant subset of the space of possible orderings is clearly intractable, owing to the factorial growth of

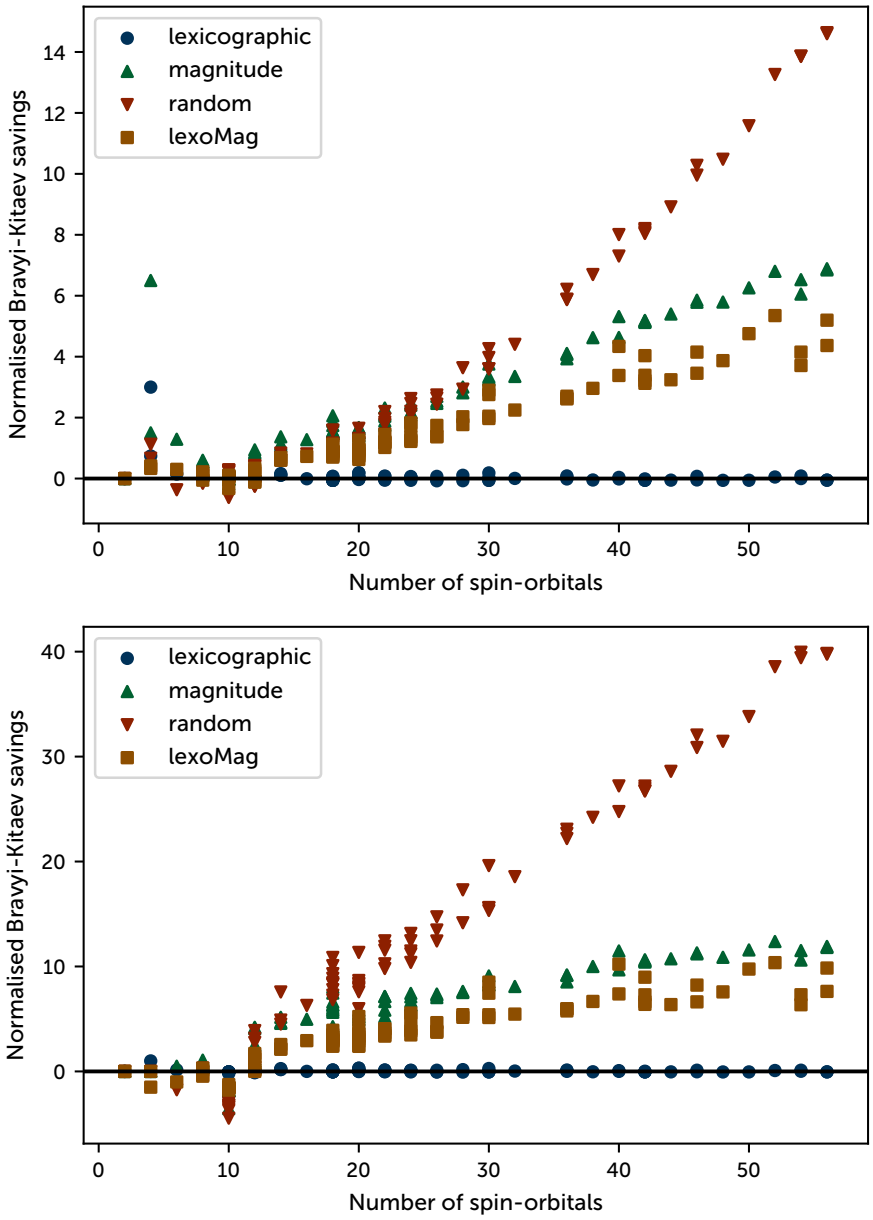



Figure 5.8.:
Gate savings associated with the Bravyi-Kitaev mapping; normalised by the gate savings acquired by using the same optimisation with the Jordan-Wigner mapping $(\frac{Gates_{Opt.BK} - Gates_{Opt.JW}}{Gates_{JW} - Gates_{Opt.JW}})$ for various ordering schemes. A value of 0 indicates that the optimised Bravyi-Kitaev and optimised Jordan-Wigner circuits have equal number of gates. A value of 1 indicates that the optimised Bravyi-Kitaev mapping outperforms the optimised Jordan-Wigner mapping by a number of gates equal to that saved by performing optimisation on the raw Jordan-Wigner circuit. Upper: total gates. The Bravyi-Kitaev mapping is significantly, and increasingly, superior for all orderings other than lexicographic. Lower: entangling gates. The same trend is observed as for total, but with substantially larger savings.

the number of possibilities. For each system, the random ordering is only one of these myriad choices. Consequentially, it does not represent a statistically meaningful representation of the bulk of the possible orderings. Nonetheless, it is interesting that this random choice qualitatively manifests the same trend as the other ordering schemes to a very large extent. Quantitatively, the advantage associated with the Bravyi-Kitaev mapping when using these random orderings is dramatic - for the largest example, the reduction in CNOT count using the Bravyi-Kitaev mapping was 40 times the reduction using gate cancellation alone. In the results, the use of a tailored ordering scheme (whether it be ordering lexicographically, by magnitude or otherwise) results in a reduction of advantage for the Bravyi-Kitaev mapping.

Although these figures suggest that the use of the Bravyi-Kitaev mapping results in shorter circuits when using most possible orderings, it should be emphasised that the lexicographic ordering dramatically decreases gate count through cancellation irrespective of mapping strategy. It should be restated that using this strategy, Figure 5.7 shows that employing the Jordan-Wigner mapping results in marginally shorter circuits than the Bravyi-Kitaev mapping.

The choice of Trotter ordering may be dictated by other factors - for instance, architecture constraints. In these circumstances, calculations will be dramatically shortened by the use of the Bravyi-Kitaev mapping, as this mapping results in reduced circuit length for all of the non-lexicographic orderings considered.

However, more importantly, and as discussed above and in Chapter 6, the choice of Trotter ordering has significant impact on the Trotter error.[77,79] Potentially, minimisation of Trotter error may therefore require an ordering being chosen which is suboptimal in terms of single Trotter step gate count, where the Bravyi-Kitaev holds a significant advantage.

As such, consideration of the Bravyi-Kitaev mapping within the context of different ordering schemes *requires* an estimation of the associated Trotter error. Studies on the hydrogen and methane molecules, described above and in[79,80], indicate that in these systems the use of the Bravyi-Kitaev mapping can result in a reduced Trotter error – although, for methane, insufficiently to decrease the number of Trotter steps required for accurate simulation. This motivates the analysis of Chapter 6, regarding Trotter error.

In general, we conclude here that the use of the Bravyi-Kitaev mapping does not result in a predictable improvement in gate count when using an ordering optimised for gate cancellation, ignoring Trotter error implications. This is in contrast with the substantial and predictable improvement observed with other orderings, particularly when a random ordering is used.

The fact that the Bravyi-Kitaev mapping is superior in all ordering schemes other than the lexicographic ordering reinforces the necessity for an analysis of Trotter error. While it could be argued that the Bravyi-Kitaev mapping should be the "default" choice (as even in a lexicographic ordering scheme, the overhead incurred is marginal), it could also be suggested that there is typically no reason to prefer any other ordering scheme due to the substantial

reduction in gate count observed; thus negating any benefit of the Bravyi-Kitaev mapping. However, for this argument to hold, it is necessary to ensure that the Trotter error of the lexicographic ordering scheme is not systematically worse, as this would increase the number of Trotter steps required for chemical precision, thereby increasing the overall gate count and potentially nullifying the benefit of gate cancellation. Such an analysis of Trotter error is presented in the next chapter.

# 6 Trotterization Ordering Schemes

In Chapter 2, we discussed the canonical method for the transformation of the evolution operator $\hat{U}(t)$ into individual, circuit-describable unitaries $\hat{U}_i(t)$ by the invocation of a Trotter-Suzuki approximation. This incurs an error – the *Trotter error* – which decreases with increasing *Trotter number*, i.e. the number of steps the overall evolution is divided into.

The error incurred in this approximation can be quantified by using the expectation value of the Trotter error operator with the ground state. For example, for the second order Trotter-Suzuki approximant (given by eqn. (2.4.0.7)), this expectation value is given by[93]:

$$\Delta E_i = -\frac{\Delta_t^2}{12} \sum_{\alpha \leq \beta} \sum_{\beta} \sum_{\gamma < \beta} \langle \psi_i | \left[ \hat{H}_\alpha \left( 1 - \frac{\delta_{\alpha,\beta}}{2} \right), \left[ \hat{H}_\beta, \hat{H}_\gamma \right] \right] | \psi_i \rangle \,, \tag{6.0.0.1}$$

where $\Delta E_i$ is the expected error (ignoring higher order terms), $\Delta_t^2$ is the Trotter step size, $|\psi_i\rangle$ is the ground state , $\delta$ is the Kronecker delta function and $\hat{H}$ are Hamiltonian terms.[93]

Examining this, it is evident that two factors primarily affect the degree of error introduced by the use of Trotter's approximation. Firstly, the error will decrease with the number of time steps used. The use of higher order Trotter-Suzuki approximants improves the polynomial degree of the error, but with increased overhead. Additionally, as the sums in eqn. (6.0.0.1) are ordered - being sums over conditional indices, rather than the whole range - the order in which terms are applied will impact the error incurred.

The overall number of gates required for implementation increases linearly with the number of time steps, $N_{trotter}$. Consequently, given a quantum computer capable of implementing an extremely large number of gates coherently, a large number of time steps could be used in order to guarantee an accurate result. However, devices of the near- to medium- term future are likely to be constrained significantly by the number of gates that can be performed while maintaining coherence. In order to perform practical calculations in this time-frame, it is thus desirable to consider practical methods of reducing the number of time steps required to reach a desired precision.

The results in Chapter 5 suggest that – at least for systems of less than 100 spin-orbitals – only a single-digit number of Trotter steps would be required for the first bit of precision in the ground state energy. However, the number of Trotter steps required for the reliable determination of higher bits of

precision for larger systems is unknown. While analytic bounds on the Trotter error exist[77], Babbush, McClean, Wecker, Aspuru-Guzik and Wiebe[93], by computing the expectation value of the error operator in an exact ground state, demonstrated that these can overestimate the Trotter error by up to sixteen orders of magnitude. Indeed, as numerically computing the exact Trotter error requires solving the full configuration interaction problem, it is likely that such a task could only be tractably performed on a scalable quantum computer itself.

Previous work has suggested that the Trotter ordering can dramatically impact the number of Trotter steps required for constant precision.[79,106] As such, the focus here was to consider Trotter ordering schemes, in an attempt to find a method which minimises the overall quantum computational cost of simulation.

Babbush, McClean, Wecker, Aspuru-Guzik and Wiebe[93] used the norm of the error operator to study its dependence on the underlying chemistry of the system, noting a remarkable correlation between the error operator norm and the maximum nuclear charge of the atoms of the system. The error operator norm can be determined tractably on a classical device. However, they show that while the error operator norm accurately reproduces the qualitative behaviour of the Trotter error, it quantitatively overestimates the error.

As such, we instead consider the Trotter error through explicit calculation. As discussed in Chapter 3, the code can use the `sparseFermions` mod-

ule to generate a sparse matrix representation of the Hamiltonian, which can be diagonalised to find an exact ground-state eigenvector and eigenvalue. The `directFermions` module can simulate the action of a Trotterized operator (at a given Trotter order and number of steps), and compare this against the exact eigenvalue to estimate the Trotter error.

It should be emphasised that this can only be performed (with reasonable computational cost) for relatively small systems. Using the methods described here, the largest practical calculations can be performed on systems of approximately 30 qubits. The eigenvalues generated here can be compared against traditional full configuration interaction calculations obtained through direct diagonalisation of the Hamiltonian for verification purposes. This eliminates the problem of requiring an explicit matrix representation of the unitary evolution operator. Nonetheless, the initial determination of ground state eigenvectors remains exponentially difficult, requiring on the order of hundreds of gigabytes of RAM for examples with more than 20 spin-orbitals.

This approach is, however, still appropriate for smaller systems. Our method here, then, is to analyse such small systems, look for patterns, and use these to inform development of algorithmic approaches to Trotter ordering. In this chapter, we first discuss such an analysis in an attempt to derive an optimal strategy for Trotter order. We then return to the Bravyi-Kitaev mapping, and discuss how this mapping performs with regard to Trotter ordering.

## 6.1 Molecular hydrogen

Molecular hydrogen is the smallest and simplest chemically interesting electronic structure problem – barring its cationic form, which lacks two-electron interactions. Notably, quantum simulation of its electronic structure has been experimentally demonstrated, as discussed in Chapter 1. In its most naive form, using the Jordan-Wigner transformation, it consists of 15 terms:

$$\begin{aligned}\hat{H} = &- 0.81261I + 0.17120\sigma_0^z + 0.17120\sigma_1^z - 0.22280\sigma_2^z \\ &- 0.22280\sigma_3^z + 0.16862\sigma_1^z\sigma_0^z + 0.12055\sigma_2^z\sigma_0^z + 0.16587\sigma_3^z\sigma_0^z \\ &+ 0.16587\sigma_2^z\sigma_1^z + 0.12055\sigma_3^z\sigma_1^z + 0.17435\sigma_3^z\sigma_2^z \\ &- 0.04532\sigma_3^y\sigma_2^y\sigma_1^x\sigma_0^x + 0.04532\sigma_3^x\sigma_2^y\sigma_1^y\sigma_0^x \\ &+ 0.04532\sigma_3^y\sigma_2^x\sigma_1^x\sigma_0^y - 0.04532\sigma_3^x\sigma_2^x\sigma_1^y\sigma_0^y\,. \end{aligned} \tag{6.1.0.1}$$

Unfortunately, even a Hamiltonian of 15 terms provides on the order of $10^{12}$ possible orderings. It is thus infeasible to consider all orderings for the full Hamiltonian, even for the hydrogen molecule in a minimal basis. The second order Trotter error operator (eqn. (6.0.0.1)) consists of a sum of many triple commutators between Hamiltonian terms. As such, it is prudent to examine the commutativity structure of the hydrogen Hamiltonian.

Figure 6.1 demonstrates this structure graphically. Here, nodes correspond to Hamiltonian terms, with edges drawn between terms that do not commute. It is immediately apparent that seven terms commute with all other

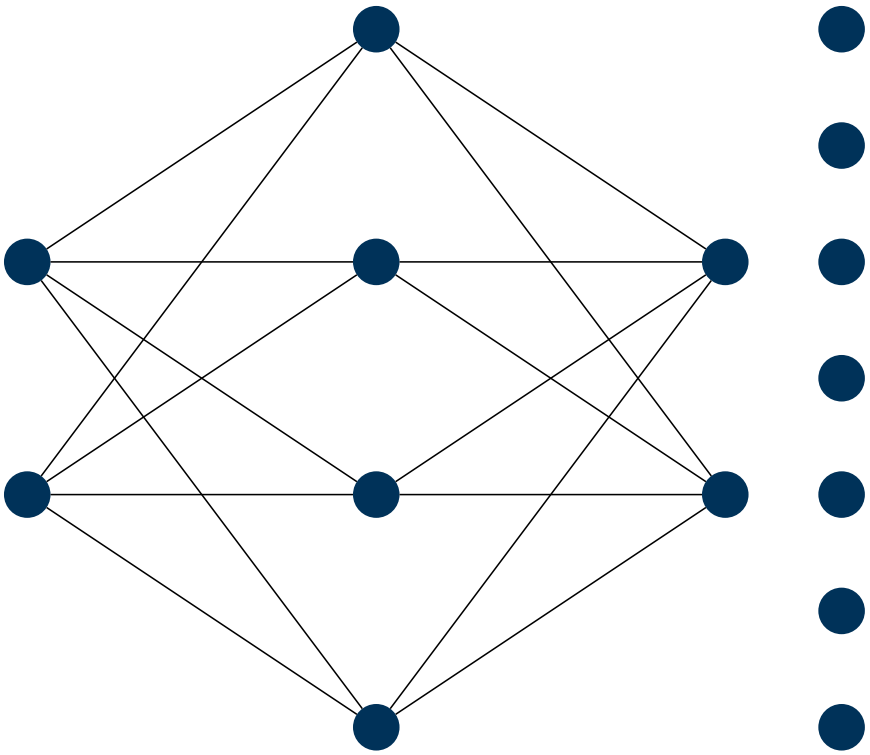


Figure 6.1.:
The commutativity graph of both the Jordan-Wigner and Bravyi-Kitaev Hamiltonians. Nodes correspond to Hamiltonian terms, edges correspond to non-commutativity between terms. Note the set of isolated nodes on the right-hand side, containing the terms that commute with all other terms.

terms, forming a *totally commuting set*. Because they commute with all terms, their exponentials commute with the exponentials of all other terms, and consequentially they can be freely moved around a Trotterized evolution operator. As such, all terms in this set can be moved to the front of the operator, combined, and simulated as an entirely independent operator

$$e^{\frac{-i\hat{H}t}{\hbar}} = e^{\frac{-i\hat{H}_C}{\hbar}} T\left(\hat{H}_{NC}\right) , \tag{6.1.0.2}$$

where $\hat{H}_C$ is the sum of all totally commuting terms, and $T\left(\hat{H}_{NC}\right)$ is a Trotterization of the other terms. In other words, we do not need to Trotterize the operators within this set, and thus only need to Trotterize the terms that do not commute with all terms.

This analysis has two principal advantages. Firstly, as the totally commuting set does not require a Trotter-Suzuki approximation, these terms can be simulated in one time step. This dramatically cuts down on the number of gates required for implementation on a quantum simulator. It should be noted here that for larger systems the size of the totally commuting set rapidly drops to one term (the identity operator). As such, this advantage is not scalable, and therefore this technique cannot be used directly in the analysis of larger systems.

More importantly for the purposes of this discussion, as we do not need to Trotterize the terms in the totally commuting set, the space of possible orderings is dramatically reduced to 40320. As the simulation of each ordering

takes less than half a second on an average laptop computer, it is thus feasible to simply brute force the entire space of orderings in order to study the distribution of errors.

In the simulations, integral data was generated using the Psi4[104] quantum chemistry package, with a standard Hartree-Fock basis used to express the molecular Hamiltonian. The bond length was set at 0.74Å and a STO-3G basis used. According to the methodology described in Chapter 3, the code was used to generate both Jordan-Wigner and Bravyi-Kitaev Hamiltonians in oplist format. The `sparseFermions` module was then used to find the exact ground state and its corresponding energy. Similarly, the Trotter error for every possible ordering was determined, with the overall evolution time and the number of Trotter steps set to unity.

Figure 6.2 is a cumulative density plot showing the distribution of Trotterization errors which are obtained as a result of ordering variations. Errors are given relative to the true eigenvalue of the Hamiltonian with totally commuting terms removed. The same distribution is obtained regardless of whether the Jordan-Wigner or Bravyi-Kitaev Hamiltonian is simulated. This is unsurprising, given the direct relationship between the two mapping schemes. The distribution is heavily weighted towards the low error region. This is promising, as it implies that a random choice of ordering is likely to introduce a low amount of ordering-dependent error. There is, however, over an order of magnitude difference between the optimal and poorest orderings. In this case, the difference is such that only three Trotter steps are required to reach an accuracy of 0.0001*a.u.* for the optimal ordering, whereas

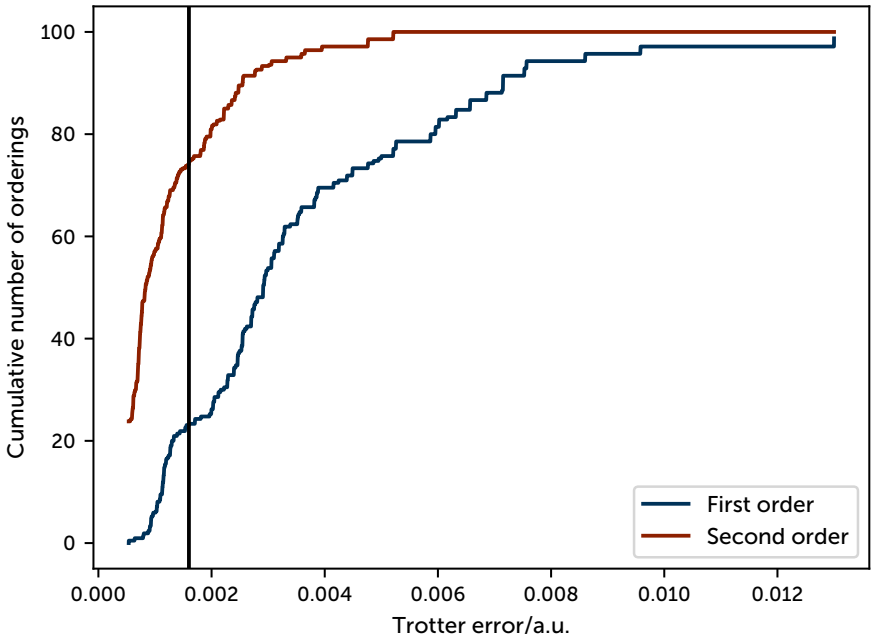



Figure 6.2.:
Cumulative density plot of hydrogen ordering errors for one Trotter step. The vertical line denotes an error of 1 kcal/mol. Approximately 20% of the orderings achieve this error or lower for the first-order Trotter-Suzuki approximation. Around 80% of orderings achieve an error of 0.005 Hartree, approximately half that of the worst possible ordering.

seven Trotter steps are required for the worst ordering. This implies that – at least for the hydrogen molecule – picking a bad ordering could result in a dramatically increased gate count in actual quantum simulations. This is important despite the high likelihood of random orderings being accurate, as if a systematic approach to ordering is taken, it must be ensured that this does not result in bad orderings being selected.

Using a second order Trotter-Suzuki approximation yields a similar trend as the first order approach, although the curve "peaks" at a lower error. In a sense, this is indicative of the second order approach being more robust to

ordering choice. While this is encouraging, it should be remembered that the second order approximant doubles the cost of simulation per Trotter step.

We now consider the optimal ordering strategy for the hydrogen molecule. For this, we re-examine the commutativity structure graph discussed above. Figure 6.3 shows this with totally commuting terms removed. With this, it is apparent that the commutativity graph is bipartite – it can be easily coloured into two independent sets. As such, the Hamiltonian can be subdivided into two sets of terms. Within each set, all terms mutually commute. No term in a given set commutes with any term in the other set. One set consists of all of the terms with Pauli Z operators, whereas the other set consists of terms with Pauli X and Y operators. Using the Jordan-Wigner transformation, the optimal ordering is given by

$$\begin{aligned}\hat{H}_1 =& +0.04532\sigma_3^x\sigma_2^y\sigma_1^y\sigma_0^x - 0.22280\sigma_2^z - 0.04532\sigma_3^y\sigma_2^y\sigma_1^x\sigma_0^x \\ & - 0.22280\sigma_3^z - 0.04532\sigma_3^x\sigma_2^x\sigma_1^y\sigma_0^y + 0.17120\sigma_1^z \\ & + 0.17120\sigma_0^z + 0.04532\sigma_3^y\sigma_2^x\sigma_1^x\sigma_0^y\,, \end{aligned} \tag{6.1.0.3}$$

where the leftmost term is simulated first, the second term is simulated second, and so on. Here, the blue terms are members of the XY set, and the red terms are members of the Z set. Examining this, it is evident that the optimal ordering strategy is given by alternating between sets in order of descending coefficient magnitude. Although instinct may suggest to group commuting terms together, an examination of the error operator (eqn. (6.0.0.1)) suggests

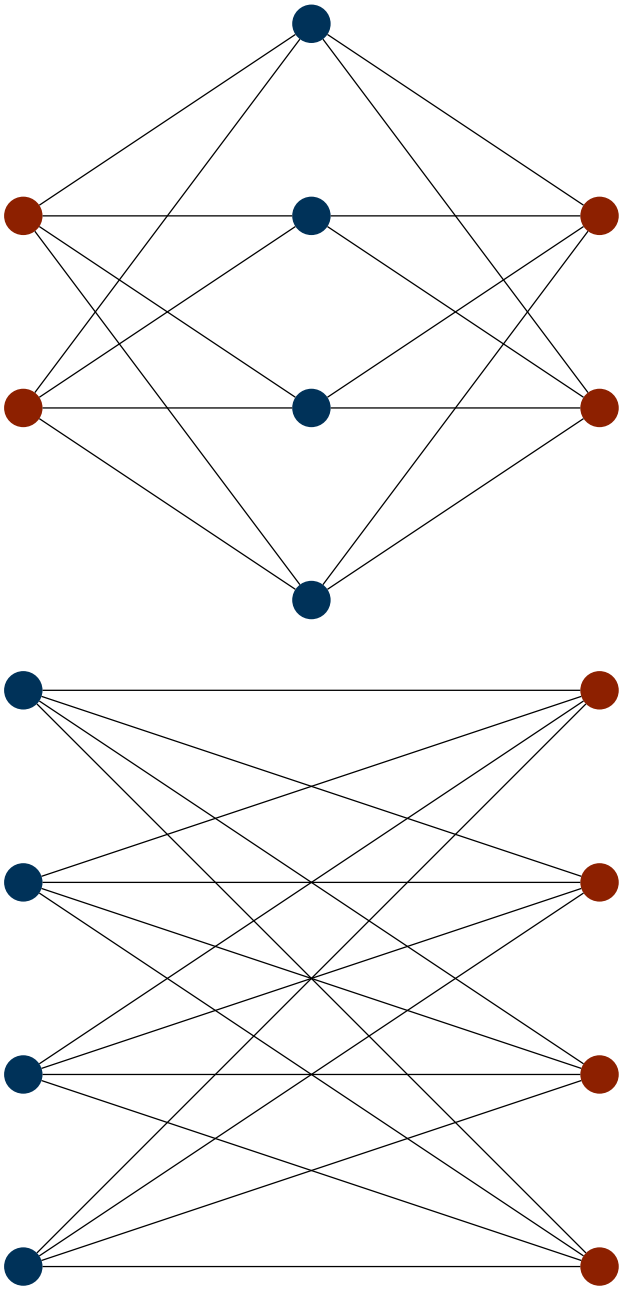


Figure 6.3.:
Two representations of the commutativity graph of the Jordan-Wigner and Bravyi-Kitaev Hamiltonians, with the totally commuting set removed. Nodes correspond to Hamiltonian terms, edges correspond to non-commutativity between terms. The lower diagram clearly reveals the two independent sets, with the XY set coloured blue and the Z set coloured red.

an intuitive explanation for why this is incorrect. As the non-commutativity is "spread out" through the Hamiltonian, earlier terms will feed less terms into the error operator. This strategy is optimal for both Jordan-Wigner and Bravyi-Kitaev Hamiltonians. However, other systematic approaches to ordering strategies for this Hamiltonian produce differing results for the two mapping techniques. This emphasises the importance of ordering analysis, and the difficulty of comparing the techniques without such considerations.

Our explicit discussion of the commutativity structure of the hydrogen molecule Hamiltonian is, of course, only possible due to its extremely small size and low complexity. Nonetheless, it is an important system, particularly in the context of experimental demonstrations of quantum simulation. The Bravyi-Kitaev Hamiltonian for the hydrogen molecule

$$\begin{aligned}\hat{H} = &- 0.81261I + 0.17120\sigma_0^z + 0.16862\sigma_1^z - 0.22280\sigma_2^z \\ &+ 0.17120\sigma_1^z\sigma_0^z + 0.12055\sigma_2^z\sigma_0^z + 0.17435\sigma_3^z\sigma_1^z \\ &+ 0.04532\sigma_2^x\sigma_1^z\sigma_0^x + 0.04532\sigma_2^y\sigma_1^z\sigma_0^y + 0.165868\sigma_2^z\sigma_1^z\sigma_0^z \\ &- 0.12055\sigma_3^z\sigma_2^z\sigma_0^z - 0.22280\sigma_3^z\sigma_2^z\sigma_1^z + 0.04532\sigma_3^z\sigma_2^x\sigma_1^z\sigma_0^x \\ &+ 0.04532\sigma_3^z\sigma_2^y\sigma_1^z\sigma_0^y + 0.16587\sigma_3^z\sigma_2^z\sigma_1^z\sigma_0^z \end{aligned} \tag{6.1.0.4}$$

can be examined. The above analysis showed that the distribution of Trotter errors with respect to ordering is the same here as for the Jordan-Wigner Hamiltonian. However, in subsequent collaborative work[62] it was noted that, due to the specific structure of the hydrogen Hamiltonian, the Bravyi-Kitaev Hamiltonian acts off-diagonally on only the 0 and 2 qubits. This al-

lows for the Hamiltonian to be scalably reformulated as a two qubit, six term Hamiltonian. This strategy of reducing the number of qubits was later generalised by Bravyi, Gambetta, Mezzacapo and Temme[116], which subsequently enabled the simulation of the beryillium hydride molecule by Kandala, Mezzacapo, Temme, Takita, Brink, Chow and Gambetta.[63]

Following the above method, the totally commuting terms can be removed from the two qubit hydrogen Hamiltonian, to reduce the size of the two qubit Hamiltonian to four terms. In keeping with the full hydrogen Hamiltonian, the number of possible orderings for the four term Hamiltonian can then be brute forced to find the optimal ordering scheme with regards to Trotter error. This enabled the simulation of the hydrogen molecule in reference [62].

## 6.2 Generalised ordering strategies

While analysis of the hydrogen molecule yielded interesting results regarding the spread of ordering strategies, it would be desirable to find an ordering strategy that is effective in the general case. For this, we require an analysis of a variety of systems, and ideally those of a chemically interesting size. While the hydrogen study does not provide this, the optimal ordering strategy observed does provide us with a potential starting point for our investigation.

We observed in the previous section that the optimal ordering strategy for molecular hydrogen was to divide the Hamiltonian into two mutually commuting subsets, and intersperse them so as to minimise the number of nonzero terms in the error operator. This procedure is not viable for larger systems, as there is no guarantee as to the number of mutually commuting sets. Although a graph describing the commutativity structure of the Hamiltonian terms can be generated tractably (if slowly), finding mutually commuting sets is essentially a process of graph colouring. This introduces a host of computational problems, well studied in graph theory. Finding the chromatic number of a graph (in our case, the *minimal number of mutually commuting sets*) is known to be NP-hard.[117]

Graph colouring schemes could be used to divide the Hamiltonian into mutually commuting sets, for example through a greedy colouring algorithm. It remains an open question as to the efficacy of this with regards to Trotter error. A variety of factors could be considered here. For example, it is not immediately obvious whether it would be preferable to split the graph into few large independent sets, or many smaller ones. Future work would be interesting in this regard; however, an alternative strategy is instead discussed here.

The present scheme is still inspired by the optimal hydrogen ordering. However, rather than attempting to find specific sets, it instead attempts to order by minimising the amount of commutativity in the ordered Hamiltonian at each step.

We start by ordering the Hamiltonian by coefficient magnitude. The highest magnitude term is then assigned to be the first term in the ordered Hamiltonian. We then consider the terms that do not commute with this, and the highest magnitude of these is appended to the ordered Hamiltonian. We keep track of the number of terms in the ordered Hamiltonian with which each term in the unordered Hamiltonian commutes, and we consider the subset of unordered Hamiltonian terms that minimise this number. Of these, the term with highest magnitude is chosen as the next term in the ordered Hamiltonian. This process is repeated until the unordered Hamiltonian is exhausted. The left-hand chart of Figure 6.4 demonstrates this process diagrammatically. We refer to this scheme as a *commutator* ordering.

It should be noted that this process does not reproduce the exact optimal strategy for the molecular hydrogen test case, as in this instance the lower magnitude XY term group instead begins the ordering process. Nonetheless, it does produce an ordering which performs almost equivalently.

In addition to this scheme, a slightly modified scheme was considered. In this scheme – the *reverseCommutator* ordering – we again start with an unordered Hamiltonian comprising of a list of terms. At each stage, instead of counting the number of terms in the ordered Hamiltonian that the candidate terms commute with, we count the number of terms in the *unordered* Hamiltonian that the candidate terms commute with. We then pick the terms that *maximise* this number, and append the highest magnitude of these. This process is depicted in the right-hand chart of Figure 6.4.

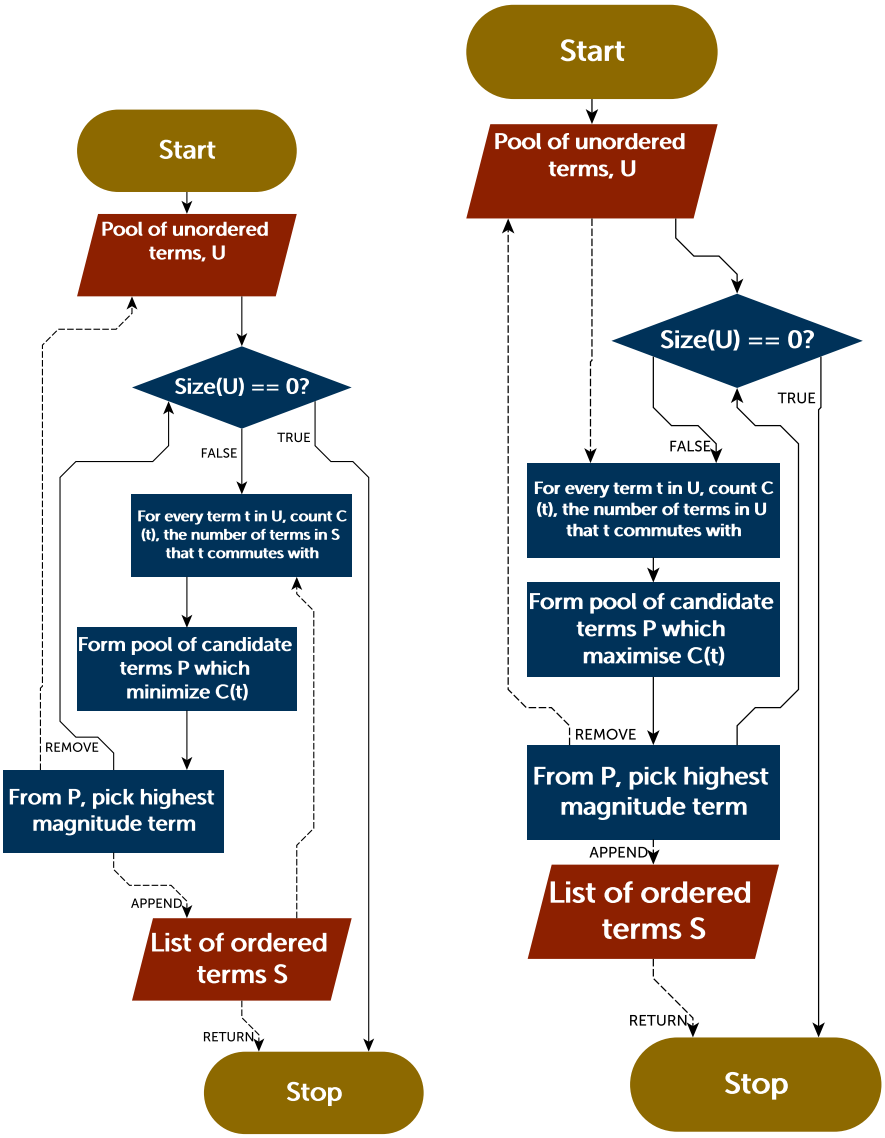



Figure 6.4.: Flowcharts representing the commutator (left) and reverseCommutator (right) ordering schemes.

Both of these algorithms are essentially greedy. The *commutator* ordering at each stage attempts to maximise the non-commutativity of the ordered Hamiltonian, whereas the *reverseCommutator* ordering at each stage attempts to minimise the non-commutativity of the unordered Hamiltonian.

### 6.2.1 *Other ordering schemes*

In order to contrast our ordering schemes against other possible alternatives, it is necessary to briefly review the conventions we use to describe other ordering schemes. A comparison of the Jordan-Wigner molecular hydrogen Hamiltonian in oplist format in each of these cases is included as Appendix C.

Perhaps the most immediately obvious ordering scheme is the *magnitude* ordering. Here, terms are ordered according to the magnitude of their coefficient, from largest to smallest – the term with the largest magnitude coefficient is simulated first, followed by the second largest magnitude coefficient, and so on. This ordering scheme has the immediate appeal that high magnitude terms are simulated first. As a result of this process, one could expect a reduction in the number of high magnitude terms in the error operator. However, this approach does not take into consideration the structure of the error operator – high magnitude terms may not result in a meaningful increase in error if, for example, they commute with many other terms.

The *lexicographic* ordering is an ordering scheme which attempts to maximise the similarity of the Pauli strings of adjacent terms. In the oplist format, this is essentially a numerical ordering with respect to the Pauli strings. While this scheme may seem arbitrary, it is known to result in a maximum amount of gate cancellation, with commensurate reduction in overall quantum computational cost.[106] However, there is little reason to suspect that this ordering would be beneficial for the purposes of Trotter error. Such an increase could lead to the requirement for a greater number of Trotter steps, undermining its advantages.

It should be noted that our strategy for ordering terms lexicographically is a small modification from that used in other work.[106] As discussed in Chapter 3, whereas other approaches have grouped the qubit Hamiltonian terms by their fermionic operator, our approach does not maintain this structure, and instead stores the qubit Hamiltonian as a list of individual weighted Pauli string terms. As such, we do not treat any terms specially based on their fermionic role. Similarly, our ordering is based purely on the structure of the individual Pauli strings, rather than the fermionic terms they correspond to.

Finally, while we do not use it in our comparison to the commutator and reverseCommutator orderings for clarity, a *lexoMag* ordering can be made by interleaving the magnitude and lexicographic orderings, as mentioned in Chapter 5. We do not contend that this ordering should provide any improvement over the other ordering schemes discussed. Instead, it is used to contrast the methods described, in an attempt to see if the behaviour of the

ordering schemes can be "mixed". This ordering scheme is considered again in the second half of this chapter, dealing with the relationship of the Bravyi-Kitaev mapping to ordering-dependent Trotter error.

### 6.2.2 *Testing – random Hamiltonians*

As determining the Trotter error is classically computationally intractable for Hamiltonians with large numbers of terms, testing the ordering strategies proved non-trivial. Our initial approach was inspired by the brute force strategy with respect to molecular hydrogen, and focussed on systems whereby the error incurred by all possible orderings could be feasibly determined, for the purpose of comparison.

The `directOrderings` module allows for the generation of random Hamiltonians in the oplist format, as discussed in Chapter 3. These random Hamiltonians are subject to almost no constraints with respect to their physicality. They consist of oplists with random coefficient (in the [0.0, 1.0) interval) and random choice of Pauli operation (or identity) applied to each qubit, rejecting only oplists that are undiagonalisable. It should be emphasised that these arbitrary Hamiltonians bear resemblance to real chemical electronic Hamiltonians only in the sense that they are both weighted sums of strings of Pauli operations. Their use here is due to the fact that the brute-force approach is intractable for almost all real chemical systems. This approach suffers from the fact that any unique behaviour of real electronic Hamiltonians will not be present in this analysis – it has been shown[77] that this can

Table 6.1.: The number of random Hamiltonians generated for each value of $N$, the number of terms in each Hamiltonian.

| Terms | # Systems |
|---|---|
| 3 | 10000 |
| 4 | 10000 |
| 5 | 10000 |
| 6 | 10000 |
| 7 | 1000 |
| 8 | 500 |

produce misleading results. However, examining the error incurred across all possible orderings allows for an examination of the distribution of ordering errors, as studied for the hydrogen molecule in Section 6.1.

To this end, a number of six qubit Hamiltonians were generated, of number of terms $N \in \{3, 4, 5, 6, 7, 8\}$. The number of Hamiltonians generated for each value of $N$ is summarised in Table 6.1. 10, 000 Hamitonians were generated for every $N \in \{3, 4, 5, 6\}$, but for larger $N$, a reduced number of Hamiltonians are generated due to the factorial growth in computational cost. These numbers were tailored so as to reduce the overall runtime to less than a day on a standard laptop computer. Clearly, a larger study would be possible with more substantial computing resources, however the problematic scaling would prohibit study of increased $N$, and as such the present breadth of systems was considered sufficient for this preliminary analysis.

In keeping with the analysis of the hydrogen molecule, the Trotter error incurred using every possible ordering scheme for each of these Hamiltonians was calculated using the `directFermions` module. The Trotter error incurred by using a commutator, reverseCommutator, magnitude or lexicographic ordering was then picked out from this collection. Additionally, the average Trotter error – the Trotter error for a given Hamiltonian, averaged across the space of all possible orderings – was determined.

In order to account for variation in the average ordering error between Hamiltonian, for each Hamiltonian and each ordering strategy we then define the relative performance

$$P = \frac{E_{ordering}}{E_{average}} \tag{6.2.2.1}$$

of that ordering, where $E_{ordering}$ is the Trotter error of the ordering in question, and $E_{average}$ is the average Trotter error over all possible orderings for the Hamiltonian. A $P$ of less than 1 thus indicates that the ordering performed better than average, whereas a $P$ of greater than 1 indicates that the ordering performed worse than average.

We consider the results of this analysis in Figure 6.5. The first consideration is the average performance of each ordering scheme across each value of $N$, shown in the upper plot of Figure 6.5. An inspection here initially seems positive. For $N \in \{3, 4, 5, 6\}$, all ordering schemes yield an error in worst case less than 1% greater than the average Trotter error. In most cases, we observe an error reduction of up to approximately 8%. The results for $N \in \{7, 8\}$, are less suggestive. We observe roughly consistent performance using

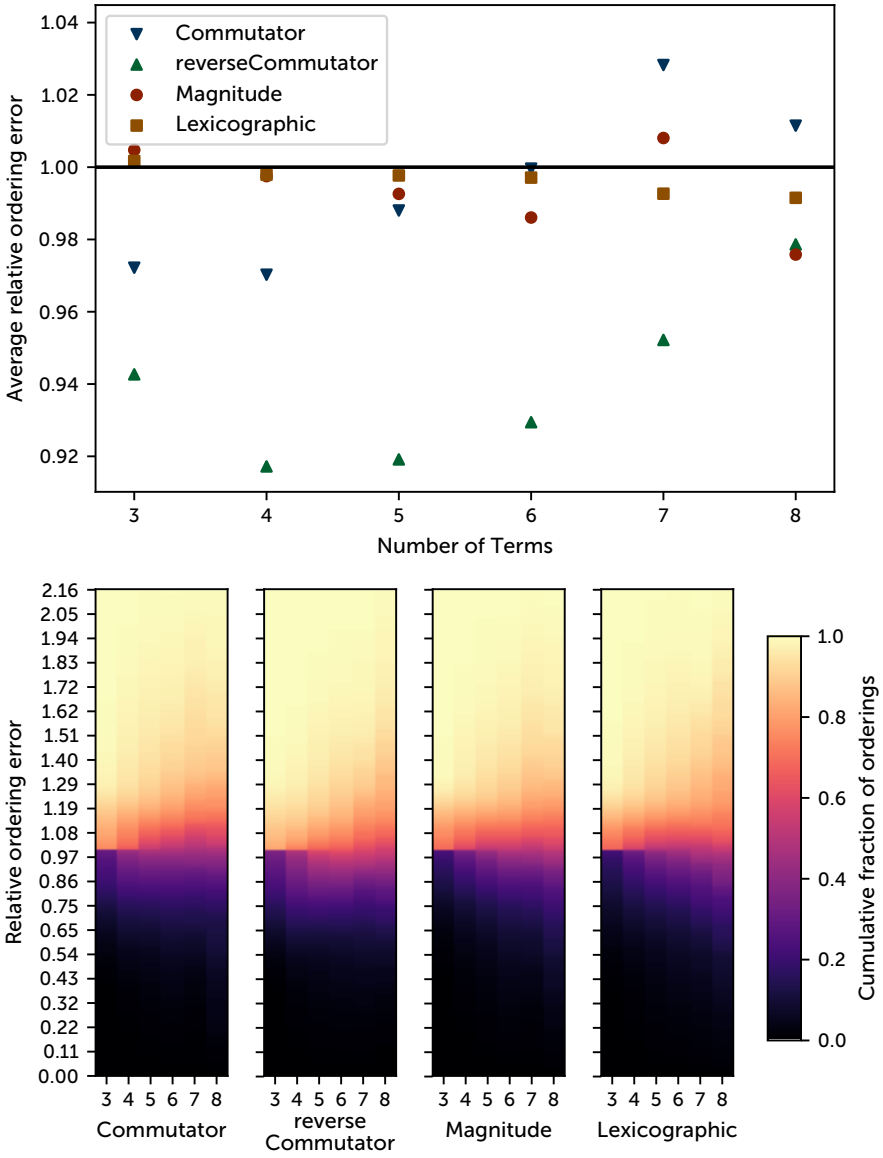



Figure 6.5.:
A comparison of the performance of each ordering, being the Trotter error of each ordering normalised by the Trotter error averaged over all possible orderings for each Hamiltonian. Upper: performances averaged across each value of the number of terms in each Hamiltonian. The reverseCommutator ordering appears to perform well. Lower: a heat map showing the cumulative density of performances for each ordering scheme. The reverseCommutator ordering appears to perform well, although all schemes result in significant variance for higher numbers of terms.

a lexicographic ordering, a pattern curiously reversed in the analysis of real molecules below. A magnitude ordering yields remarkably increased error for $N = 7$, in contrast to its roughly decreasing error trend for other values of $N$. To an extent, this could be an artefact of the reduction in sample size for $N \in \{7, 8\}$.

Comparing ordering schemes, the "reverseCommutator" scheme displays notably reduced Trotter error in contrast to a magnitude or lexicographic ordering in all cases other than $N = 8$, where it performs roughly commensurately with a magnitude ordering. Although this is encouraging, we do observe an increase in average error with increasing $N$.

While for small values of $N$, the commutator scheme yields remarkably good performance in comparison to the other ordering schemes, for large values of $N$ this trend is not maintained. Comparing our two ordering schemes, it is clear that in these results, the reverseCommutator ordering performed significantly better. This could be due to the fact that, for early terms in the ordering process, there is a greater ability to differentiate between terms. This is because the reverse commutator ordering compares candidate terms against the pool of *unordered* terms, rather than the ordered terms. In initial stages of the ordering process, the pool of unordered terms is substantial, whereas the pool of ordered terms is small.

Obscured by the upper plot of Figure 6.5 is the fact that there is an extremely high degree of variance within the relative ordering performance within each class of Hamiltonians. The lower plot of Figure 6.5 demonstrates this by

plotting the cumulative density of the performance $P$ for each ordering strategy. Immediately apparent is that the variance increases substantially with the number of terms $N$, due to the rapid growth of the space of possible orderings. An inspection of the lower plot of Figure 6.5 does, however, seem to confirm that – for these small, random Hamiltonians – the reverse commutator ordering results in marginally smaller errors than the other ordering schemes. A point of interest revealed by this analysis is the large spread of performances yielded by the lexicographic ordering. This could be attributed to the fact that there is no inherent attempt at error-reduction in this scheme, rather focusing on maximising the similarity of neighbouring Pauli strings.

### 6.2.3 *Testing – molecular Hamiltonians*

It has been established that testing on random Hamiltonians does not realistically capture the behaviour of real molecular Hamiltonians when considering Trotter errors. As such, although the previous analysis was useful to assess the ordering schemes in a context where every ordering can be ascertained, a more rigorous analysis requires the use of real chemical Hamiltonians. As such, we initially repeated the analysis using the SMALL_DATA data set. Given that the task of exactly determining Trotter error is exponentially hard, we use a subset of the SMALL_DATA set, as indicated by Appendix A.

Clearly, as this includes systems with more than 8 qubits, an exhaustive search of all possible orderings is not possible for all but the smallest of the systems

included. Instead, an approach of sampling the possible ordering space was adopted, and averaging the Trotter error of the sample. It should be emphasised that as the space of possible orderings grows factorially, the statistical significance of this sample rapidly decays towards zero. As such, these sample averages should not be taken to represent a meaningful approximation to the true average error across all orderings. Instead, they serve mostly as a fixed reference point by which to compare the systematic ordering schemes described above. It should be noted that if the analysis of the hydrogen molecule holds for larger systems, then the random orderings are likely to fall into the dominating region whereby the Trotter error is relatively low. However, verifying whether this is the case (at least by brute force methods) is intractable, even on a quantum device.

We again consider the performance $P$ of each ordering, defined here as the ordering for that error normalised by the error of the random sample of orderings. Instead, here, we consider the performance of each ordering in terms of the number of spin-orbitals considered in the molecular system. Figure 6.6 shows this. It is somewhat difficult to draw conclusions from this data. Reassuringly, again the bulk of all of the performances calculated are superior to those obtained by the random samples, although it should be emphasised that these samples may not be representative.

Much like the analysis of random Hamiltonians, we observe reasonably consistent better-than-average performance associated with the magnitude ordering. Additionally, we yield a similar disperse pattern for the lexicographic ordering as seen in the random Hamiltonians. This is somewhat troubling,

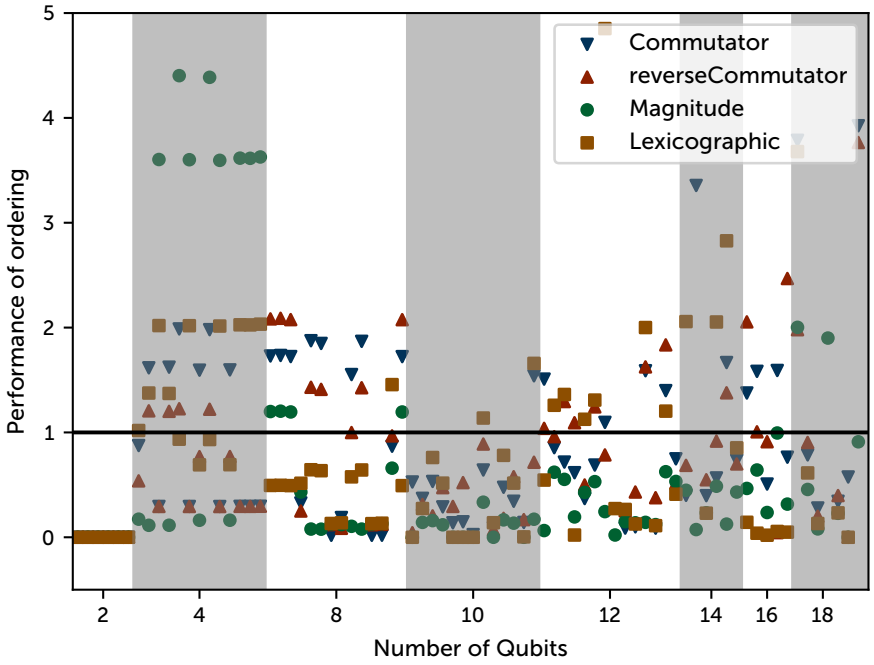



Figure 6.6.: Performance of ordering schemes for the small_data set, with respect to number of qubits. $x$-axis values are arbitrary within each category of number of qubits. The commutator and reverseCommutator orderings do not appear to consistently reduce error. The magnitude ordering is consistently the most effective ordering in terms of Trotter error.

as it indicates that the use of a lexicographic ordering could result in unpredictable error rates, and consequent fixed-precision simulation gate counts. The commutator and reverseCommutator orderings do not fare well in these results. They do not appear to achieve consistent improvement versus a magnitude ordering, and are quite disperse (although arguably not as much as the lexicographic ordering). In contrast to the analysis of random Hamiltonians, the reverseCommutator ordering does not appear to outperform the commutator ordering.

As these results were somewhat inconclusive, it was desirable to repeat the analysis using the LARGE_DATA set of systems. Due to computational cost, we restricted the error analysis to these smaller systems, as indicated by Table 3.2 and Appendix A. The increased size of this data set resulted in commensurately longer compute times for the determination of Trotter errors, on the order of days for each ordering and system. As such, the random sampling approach was abandoned in favour of directly comparing the Trotter error of each ordering scheme.

Figure 6.7 demonstrates the results of this analysis. As before, we observe relatively consistent error performance associated with a magnitude ordering, and relatively inconsistent performance with respect to the lexicographic ordering. Once again, the commutator and reverseCommutator orderings do not appear to yield consistently improved error estimates. This suggests that the commutator and reverseCommutator orderings in Section 6.2 do not appear to consistently reduce Trotter error. The reliability of a magnitude ordering scheme does strongly suggest the reason for this. Our schemes used the magnitude of terms as only a tie-breaker in cases where candidate terms yielded a common level of non-commutativity with regards to the ordered or unordered terms. The fact that the magnitude ordering yielded comparatively low error in most cases emphasises the importance of the magnitude of the resultant terms in the error operator, rather than their mere presence. As such, future work in tailoring ordering schemes could integrate a weighting factor based on each term's magnitude. The efficacy of such schemes remains an open question.

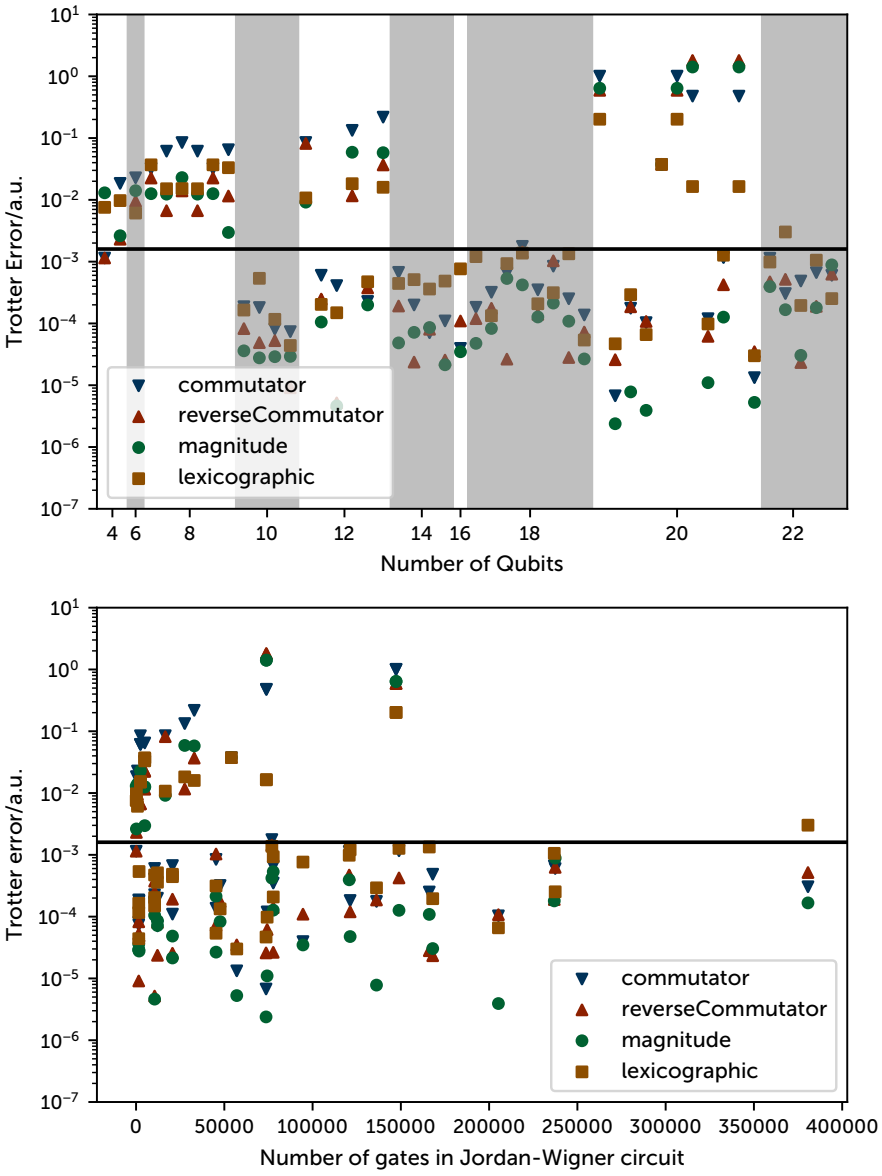



Figure 6.7.:
Trotter error of the different ordering schemes using the LARGE_DATA set. The horizontal line denotes an error of 1 kcal/mol. Upper: by number of qubits. Again, the magnitude ordering is results in the most consistently reduced Trotter error. Lower: by gate count of the unoptimised Jordan-Wigner circuit. Note the instances of worse-than-chemical-accuracy Trotter errors are clustered around very short circuits.

## 6.3 Trotter error and the Bravyi-Kitaev mapping

The discussion in Chapter 5 concluded by noting that a thorough discussion of the efficacy of the Bravyi-Kitaev mapping is not possible without similarly considering the Trotter error incurred by its use, particularly when differing ordering strategies are applied.

The analysis of the hydrogen molecule in Section 6.1 showed that if all orderings are considered, there exists an ordering for the Bravyi-Kitaev mapping that yields the same error as every ordering for the Jordan-Wigner mapping. However, the analysis of the methane molecule in Section 5.1 and in reference [80], along with other work[79], observed lower Trotter error associated with the Bravyi-Kitaev mapping. This suggests that while the distribution of errors across possible orderings may be the same for both mapping schemes, the error *for a given ordering scheme* (lexicographic, magnitude, etc.) will vary. As no algorithm yet exists that will even approximately obtain the optimal ordering scheme for either mapping, it is therefore necessary to compare the errors yielded for each ordering scheme using *both* the Jordan-Wigner and Bravyi-Kitaev mappings. An ordering which is good for one mapping may be bad when applied to the other, and so on.

As such, using the `directFermions` module as described previously in this chapter, we determined the Trotter error for 34 for the systems in the

large_data set. As for the methane molecule, the calculations were performed for a variety of choices of Trotter order and step number. Nonetheless, it is remarkable that limiting our discussion to trials of one Trotter step of first order suffices to yield chemical accuracy (i.e. to within 1 kcal/mol of the FCI energy) for a single application of the evolution operator. For simulation of the full phase estimation procedure, determination of the error incurred in the application of higher powers of this operator would be necessary. However, this does not qualitatively affect the ordering comparison.

Figure 6.8 demonstrates the results of these simulations. Encouragingly, in the overwhelming majority of systems considered, the Trotter error is extremely small even with only one Trotter step: it is often less than 0.001 Hartree. It is possible that this is an artefact of the small number of spin orbitals in the systems considered in the error analysis. It is also worth noting that these errors will be compounded when considering higher bits of precision in a full phase estimation procedure. Nonetheless, it does suggest that the number of Trotter steps required for chemically accurate simulation of larger systems will be relatively modest, potentially less than 10 Trotter steps for the first bit of precision.

Figure 6.8 additionally demonstrates the ordering dependence of the Trotter error. We consider the Trotter error of each systematic (i.e. non-random) ordering normalised by the Trotter error of the lexicographic ordering for each system.

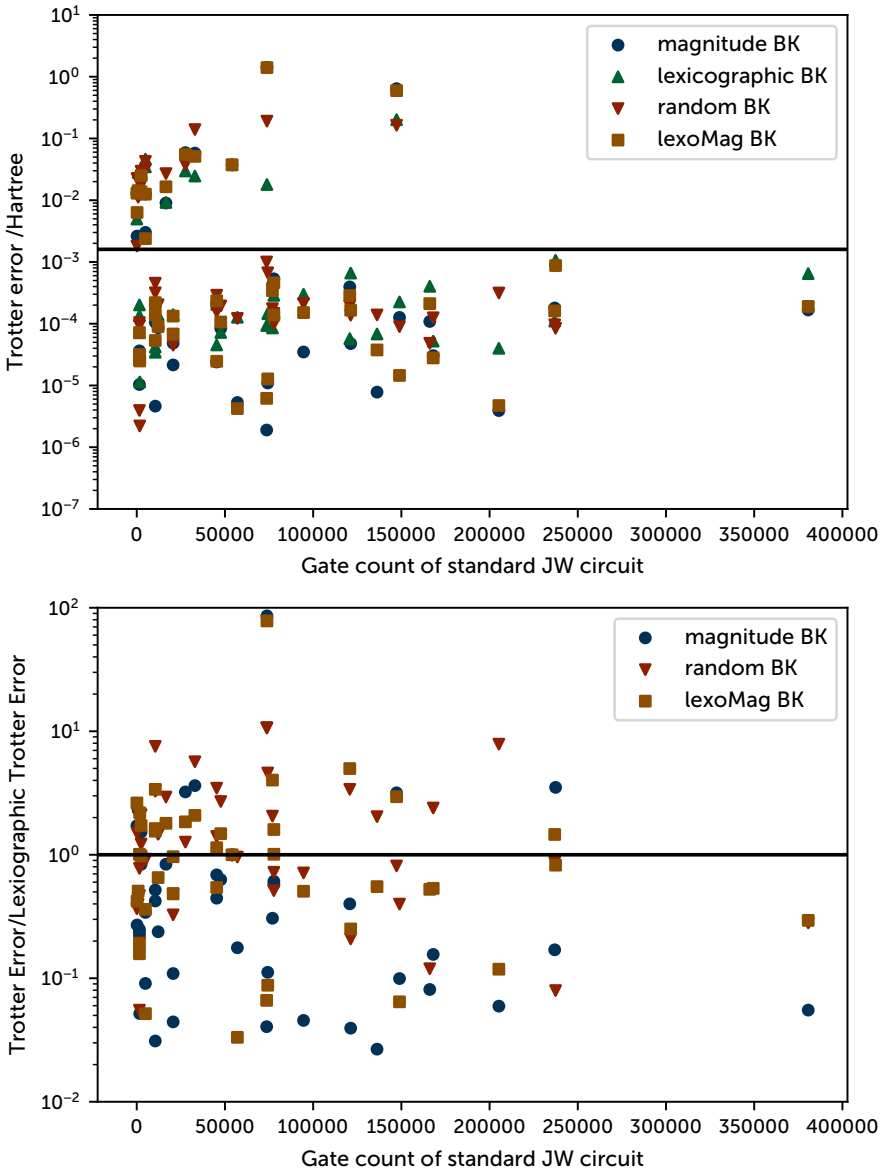



Figure 6.8.:
Trotter error using the Bravyi-Kitaev mapping as a function of Jordan-Wigner circuit length, for different ordering schemes. Upper: Absolute Trotter error. Middle: Trotter error relative to error of lexicographic ordering. Note this excludes two instances where the relative Trotter error was > 20. The red line indicates a relative performance of 1 - i.e. below the line, the ordering results in a lower Trotter error than a lexicographic ordering. The magnitude ordering usually results in a substantially lower error, however in most of these cases the Trotter error was already around $10^{-4}$.

In most cases, the magnitude ordering appears to achieve a significantly lower Trotter error than the lexicographic ordering. In some cases, this difference exceeds an order of magnitude. It is important to note that this represents a large variance on an exceptionally small error. Noting that one Trotter step was sufficient for chemical accuracy in most of the systems studied here, we do not argue that this necessarily indicates that a magnitude ordering achieves a useful reduction in error compared to a lexicographic ordering. However, it is likely that this error would be magnified when determining higher bits of precision in the full phase estimation procedure. Future work is required to assess this, as in these circumstances this effect could become sufficiently significant to determine ordering choice.

An examination of the relative performance of the Bravyi-Kitaev mapping and the Jordan-Wigner mapping in the context of ordering strategies is included as Figure 6.9. Again, the distinction between the two mappings is in most cases not as substantial as the differences observed for single Trotter step circuit length. In the majority of systems, the normalised difference between the two errors is between 1 and -1 – that is, the error associated with one mapping is very rarely more than double that of the other. Several systems display substantially increased error associated with the Bravyi-Kitaev mapping (including two not shown on Figure 6.9 for scale clarity); however, in the general case no such pattern emerges.

Using a lexicographic ordering, a preference for the Bravyi-Kitaev mapping is observable in most systems. In some cases, the error is almost halved by the use of the Bravyi-Kitaev mapping. At larger system sizes this could become

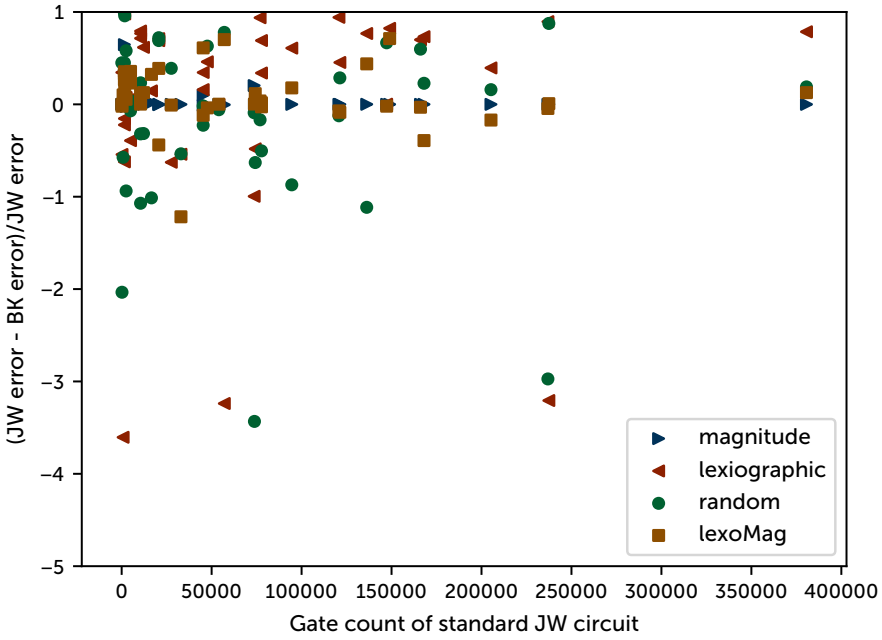



Figure 6.9.: Relative Trotter error of Jordan-Wigner and Bravyi-Kitaev mappings. The difference in error is normalised by the Jordan-Wigner error, such that a value of 0 indicates equivalent performance, with negative values implying high Bravyi-Kitaev error. Both schemes display remarkably similar errors for the magnitude ordering. For a lexicographic ordering, the Bravyi-Kitaev mapping shows lower error in most of the systems studied.

a more substantial effect; however, we do not contend that the data provides concrete evidence as to whether this is true. For a magnitude ordering the Jordan-Wigner and Bravyi-Kitaev mappings yield nearly identical Trotter error in almost all cases. In a sense, this could be attributed to the more directly physical nature of the magnitude ordering. Important terms will intrinsically be simulated earlier in sequence using both mappings. As such, the error of both is likely to be similar in this case. As to be expected, the "lexoMag" ordering performs roughly as a combination of the magnitude and lexicographic orderings.

As above, the impact of this variation in error could prove substantial when propagated through the entire phase estimation algorithm. Using a lexicographic ordering – optimal for gate cancellation – the Bravyi-Kitaev mapping outperforms the Jordan-Wigner mapping in most cases. If this superior error performance scales to above 30 qubit systems, this could result in a reduction of the necessary Trotter steps for simulation. The consequent reduction in circuit length could counterbalance the marginally increased individual Trotter step cost of using the Bravyi-Kitaev mapping in a lexicographic ordering. Further work examining the entire procedure should assess this. Nonetheless, it should not be forgotten that an exact determination of the Trotterization error is equivalent to the solution of the exponentially hard eigenvalue problem itself. Consequentially, this approach may prove intractable. Qualitatively, an examination of the norm of the Trotter error operator may prove instructive.[93]

Overall, we conclude here that the Bravyi-Kitaev mapping does show a great deal of promise when performing the quantum simulation of electronic structure. Chapter 5 concluded that improvement was observed using this mapping for all orderings other than the lexicographic ordering, which is optimal for gate cancellation. Here, however, we observe consistently superior Trotter error performance associated with the Bravyi-Kitaev mapping when using the lexicographic ordering, which is likely to result in shorter circuits, even accounting for the aforementioned marginally increased overhead. Future work is, however, required to assess this in the context of the full phase estimation procedure.

### 6.3.1 *Commutator based orderings*

In Section 6.2 we discussed attempts to generate an optimised ordering strategy with respect to Trotter error. While the orderings derived from this did not achieve consistent reduction of Trotter error, it did yield information which would be useful for future attempts.

As with the Jordan-Wigner mapping, for the hydrogen molecule in a minimal basis, our schemes reproduce the optimal ordering strategy when using the Bravyi-Kitaev mapping. Following the logic of Section 6.2, an analysis of the commutator and reverseCommutator ordering schemes when using the Bravyi-Kitaev mapping could be instructive. As such, we again used the `directFermions` module to calculate the Trotter errors as a result of a single Trotter step for a subset of the LARGE_DATA set using a commutator and reverseCommutator ordering, this time using a Bravyi-Kitaev mapping.

The results here are depicted in Figure 6.10. As with the Jordan-Wigner transformation, we again do not witness a reliable degree of improvement associated with the commutator and reverseCommutator orderings, in terms of Trotter error. The reverseCommutator ordering is typically marginally superior to the commutator ordering, and is likewise better than a lexicographic ordering in most cases, although this is not consistent. In a few cases, the reverseCommutator ordering does provide the lowest error (beating a magnitude ordering), although these are rare and should not be relied

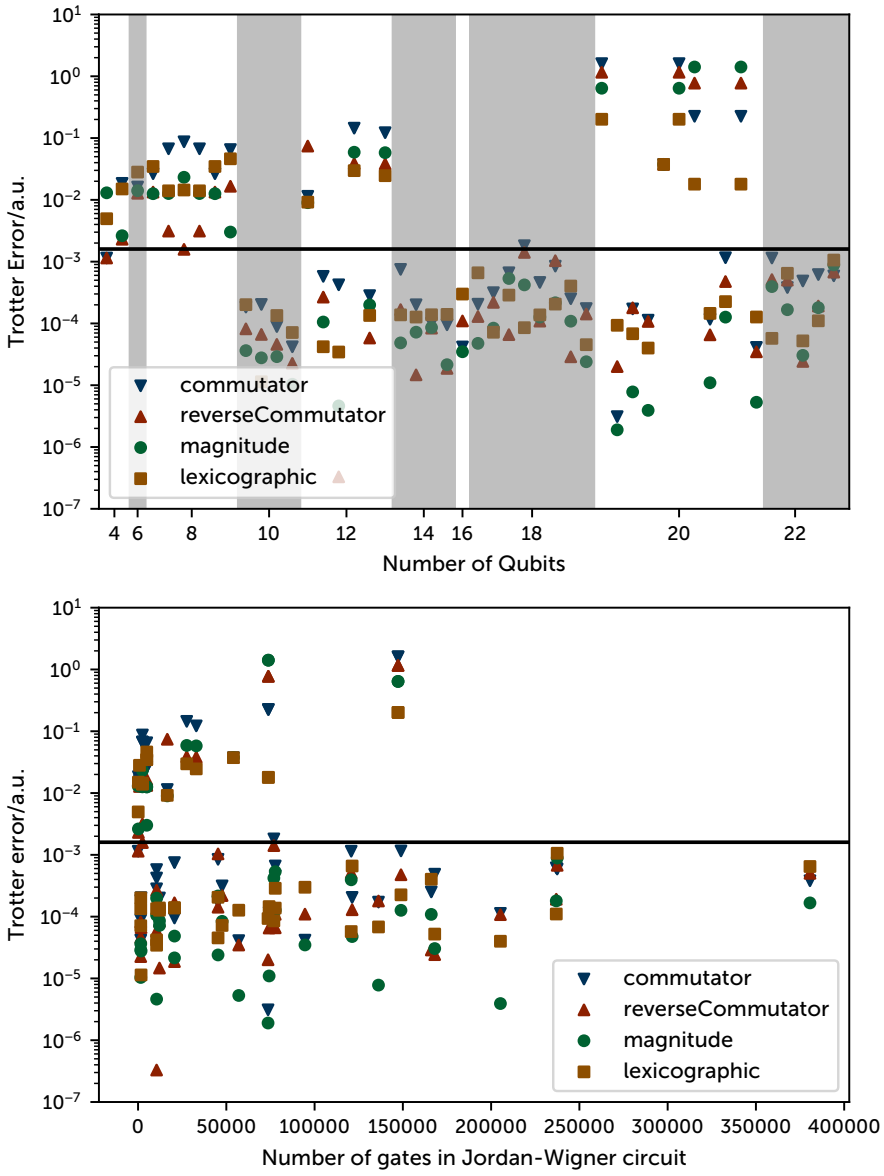



Figure 6.10.:
Error performance of the Bravyi-Kitaev mapping using a commutator and reverseCommutator ordering. Upper: by number of qubits. Lower: by length of unoptimised circuit. In both cases, a similar trend to that observed using the Jordan-Wigner mapping is apparent.

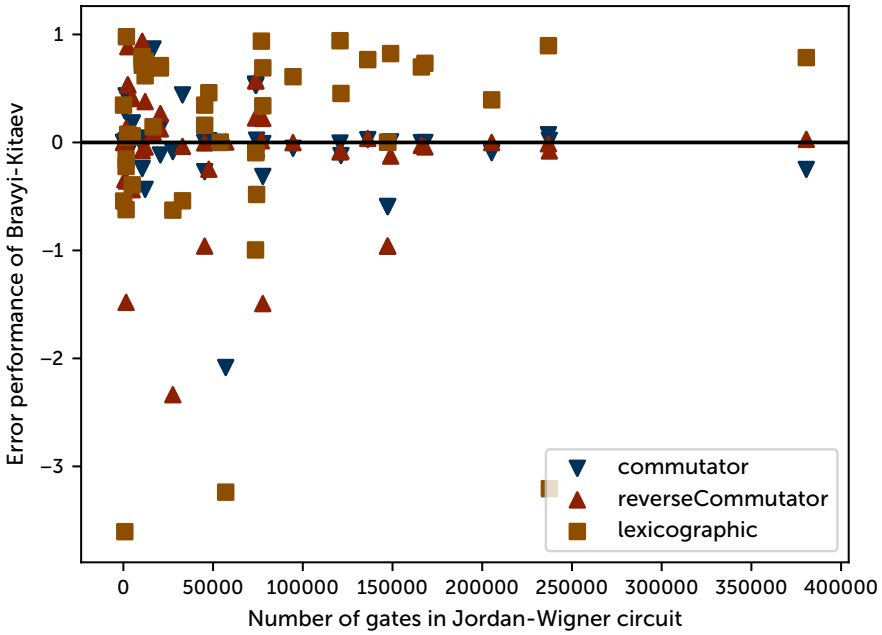



Figure 6.11.: Relative performance of Bravyi-Kitaev and Jordan-Wigner mappings when using the commutator or reverseCommutator orderings. The commutator and reverseCommutator orderings display roughly equivalent performance for both mapping schemes, similar to the magnitude ordering rather than the Bravyi-Kitaev-favouring lexicographic ordering.

upon. The commutator ordering fares even worse, frequently resulting in the *highest* error rates of each ordering scheme.

Examining the Bravyi-Kitaev error rates alone indicates a similar picture to that observed when using the Jordan-Wigner mapping; however, a comparison of the difference in error between the two mapping schemes for each ordering is important. This indicates which scheme would be preferred when using a specific ordering, with respect to Trotter error. We present this in Figure 6.11. Curiously, this shows that – especially for larger circuit lengths – the Trotter error incurred when using the commutator or reverseCommut-

ator orderings is in most cases roughly independent of the choice of mapping scheme. This is in contrast to the results when using a lexicographic ordering, where the Bravyi-Kitaev mapping consistently results in reduced error. In this manner, it is more akin to a magnitude ordering, where both mapping schemes perform roughly equivalently in terms of error.

Initially, this may seem counter-intuitive, as the commutator and reverseCommutator ordering schemes employ the term magnitude only as tiebreakers. However, they are based on the commutativity structure of the Hamiltonian. The commutativity of the terms within the Bravyi-Kitaev or Jordan-Wigner Hamiltonian is directly related to the commutativity of the terms within the electronic Hamiltonian from whence they arise. As such, we can expect these orderings to result in a more consistent error rate with respect to choice of mapping, in comparison to the arguably less physical lexicographic ordering.

While the reverseCommutator ordering initially showed promise when studying random Hamiltonians, both it and the commutator ordering failed to yield consistent improvement with regard to Trotter error using either the Jordan-Wigner or Bravyi-Kitaev mapping. Nonetheless, important information was yielded through this process.

Though care should be taken here, brute force analysis of the hydrogen molecule suggested that while there is a substantial difference in error performance between worst and best orderings, the distribution of errors with respect to ordering is clustered towards the low error region – suggesting that

ordering choice should focus on eliminating bad orderings, rather than finding optimal ones. This analysis also aided in experimental demonstrations of these techniques.

The Trotter error superiority of the magnitude mapping emphasises that the commutativity structure cannot be the only thing taken into account when choosing a Trotter ordering, and that approaches need to weigh this against the magnitude of each term. Finally, the results discussed here confirmed previous results[79,80] that the Bravyi-Kitaev mapping results in a reduction in Trotter error. Future work is required to assess this within the context of the full phase estimation algorithm, although this may involve substantial computational efforts.

# 7 Conclusions and Outlook

In this thesis we have discussed attempts to characterise improvements to the canonical method of performing electronic structure calculations on digital quantum computers. The focus was particularly on two areas: the Bravyi-Kitaev mapping, and ordering schemes when applying a Trotter-Suzuki expansion.

Analysis of the Bravyi-Kitaev mapping revealed a previously unknown variability to the impact of the technique with regard to simulation gate counts, and thus the quantum computational cost. Previous work[79] suggested that there was a deterministic improvement associated with its use, which would be expected to be maintained when using other forms of circuit optimisation. Initial analysis of the methane molecule agreed with this, observing superior performance from the Bravyi-Kitaev mapping in both circuit length and Trotter error, although by a lower margin than previously expected.

However, more detailed analysis involving a large set of molecular systems and optimised circuits revealed that the gate count advantage displayed by the Bravyi-Kitaev mapping *strongly depends on the Trotter order that is chosen*.

For some orderings, substantial improvement was observed; however, when an ordering optimal for gate cancellation is used, the advantage disappears, and is (in terms of gate count) worse than the Jordan-Wigner mapping – albeit to within a very low margin. This emphasises the fact that these optimisation schemes are not independent, and that an inherently optimal choice is perhaps impossible. Future work should bear this in mind when designing circuits to be performed on quantum devices.

Trotter error analysis did suggest that reduced Trotter error is associated with the Bravyi-Kitaev mapping. This is encouraging, as a reduction in the number of Trotter steps required would potentially result in a large reduction in cost for an overall simulation. While the Trotter error difference between the Bravyi-Kitaev and Jordan-Wigner mappings was low for the first bit of the ground state energy, this could result in a meaningful distinction when propagated through the entire phase estimation procedure. Unfortunately, it was not possible to calculate this, due to the substantial computational difficulty.

This is, to an extent, inherent within the task, as determining the exact Trotter error is exponentially difficult. Nonetheless, the simulation of 49 and (with some limitations) 56 qubit systems has been demonstrated on classical computers.[43] It thus seems likely that, with specialised computing resources, a thorough analysis of Trotter error in systems of perhaps $40 - 50$ spin-orbitals could be performed. Although potentially expensive, such an analysis could prove invaluable in determining the Trotter error implications of the Bravyi-Kitaev mapping. Similarly, an analysis involving consideration

of the norm of the Trotter error operator could prove instructive, as in other work.[93] As this quantitatively substantially overestimates the exact Trotter error, this may obscure the distinction between the mapping schemes. However, if a qualitative relationship between this and the exact Trotter error considering each mapping scheme can be ascertained, this could be a useful way to generalise to much larger system sizes.

The analysis of Trotter ordering consisted of several aspects. A brute-force search of Trotter orderings for the hydrogen molecule yielded some insight that proved useful experimentally. As has been observed in other work[79], we confirmed that the choice of ordering can strongly influence the Trotter error incurred in the study of the hydrogen molecule.

This section did, however, also suggest that the bulk of the orderings yield a relatively low error, in comparison to the error obtained by the worst possible ordering. This is encouraging, as it does suggest that the emphasis should perhaps not be on finding the optimal possible ordering, but rather an ordering scheme that is *good enough*. So long as the worst possible ordering scheme is not systematically chosen, this allows for greater flexibility to tailor the ordering scheme by other criteria (such as the efficacy of gate cancellation or hardware constraints).

This argument should be tempered by the fact that although a brute-force approach was effective for the hydrogen molecule, it is dangerous to generalise too strongly from this to larger systems. Unlike the analysis on the Trotter error of the Bravyi-Kitaev mapping, it is unlikely that much further

computational work will be possible utilising this technique. The factorial scaling of the number of possible orderings with regards to the number of terms in the Hamiltonian will necessitate another approach for anything but trivial systems, regardless of the amount of computational effort thrown at the problem.

Attempts to determine an optimal ordering scheme for Trotterization unfortunately did not appear to yield a consistently superior ordering strategy in terms of Trotter error. However, useful information was gleaned with regards to the emphasis that must be paid to both the commutativity relationships between the terms and the magnitude of those terms. The impressive result of the magnitude ordering with regards to Trotter error emphasises the importance of the latter of these. Future attempts at developing optimal ordering strategies should take this into consideration. Other work[106] has examined the *importance* of a term with respect to its impact on the Hartree-Fock ground state, as opposed to the raw magnitude of the term. The commutator-based strategies would likely be improved by an approach similar to this, although this would need to be tempered by an increased dependence of the ordering on importance, rather than commutativity.

The continued development of classical computing over the past century has revolutionised not just science, but the entirety of human society, at a pace which is historically unprecedented. A major driving force behind this is not just the availability of devices, but their rapid growth in computing power. This growth is frequently described with Moore's law[118], a statement of the observed exponential growth in CPU transistor count.

In chemistry, this explosion of computing power has resulted in profound effects. Calculations which would have seemed preposterously impractical at the birth of quantum mechanics – the generation of the integrals used in this project, as a simple example – are now routine. The fundamental quantum mechanical nature of chemistry is emphasised and studied, even in fields relating to large molecules or complex reactions.

For quantum computing, the historical comparison to the development of classical computing is difficult to avoid.[119] At present, qubits are the domain of teams of researchers specialised in the task of making a scalable device exist, and the software to run on it. This is not dissimilar to the state of classical computing in the 1940s, and is a far cry from the ubiquity of – for instance – computational chemical methods today.

However, the rapid increase of the pace of progress is striking, even on a time-scale as short as the work described in this thesis. The first experimental demonstration of the simulation of quantum chemistry, on a two qubit device, was reported in 2010 – less than a decade prior to the time of writing. In 2017 and 2018, prototypes of 50 and 72 qubit devices were reported. It has been argued that the first demonstration of a quantum device outperforming all classical supercomputers will be possible with such technology.[46]

The largest molecular simulation on a quantum device was of beryllium hydride in 2017.[63] While this represents an impressive and major step forward in the use of quantum computing for chemical problems, it remains trivial in comparison to calculations that can be performed classically, even at a full

configuration interaction level. It is clear that algorithmic improvements remain of crucial importance to the development of the practical quantum simulation of quantum chemistry.

Such improvements continue to be made. Some have been discussed in depth in the bulk of this thesis, and others were briefly reviewed in Chapter 1. In a limited sense, the array of quantum methodologies for the simulation of quantum chemistry begins to resemble the myriad of classical methodologies. As in the development of computational chemistry on classical computers, it is likely that each of these methodologies and optimisations will find its own use case. Benchmarking and categorisation attempts – such as those described in this thesis – will be required in this process.

Quantum computers are unlikely to ever fully replace classical computers in the study of quantum chemistry. However, their fundamental promise remains the possibility of performing calculations that could never be performed on a classical device. We speculate that this could open entirely new avenues to the study of theoretical chemistry, to complement those yielded by classical devices. If quantum computing today is comparable to the state of classical computing in the post-war era, then those avenues could stretch far indeed.

# Appendices

# Data sets



This Appendix contains a detailed list of the systems in the SMALL_DATA and LARGE_DATA sets, as discussed in Chapter 3.

Table A.1.: The systems included in the SMALL_DATA set. Level corresponds to whether or not our ordering analysis was performed for the system – N indicates that this analysis was performed for the system, Y indicates that it was not. MO basis gives the molecular orbital basis used – CMO for the canonical Hartree-Fock molecular orbitals, OAO for Löwdin Orthogonalised atomic orbitals, and NMO for natural molecular orbitals.

| Name | Charge | Mult. | Basis | MO Basis | N. Qubits | Level |
|---|---|---|---|---|---|---|
| Al | 0 | 2 | STO-6G | CMO | 18 | Y |
| Al | 0 | 2 | STO-6G | NMO | 18 | Y |
| Al | 0 | 2 | STO-6G | OAO | 18 | Y |
| Ar | 0 | 1 | STO-6G | NMO | 18 | Y |
| Ar | 0 | 1 | STO-6G | OAO | 18 | Y |
| B | 0 | 2 | STO-6G | CMO | 10 | Y |
| B | 0 | 2 | ECP | CMO | 16 | N |

| | | | | | | |
|---|---|---|---|---|---|---|
| B | 0 | 2 | ECP | NMO | 16 | N |
| B | 0 | 2 | ECP | OAO | 16 | N |
| B | 0 | 2 | STO-6G | NMO | 10 | Y |
| B | 0 | 2 | STO-6G | OAO | 10 | N |
| Be | 0 | 1 | STO-6G | CMO | 10 | Y |
| Be | 0 | 1 | ECP | CMO | 16 | Y |
| Be | 0 | 1 | ECP | NMO | 16 | Y |
| Be | 0 | 1 | ECP | OAO | 16 | Y |
| Be | 0 | 1 | STO-6G | NMO | 10 | Y |
| Be | 0 | 1 | STO-6G | OAO | 10 | Y |
| $BeH_2$ | 0 | 1 | STO-6G | CMO | 14 | Y |
| $BeH_2$ | 0 | 1 | STO-6G | NMO | 14 | Y |
| $BeH_2$ | 0 | 1 | STO-6G | OAO | 14 | Y |
| C | 0 | 3 | STO-6G | CMO | 10 | N |
| C | 0 | 3 | ECP | CMO | 16 | N |
| C | 0 | 3 | ECP | NMO | 16 | N |
| C | 0 | 3 | ECP | OAO | 16 | N |
| C | 0 | 3 | STO-6G | NMO | 10 | N |
| C | 0 | 3 | STO-6G | OAO | 10 | N |
| $CH_2$ | 0 | 3 | 321G | CMO | 26 | N |
| $CH_2$ | 0 | 3 | 321G | NMO | 26 | N |
| $CH_2$ | 0 | 3 | 321G | OAO | 26 | N |
| $CH_2$ | 0 | 3 | STO-6G | CMO | 14 | N |
| $CH_2$ | 0 | 3 | ECP | CMO | 24 | N |

| | | | | | | |
|---|---|---|---|---|---|---|
| $CH_2$ | 0 | 3 | ECP | NMO | 24 | N |
| $CH_2$ | 0 | 3 | ECP | OAO | 24 | N |
| $CH_2$ | 0 | 3 | STO-6G | NMO | 14 | N |
| $CH_2$ | 0 | 3 | STO-6G | OAO | 14 | N |
| Cl | 0 | 2 | STO-6G | CMO | 18 | N |
| Cl | 0 | 2 | STO-6G | NMO | 18 | N |
| Cl | 0 | 2 | STO-6G | OAO | 18 | N |
| F | 0 | 2 | STO-6G | CMO | 10 | N |
| F | 0 | 2 | ECP | CMO | 16 | N |
| F | 0 | 2 | ECP | NMO | 16 | N |
| F | 0 | 2 | ECP | OAO | 16 | N |
| F | 0 | 2 | STO-6G | NMO | 10 | N |
| F | 0 | 2 | STO-6G | OAO | 10 | Y |
| H | 0 | 2 | STO-6G | CMO | 2 | Y |
| H | 0 | 2 | ECP | CMO | 4 | N |
| H | 0 | 2 | ECP | NMO | 4 | N |
| H | 0 | 2 | ECP | OAO | 4 | N |
| H | 0 | 2 | STO-6G | NMO | 2 | Y |
| H | 0 | 2 | STO-6G | OAO | 2 | Y |
| $H_2$ | 0 | 1 | 321G | CMO | 8 | Y |
| $H_2$ | 0 | 1 | 321G | NMO | 8 | Y |
| $H_2$ | 0 | 1 | 321G | OAO | 8 | Y |
| $H_2$ | 0 | 1 | 631++G | CMO | 12 | Y |
| $H_2$ | 0 | 1 | 631++G | NMO | 12 | Y |

| | | | | | | |
|---|---|---|---|---|---|---|
| $H_2$ | 0 | 1 | 631++G | OAO | 12 | Y |
| $H_2$ | 0 | 1 | 631++Gss | CMO | 24 | N |
| $H_2$ | 0 | 1 | 631++Gss | NMO | 24 | N |
| $H_2$ | 0 | 1 | 631++Gss | OAO | 24 | N |
| $H_2$ | 0 | 1 | 6311++Gss | CMO | 28 | N |
| $H_2$ | 0 | 1 | 6311++Gss | NMO | 28 | N |
| $H_2$ | 0 | 1 | 6311++Gss | OAO | 28 | N |
| $H_2$ | 0 | 1 | 6311G | CMO | 12 | Y |
| $H_2$ | 0 | 1 | 6311G | NMO | 12 | Y |
| $H_2$ | 0 | 1 | 6311G | OAO | 12 | Y |
| $H_2$ | 0 | 1 | 6311GU | CMO | 16 | N |
| $H_2$ | 0 | 1 | 6311GU | NMO | 16 | N |
| $H_2$ | 0 | 1 | 6311GU | OAO | 16 | N |
| $H_2$ | 0 | 1 | 631G | CMO | 8 | Y |
| $H_2$ | 0 | 1 | 631G | NMO | 8 | Y |
| $H_2$ | 0 | 1 | 631G | OAO | 8 | Y |
| $H_2$ | 0 | 1 | 631Gss | CMO | 20 | N |
| $H_2$ | 0 | 1 | 631Gss | NMO | 20 | N |
| $H_2$ | 0 | 1 | 631Gss | OAO | 20 | N |
| $H_2$ | 0 | 1 | STO-6G | CMO | 4 | Y |
| $H_2$ | 0 | 1 | ECP | CMO | 8 | Y |
| $H_2$ | 0 | 1 | ECP | NMO | 8 | Y |
| $H_2$ | 0 | 1 | ECP | OAO | 8 | Y |
| $H_2$ | 0 | 1 | 12 | NMO | 12 | Y |

| | | | | | | |
|---|---|---|---|---|---|---|
| $H_2$ | 0 | 1 | 2 | NMO | 4 | Y |
| $H_2$ | 0 | 1 | 20 | NMO | 20 | N |
| $H_2$ | 0 | 1 | 4 | NMO | 4 | Y |
| $H_2$ | 0 | 1 | 8 | NMO | 8 | Y |
| $H_2$ | 0 | 1 | STO-6G | NMO | 4 | Y |
| $H_2$ | 0 | 1 | STO-6G | OAO | 4 | Y |
| $H_2$ | 0 | 1 | SBKJC-ECP | CMO | 8 | Y |
| $H_2$ | 0 | 1 | SBKJC-ECP | NMO | 8 | Y |
| $H_2$ | 0 | 1 | SBKJC-ECP | OAO | 8 | Y |
| $H_2$ | 0 | 1 | STO-6G | CMO | 4 | Y |
| $H_2$ | 0 | 1 | STO-6G | NMO | 4 | Y |
| $H_2$ | 0 | 1 | STO-6G | OAO | 4 | Y |
| $H_2O$ | 0 | 1 | 321G | CMO | 26 | N |
| $H_2O$ | 0 | 1 | 321G | NMO | 26 | N |
| $H_2O$ | 0 | 1 | 321G | OAO | 26 | N |
| $H_2O$ | 0 | 1 | STO-6G | CMO | 14 | Y |
| $H_2O$ | 0 | 1 | STO-6G | NMO | 14 | Y |
| $H_2O$ | 0 | 1 | STO-6G | OAO | 14 | Y |
| HF | 0 | 1 | 321G | CMO | 22 | N |
| HF | 0 | 1 | 321G | NMO | 22 | N |
| HF | 0 | 1 | 321G | OAO | 22 | N |
| HF | 0 | 1 | STO-6G | CMO | 12 | Y |
| HF | 0 | 1 | ECP | CMO | 20 | N |
| HF | 0 | 1 | ECP | NMO | 20 | N |

| | | | | | | |
|---|---|---|---|---|---|---|
| HF | 0 | 1 | ECP | OAO | 12 | N |
| HF | 0 | 1 | STO-6G | NMO | 12 | Y |
| HF | 0 | 1 | STO-6G | OAO | 12 | Y |
| He | 0 | 1 | STO-6G | CMO | 2 | Y |
| He | 0 | 1 | ECP | CMO | 4 | Y |
| He | 0 | 1 | ECP | NMO | 4 | Y |
| He | 0 | 1 | ECP | OAO | 4 | Y |
| He | 0 | 1 | STO-6G | NMO | 2 | Y |
| He | 0 | 1 | STO-6G | OAO | 2 | Y |
| $HeH^+$ | 1 | 1 | 321G | CMO | 8 | N |
| $HeH^+$ | 1 | 1 | 321G | NMO | 8 | N |
| $HeH^+$ | 1 | 1 | 321G | OAO | 8 | N |
| $HeH^+$ | 1 | 1 | 6311G | CMO | 12 | N |
| $HeH^+$ | 1 | 1 | 6311G | NMO | 12 | N |
| $HeH^+$ | 1 | 1 | 6311G | OAO | 12 | N |
| $HeH^+$ | 1 | 1 | 6311GU | CMO | 16 | N |
| $HeH^+$ | 1 | 1 | 6311GU | NMO | 16 | N |
| $HeH^+$ | 1 | 1 | 6311GU | OAO | 16 | N |
| $HeH^+$ | 1 | 1 | 631G | CMO | 8 | N |
| $HeH^+$ | 1 | 1 | 631G | NMO | 8 | N |
| $HeH^+$ | 1 | 1 | 631G | OAO | 8 | N |
| $HeH^+$ | 1 | 1 | STO-6G | CMO | 4 | N |
| $HeH^+$ | 1 | 1 | ECP | CMO | 8 | N |
| $HeH^+$ | 1 | 1 | ECP | NMO | 8 | N |

| | | | | | | |
|---|---|---|---|---|---|---|
| $HeH^+$ | 1 | 1 | ECP | OAO | 8 | N |
| $HeH^+$ | 1 | 1 | STO-6G | NMO | 4 | Y |
| $HeH^+$ | 1 | 1 | STO-6G | OAO | 4 | N |
| $HeH^+$ | 1 | 1 | SBKJC-ECP | CMO | 8 | N |
| $HeH^+$ | 1 | 1 | SBKJC-ECP | NMO | 8 | N |
| $HeH^+$ | 1 | 1 | SBKJC-ECP | OAO | 8 | N |
| $HeH^+$ | 1 | 1 | STO-6G | CMO | 4 | N |
| $HeH^+$ | 1 | 1 | STO-6G | NMO | 4 | Y |
| $HeH^+$ | 1 | 1 | STO-6G | OAO | 4 | N |
| Li | 0 | 2 | STO-6G | CMO | 10 | N |
| Li | 0 | 2 | ECP | CMO | 16 | Y |
| Li | 0 | 2 | ECP | NMO | 16 | Y |
| Li | 0 | 2 | ECP | OAO | 16 | N |
| Li | 0 | 2 | STO-6G | NMO | 10 | Y |
| Li | 0 | 2 | STO-6G | OAO | 10 | Y |
| LiH | 0 | 1 | 321G | CMO | 22 | N |
| LiH | 0 | 1 | 321G | NMO | 22 | N |
| LiH | 0 | 1 | 321G | OAO | 22 | N |
| LiH | 0 | 1 | 6 | CMO | 6 | Y |
| LiH | 0 | 1 | STO-6G | CMO | 12 | Y |
| LiH | 0 | 1 | STO-6G | CMO6 | 12 | Y |
| LiH | 0 | 1 | 6 | NMO | 6 | Y |
| LiH | 0 | 1 | 8 | NMO | 8 | Y |
| LiH | 0 | 1 | STO-6G | NMO | 12 | Y |

| | | | | | | |
|---|---|---|---|---|---|---|
| LiH | 0 | 1 | STO-6G | OAO | 12 | Y |
| Mg | 0 | 1 | STO-6G | CMO | 18 | Y |
| Mg | 0 | 1 | STO-6G | NMO | 18 | Y |
| Mg | 0 | 1 | STO-6G | OAO | 18 | Y |
| N | 0 | 4 | STO-6G | CMO | 10 | Y |
| N | 0 | 4 | ECP | CMO | 16 | N |
| N | 0 | 4 | ECP | NMO | 16 | N |
| N | 0 | 4 | ECP | OAO | 16 | N |
| N | 0 | 4 | STO-6G | NMO | 10 | N |
| N | 0 | 4 | STO-6G | OAO | 10 | Y |
| Na | 0 | 2 | STO-6G | CMO | 18 | Y |
| Na | 0 | 2 | STO-6G | NMO | 18 | Y |
| Na | 0 | 2 | STO-6G | OAO | 18 | Y |
| Ne | 0 | 1 | STO-6G | CMO | 10 | Y |
| Ne | 0 | 1 | ECP | CMO | 16 | N |
| Ne | 0 | 1 | ECP | NMO | 16 | N |
| Ne | 0 | 1 | ECP | OAO | 16 | N |
| Ne | 0 | 1 | STO-6G | NMO | 10 | Y |
| Ne | 0 | 1 | STO-6G | OAO | 10 | Y |
| O | 0 | 3 | STO-6G | CMO | 10 | N |
| O | 0 | 3 | ECP | CMO | 16 | N |
| O | 0 | 3 | ECP | NMO | 16 | N |
| O | 0 | 3 | ECP | OAO | 16 | N |
| O | 0 | 3 | STO-6G | NMO | 10 | N |

| | | | | | | |
|---|---|---|---|---|---|---|
| O | 0 | 3 | STO-6G | OAO | 10 | N |
| P | 0 | 4 | STO-6G | CMO | 18 | N |
| P | 0 | 4 | STO-6G | NMO | 18 | N |
| P | 0 | 4 | STO-6G | OAO | 18 | N |
| S | 0 | 3 | STO-6G | CMO | 18 | N |
| S | 0 | 3 | STO-6G | NMO | 18 | N |
| S | 0 | 3 | STO-6G | OAO | 18 | N |
| Si | 0 | 3 | STO-6G | CMO | 18 | N |
| Si | 0 | 3 | STO-6G | NMO | 18 | N |
| Si | 0 | 3 | STO-6G | OAO | 18 | N |

Table A.2.: The systems included in the large_data set. Level corresponds to whether or not the error incurred in Trotterization was considered – N indicates that this analysis was not performed for this system, Y indicates that it was.

| Name | Charge | Mult. | Basis | N. Qubits | Level |
|---|---|---|---|---|---|
| Methane | 0 | 1 | STO-3G | 18 | Y |
| Ethane | 0 | 1 | STO-3G | 32 | N |
| Propane | 0 | 1 | STO-3G | 46 | N |
| Ethene | 0 | 1 | STO-3G | 28 | N |
| Propene | 0 | 1 | STO-3G | 42 | N |
| Ethyne | 0 | 1 | STO-3G | 24 | N |
| Methanol | 0 | 1 | STO-3G | 28 | N |
| Ethanol | 0 | 1 | STO-3G | 42 | N |
| Isopropanol | 0 | 1 | STO-3G | 56 | N |
| 1-Propanol | 0 | 1 | STO-3G | 56 | N |
| Propanone | 0 | 1 | STO-3G | 52 | N |
| Methanal | 0 | 1 | STO-3G | 24 | N |
| Ethanal | 0 | 1 | STO-3G | 38 | N |
| Ethanoate Ion | -1 | 1 | STO-3G | 46 | N |
| Ethanoic Acid | 0 | 1 | STO-3G | 48 | N |
| Hydrogen Peroxide | 0 | 1 | STO-3G | 24 | N |
| Ethanamide | 0 | 1 | STO-3G | 50 | N |
| Methylamine | 0 | 1 | STO-3G | 30 | N |
| Dimethylamine | 0 | 1 | STO-3G | 44 | N |

| | | | | | |
|---|---|---|---|---|---|
| Ammonia | 0 | 1 | STO-3G | 16 | Y |
| Ammonium | 1 | 1 | STO-3G | 18 | Y |
| Nitrogen Dioxide | 0 | 2 | STO-3G | 30 | N |
| Lithium Hydroxide | 0 | 1 | STO-3G | 22 | Y |
| Sodium Hydroxide | 0 | 1 | STO-3G | 30 | N |
| $H_2$ | 0 | 1 | STO-3G | 4 | Y |
| Lithium Hydride | 0 | 1 | STO-3G | 12 | Y |
| Beryllium Hydride | 0 | 1 | STO-3G | 14 | Y |
| $N_2$ | 0 | 1 | STO-3G | 20 | Y |
| $O_2$ | 0 | 3 | STO-3G | 20 | Y |
| $O_2$ | 0 | 1 | STO-3G | 20 | Y |
| $F_2$ | 0 | 1 | STO-3G | 20 | Y |
| Sodium Hydride | 0 | 1 | STO-3G | 20 | Y |
| Magnesium Hydride | 0 | 1 | STO-3G | 22 | Y |
| $Cl_2$ | 0 | 1 | STO-3G | 36 | N |
| HCl | 0 | 1 | STO-3G | 20 | Y |
| HF | 0 | 1 | STO-3G | 12 | Y |
| Carbon Dioxide | 0 | 1 | STO-3G | 30 | N |
| Carbon Monoxide | 0 | 1 | STO-3G | 20 | Y |
| Water | 0 | 1 | STO-3G | 14 | Y |
| Methylene | 0 | 3 | STO-3G | 14 | Y |
| Hydroxide | -1 | 1 | STO-3G | 12 | Y |
| $HNO_3$ | 0 | 1 | STO-3G | 42 | N |
| Nitrate | -1 | 1 | STO-3G | 40 | N |

| | | | | | |
|---|---|---|---|---|---|
| H | 0 | 2 | STO-3G | 2 | N |
| He | 0 | 1 | STO-3G | 2 | N |
| Li | 0 | 2 | STO-3G | 10 | Y |
| Be | 0 | 1 | STO-3G | 10 | Y |
| B | 0 | 2 | STO-3G | 10 | Y |
| C | 0 | 3 | STO-3G | 10 | Y |
| N | 0 | 4 | STO-3G | 10 | N |
| O | 0 | 3 | STO-3G | 10 | N |
| F | 0 | 2 | STO-3G | 10 | N |
| Ne | 0 | 1 | STO-3G | 10 | N |
| Na | 0 | 2 | STO-3G | 18 | Y |
| Mg | 0 | 1 | STO-3G | 18 | Y |
| Si | 0 | 3 | STO-3G | 18 | Y |
| P | 0 | 4 | STO-3G | 18 | Y |
| S | 0 | 3 | STO-3G | 18 | Y |
| Cl | 0 | 2 | STO-3G | 18 | Y |
| Ar | 0 | 1 | STO-3G | 18 | N |
| K | 0 | 2 | STO-3G | 26 | N |
| I | 0 | 2 | STO-3G | 54 | N |
| Br | 0 | 2 | STO-3G | 36 | N |
| $I^-$ | -1 | 1 | STO-3G | 54 | N |
| $Br^-$ | -1 | 1 | STO-3G | 36 | N |
| $Cl^-$ | -1 | 1 | STO-3G | 18 | N |
| $H_2$ | 0 | 1 | 3-21G | 8 | Y |

| | | | | | |
|---|---|---|---|---|---|
| $H_2$ | 0 | 1 | 6-31G | 8 | Y |
| $H_2$ | 0 | 1 | 6-31G** | 20 | Y |
| $H_2$ | 0 | 1 | 6-311G* | 12 | Y |
| $H_2$ | 0 | 1 | 6-311G** | 24 | N |
| $HeH^+$ | 1 | 1 | STO-3G | 4 | Y |
| $HeH^+$ | 1 | 1 | 3-21G | 8 | Y |
| $HeH^+$ | 1 | 1 | 6-31G | 8 | Y |
| $HeH^+$ | 1 | 1 | 6-31G** | 20 | Y |
| $HeH^+$ | 1 | 1 | 6-311G* | 12 | Y |
| $HeH^+$ | 1 | 1 | 6-311G** | 24 | N |
| Methylene | 0 | 3 | 3-21G | 14 | N |
| HF | 0 | 1 | 3-21G | 22 | Y |
| Lithium Hydride | 0 | 1 | 3-21G | 22 | Y |
| $H_2O$ | 0 | 1 | 3-21G | 26 | N |
| $H3^+$ | 1 | 1 | STO-3G | 6 | Y |
| $H3^+$ | 1 | 1 | 3-21G | 12 | Y |
| $CO_3$ | 0 | 1 | STO-3G | 40 | N |
| Magnesium Hydroxide | 0 | 1 | STO-3G | 42 | N |
| $H_2S$ | 0 | 1 | STO-3G | 22 | Y |

# Gate.act()

# B

This appendix contains the code for the `act` method of `Gate` objects. When provided a vector representing a state of a register of qubits, this returns a vector representing the state of the register after the gate has acted upon it.

Listing B.1: The `act` method of `Gate`

```
def act(self,state,debugOn=0):
    '''act the gate on a state vector, return the
       state vector.

    args:
    state -- a dense or sparse state vector to be
       acted upon by the gate
    debugOn -- print debug info?

    '''
    if debugOn:
        initialNorm = state.conjugate().transpose()
            * state
    state = scipy.sparse.csr_matrix(state,dtype=
       numpy.complex128)
```

```
numQubits = int(math.log(state.shape[0],2))
if self.type == 'i':
    newState = state * numpy.exp(self.angle  *
        -2j * numpy.pi)

elif self.type == 'cnot':
    controlQubit = self.qubits[0]
    controlQubitDec = 2**controlQubit
    targetQubit = self.qubits[1]
    targetQubitDec = 2**targetQubit
    wtf = [x for x in range(2**numQubits) if (
        controlQubitDec & x != 0)]
    unchangedIndices = [x for x in range(2**
        numQubits) if (x not in wtf)]
    newState2 = [state[x,0] if x in
        unchangedIndices else state[x^
        targetQubitDec, 0] for x in range(2**
        numQubits)]
    newState = scipy.sparse.csr_matrix(newState2
        ,dtype=numpy.complex128).transpose()


elif self.type in ['rz','rx','ry']:
    realAngle = self.angle*2*numpy.pi
    identityCoefficient = numpy.cos(realAngle/2)
    pauliCoefficient = -1j * numpy.sin(realAngle
        /2)
    stateCopy = copy.deepcopy(state)
```

```
        firstComponent = identityCoefficient *
            stateCopy
        if self.type == 'rz':
            secondComponent = pauliCoefficient *
                directFermions.oneZOnState(state,self
                .qubits)
        newState = firstComponent + secondComponent

    elif self.type == 'cz':
        firstQubitMask = 2**self.qubits[0]
        secondQubitMask = 2**self.qubits[1]
        mask = firstQubitMask + secondQubitMask
        indicesToPhase = [x for x in range(2**
            numQubits) if x & mask == mask]
        for index in indicesToPhase:
            state[index,0] = state[index,0] * -1
        newState = state

    elif self.type == 'h':
        state2 = copy.deepcopy(state)
        xComponent = directFermions.oneXOnState(
            state,self.qubits)
        zComponent = directFermions.oneZOnState(
            state2,self.qubits)
        newState = (2.**-(1./2.)) * (xComponent +
            zComponent)

    elif self.type == 'hy':
        '''hy = 1/sqrt(2)  I + iX'''
        firstTerm = copy.deepcopy(state)
```

```
            zComponent = 1j * directFermions.oneXOnState
                (state,self.qubits)
            newState = (2.**-(1./2.)) * (firstTerm +
                zComponent)

        elif self.type == 'hyinv':
            '''hy = 1/sqrt(") I - iX'''
            firstTerm = copy.deepcopy(state)
            zComponent = -1j * directFermions.
                oneXOnState(state,self.qubits)
            newState = (2.**-(1./2.)) * (firstTerm +
                zComponent)

        else:
            print('error, no action for gate ' + self.
                type + ' found')

        if debugOn:
            newNorm = newState.conjugate().transpose() *
                 newState
            if newNorm - initialNorm >= 0.01:
                print ('Error - gate ' + self.type + '
                    on qubit ' + str(self.qubits) + '
                    increasing norm')
                print ('norm ' + str(newNorm -
                    initialNorm))
            if initialNorm - newNorm >= 0.01:
                print ('Error - gate ' + self.type + '
                    on qubit ' + str(self.qubits) + '
                    decreasing norm')
```

```
            print ('norm ' + str(newNorm -
                initialNorm))

    return newState
```

# C Hydrogen Hamiltonian orderings

This appendix shows the oplist format of molecular hydrogen in an STO-3G basis, in a variety of ordering schemes. The oplist format is described in Chapter 3.

Listing C.1: The oplist describing the Jordan-Wigner Hamiltonian for molecular Hydrogen in an STO-3G basis, using a magnitude ordering.

```
[[(-0.8126099824999999+0j), [0, 0, 0, 0]],
 [(-0.222796615+0j), [0, 3, 0, 0]],
 [(-0.22279661499999995+0j), [3, 0, 0, 0]],
 [(0.174349485+0j), [3, 3, 0, 0]],
 [(0.17120109750000007+0j), [0, 0, 0, 3]],
 [(0.17120109750000007+0j), [0, 0, 3, 0]],
 [(0.1686232725+0j), [0, 0, 3, 3]],
 [(0.1658680075+0j), [0, 3, 3, 0]],
 [(0.1658680075+0j), [3, 0, 0, 3]],
 [(0.12054612249999999+0j), [0, 3, 0, 3]],
 [(0.12054612249999999+0j), [3, 0, 3, 0]],
 [(-0.045321885+0j), [1, 1, 2, 2]],
```

```
 [(0.045321885+0j), [1, 2, 2, 1]],
 [(0.045321885+0j), [2, 1, 1, 2]],
 [(-0.045321885+0j), [2, 2, 1, 1]]]
```

Listing C.2: The oplist describing the Jordan-Wigner Hamiltonian for molecular Hydrogen in an STO-3G basis, using a lexicographic ordering.

```
[[(-0.8126099824999999+0j), [0, 0, 0, 0]],
 [(-0.22279661499999995+0j), [3, 0, 0, 0]],
 [(-0.222796615+0j), [0, 3, 0, 0]],
 [(0.174349485+0j), [3, 3, 0, 0]],
 [(0.17120109750000007+0j), [0, 0, 3, 0]],
 [(0.12054612249999999+0j), [3, 0, 3, 0]],
 [(0.1658680075+0j), [0, 3, 3, 0]],
 [(-0.045321885+0j), [2, 2, 1, 1]],
 [(0.045321885+0j), [1, 2, 2, 1]],
 [(0.045321885+0j), [2, 1, 1, 2]],
 [(-0.045321885+0j), [1, 1, 2, 2]],
 [(0.17120109750000007+0j), [0, 0, 0, 3]],
 [(0.1658680075+0j), [3, 0, 0, 3]],
 [(0.12054612249999999+0j), [0, 3, 0, 3]],
 [(0.1686232725+0j), [0, 0, 3, 3]]]
```

Listing C.3: The oplist describing the Jordan-Wigner Hamiltonian for molecular Hydrogen in an STO-3G basis, using a lexoMag ordering.

```
[[(-0.8126099824999999+0j), [0, 0, 0, 0]],
 [(-0.22279661499999995+0j), [3, 0, 0, 0]],
 [(-0.222796615+0j), [0, 3, 0, 0]],
```

```
 [(0.174349485+0j), [3, 3, 0, 0]],
 [(0.17120109750000007+0j), [0, 0, 3, 0]],
 [(0.12054612249999999+0j), [3, 0, 3, 0]],
 [(0.17120109750000007+0j), [0, 0, 0, 3]],
 [(0.1658680075+0j), [0, 3, 3, 0]],
 [(0.1686232725+0j), [0, 0, 3, 3]],
 [(-0.045321885+0j), [2, 2, 1, 1]],
 [(0.1658680075+0j), [3, 0, 0, 3]],
 [(0.045321885+0j), [1, 2, 2, 1]],
 [(0.12054612249999999+0j), [0, 3, 0, 3]],
 [(0.045321885+0j), [2, 1, 1, 2]],
 [(-0.045321885+0j), [1, 1, 2, 2]]]
```

Listing C.4: The oplist describing the Jordan-Wigner Hamiltonian for molecular Hydrogen in an STO-3G basis, using a commutator ordering.

```
[[(-0.8126099824999999+0j), [0, 0, 0, 0]],
 [(-0.222796615+0j), [0, 3, 0, 0]],
 [(-0.045321885+0j), [2, 2, 1, 1]],
 [(-0.22279661499999995+0j), [3, 0, 0, 0]],
 [(0.045321885+0j), [1, 2, 2, 1]],
 [(0.17120109750000007+0j), [0, 0, 3, 0]],
 [(0.045321885+0j), [2, 1, 1, 2]],
 [(0.17120109750000007+0j), [0, 0, 0, 3]],
 [(-0.045321885+0j), [1, 1, 2, 2]],
 [(0.174349485+0j), [3, 3, 0, 0]],
 [(0.1686232725+0j), [0, 0, 3, 3]],
 [(0.1658680075+0j), [0, 3, 3, 0]],
 [(0.1658680075+0j), [3, 0, 0, 3]],
 [(0.12054612249999999+0j), [3, 0, 3, 0]],
```

```
 [(0.12054612249999999+0j), [0, 3, 0, 3]]]
```

Listing C.5: The oplist describing the Jordan-Wigner Hamiltonian for molecular Hydrogen in an STO-3G basis, using a reverseCommutator ordering.

```
[[(-0.8126099824999999+0j), [0, 0, 0, 0]],
 [(0.174349485+0j), [3, 3, 0, 0]],
 [(0.1686232725+0j), [0, 0, 3, 3]],
 [(0.1658680075+0j), [0, 3, 3, 0]],
 [(0.1658680075+0j), [3, 0, 0, 3]],
 [(0.12054612249999999+0j), [3, 0, 3, 0]],
 [(0.12054612249999999+0j), [0, 3, 0, 3]],
 [(-0.222796615+0j), [0, 3, 0, 0]],
 [(-0.045321885+0j), [2, 2, 1, 1]],
 [(-0.22279661499999995+0j), [3, 0, 0, 0]],
 [(0.045321885+0j), [1, 2, 2, 1]],
 [(0.17120109750000007+0j), [0, 0, 3, 0]],
 [(0.045321885+0j), [2, 1, 1, 2]],
 [(0.17120109750000007+0j), [0, 0, 0, 3]],
 [(-0.045321885+0j), [1, 1, 2, 2]]]
```

# Bibliography


1. Kortagere, S., Lill, M. & Kerrigan, J. "Role of Computational Methods in Pharmaceutical Sciences". *Methods in Molecular Biology* 929, 21–48. (Aug. 2012).
2. Deglmann, P., Schäfer, A. & Lennartz, C. "Application of Quantum Calculations in the Chemical Industry—An Overview". *International Journal of Quantum Chemistry* 115, 107–136. (Feb. 2015).
3. Villaverde, J. J., López-Goti, C., Alcamí, M., Lamsabhi, A. M., Alonso-Prados, J. L. & Sandín-España, P. "Quantum Chemistry in Environmental Pesticide Risk Assessment". *Pest Management Science* 73, 2199–2202. (Nov. 2017).
4. Pople, J. A. *A Priori Geometry Predictions* in *Applications of Electronic Structure Theory* (ed Schaefer, H. F.) 1–27 (Springer US, Boston, MA, 1977). ISBN: 978-1-4684-8541-7.
5. Schlegel, H. B. "Optimization of Equilibrium Geometries and Transition Structures". *Journal of Computational Chemistry* 3, 214–218. (June 1982).

6. Withnall, R., Chowdhry, B. Z., Bell, S. & Dines, T. J. "Computational Chemistry Using Modern Electronic Structure Methods". *Journal of Chemical Education* 84, 1364. (Aug. 2007).

7. Szabo, A. & Ostlund, N. S. *Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory* 466 pp. ISBN: 978-0-486-69186-2 (Dover Publications, Mineola, N.Y, 1996).

8. Hess, B. A., Schaad, L. J., Carsky, P. & Zahradnik, R. "Ab Initio Calculations of Vibrational Spectra and Their Use in the Identification of Unusual Molecules". *Chemical Reviews* 86, 709–730. (Aug. 1986).

9. Bouř, P., Sopková, J., Bednárová, L., Maloň, P. & Keiderling, T. A. "Transfer of Molecular Property Tensors in Cartesian Coordinates: A New Algorithm for Simulation of Vibrational Spectra". *Journal of Computational Chemistry* 18, 646–659. (Apr. 1997).

10. Crawford, T. D., Tam, M. C. & Abrams, M. L. "The Current State of Ab Initio Calculations of Optical Rotation and Electronic Circular Dichroism Spectra". *The Journal of Physical Chemistry A* 111, 12057–12068. (Dec. 2007).

11. Mayer, I. "Bond Orders and Valences from Ab Initio Wave Functions". *International Journal of Quantum Chemistry* 29, 477–483. (Mar. 1986).

12. Bernardi, F., Olivucci, M. & Robb, M. "Modelling Photochemical Reactivity of Organic Systems — A New Challenge to Quantum

Computational Chemistry". *Israel Journal of Chemistry* 33, 265–276. (Jan. 1993).

13. Baldridge, K. K., Gordon, M. S., Steckler, R. & Truhlar, D. G. "Ab Initio Reaction Paths and Direct Dynamics Calculations". *The Journal of Physical Chemistry* 93, 5107–5119. (June 1989).
14. Althorpe, S. C. & Clary, D. C. "Quantum Scattering Calculations on Chemical Reactions". *Annual Review of Physical Chemistry* 54, 493–529. (Oct. 2003).
15. Nyman, G. "Computational Methods of Quantum Reaction Dynamics". *International Journal of Quantum Chemistry* 114, 1183–1198. (Sept. 2014).
16. Dirac, P. A. M. "Quantum Mechanics of Many-Electron Systems". *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* 123, 714–733. (Apr. 1929).
17. Knowles, P. J. & Handy, N. C. "A New Determinant-Based Full Configuration Interaction Method". *Chemical Physics Letters* 111, 315–321. (Nov. 1984).
18. Parr, R. G. & Yang, W. *Density-Functional Theory of Atoms and Molecules* ISBN: 978-0-19-535773-8 (Oxford University Press, Apr. 1989).
19. Stewart, J. J. P. *Semiempirical Molecular Orbital Methods* in *Reviews in Computational Chemistry* (eds Lipkowitz, K. B. & Boyd, D. B.) 45–81 (John Wiley & Sons, Inc., 1990). ISBN: 978-0-470-12578-6.

20. Cohen, A. J., Mori-Sánchez, P. & Yang, W. "Insights into Current Limitations of Density Functional Theory". *Science* 321, 792–794. (Aug. 2008).

21. Perdomo-Ortiz, A., Dickson, N., Drew-Brook, M., Rose, G. & Aspuru-Guzik, A. "Finding Low-Energy Conformations of Lattice Protein Models by Quantum Annealing". *Scientific Reports* 2, 571. (Aug. 2012).

22. Hartree, D. R. "The Wave Mechanics of an Atom with a Non-Coulomb Central Field. Part I. Theory and Methods". *Mathematical Proceedings of the Cambridge Philosophical Society* 24, 89–110. (Jan. 1928).

23. Fock, V. "Näherungsmethode zur Lösung des quantenmechanischen Mehrkörperproblems". *Zeitschrift für Physik* 61, 126–148. (Jan. 1930).

24. Slater, J. C. "The Theory of Complex Spectra". *Physical Review* 34, 1293–1322. (Nov. 1929).

25. Møller, C. & Plesset, M. S. "Note on an Approximation Treatment for Many-Electron Systems". *Physical Review* 46, 618. (Oct. 1934).

26. Bartlett, R. J. & Silver, D. M. "Correlation Energy in LiH, BH, and HF with Many-Body Perturbation Theory Using Slater-Type Atomic Orbitals". *International Journal of Quantum Chemistry* 8, 271–276. (Jan. 1974).

27. Handy, N. C. *Full Configuration Interaction and Møller-Plesset Theory* in *Relativistic and Electron Correlation Effects in Molecules and Solids* (ed Malli, G. L.) 133–160 (Springer US, Boston, MA, 1994). ISBN: 978-1-4899-1340-1.

28. Szabo, A. & Ostlund, N. S. *Truncated CI and the Size-Consistency Problem* in *Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory* 261–269 (Dover Publications, Mineola, N.Y, 1996). ISBN: 978-0-486-69186-2.

29. Raghavachari, K., Trucks, G. W., Pople, J. A. & Head-Gordon, M. "A Fifth-Order Perturbation Comparison of Electron Correlation Theories". *Chemical Physics Letters* 157, 479–483. (May 1989).

30. Řezáč, J. & Hobza, P. "Describing Noncovalent Interactions beyond the Common Approximations: How Accurate Is the "Gold Standard," CCSD(T) at the Complete Basis Set Limit?" *Journal of Chemical Theory and Computation* 9, 2151–2155. (May 2013).

31. Boys, S. F., Cook, G. B., Reeves, C. M. & Shavitt, I. "Automatic Fundamental Calculations of Molecular Structure". *Nature* 178, 1207–1209. (Dec. 1956).

32. Hutchings, J. *Engineering and Science, Volume 23:5, February 1960* http://resolver.caltech.edu/CaltechES:23.5.0, Accessed: 14/06/2018.

33. Benioff, P. "The Computer as a Physical System: A Microscopic Quantum Mechanical Hamiltonian Model of Computers as Repres-

ented by Turing Machines". *Journal of Statistical Physics* 22, 563–591. (May 1980).

34. Deutsch, D. "Quantum Theory, the Church-Turing Principle and the Universal Quantum Computer". *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* 400, 97–117. (July 1985).

35. Shor, P. "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer". *SIAM Journal on Computing* 26, 1484–1509. (Oct. 1997).

36. Nielsen, M. A. & Chuang, I. L. *Quantum Computation and Quantum Information: 10th Anniversary Edition* 709 pp. ISBN: 978-1-139-49548-6 (Cambridge University Press, Dec. 2010).

37. DiVincenzo, D. P. "The Physical Implementation of Quantum Computation". *Fortschritte der Physik* 48, 771–783. (Sept. 2000).

38. Häffner, H., Roos, C. F. & Blatt, R. "Quantum Computing with Trapped Ions". *Physics Reports* 469, 155–203. (Dec. 2008).

39. Kok, P., Munro, W. J., Nemoto, K., Ralph, T. C., Dowling, J. P. & Milburn, G. J. "Linear Optical Quantum Computing with Photonic Qubits". *Reviews of Modern Physics* 79, 135–174. (Jan. 2007).

40. Vandersypen, L. M. K. & Chuang, I. L. "NMR Techniques for Quantum Control and Computation". *Reviews of Modern Physics* 76, 1037–1069. (Jan. 2005).

41. Kloeffel, C. & Loss, D. "Prospects for Spin-Based Quantum Computing in Quantum Dots". *Annual Review of Condensed Matter Physics* 4, 51–81. (Mar. 2013).

42. Childress, L. & Hanson, R. "Diamond NV Centers for Quantum Computing and Quantum Networks". *MRS Bulletin* 38, 134–138. (Feb. 2013).

43. Pednault, E., Gunnels, J. A., Nannicini, G., Horesh, L., Magerlein, T., Solomonik, E. & Wisnieff, R. "Breaking the 49-Qubit Barrier in the Simulation of Quantum Circuits". *arXiv:1710.05867 [quant-ph].* (Oct. 2017).

44. Mohseni, M., Read, P., Neven, H., Boixo, S., Denchev, V., Babbush, R., Fowler, A., Smelyanskiy, V. & Martinis, J. "Commercialize Quantum Technologies in Five Years". *Nature News* 543, 171. (Mar. 2017).

45. Reiher, M., Wiebe, N., Svore, K. M., Wecker, D. & Troyer, M. "Elucidating Reaction Mechanisms on Quantum Computers". *Proceedings of the National Academy of Sciences of the United States of America* 114, 7555–7560. (July 2017).

46. Boixo, S., Isakov, S. V., Smelyanskiy, V. N., Babbush, R., Ding, N., Jiang, Z., Bremner, M. J., Martinis, J. M. & Neven, H. "Characterizing Quantum Supremacy in Near-Term Devices". *Nature Physics* 14, 595–600. (June 2018).

47. Whitney, M. G., Isailovic, N., Patel, Y. & Kubiatowicz, J. *A Fault Tolerant, Area Efficient Architecture for Shor's Factoring Algorithm* in *Proceedings of the 36th Annual International Symposium on Computer Architecture* (ACM, New York, NY, USA, 2009), 383–394. ISBN: 978-1-60558-526-0.

48. Feynman, R. P. "Simulating Physics with Computers". *International Journal of Theoretical Physics* 21, 467–488. (June 1982).

49. Lloyd, S. "Universal Quantum Simulators". *Science* 273, 1073–1078. (Aug. 1996).

50. Abrams, D. S. & Lloyd, S. "Simulation of Many-Body Fermi Systems on a Universal Quantum Computer". *Physical Review Letters* 79, 2586–2589. (Sept. 1997).

51. Aspuru-Guzik, A., Dutoi, A. D., Love, P. J. & Head-Gordon, M. "Simulated Quantum Computation of Molecular Energies". *Science* 309, 1704–1707. (Sept. 2005).

52. Olson, J., Cao, Y., Romero, J., Johnson, P., Dallaire-Demers, P.-L., Sawaya, N., Narang, P., Kivlichan, I., Wasielewski, M. & Aspuru-Guzik, A. "Quantum Information and Computation for Chemistry". *arXiv:1706.05413 [physics, physics:quant-ph]*. (June 2017).

53. Smith, B. E. "Nitrogenase Reveals Its Inner Secrets". *Science* 297, 1654–1655. (Sept. 2002).

54. Booth, G. H. & Alavi, A. "Approaching Chemical Accuracy Using Full Configuration-Interaction Quantum Monte Carlo: A Study of Ionization Potentials". *The Journal of Chemical Physics* 132, 174104. (May 2010).

55. Bourzac, K. "Chemistry Is Quantum Computing's Killer App". *C&EN Global Enterprise* 95, 27–31. (Oct. 2017).

56. Lidar, D. A. & Wang, H. "Calculating the Thermal Rate Constant with Exponential Speedup on a Quantum Computer". *Physical Review E* 59, 2429–2438. (Feb. 1999).

57. Kassal, I. & Aspuru-Guzik, A. "Quantum Algorithm for Molecular Properties and Geometry Optimization". *The Journal of Chemical Physics* 131, 224102. (Dec. 2009).

58. Kassal, I., Jordan, S. P., Love, P. J., Mohseni, M. & Aspuru-Guzik, A. "Polynomial-Time Quantum Algorithm for the Simulation of Chemical Dynamics". *Proceedings of the National Academy of Sciences* 105, 18681–18686. (Feb. 2008).

59. Lanyon, B. P., Whitfield, J. D., Gillett, G. G., Goggin, M. E., Almeida, M. P., Kassal, I., Biamonte, J. D., Mohseni, M., Powell, B. J., Barbieri, M., Aspuru-Guzik, A. & White, A. G. "Towards Quantum Chemistry on a Quantum Computer". *Nature Chemistry* 2, 106–111. (Jan. 2010).

60. Du, J., Xu, N., Peng, X., Wang, P., Wu, S. & Lu, D. "NMR Implementation of a Molecular Hydrogen Quantum Simulation with

Adiabatic State Preparation". *Physical Review Letters* 104, 030502. (Jan. 2010).

61. Wang, Y., Dolde, F., Biamonte, J., Babbush, R., Bergholm, V., Yang, S., Jakobi, I., Neumann, P., Aspuru-Guzik, A., Whitfield, J. D. & Wrachtrup, J. "Quantum Simulation of Helium Hydride Cation in a Solid-State Spin Register". *ACS Nano* 9, 7769–7774. (Aug. 2015).

62. O'Malley, P. J. J., Babbush, R., Kivlichan, I. D., Romero, J., McClean, J. R., Barends, R., Kelly, J., Roushan, P., Tranter, A., Ding, N., Campbell, B., Chen, Y., Chen, Z., Chiaro, B., Dunsworth, A., Fowler, A. G., Jeffrey, E., Lucero, E., Megrant, A., Mutus, J. Y., Neeley, M., Neill, C., Quintana, C., Sank, D., Vainsencher, A., Wenner, J., White, T. C., Coveney, P. V., Love, P. J., Neven, H., Aspuru-Guzik, A. & Martinis, J. M. "Scalable Quantum Simulation of Molecular Energies". *Physical Review X* 6, 031007. (July 2016).

63. Kandala, A., Mezzacapo, A., Temme, K., Takita, M., Brink, M., Chow, J. M. & Gambetta, J. M. "Hardware-Efficient Variational Quantum Eigensolver for Small Molecules and Quantum Magnets". *Nature* 549, 242–246. (Sept. 2017).

64. McClean, J. R., Romero, J., Babbush, R. & Aspuru-Guzik, A. "The Theory of Variational Hybrid Quantum-Classical Algorithms". *New Journal of Physics* 18, 023023. (Feb. 2016).

65. Peruzzo, A., McClean, J., Shadbolt, P., Yung, M.-H., Zhou, X.-Q., Love, P. J., Aspuru-Guzik, A. & O'Brien, J. L. "A Variational Eigen-

value Solver on a Photonic Quantum Processor". *Nature Communications* 5, 4213. (July 2014).

66. Yung, M.-H., Casanova, J., Mezzacapo, A., McClean, J., Lamata, L., Aspuru-Guzik, A. & Solano, E. "From Transistor to Trapped-Ion Computers for Quantum Chemistry". *Scientific Reports* 4, 3589. (Jan. 2014).

67. Taube, A. G. & Bartlett, R. J. "New Perspectives on Unitary Coupled-Cluster Theory". *International Journal of Quantum Chemistry* 106, 3393–3401. (Aug. 2006).

68. Babbush, R., Love, P. J. & Aspuru-Guzik, A. "Adiabatic Quantum Simulation of Quantum Chemistry". *Scientific Reports* 4, 6603. (Oct. 2014).

69. Cao, Y., Babbush, R., Biamonte, J. & Kais, S. "Hamiltonian Gadgets with Reduced Resource Requirements". *Physical Review A* 91, 012315. (Jan. 2015).

70. Aharonov, D., van Dam, W., Kempe, J., Landau, Z., Lloyd, S. & Regev, O. "Adiabatic Quantum Computation Is Equivalent to Standard Quantum Computation". *SIAM Journal on Computing* 37, 166–194. (Jan. 2007).

71. Babbush, R., Berry, D. W., Kivlichan, I. D., Wei, A. Y., Love, P. J. & Aspuru-Guzik, A. "Exponentially More Precise Quantum Simulation of Fermions in Second Quantization". *New Journal of Physics* 18, 033032. (Mar. 2016).

72. Toloui, B. & Love, P. J. "Quantum Algorithms for Quantum Chemistry Based on the Sparsity of the CI-Matrix". *arXiv:1312.2579 [quant-ph]*. (Dec. 2013).

73. Babbush, R., Berry, D. W., Sanders, Y. R., Kivlichan, I. D., Scherer, A., Wei, A. Y., Love, P. J. & Aspuru-Guzik, A. "Exponentially More Precise Quantum Simulation of Fermions in the Configuration Interaction Representation". *Quantum Science and Technology* 3, 015006. (Dec. 2017).

74. Zalka, C. "Simulating Quantum Systems on a Quantum Computer". *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* 454, 313–322. (Aug. 1998).

75. Kivlichan, I. D., Wiebe, N., Babbush, R. & Aspuru-Guzik, A. "Bounding the Costs of Quantum Simulation of Many-Body Physics in Real Space". *Journal of Physics A: Mathematical and Theoretical* 50, 305301. (June 2017).

76. Wecker, D., Bauer, B., Clark, B. K., Hastings, M. B. & Troyer, M. "Can Quantum Chemistry Be Performed on a Small Quantum Computer?" *arXiv:1312.1695 [physics, physics:quant-ph]*. (Dec. 2013).

77. Poulin, D., Hastings, M. B., Wecker, D., Wiebe, N., Doberty, A. C. & Troyer, M. "The Trotter Step Size Required for Accurate Quantum Simulation of Quantum Chemistry". *Quantum Information & Computation* 15, 361–384. (Apr. 2015).

78. Bravyi, S. B. & Kitaev, A. Y. "Fermionic Quantum Computation". *Annals of Physics* 298, 210–226. (May 2002).

79. Seeley, J. T., Richard, M. J. & Love, P. J. "The Bravyi-Kitaev Transformation for Quantum Computation of Electronic Structure". *The Journal of Chemical Physics* 137, 224109. (Dec. 2012).

80. Tranter, A., Sofia, S., Seeley, J., Kaicher, M., McClean, J., Babbush, R., Coveney, P. V., Mintert, F., Wilhelm, F. & Love, P. J. "The Bravyi-Kitaev Transformation: Properties and Applications". *International Journal of Quantum Chemistry* 115, 1431–1441. (Oct. 2015).

81. Troyer, M., Wecker, D. B., Clark, B. & Smith, B. J. *US Patent* 20150006597A1 (2015).

82. Setia, K. & Whitfield, J. D. "Bravyi-Kitaev Superfast Simulation of Fermions on a Quantum Computer". *arXiv:1712.00446 [quant-ph].* (Dec. 2017).

83. Kresse, G. & Furthmüller, J. "Efficient Iterative Schemes for Ab Initio Total-Energy Calculations Using a Plane-Wave Basis Set". *Physical Review B* 54, 11169–11186. (Oct. 1996).

84. Shang, H., Xiang, H., Li, Z. & Yang, J. "Linear Scaling Electronic Structure Calculations with Numerical Atomic Basis Set". *International Reviews in Physical Chemistry* 29, 665–691. (Oct. 2010).

85. Binkley, J. S., Pople, J. A. & Hehre, W. J. "Self-Consistent Molecular Orbital Methods. 21. Small Split-Valence Basis Sets for First-Row Ele-

ments". *Journal of the American Chemical Society* 102, 939–947. (Jan. 1980).

86. Dunning, T. H. "Gaussian Basis Sets for Use in Correlated Molecular Calculations. I. The Atoms Boron through Neon and Hydrogen". *The Journal of Chemical Physics* 90, 1007–1023. (Jan. 1989).

87. Löwdin, P. "On the Non-Orthogonality Problem Connected with the Use of Atomic Wave Functions in the Theory of Molecules and Crystals". *The Journal of Chemical Physics* 18, 365–375. (Mar. 1950).

88. McClean, J. R., Babbush, R., Love, P. J. & Aspuru-Guzik, A. "Exploiting Locality in Quantum Computation for Quantum Chemistry". *The Journal of Physical Chemistry Letters* 5, 4368–4380. (Dec. 2014).

89. Hartree, D. R. "The Calculation of Atomic Structures". *Reports on Progress in Physics* 11, 113. (1947).

90. Szabo, A. & Ostlund, N. S. *The Hartree-Fock Approximation* in *Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory* 180–230 (Dover Publications, Mineola, N.Y, 1996). ISBN: 978-0-486-69186-2.

91. Koopmans, T. "Über Die Zuordnung von Wellenfunktionen Und Eigenwerten Zu Den Einzelnen Elektronen Eines Atoms". *Physica* 1, 104–113. (Jan. 1934).

92. Strout, D. L. & Scuseria, G. E. "A Quantitative Study of the Scaling Properties of the Hartree–Fock Method". *The Journal of Chemical Physics* 102, 8448–8452. (June 1995).

93. Babbush, R., McClean, J., Wecker, D., Aspuru-Guzik, A. & Wiebe, N. "Chemical Basis of Trotter-Suzuki Errors in Quantum Chemistry Simulation". *Physical Review A* 91, 022311. (Feb. 2015).

94. Whitfield, J. D., Biamonte, J. & Aspuru-Guzik, A. "Simulation of Electronic Structure Hamiltonians Using Quantum Computers". *Molecular Physics* 109, 735–750. (Mar. 2011).

95. Havlíček, V., Troyer, M. & Whitfield, J. D. "Operator Locality in Quantum Simulation of Fermionic Models". *Physical Review A* 95. (Mar. 2017).

96. Zanardi, P. "Quantum Entanglement in Fermionic Lattices". *Physical Review A* 65, 042101. (Mar. 2002).

97. Derzhko, O. "Jordan-Wigner Fermionization for Spin-1/2 Systems in Two Dimensions: A Brief Review". *arXiv:cond-mat/0101188*. (Jan. 2001).

98. Hatano, N. & Suzuki, M. *Finding Exponential Product Formulas of Higher Orders* in *Quantum Annealing and Other Optimization Methods* (eds Das, A. & Chakrabarti, B. K.) *Lecture Notes in Physics* 679, 37–68 (Springer Berlin Heidelberg, Jan. 2005). ISBN: 978-3-540-27987-7 978-3-540-31515-5.

99. Suzuki, M. "Generalized Trotter's Formula and Systematic Approximants of Exponential Operators and Inner Derivations with Applications to Many-Body Problems". *Communications in Mathematical Physics* 51, 183–190. (June 1976).

100. Kitaev, A. Y. "Quantum Measurements and the Abelian Stabilizer Problem". *arXiv:quant-ph/9511026*. (Nov. 1995).

101. Veis, L. & Pittner, J. "Adiabatic State Preparation Study of Methylene". *The Journal of Chemical Physics* 140, 214111. (June 2014).

102. Dobšíček, M., Johansson, G., Shumeiko, V. & Wendin, G. "Arbitrary Accuracy Iterative Quantum Phase Estimation Algorithm Using a Single Ancillary Qubit: A Two-Qubit Benchmark". *Physical Review A* 76, 030306. (Sept. 2007).

103. McClean, J. R., Kivlichan, I. D., Sung, K. J., Steiger, D. S., Cao, Y., Dai, C., Fried, E. S., Gidney, C., Gimby, B., Häner, T., Hardikar, T., Havlíček, V., Huang, C., Jiang, Z., Neeley, M., O'Brien, T., Ozfidan, I., Radin, M. D., Romero, J., Rubin, N., Sawaya, N. P. D., Setia, K., Sim, S., Steudtner, M., Sun, W., Zhang, F. & Babbush, R. "OpenFermion: The Electronic Structure Package for Quantum Computers". *arXiv:1710.07629 [physics, physics:quant-ph]*. (Oct. 2017).

104. Parrish, R. M., Burns, L. A., Smith, D. G. A., Simmonett, A. C., DePrince, A. E., Hohenstein, E. G., Bozkaya, U., Sokolov, A. Y., Di Remigio, R., Richard, R. M., Gonthier, J. F., James, A. M., McAlexander, H. R., Kumar, A., Saitow, M., Wang, X., Pritchard, B. P., Verma, P., Schaefer, H. F., Patkowski, K., King, R. A., Valeev, E. F., Evangelista, F. A., Turney, J. M., Crawford, T. D. & Sherrill, C. D. "Psi4 1.1: An Open-Source Electronic Structure Program Emphasizing Automation, Advanced Libraries, and Interoperability". *Journal of Chemical Theory and Computation* 13, 3185–3197. (July 2017).

105. Johnson III, R. D. *NIST Computational Chemistry Comparison and Benchmark Database NIST Standard Reference Database Number 101 Release 18* (Oct. 2016).

106. Hastings, M. B., Wecker, D., Bauer, B. & Troyer, M. "Improving Quantum Algorithms for Quantum Chemistry". *Quantum Information & Computation* 15, 1–21. (Jan. 2015).

107. Jones, E., Oliphant, T., Peterson, P. *et al.* "SciPy: Open Source Scientific Tools for Python". [Online; accessed 2018-05-02]. (2001–).

108. Cross, A. W., Bishop, L. S., Smolin, J. A. & Gambetta, J. M. "Open Quantum Assembly Language". *arXiv:1707.03429 [quant-ph]*. (July 2017).

109. Chuang, I. L. *Quantum Architectures: Qasm2circ* https://www.media.mit.edu/quanta/qasm2circ/, Accessed: 21/06/2018.

110. Fenwick, P. M. "A New Data Structure for Cumulative Frequency Tables". *Software: Practice and Experience* 24, 327–336. (Mar. 1994).

111. Tranter, A., Love, P. J., Mintert, F. & Coveney, P. V. "A Comparison of the Bravyi–Kitaev and Jordan–Wigner Transformations for the Quantum Simulation of Quantum Chemistry". *Journal of Chemical Theory and Computation* 14, 5617–5630. (Nov. 2018).

112. Tranter, A. *Quantum Computation for Electronic Structure Theory* MRes Thesis (Imperial College London, Sept. 2014).

113. Häner, T. & Steiger, D. S. *0.5 Petabyte Simulation of a 45-Qubit Quantum Circuit* in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis* (ACM, New York, NY, USA, 2017), 33:1–33:10. ISBN: 978-1-4503-5114-0.

114. Smelyanskiy, M., Sawaya, N. P. D. & Aspuru-Guzik, A. "qHiPSTER: The Quantum High Performance Software Testing Environment". *arXiv:1601.07195 [quant-ph]*. (Jan. 2016).

115. Rossi, E., Bendazzoli, G. L., Evangelisti, S. & Maynau, D. "A Full-Configuration Benchmark for the N2 Molecule". *Chemical Physics Letters* 310, 530–536. (Sept. 1999).

116. Bravyi, S., Gambetta, J. M., Mezzacapo, A. & Temme, K. "Tapering off Qubits to Simulate Fermionic Hamiltonians". *arXiv:1701.08213 [quant-ph]*. (Jan. 2017).

117. Garey, M. R., Johnson, D. S. & Stockmeyer, L. *Some Simplified NP-Complete Problems* in *Proceedings of the Sixth Annual ACM Symposium on Theory of Computing* (ACM, New York, NY, USA, 1974), 47–63.

118. Moore, G. E. "Cramming More Components Onto Integrated Circuits". *Proceedings of the IEEE* 86, 82–85. (Jan. 1998).

119. Love, P. J. *Back to The Future: A Roadmap for Quantum Simulation From Vintage Quantum Chemistry* in *Quantum Information*

*and Computation for Chemistry* 39–66 (Wiley-Blackwell, Mar. 2014). ISBN: 978-1-118-74263-1.