Multideterminant Wavefunction Development

Systematic Approach to Multideterminant Wavefunction Development

By Taewon David Kim, B.Sc.

A Thesis Submitted to the School of Graduate Studies of McMaster University

in Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

McMaster University

Doctor of Philosophy (2020)

Hamilton, Ontario (Department of Chemistry and Chemical Biology)

TITLE: Systematic Approach to Multideterminant Wavefunction Development

AUTHOR: Taewon David Kim, B.Sc. (McMaster University)

SUPERVISOR: Prof. Paul W. Ayers

PAGES: 1, 212

Abstract

Electronic structure methods aim to accurately describe the behaviour of the electrons in molecules and materials. To be applicable to arbitrary systems, these methods cannot depend on observations of specific chemical phenomena and must be derived solely from the fundamental physical constants and laws that govern all electrons. Such methods are called *ab initio* methods. *Ab initio* methods directly solve the electronic Schrödinger equation to obtain the electronic energy and wavefunction. For more than one electron, solving the electronic Schrödinger equation is impossible, so it is imperative to develop approximate methods that cater to the needs of their users, which can vary depending on the chemical systems under study, the available computational resources and time, and the desired level of accuracy. The most accessible *ab initio* approaches, including Hartree-Fock methods and Kohn-Sham density functional theory methods, assume that only one electronic configuration is needed to describe the system. While these single-reference methods are successful when describing systems where a single electron configuration dominates, like most closed-shell ground-state organic molecules in their equilibrium geometries, single-reference methods are unreliable for molecules in nonequilibrium geometries (e.g., transition states) and molecules containing unpaired electrons (e.g., transition metal complexes and radicals). For these types of multireference systems, accurate results can only be obtained if multiple electronic configurations are accounted for. Wavefunctions that incorporate many electronic configurations are called multideterminant wavefunctions.

This thesis presents a systematic approach to developing multideterminant wavefunctions. First, we establish a framework that outlines the structural components of a multideterminant wavefunction and propose several novel wavefunction ansätze. Then, we present a software package that is designed to aid the development of new wavefunctions and algorithms. Using this approach, we develop an algorithm for evaluating the geminal wavefunctions, a class of multideterminant wavefunctions that are expressed with respect to electron pairs. Finally, we explore using machine learning to solve the Schrödinger equation by presenting a neural network wavefunction ansatz and optimizing its parameters using stochastic gradient descent.

To my parents, 김종희 and 오완, whose love and sacrifice made all this possible.

Acknowledgements

First and foremost, I would like to thank Paul W. Ayers for all his guidance over the past decade. This work would not be possible without his tireless patience in answering our questions and his selfless commitment in helping us develop as scientists and researchers. Thank you for believing in me, for being my role model, and for the many late-night McDonald runs.

I would like to thank to Peter Limacher who was my guide through the early stages of my graduate studies and set the foundation for much of the content in this work. In addition, I would like to thank Patrick Bultinck and Toon Verstraelen for their help during my excursion to Ghent, and the members of CMM who made me feel welcome.

I am forever grateful to my loving parents, 김종회 and 오완, whose love, support, and selfless sacrifice remain a source of inspiration, and to my sister, Anna, who supported me during difficult times.

Finally, I'd like to thank the members of Ayers lab and fellow graduate students, who were often like a family away from home. I express my heartfelt gratitude to Xiaotian Yang, Xiaomin Huang, Kumru Dikmenli, Farnaz Heidar-Zadeh, Stijn Fias, Paul Johnson, Matthew Chan, Lian Pharoah, Wil Adams, Yingxing Cheng, Cristina Elizabeth González Espinoza, and Ramón Alain Miranda Quintana. And of course, my friends outside, Roxanne Ban, Rubhen Raiju Murugaanandan, Diem Le, and Diego Berrocal, who leant their shoulders in times of need. Thank you all for the pleasant memories.

Contents

List of Figures

Chapter 1

Introduction

While the properties of chemical substances can be observed experimentally, directly experimental observation of the underlying behaviour of atoms and molecules is typically impossible. Solving the Schrödinger equation for the ground- and excited-state wavefunctions and energies provides a complete representation of a chemical system, and can be used to predict properties, thereby aiding the interpretation of experiment. Hypothetically, given enough time and computational resources, we can examine all chemical systems without experimentation, which is especially useful when the systems under study are not amenable to experimental observations. However, solving the Schrödinger equation is not practical for many systems of interest to chemists. In this thesis, we discuss why it is difficult to solve the Schrödinger equation and present new approaches for solving the Schrödinger equation in the context of molecular electronic structure theory.

Solving the Schrödinger equation is not the only way to model chemical systems. There are a wide range of theoretical methods that do not rely on quantum mechanics, but instead on thermodynamics and classical mechanics. In molecular mechanics, for example, each atom, functional group, and moeity interact with another via a classical force field that is parameterized to fit experimental or computational benchmark data[1]. Thermodynamic properties can then be estimated by averaging either stochastically (Monte Carlo)[2] or in time (molecular dynamics, based on classical mechanics)[3]. In practice, molecular mechanics methods perform poorly when the electronic structure of molecules changes, as they do in a chemical reaction.

In addition to molecular mechanics methods, there are quantum mechanical methods based on fitting to empirical data[3] and emerging data-based machine-learning approaches[4]. These empirical (or semi-empirical, when physical reasoning and chemical data are combined) models are limited by their training data, so it is not possible to systematically increase the accuracy of the calculations without reinventing the model. Since semi-empirical models do not converge to the exact result in any systematic way, it is difficult to determine the accuracy of these methods without comparing to experimental results, so chemists usually interpret the results by appeal to intuition, which can be slow and unreliable. For chemical systems that are complex or novel, and therefore difficult to understand, chemical intuition is notoriously unreliable. For example, the reactivity of metal complexes is difficult to assess with semiempirical methods, especially for complexes containing multiple, inequivalent, redox centres[5–7]. In contrast, *ab initio* methods solve the Schrödinger equation with little to no parameterization beyond fundamental physical constants. Many *ab initio* methods can be improved systematically, approaching the exact solution in some limit, although that limit is generally computationally intractable. *Ab initio* methods, however, are usually much more expensive than empirical and semiempirical methods[8–10].

1.1 Approximations

It is not possible to exactly solve the Schrödinger equation for the electrons in a molecule except for the smallest model systems. Moreover, very accurate approximate solutions are prohibitively expensive for most systems with more than about 50 electrons. This motivates research into approximations that simplify the process of solving Schrödinger equation. In this thesis, we always assume that the electrons in a molecule move non-relativistically, that the potentials that bind the electrons are time-independent, that positions of the atomic nuclei are fixed, and that the temperature is absolute zero. In addition to the aforementioned physical assumptions, we make one convenient mathematical assumption, namely that the electronic Hamiltonian and its many-electron wavefunction can be accurately modelled using a one-electron basis set. That is, we consider only the second-quantized time-independent electronic Schrödinger equation in the Born-Oppenheimer (clamped nuclei) approximation,

$$\hat{H} |\Psi\rangle = E |\Psi\rangle \tag{1.1}$$

This is a Hermitian eigenproblem, but practical computational methods need to exploit special features that are specific to the electrons in molecules.

Relativistic effects are important for modelling systems with heavier elements, including Lanthanides, Actinides, and second- and third-row transition metals. The correct treatment of relativistic effects requires replacing the Schrödinger equation with the Dirac equation[11], but various approximations, which are often adequate for chemical applications, allow the Schrödinger equation to be used even for relativistic systems[12].

The most important ways to incorporate relativistic effects into the Schrödinger equation include pseudopotentials[13], relativistic corrections to the kinetic energy operators like the zero-order regular approximation (ZORA)[14, 15], and two-component approaches based on the Foldy-Wouthuysen transformation of the Dirac equation[16]. The methods in this thesis are directly applicable to these approximate relativistic Schrödinger equations and could be extended to treat the full four-component Dirac equation.

When the electrons in a molecule experience time-dependent forces, the time-dependent Schrödinger equation needs to be solved. This is important for computing molecular response properties and modelling spectroscopy[17]. Many time-dependent effects can be modelled by solving the time-independent Schrödinger equation and then using time-dependent perturbation theory[18]. For computing thermodynamic properties of molecules, including temperature dependence is important. Temperature-dependent properties can be modelled explicitly by Boltzmann-weighting ground- and excited-state properties obtained by solving the time-independent Schrödinger equation[2], or by solving an effective time-dependent Schrödinger equation with doubled dimensionality in the thermofield dynamics approach[19]. Though we will not consider it, the methods in this thesis can be extended to include time-dependent and temperature-dependent phenomena.

We solve the time-independent electronic Schrödinger equation, thereby neglecting nuclear quantum effects except insofar as they can be encoded with the adiabatic approximation. Neglecting the nuclear kinetic-energy operator is justified since the nuclei are much more massive than the electrons, but this approximation can be remedied by solving the nuclear Schrödinger equation or modern alternative approaches[20]. The most important nuclear quantum effects (e.g., zero-point energies and tunnelling corrections) can often be mimicked by post-facto corrections on top of solutions to the electronic Schrödinger equation[10, 21, 22]. Most of the methods presented in this thesis are designed for the electronic structure problem, but they could be adapted to include nuclear quantum effects.

When solving the time-independent zero-temperature electronic Schrödinger equation numerically, it is necessary to discretize the problem. We do this by selecting a finite one-electron basis set, in which we expand the Hamiltonian operator and its associated eigenfunctions. The accuracy of results can be systematically improved by increasing the size of the basis set, though the computational cost also grows with basis-set size[8, 9, 23, 24]. Throughout this thesis, an orthonormal basis set consisting of molecular orbitals obtained from the (approximate) Hartree-Fock solution to the electronic Schrödinger equation will be used. For this reason, a one-electron basis function will often be referred to as a spin-orbital or orbital, in short. Unlike the preceding assumptions, the assumption of a finite (but perhaps very large) one-electron basis set is essential to the methods proposed in this thesis.

1.2 Slater Determinant

According to the Pauli principle, the wavefunction is antisymmetric with respect to the interchange of any two fermions and, conversely, symmetric with respect to the interchange of any two bosons. Electrons are fermions, so the simplest possible N-electron wavefunction, $\Phi(\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_N)$, is an antisymmetrized product of N distinct one-electron basis functions (spin-orbitals), $\{\chi_{m_1}, \ldots, \chi_{m_N}\}$. This wavefunction is called a Slater determinant[23, 24]:

$$\begin{aligned} |\Phi(\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_N)\rangle &= \frac{1}{N!} \begin{vmatrix} \chi_{m_1}(\mathbf{r}_1) & \chi_{m_1}(\mathbf{r}_2) & \cdots & \chi_{m_1}\mathbf{r}_N) \\ \chi_{m_2}(\mathbf{r}_1) & \chi_{m_2}(\mathbf{r}_2) & \cdots & \chi_{m_2}(\mathbf{r}_N) \\ \vdots & \vdots & \ddots & \vdots \\ \chi_{m_N}(\mathbf{r}_1) & \chi_{m_N}(\mathbf{r}_2) & \cdots & \chi_{m_N}(\mathbf{r}_N) \end{vmatrix}^{-} \\ &= \sum_{\sigma \in S_N} \operatorname{sgn}(\sigma) \prod_{i=1} \chi_i(\mathbf{r}_{\sigma(i)}) \end{aligned} \tag{1.2}$$

Hereon, we will denote a Slater determinant with Φ or, equivalently, with the occupation vector, $\mathbf{m}$, listing the indices of the occupied spin-orbitals in the Slater determinant.

The Slater determinant is the mathematical representation for an electron configuration: each orbital is considered occupied if it is used to build the Slater determinant. In accordance with the Pauli exclusion principle, each spin-orbital can contain up to at most one electron and each spatial orbital can contain up to two electrons, with opposing spins. A spatial orbital is a set of two spin-orbitals that have the same spatial component but different spin. By construction, the Slater determinant satisfies the Pauli exclusion principle because a determinant with identical rows (doubly-occupied spin-orbitals) or columns (two electrons with the same spin at the same location) is zero[23]. The formulas and derivation that follow are expressed with respect to spin-orbitals for clarity and generality.

It is important that the one-electron basis functions are orthonormal to one another, i.e.,

$$\int \chi_i(\mathbf{r})\chi_j(\mathbf{r})d\mathbf{r} = \delta_{ij} \tag{1.3}$$

Otherwise, the Slater determinants and the related integrals become too expensive (but not impossible) to compute. Nonorthogonal orbitals are the hallmark of valence-bond approaches to the electronic structure theory problem[25].

In most single-determinant methods, the goal is to find the set of orthonormal orbitals such that the corresponding Slater determinant(s) optimally solve the Schrödinger equation, albeit perhaps with a modified Hamiltonian. In both Hartree-Fock (HF; where the true Hamiltonian is used)[23] and Kohn-Sham density-functional theory (KS-DFT; where a noninteracting model system is considered)[26], the orthogonal spin-orbitals are transformed with a unitary matrix, and the unitary matrix is chosen to minimize the energy. Because these methods provide an effective one-electron Hamiltonian, they produce one orbital for every function in the one-electron basis set, which is usually far larger than the number of electrons. Orbitals that are not occupied in the lowest-energy Slater determinant are called virtual orbitals. Excited states can be modelled by a Slater determinant where orbitals that were occupied in the ground-state Slater determinant are replaced by virtual orbitals. Obviously, the number of Slater determinants (and excitations) increases combinatorially with the size of the basis set and the number of electrons. If there are $2K$ spin-orbitals and N electrons, then there are $\binom{2K}{N}$ possible Slater determinants. The focus of this thesis is on multideterminant wavefunctions: linear combinations of multiple Slater determinants which are used to model systems that cannot be described with the requisite accuracy using only a single configuration[23].

Though the methods we present are applicable to any orthonormal one-electron basis set, we will often assume that the basis sets are composed of the molecular orbitals obtained from HF calculations.

1.3 Second Quantization

Especially when discussing multideterminant wavefunctions, it is helpful to use bra-ket notation and second quantization. The bra-ket notation is a shorthand for writing the wavefunction and the associated integrals that makes the vector-space nature of the Hilbert space of N-electron wavefunctions explicit[24, 26]:

$$\begin{aligned}
\langle\Psi| &= \Psi^*(\mathbf{r}_1, \dots, \mathbf{r}_N)\\
|\Psi\rangle &= \Psi(\mathbf{r}_1, \dots, \mathbf{r}_N)\\
\langle\Phi|\Psi\rangle &= \int\cdots\int \Phi^*(\mathbf{r}_1, \dots, \mathbf{r}_N)\Psi(\mathbf{r}_1, \dots, \mathbf{r}_N)d\mathbf{r}_1 \dots d\mathbf{r}_N\\
\langle\Phi|\hat{H}|\Psi\rangle &= \int\cdots\int \Phi^*(\mathbf{r}_1, \dots, \mathbf{r}_N)\hat{H}\Psi(\mathbf{r}_1, \dots, \mathbf{r}_N)d\mathbf{r}_1 \dots d\mathbf{r}_N
\end{aligned} \tag{1.4}$$

Note that the positions of the electrons are generally omitted in bra-ket notation and that the bra is a complex conjugate of the ket and vice versa. The vacuum state is a quantum state without any particles and is described with an empty bra, $\langle|$, and ket, $|\rangle$. By convention, $\langle|\rangle = 1$. Occasionally it is useful to use a pseudo-vacuum, but this does not affect the notation, only the computational implementation of the resulting equations.

In this thesis, we use the following convention for the Slater determinant:

$$\begin{aligned}
|\Phi(\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N)\rangle &= \begin{vmatrix}
\chi_{m_1}(\mathbf{r}_1) & \chi_{m_1}(\mathbf{r}_2) & \dots & \chi_{m_1}(\mathbf{r}_N) \\
\chi_{m_2}(\mathbf{r}_1) & \chi_{m_2}(\mathbf{r}_2) & \dots & \chi_{m_2}(\mathbf{r}_N) \\
\vdots & \vdots & \ddots & \vdots \\
\chi_{m_{N-1}}(\mathbf{r}_1) & \chi_{m_{N-1}}(\mathbf{r}_2) & \dots & \chi_{m_{N-1}}(\mathbf{r}_N) \\
\chi_{m_N}(\mathbf{r}_1) & \chi_{m_N}(\mathbf{r}_2) & \dots & \chi_{m_N}(\mathbf{r}_N)
\end{vmatrix}^{-} \\
&= |m_1 m_2 \dots m_{N-1} m_N\rangle \\
&= |\mathbf{m}\rangle
\end{aligned} \tag{1.5}$$

where $\mathbf{m} = (m_1 m_2 \dots m_{N-1} m_N)$ is the occupation vector. In second quantization, the occupied spin-orbitals are added to the Slater determinant with a creation operator, $a_i^\dagger$, and removed with an annihilation operator, a_i. For example,

$$\begin{aligned}
a_i^\dagger |\rangle &= |i\rangle \\
a_i |i\rangle &= |\rangle
\end{aligned} \tag{1.6}$$

When the occupation vector contains many indices, the creation operators are created in the opposite order of the occupation vector, i.e. the right-most index in the occupation vector is created first and the left-most index is created last.

$$|m_1 m_2 \dots m_{N-1} m_N\rangle = a_{m_1}^\dagger a_{m_2}^\dagger \dots a_{m_{N-1}}^\dagger a_{m_N}^\dagger |\rangle \tag{1.7}$$

The important characteristics of fermionic wavefunctions, specifically the Pauli exclusion principle and the antisymmetry property, are ensured by defining the behaviour of the second-quantized operators. The Pauli exclusion principle is enforced by ensuring that a spin-orbital cannot be created twice, i.e. creation operator applied to itself is zero:

$$a_i^\dagger a_i^\dagger = 0 \tag{1.8}$$

Likewise, a spin-orbital cannot be removed twice:

$$a_i a_i = 0 \tag{1.9}$$

In fact, annihilating any spin-orbital that is not occupied in the Slater determinant will result in a zero:

$$a_{i \notin \{m_1, m_2, \ldots, m_N\}} \left| m_1 m_2 \ldots m_N \right\rangle = 0 \tag{1.10}$$

Note that Equation 1.10 is valid for any wavefunction in which spin-orbital i is unoccupied.

In second quantization, antisymmetry is enforced by the anticommutation relations (Equation 1.11).

$$\begin{aligned} a_i^\dagger a_j^\dagger + a_i^\dagger a_j^\dagger &= 0 \\ a_i a_j + a_i a_j &= 0 \end{aligned} \tag{1.11}$$

The final anticommutation relation ensures that every spin-orbital is either occupied or empty,

$$a_i^\dagger a_j + a_j a_i^\dagger = \delta_{ij} \tag{1.12}$$

The following example illustrates the off-diagonal and diagonal implications of this anticommutation relation by applying $a_2^\dagger a_1$ and $a_1^\dagger a_1$ to the Slater determinant $|1\rangle$.

$$\begin{aligned}
a_2^\dagger a_1 |1\rangle &= (\delta_{12} - a_1 a_2^\dagger) |1\rangle \\
&= 0 - a_1 a_2^\dagger a_1^\dagger |\rangle \\
&= a_1 a_1^\dagger a_2^\dagger |\rangle \\
&= |2\rangle \\
a_1^\dagger a_1 |1\rangle &= (\delta_{11} - a_1 a_1^\dagger) |1\rangle \\
&= |1\rangle - 0 \\
&= |1\rangle
\end{aligned} \tag{1.13}$$

Finally, the conjugate transpose of the creation operator is the annihilation operator, and vice versa (Equation 1.14).

$$\begin{aligned}
(a_i^\dagger)^\dagger &= a_i \\
(a_i)^\dagger &= a_i^\dagger
\end{aligned} \tag{1.14}$$

Moreover,

$$\begin{aligned}
(a_i^\dagger a_j^\dagger)^\dagger &= a_j a_i \\
(a_i a_j)^\dagger &= a_j^\dagger a_i^\dagger
\end{aligned} \tag{1.15}$$

1.4 Multideterminant Wavefunctions

Because the set of all N-electron Slater determinants is a complete basis for the N-electron wavefunction, all N-electron wavefunctions can be represented exactly (within the strictures of the presupposed one-electron basis set) as a linear combination of Slater determinants. Often the exact wavefunction for a chemical system has a single dominant determinant; such systems are called single-reference[27]. Closed-shell organic molecules near their equilibrium geometry are often single reference. However, for chemical systems that have many valence orbitals that are nearly degenerate in energy, like transition-metal complexes, the ground-state electron configuration is ambiguous and a single electron configuration cannot sufficiently describe the system, often leading to catastrophically wrong results. Building a N-electron wavefunction for this type of multiconfigurational molecule requires multiple Slater determinants.

The simplest multideterminant wavefunction is the Configuration Interaction (CI) wavefunction[23, 24]. It is a simple linear combination of Slater determinants,

$$|\Psi\rangle = \sum_{\mathbf{m} \in S} c_{\mathbf{m}} |\mathbf{m}\rangle \tag{1.16}$$

where S is the set of the Slater determinants used and $c_{\mathbf{m}}$ is the coefficient of the Slater determinant $\mathbf{m}$. Since every Slater determinant satisfies the Pauli principle, so do CI wavefunctions. The cost and the accuracy of a CI wavefunction are controlled by the number of Slater determinants: the accuracy and cost increase as more Slater determinants are added.

CI methods generally supersede methods based on a single Slater determinant. In fact, an optimized Full CI (FCI) wavefunction, where all $\binom{2K}{N}$ determinants are included in Equation 1.16, is the exact solution to the Schrödinger equation within a given one-electron basis set. The primary task of molecular electronic structure theory, then, is to find computationally feasible approximations to the FCI method. To avoid the prohibitive cost of FCI, Slater determinants are often systematically selected according to the orders of excitation from the ground-state configuration or are hand-selected based on chemical intuition or a mathematical estimate of their likely importance[28–38]. Unfortunately, as the system gets larger, the number of Slater determinants required for a given accuracy grows quickly and accurate CI calculations become infeasible. In particular, there are a large number of Slater determinants that have relatively small coefficients, but which are nonetheless essential, in aggregate, for obtaining quantitative accuracy. Including (or at least approximating) these minor contributions is an important target for research in quantum chemistry.

A particular problem of CI methods is that, except for full-CI, they are not *size consistent*[39]. That is, the wavefunction of two noninteracting subsystems will not be the antisymmetrized product of the subsystems' wavefunctions, which is a particular problem when using CI methods to describe molecular dissociation or extended systems. Size consistency can be restored and contributions from small-coefficient determinants can be included by parameterizing the CI coefficients with a function, $f(\mathbf{m})$[40]:

$$|\Psi\rangle = \sum_{\mathbf{m}\in S} f(\mathbf{m}) |\mathbf{m}\rangle \tag{1.17}$$

This function is the overlap of the wavefunction with the Slater determinant, $f(\mathbf{m}) =$

$\langle \mathbf{m} | \Psi \rangle$. By carefully selecting f, one can ensure size-consistency, along with other desirable mathematical and chemical properties. We'll discuss this approach further in Chapter 2.

1.5 Hamiltonian

The FCI method is exact for any N-fermion system. Different systems are distinguished, then, not by the mathematical structure of their wavefunction but by their Hamiltonians. As mentioned previously, in this thesis we are concerned only with the non-relativistic electronic Hamiltonian of molecular systems[23] which, in atomic units, is given by

$$\begin{aligned}\hat{H} &= -\frac{1}{2}\sum_i \nabla_i^2 - \sum_i \sum_A \frac{Z_A}{r_{iA}} + \sum_{i<j} \frac{1}{r_{ij}} \\ &= \sum_i \hat{h}_i + \sum_{i<j} \hat{g}_{ij}\end{aligned} \tag{1.18}$$

Here ∇_i is the gradient with respect to the position of electron i, Z_A is the charge of the atomic nucleus A, r_{iA} is the distance between electron i and nucleus A, and r_{ij} is the distance between electrons i and j. For each electron i, the operators for the kinetic energy, $-\frac{1}{2}\nabla_i^2$, and the nuclear electron attraction, $-\sum_A \frac{Z_A}{r_{iA}}$, are grouped together to form the one-electron operator, $\hat{h}_i$. Likewise, the electron-electron repulsion operator, $\frac{1}{r_{ij}}$, is abbreviated as the two-electron operator, $\hat{g}_{ij}$.

Since the wavefunction is a function of the positions of the electrons, the electron indices, i and j, are needed to keep track of the distinct electronic positions even though all electrons are equivalent particles. As before, it is helpful to project the Hamiltonian onto the orbital basis set and reexpress it with second quantization, so that the electronic positions need not be tracked explicitly. The matrix representations of the one- and two-electron operators in the orbital basis set are denoted as one-electron integrals, h_{ij}, and two-electron integrals, g_{ijkl}, respectively. Explicitly,

$$\begin{aligned} h_{ij} &= \int \phi_i^*(\mathbf{r}_1)\hat{h}\phi_j(\mathbf{r}_1)d\mathbf{r}_1 \\ &= \int \phi_i^*(\mathbf{r}_1)\left(-\frac{1}{2}\nabla^2 - \sum_A \frac{Z_A}{|\mathbf{r}_1 - \mathbf{r}_A|}\right)\phi_j(\mathbf{r}_1)d\mathbf{r}_1 \end{aligned} \tag{1.19}$$

$$\begin{aligned} g_{ijkl} &= \int \phi_i^*(\mathbf{r}_1)\phi_j^*(\mathbf{r}_2)\hat{g}\phi_k(\mathbf{r}_1)\phi_l(\mathbf{r}_2)d\mathbf{r}_1 d\mathbf{r}_2 \\ &= \int \phi_i^*(\mathbf{r}_1)\phi_j^*(\mathbf{r}_2)\frac{1}{|\mathbf{r}_1 - \mathbf{r}_2|}\phi_k(\mathbf{r}_1)\phi_l(\mathbf{r}_2)d\mathbf{r}_1 d\mathbf{r}_2 \end{aligned} \tag{1.20}$$

Note that in the literature two-electron integrals are sometimes denoted using physicists' notation (Equation 1.20) and sometimes using chemists' notation (Equation 1.21).

$$g_{ijkl} = \int \phi_i^*(\mathbf{r}_1)\phi_k^*(\mathbf{r}_2)\hat{g}\phi_j(\mathbf{r}_1)\phi_l(\mathbf{r}_2)d\mathbf{r}_1 d\mathbf{r}_2 \tag{1.21}$$

Throughout this work, physicists' notation will be used. Using the one- and two-electron integrals, the second-quantized Hamiltonian[24] is:

$$\hat{H} = \sum_{ij} h_{ij} a_i^\dagger a_j + \frac{1}{2}\sum_{ijkl} g_{ijkl} a_i^\dagger a_j^\dagger a_l a_k \tag{1.22}$$

Note that the second-quantized Hamiltonian is valid for any number of fermions, interacting by any type of 1-body and 2-body forces. Thus Equation 1.22, and the numerical methods used to solve it, are applicable not only to molecules but to any other system of interacting elementary fermionic particles.

The Hamiltonian is sometimes modified so that solving the Schrödinger equation is simpler. For example, the Fock operator is an approximate Hamiltonian where the two-body operator is subsumed into effective one-body terms[24]. Different single-determinant methods are associated with different Fock operators, which are in turn associated with different ways of "averaging" the two-body terms into effective one-body interactions. For a given Fock operator, the N-fermion Schrödinger equation is separable, and the occupied orbitals are obtained by simply diagonalizing the Fock operator. However, because the average two-body interaction depends on the choice of occupied orbitals, and the choice of occupied orbitals depends on the effective one-body interaction, computational methods based on Fock operators are solved iteratively, where the Fock operator is diagonalized to find the occupied orbitals and the effective one-body interaction is accordingly updated until the occupied orbitals and the effective one-body interaction are self-consistent. This self-consistent-field (SCF) approach[23] is much cheaper and easier than solving the molecular Hamiltonian. However, SCF approaches discard important details about the two-electron interactions and always have a single-determinant ground-state wavefunction, which is qualitatively incorrect for strongly correlated systems like metal complexes and transition states. Instead of discarding the two-body operator altogether, one can try to simplify it so that the Schrödinger equation is easier to solve. This is done, for example, in the Hubbard model, where the Hamiltonian is simplified so that only repulsions between electrons in the same spatial orbital are included[41]. Though the Hubbard model is often used to model strongly-correlated systems and to study magnetism and superconductivity, it is limited to lattice-like systems and is rarely adequate for making quantitative predictions about real chemical/physical systems[42, 43].

1.6 Methods for Determining Electronic Wavefunctions

Once the Hamiltonian is constructed and the wavefunction model is selected, the Schrödinger equation can be solved. The traditional form of the Schrödinger equation (Equation 1.1) is rarely useful because it involves solving a many-body nonseparable partial differential equation, which is intractable except for one-particle systems and a few, unphysical, N-particle model Hamiltonians[23, 24]. Once a finite one-electron basis set is selected, solving the Schrödinger equation becomes equivalent to a $\binom{2K}{N}$-dimensional ordinary eigenvalue problem. This is still intractable for almost all systems of chemical interest. This motivates the development of alternative approaches for determining the wavefunction.

Multiplying the Schrödinger equation on the left by the complex conjugate of the wavefunction and integrating gives an explicit expression for the expectation value of the energy for the wavefunction in question:

$$\begin{aligned} \langle\Psi|\hat{H}|\Psi\rangle &= E\,\langle\Psi|\Psi\rangle \\ E &= \frac{\langle\Psi|\hat{H}|\Psi\rangle}{\langle\Psi|\Psi\rangle} \end{aligned} \tag{1.23}$$

It is easy to show that Equation 1.23 is always greater than or equal to the true energy, and equal to the true energy only when the wavefunction model gives the exact ground state[10]. This means that the parameters in the wavefunction model can be determined by minimizing the energy expression. This strategy works well for selected CI methods, but for parameterized CI methods, Equation 1.17, the energy is typically a non-convex function of the wavefunction parameters, and many local minima exist. Global optimization algorithms are usually intractable except for systems so small that FCI is tractable. Nonetheless, local minima of the energy functional (Equation 1.23) often provide reasonable approximate energies, especially when good initial guesses are available.

Variational methods are only feasible when the normalization of the wavefunction $\langle\Psi|\Psi\rangle$ and the associated matrix elements of the 2-body operator $\langle\Psi|a_i^\dagger a_j^\dagger a_l a_k|\Psi\rangle$ can be computed efficiently. If one denotes the cost of evaluating matrix elements like these as C_{wfn}, then the cost of evaluating the energy expectation value scales as $\mathcal{O}(C_{\mathrm{wfn}}K^4)$, where K is the size of the orbital basis. This is the case for single-determinant wavefunctions like Hartree-Fock and for special multideterminant wavefunctions like Matrix Product States (MPS)[44].

Even when the wavefunction cannot be integrated efficiently with itself, it is sometimes efficient to compute the coefficients of the expansion of the wavefunction in Slater determinants $\langle\mathbf{m}|\Psi\rangle$. In this case it is useful to reformulate the energy expression (Equation 1.23) to include a projection operator $\mathcal{P} = \sum_{\mathbf{m}} |\mathbf{m}\rangle \langle\mathbf{m}|$, where the sum is over all

M determinants that have nonzero coefficients in the model wavefunction $|\Psi\rangle$:

$$\begin{aligned} E &= \frac{\langle \Psi | \mathcal{P} \hat{H} | \Psi \rangle}{\langle \Psi | \mathcal{P} | \Psi \rangle} \\ &= \frac{\sum_{\mathbf{m}} \langle \Psi | \mathbf{m} \rangle \langle \mathbf{m} | \hat{H} | \Psi \rangle}{\sum_{\mathbf{n}} \langle \Psi | \mathbf{n} \rangle \langle \mathbf{n} | \Psi \rangle} \\ &= \frac{\sum_{\mathbf{m}} \langle \Psi | \mathbf{m} \rangle \langle \hat{H} \mathbf{m} | \Psi \rangle}{\sum_{\mathbf{n}} |\langle \Psi | \mathbf{n} \rangle|^2} \end{aligned} \tag{1.24}$$

In the last line we explicitly indicate that the Hamiltonian is applied to the Slater determinants, rather than to the complicated multideterminant wavefunction. The cost of evaluating Equation 1.24 scales as $\mathcal{O}(C_{\text{sd}} M K^4)$. This is preferred to the direct evaluation of the variational energy expression when $\mathcal{O}(C_{\text{sd}} M) < \mathcal{O}(C_{\text{wfn}})$.

Unfortunately, as we mentioned earlier, accurate multideterminant wavefunctions always include an enormous number of Slater determinants, most of which are individually relatively unimportant, but collectively critical for quantitative accuracy. Equation 1.24 is thus prohibitively expensive except for very small systems. This suggests that only a subset of the determinants with nonzero coefficients should be included in the projector, even though truncating the projector means that the energy expectation value from Equation 1.24 is no longer guaranteed to be an upper bound to the true energy.

One approach to restricting the projection space is orbital space variational Quantum Monte Carlo[45–47], which controls the cost by limiting the number of Slater determinants at each optimization step, but potentially allows all Slater determinants to contribute throughout the entire optimization. The energy

$$\begin{aligned} E &= \frac{\sum_{\mathbf{m}} \langle \Psi | \mathbf{m} \rangle \langle \mathbf{m} | \hat{H} | \Psi \rangle}{\sum_{\mathbf{n}} \left| \langle \Psi | \mathbf{n} \rangle \right|^2} \\ &= \sum_{\mathbf{m}} \frac{\left| \langle \Psi | \mathbf{m} \rangle \right|^2}{\sum_{\mathbf{n}} \left| \langle \Psi | \mathbf{n} \rangle \right|^2} \frac{\langle \mathbf{m} | \hat{H} | \Psi \rangle}{\langle \mathbf{m} | \Psi \rangle} \\ &= \mathbb{E}_{\mathbf{m} \sim p(\mathbf{m})} \left[\frac{\langle \mathbf{m} | \hat{H} | \Psi \rangle}{\langle \mathbf{m} | \Psi \rangle} \right] \\ &\approx \frac{1}{M} \sum_{\mathbf{m}} \frac{\langle \mathbf{m} | \hat{H} | \Psi \rangle}{\langle \mathbf{m} | \Psi \rangle} \end{aligned} \tag{1.25}$$

is computed by sampling the Slater determinants according to the following probability distribution in Equation 1.26.

$$p(\mathbf{m}) = \frac{\left| \langle \Psi | \mathbf{m} \rangle \right|^2}{\sum_{\mathbf{n}} \left| \langle \Psi | \mathbf{n} \rangle \right|^2} \tag{1.26}$$

The last step in Equation 1.25 assumes that the Slater determinants are sampled independently from identical distributions (i.i.d.), $p(\mathbf{m})$. If enough Slater determinants are sampled, i.e. i.i.d. approximation is valid, the energy is (approximately) variational. In orbital-space variational Quantum Monte Carlo, therefore, the variational principle is still valid, within an error that can be made arbitrarily small by sufficient sampling.

In Equation 1.24, the wavefunction is projected onto the Slater determinants that compose a projection operator $\mathcal{P}$. Similarly, the energy expression in orbital space variational quantum Monte Carlo uses an adaptive projection of the wavefunction onto a set of Slater determinants. These strategies may seem distant from the original Schrödinger eigenproblem, Equation 1.1, which did not involve any integration. To establish the link with the eigenproblem, notice that the solutions to the Schrödinger equation will also satisfy the equation after integrating on the left with an arbitrary function. In other words, if

$$\hat{H} |\Psi\rangle = E |\Psi\rangle \tag{1.27}$$

then

$$\langle\Phi|\hat{H}|\Psi\rangle = E \langle\Psi|\Phi\rangle \ \forall \ \Phi \tag{1.28}$$

Equation 1.28 is called the weak form of the eigenproblem, or the projected Schrödinger equation.

In the projected Schrödinger equation, the Schrödinger equation is integrated with a set of functions, termed the projection space, to form a system of nonlinear equations[48–50]:

$$\begin{aligned} \langle\Phi_1|\hat{H}|\Psi\rangle - E \langle\Phi_1|\Psi\rangle &= 0 \\ &\vdots \\ \langle\Phi_M|\hat{H}|\Psi\rangle - E \langle\Phi_M|\Psi\rangle &= 0 \end{aligned} \tag{1.29}$$

The equations are rearranged to equal zero so that they are more compatible with software for solving nonlinear equations. To avoid the trivial solution, $\Psi = 0$, a normalization constraint is often added as an extra equation. If all of the equations are satisfied, the wavefunction and energy are the solutions to the Schrödinger equation within the selected projection space. When the projection space is complete, i.e. it encompasses all possible Slater determinants, the solutions are exact within the basis set approximation. If the formulation of the wavefunction cannot produce a solution for the given Hamiltonian, the cost function for the system of nonlinear equations solver (typically a residual sum of squares) provides a measure of the error in the optimized wavefunction and energy. The least-squares solution for a complete projection space is an upper bound to the energy, and this energetic upper bound, minus the residual sum of squares from the equations, is a lower bound to the energy[51–53]. As before, using a complete projection space is impractical, but the projection space can be adaptively selected. Adaptively sampling of the projection space is explored in Chapter 5.

1.7 Optimizing Wavefunction Parameters

Once the Schrödinger equation is reformulated in a tractable form, iterative optimization algorithms are applied to find the (approximate) solution. There are many standard optimization algorithms that are appropriate for the equations in Section 1.6.

For the energy equations (Equations 1.23, 1.24, and 1.25), optimization algorithms for scalar functions can be used. To be efficient, some type of gradient information is necessary, as the gradient indicates how to change the wavefunction parameters to reduce the expectation value for the energy[54]. Stochastic algorithms model the potential energy surface with a probability distribution by sampling the energies at different parameters[55–57]. Typically, the stochastic algorithms are less likely to get stuck in (undesired) local minima, so they are often used for the initial stages of the optimization. Gradient-based algorithms are more efficient near local minima, where they exhibit superlinear convergence, so they are used at the end of the optimization.

Scalar optimization algorithms can be directly applied to optimize the system of nonlinear equations associated with the projected Schrödinger equation by using the residual-sum-of-squares in the nonlinear equations as an objective function. However, it is often more efficient to solve the nonlinear equations directly, using the Jacobian of the nonlinear system[54, 58].

In both variational and projected formulations, black-box algorithms have severe limitations when the optimization problem becomes sufficiently complex. Within quantum chemistry, the problem of solving the Schödinger equation is almost certainly nonlinear and, in certain situations, not even smooth. In addition, as the system gets larger, the number of parameters and Slater determinants increase, resulting in a more complex and expensive optimization process. Larger systems have larger energies, yet the energy range for chemical accuracy (approx. 1 kcal/mol) does not change with the size of the system. This means that the optimization algorithm must find a more accurate solution (relatively) as the system gets larger - the haystack grows, but the needle stays the same size.

Often, approximations that make the Schrödinger equation cheaper to solve without a significant loss in accuracy or generalizability increase the complexity of the optimization process. For example, parameterizing the coefficients of the Slater determinants (Equation 1.17) results in highly nonlinear equations. In addition, truncating the projection space may result in uneven contribution of the coefficients to the objective function. In the extreme case, if the energy (Equation 1.24) is computed by projecting onto Slater determinants that have negligible CI coefficients, the nonlinear system will be extremely ill-conditioned.

Though the exact impact of each approximation on the high-dimensional solution space is unclear, it is clear that (1) the optimization problems presented in quantum chemistry are diverse and complicated, (2) the methods must have sufficient sophistication to accurately and tractably model a wide range of systems, and (3) the wavefunctions with the highest complexity are most difficult to optimize. The difficulty in optimization may be presented as an abundance of local minima (and saddle points), discontinuity in the second derivative, or flat regions. Many of the black-box optimization algorithms will have trouble finding a good solution without preconditioning or tinkering with the optimization hyperparameters. In most cases, a good solution is guaranteed to exist, but it may be very difficult to find computationally.

Therefore, it is absolutely necessary to build a specialized optimization algorithm if the method is to be cheap and accurate for a wide range of systems. This algorithm may be specific to the approximations used in solving the Schrödinger equation (e.g., the formulations of the wavefunction, Hamiltonian, and objective) and to the systems under study. For example, the Density Matrix Renormalization Group (DMRG)[44] algorithm minimizes the energy variationally for the MPS wavefunction but is specialized for linear systems (though it can be used for other systems). In contrast, the quantum Monte Carlo algorithm[46] is more general and can be used with different wavefunctions and systems but may be more difficult to use. These methods are two of the more popular methods for benchmark calculations, providing some of the most accurate results of moderately large systems ($<$ 100 electrons). Of course, a specialized algorithm is not necessary if the problem is made simpler, by simplifying the Hamiltonian, for example, or if the method is specialized for certain systems, by building functionals in DFT[59], for example, but these methods sacrifice accuracy and reliability. A method that is reliably accurate for a wide range of systems at a reasonable cost will likely require a specialized algorithm without which many non-optimal solutions will be found, making the method neither accurate nor reliable.

In addition to specialized algorithms, a good initial guess is critical. A poor initial guess, in contrast, may result in solutions that correspond to the excited states, or worse, to unphysical local minima that arise from the mathematical formulation, rather than the system's behavior. To achieve a good initial guess, it is often best to start by optimizing a simpler, less accurate method, then use this approximate wavefunction to provide initial guesses for the parameters in a more complex wavefunction form. For example, the initial guess to a CI wavefunction can be obtained by first optimizing a CI wavefunction with a subset of the Slater determinants, and initial guesses for Coupled Cluster (CC) and geminal wavefunctions are often obtained using perturbation theory around the Hartree-Fock limit[60]. In some cases, the complexity of the method is controlled with a hyperparameter whose value controls both the cost and accuracy of the method. MPS wavefunctions use the size of the matrices, D, to control its accuracy and cost: increasing D improves accuracy at a larger cost[44]. Results from small D provide good guesses for calculations at a larger D. Along these lines, in Chapter 4, we introduce a threshold that is used to limit the number of pairing schemes in the Antisymmeterized Product of Geminal (APG) wavefunctions. These hyperparameters provides greater control over the methods and, therefore, the optimization process.

1.8 Overview of the Thesis

In *ab initio* method development, the goal is to build methods that are (relatively) cheap and accurate for a wide range of systems, especially for systems where single-determinant methods are insufficient. Unfortunately, the systems for which single-determinant methods perform poorly are often (1) large, which means that the method must have cost and accuracy that scale well with size, and (2) strongly correlated, which means that many states are nearly degenerate, which makes the correct solution more difficult to obtain. Furthermore, using a tractable method with enough complexity to be accurate may cause the final equation to be even harder to optimize.

Above, we addressed the four different components involved within an orbital-based *ab initio* method: the wavefunction model, the Hamiltonian, the objective function, and the optimization algorithm. A new method or a modification to an existing method often focuses on at least one of these components, usually improving on its cost and/or accuracy. However, modifying one of these components can affect the other components in ways that are often difficult to predict. In particular, relatively small changes to a wavefunction model can greatly enhance the difficulty of optimizing the objective function. Often the objective function and the optimization algorithm must be tweaked based on experience, which unfortunately restricts the usage of some methods to a select few.

We do not provide direct solutions to these problems in this thesis. Instead, we develop a tool to help those that wish to explore these problems further. In particular, in Chapter 2, we propose a general framework, called FANCI, from which any multideterminant wavefunction can be built and characterized. FANCI consolidates the formulations of different wavefunction ansatz into a single framework such that a wavefunction can easily be extended upon and a new wavefunction can be easily derived from existing ansätze. We believe that subsuming all multideterminant methods in a single framework will not only inspire new works but allow researchers to systematically explore the best objective functions and optimization strategies. To demonstrate the power and flexibility of this approach, we propose several new wavefunction structures motivated by existing wavefunction ansätze.

In Chapter 3, we present an open-source Python library called `Fanpy` as a platform for developing new methods in *ab inito* electronic structure theory based on the FANCI framework. Mirroring the organization of this introduction, the `Fanpy` library is divided into four independent modules: the wavefunction model, the Hamiltonian of the system, the objective function, and the optimization algorithm. `Fanpy`'s modular design, extensive documentation, and user-friendly templates and scripts allow researchers to quickly prototype new methods without an extensive understanding of the library. This library is used extensively to implement and test the ideas presented in the remainder of the thesis.

In Chapter 4, we propose an algorithm that improves the performance of the Antisymmeterized Product of Geminals (APG) wavefunction (and other geminal wavefunctions). Within the FANCI framework, the cost and accuracy of the APG wavefunction can be controlled by hyperparameters that modify the wavefunction. These hyperparameters limit the number of terms that contribute to the wavefunction so that the cost can be lowered at the expense of the accuracy. They are used by an algorithm to dynamically alter the wavefunction throughout the optimization process. Improving the algorithm may provide key insights for utilizing complex quasiparticle wavefunctions.

In Chapter 5, we look for insights into solving the Schrödinger equation from recent advances in neural networks. Over the last few decades, neural networks have been proven to be effective for modelling complex problems, such as image recognition[61–64], speech recognition[65–67], and language processing[68–71]. These advances in neural network were possible, in part, due to improvements in the models and optimization algorithms, and due to the development of accessible tools and computational resources. Despite their success in machine learning problems, we find that direct implementation of these models and algorithms are ill-suited for solving the Schrödinger equation. For example, the stochastic gradient descent algorithm is a popular algorithm for optimizing neural networks but does not seem effective when applied to the projected Schrödinger equation. However, adjusting the implementation for the projected Schrödinger equation results in an objective function similar to that of orbital space variational quantum Monte Carlo. In addition, a feed-forward neural network, with some minor modifications, can effectively model the overlap of the wavefunction but has difficulty with optimization as the network becomes deeper. This chapter provides both a demonstration of the flexibility of the FANCI strategy, a cautionary tale about the naïve application of machine-learning methods, and an indication of how future ML-based FANCI methods may be designed.

1.9 References

(1) Allinger, N. L., *Molecular structure: understanding steric and electronic effects from molecular mechanics*; John Wiley & Sons: 2010.

(2) Hill, T. L., *An introduction to statistical thermodynamics*; Courier Corporation: 1986.

(3) Car, R.; Parrinello, M. *Physical review letters* **1985**, *55*, 2471.

(4) Dral, P. O. *The Journal of Physical Chemistry Letters* **2020**, *11*, 2336–2347.

(5) Ayers, P. W. *Physical Chemistry Chemical Physics* **2006**, *8*, 3387–3390.

(6) Min, K. S.; DiPasquale, A. G.; Rheingold, A. L.; White, H. S.; Miller, J. S. *Journal of the American Chemical Society* **2009**, *131*, 6229–6236.

(7) Miller, J. S.; Min, K. S. *Angewandte Chemie International Edition* **2009**, *48*, 262–272.

(8) Jensen, F., *Introduction to computational chemistry*; John wiley & sons: 2017.

(9) Cramer, C. J., *Essentials of computational chemistry: theories and models*; John Wiley & Sons: 2013.

(10) Piela, L., *Ideas of quantum chemistry*; Elsevier: 2013.

(11) Thaller, B., *The dirac equation*; Springer Science & Business Media: 2013.

(12) Pyykko, P. *Chemical Reviews* **1988**, *88*, 563–594.

(13) Hartwigsen, C.; Gœdecker, S.; Hutter, J. *Physical Review B* **1998**, *58*, 3641.

(14) Lenthe, E. v.; Baerends, E.-J.; Snijders, J. G. *The Journal of chemical physics* **1993**, *99*, 4597–4610.

(15) Van Lenthe, E.; Baerends, E.-J.; Snijders, J. G. *The Journal of chemical physics* **1994**, *101*, 9783–9792.

(16) Costella, J. P.; McKellar, B. H. *American Journal of Physics* **1995**, *63*, 1119–1121.

(17) Ishikawa, K. L.; Sato, T. *IEEE Journal of Selected Topics in Quantum Electronics* **2015**, *21*, 1–16.

(18) Langhoff, P.; Epstein, S.; Karplus, M. *Reviews of Modern Physics* **1972**, *44*, 602.

(19) Harsha, G.; Henderson, T. M.; Scuseria, G. E. *The Journal of chemical physics* **2019**, *150*, 154109.

(20) Gidopoulos, N. I.; Gross, E. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* **2014**, *372*, 20130059.

(21) Rigolin, G.; Ortiz, G.; Ponce, V. H. *Physical Review A* **2008**, *78*, 052508.

(22) Scherrer, A.; Agostini, F.; Sebastiani, D.; Gross, E.; Vuilleumier, R. *Physical Review X* **2017**, *7*, 031035.

(23) Szabo, A.; Ostlund, N., *Modern Quantum Chemistry - Introduction to Advanced Electronic Structure Theory*; McGraw-Hill Inc.: 1989, pp 43–107.

(24) Helgaker, T.; Jørgensen, P.; Olsen, J., *Modern electronic structure theory*; Wiley: Chichester, 2000.

(25) Wu, W.; Su, P.; Shaik, S.; Hiberty, P. *Chemical Reviews* **2011**, *111*, 7557–7593.

(26) Parr, R. G.; Yang, W. *Annual review of physical chemistry* **1995**, *46*, 701–728.

(27) Dreuw, A.; Head-Gordon, M. *Chemical reviews* **2005**, *105*, 4009–4037.

(28) Tubman, N. M.; Freeman, C. D.; Levine, D. S.; Hait, D.; Head-Gordon, M.; Whaley, K. B. *Journal of chemical theory and computation* **2020**, *16*, 2139–2159.

(29) Knowles, P. J.; Werner, H.-J. *Chemical physics letters* **1988**, *145*, 514–522.

(30) Giner, E.; Scemama, A.; Caffarel, M. *Canadian Journal of Chemistry* **2013**, *91*, 879–885.

(31) Evangelista, F. A. *The Journal of Chemical Physics* **2014**, *140*, 124114.

(32) Tubman, N. M.; Lee, J.; Takeshita, T. Y.; Head-Gordon, M.; Whaley, K. B. *The Journal of chemical physics* **2016**, *145*, 044112.

(33) Holmes, A. A.; Tubman, N. M.; Umrigar, C. *Journal of chemical theory and computation* **2016**, *12*, 3674–3680.

(34) Schriber, J. B.; Evangelista, F. A. Communication: An adaptive configuration interaction approach for strongly correlated electrons with tunable accuracy., 2016.

(35) Liu, W.; Hoffmann, M. R. *Journal of chemical theory and computation* **2016**, *12*, 1169–1178.

(36) Zimmerman, P. M. *The Journal of Chemical Physics* **2017**, *146*, 104102.

(37) Garniron, Y.; Scemama, A.; Loos, P.-F.; Caffarel, M. *The Journal of Chemical Physics* **2017**, *147*, 034101.

(38) Mejuto-Zaera, C.; Tubman, N. M.; Whaley, K. B. *Physical Review B* **2019**, *100*, 125165.

(39) Nooijen, M.; Shamasundar, K.; Mukherjee, D. *Molecular Physics* **2005**, *103*, 2277–2298.

(40) Kim, T. D.; Miranda-Quintana, R. A.; Richer, M.; Ayers, P. W. Flexible Ansatz for N-body Configuration Interaction., Unpublished Manuscript, 2020.

(41) Hubbard, J. *Proceedings of the Royal Society A* **1963**, *276*, 238–257.

(42) Tasaki, H. *Journal of Physics: Condensed Matter* **1998**, *10*, 4353–4378.

(43) Arrachea, L.; Zanchi, D. *Phys. Rev. B* **2005**, *71*, 064519.

(44) Chan, G.; Sharma, S. *Annu. Rev. Phys. Chem.* **2011**, *62*, 465–81.

(45) Nightingale, M. P.; Umrigar, C. J., *Quantum Monte Carlo methods in physics and chemistry*; 525; Springer Science & Business Media: 1998.

(46) Umrigar, C. *The Journal of chemical physics* **2015**, *143*, 164105.

(47) Sabzevari, I.; Sharma, S. *Journal of chemical theory and computation* **2018**, *14*, 6276–6286.

(48) Cullen, J. *Chemical Physics* **1996**, *202*, 217–229.

(49) Limacher, P. *Journal of Chemical Physics* **2016**, *145*, 194102.

(50) Johnson, P.; Limacher, P.; Kim, T.; Richer, M.; Miranda-Quintana, R.; Heidar-Zadeh, F.; Ayers, P.; Bultinck, P.; De Baerdemacker, S.; Van Neck, D. *Computational and Theoretical Chemistry* **2017**, *1116*, 207–219.

(51) Weinstein, D. *Proceedings of the National Academy of Sciences of the United States of America* **1934**, *20*, 529.

(52) Martinazzo, R.; Pollak, E. *Proceedings of the National Academy of Sciences* **2020**, *117*, 16181–16186.

(53) Kim, T. D.; Miranda-Quintana, R. A.; Ayers, P. W. On the link between projected Schrödinger equation and variational methods., Unpublished Manuscript, 2020.

(54) Nocedal, J.; Wright, S., *Numerical optimization*; Springer Science & Business Media: 2006.

(55) Weise, T. *Self-Published Thomas Weise* **2009**.

(56) Conn, A. R.; Scheinberg, K.; Vicente, L. N., *Introduction to derivative-free optimization*; SIAM: 2009.

(57) Rios, L. M.; Sahinidis, N. V. *Journal of Global Optimization* **2013**, *56*, 1247–1293.

(58) Press, W. H.; Teukolsky, S. A.; Vetterling, W. T.; Flannery, B. P., *Numerical recipes 3rd edition: The art of scientific computing*; Cambridge university press: 2007.

(59) Mardirossian, N.; Head-Gordon, M. *Molecular Physics* **2017**, *115*, 2315–2372.

(60) Bartlett, R.; Musiał, M. *Reviews of Modern Physics* **2007**, *79*, 291–352.

(61) Krizhevsky, A.; Sutskever, I.; Hinton, G. E. In *Advances in Neural Information Processing Systems 25*, Pereira, F., Burges, C. J. C., Bottou, L., Weinberger, K. Q., Eds.; Curran Associates, Inc.: 2012, pp 1097–1105.

(62) Farabet, C.; Couprie, C.; Najman, L.; LeCun, Y. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2013**, *35*, 1915–1929.

(63) Tompson, J. J.; Jain, A.; LeCun, Y.; Bregler, C. In *Advances in Neural Information Processing Systems 27*, Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., Weinberger, K. Q., Eds.; Curran Associates, Inc.: 2014, pp 1799–1807.

(64) Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

(65) Mikolov, T.; Deoras, A.; Povey, D.; Burget, L.; Černockỳ, J. In *2011 IEEE Workshop on Automatic Speech Recognition & Understanding*, 2011, pp 196–201.

(66) Hinton, G.; Deng, L.; Yu, D.; Dahl, G. E.; Mohamed, A.-r.; Jaitly, N.; Senior, A.; Vanhoucke, V.; Nguyen, P.; Sainath, T. N., et al. *IEEE Signal processing magazine* **2012**, *29*, 82–97.

(67) Sainath, T. N.; Mohamed, A.-r.; Kingsbury, B.; Ramabhadran, B. In *2013 IEEE international conference on acoustics, speech and signal processing*, 2013, pp 8614–8618.

(68) Collobert, R.; Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K.; Kuksa, P. *Journal of machine learning research* **2011**, *12*, 2493–2537.

(69) Bordes, A.; Chopra, S.; Weston, J. *arXiv preprint arXiv:1406.3676* **2014**.

(70) Jean, S.; Cho, K.; Memisevic, R.; Bengio, Y. *arXiv preprint arXiv:1412.2007* **2014**.

(71) Sutskever, I.; Vinyals, O.; Le, Q. V. In *Advances in neural information processing systems*, 2014, pp 3104–3112.

Chapter 2

Flexible Ansatz for N-body Configuration Interaction

Flexible Ansatz for N-body Configuration Interaction

Taewon D. Kim[†] Ramón Alain Miranda-Quintana[‡]

Michael Richer[†] Paul W. Ayers[†]

September 26, 2020

[†]Department of Chemistry and Chemical Biology, McMaster University, Hamilton, Ontario, L8S-4L8, Canada

[‡]Department of Chemistry, University of Florida, Gainesville, FL 32603, USA

Abstract

We present a Flexible Ansatz for N-body Configuration Interaction (FANCI) that includes any multideterminant wavefunction. This ansatz is a generalization of the Configuration Interaction (CI) wavefunction, where the coefficients are replaced by a specified function of certain parameters. By making an appropriate choice for this function, we can reproduce popular wavefunction structures like CI, Coupled-Cluster, Tensor Network States, and geminal-product wavefunctions. The universality of this framework suggests a programming structure that allows for the easy construction and optimization of arbitrary wavefunctions. Here, we will discuss the structures of the FANCI framework and its implications for wavefunction properties, particularly accuracy, cost, and size-consistency. We

demonstrate the flexibility of this framework by reconstructing popular wavefunction ansätze and modifying them to construct novel wavefunction forms. FANCI provides a powerful framework for exploring, developing, and testing new wavefunction forms.

2.1 Introduction

In this paper, we focus on electronic systems, whose Hamiltonian can be written as

$$\hat{H}_{elec} = \sum_{ij} h_{ij} a_i^\dagger a_j + \frac{1}{2} \sum_{ijkl} g_{ijkl} a_i^\dagger a_j^\dagger a_l a_k \tag{2.1}$$

where h_{ij} and g_{ijkl} are the one- and two-electron integrals and $a_i^\dagger$ (a_i) creates (annihilates) the i^{th} spin-orbital. The exact solutions to the electronic Hamiltonian can be written as a linear combination of all possible N-electron basis functions (Slater determinants) formed from the given set of spin-orbitals. This is the Full Configuration Interaction (FCI) wavefunction[1]:

$$|\Psi_{\text{FCI}}\rangle = \sum_{\mathbf{m}}^{\binom{2K}{N}} C_{\mathbf{m}} |\mathbf{m}\rangle \tag{2.2}$$

where $2K$ is the number of spin-orbitals, N is the number of electrons, and C_m is the coefficient of the Slater determinant $|\mathbf{m}\rangle$. We can think of $\mathbf{m}$ as an occupation vector that specifies which of the $2K$ spin-orbitals are occupied to construct the N-electron basis functions. The number of parameters for the FCI wavefunction scales combinatorially with the number of orbitals and electrons, so brute-force direct calculations of the FCI wavefunctions are restricted to small systems with small basis sets.

Various approximations can be made to the Schrödinger equation to bring down its cost: (1) simplify the Hamiltonian, (2) find alternative algorithms, and (3) parameterize the FCI wavefunction[2–10]. In this article, we focus on the parameterization of the FCI wavefunction. The simplest approximation to the FCI wavefunction involves explicitly selecting (or truncating) the Slater determinants that contribute to the wavefunction. Such wavefunctions are broadly termed selected Configuration Interaction (CI) wavefunctions:

$$|\Psi_{\mathrm{CI}}\rangle = \sum_{\mathbf{m} \in S} C_{\mathbf{m}} |\mathbf{m}\rangle \tag{2.3}$$

where S is a subset of the Slater determinants within the given basis. The Slater determinants can be selected by seniority, such as the doubly occupied CI (DOCI) wavefunction[11–22], or by excitation-level relative to a reference Slater determinant, such as CI singles and doubles (CISD) wavefunction[23]. Alternatively, we can select all (or many) of the Slater determinants in a given set of orbitals. This leads to active-space methods like CASSCF[24], RASSCF [25], and MCSCF[26]. Finally, if the orbitals are localized, the Slater determinants that embody chemically intuitive concepts can be linearly combined to construct Valence Bond (VB) structures[27–31]. This leads to VBCI methods. While there are many variants of the CI wavefunction, most are not size-consistent and choosing an efficient set of orbitals/determinants is molecule and geometry dependent. For truly strongly-correlated systems, which have myriad Slater determinants with small yet significant contributions, selected CI methods generally fail.

Alternatively, the FCI wavefunction can be approximated by an alternative form (ansatz), such as a nonlinear function of parameters that are not simply the coefficients of the Slater determinants. For example, the Coupled-Cluster (CC) wavefunction parameterizes the CI wavefunction using an exponential ansatz[32–37]:

$$\begin{aligned}
|\Psi_{\mathrm{CC}}\rangle &= \exp\left(\sum_i \sum_a t_i^a \hat{E}_i^a + \sum_{i<j} \sum_{a<b} t_{ij}^{ab} \hat{E}_{ij}^{ab} + \dots\right) |\Phi_{\mathrm{HF}}\rangle \\
&= \exp\left(\sum_{\substack{\mathbf{i},\mathbf{a} \\ \hat{E}_{\mathbf{i}}^{\mathbf{a}} \in \tilde{S}_{\hat{E}}}} t_{\mathbf{i}}^{\mathbf{a}} \hat{E}_{\mathbf{i}}^{\mathbf{a}}\right) |\Phi_{\mathrm{HF}}\rangle
\end{aligned} \tag{2.4}$$

where $\hat{E}_{\mathbf{i}}^{\mathbf{a}}$ is an excitation operator that excites electrons from a set of occupied orbitals $\mathbf{i}$ to a set of virtual orbitals $\mathbf{a}$, $t_{\mathbf{i}}^{\mathbf{a}}$ is the associated coefficient/amplitude, and $\tilde{S}_{\hat{E}}$ is the set of allowed excitation operators. Just as in CI wavefunctions, the complexity of the wavefunction (and the number of parameters) can be controlled by truncating the set of excitation operators used.

Tensor Product State (TPS) wavefunctions are expressed with respect to parameters that describe the correlations between spatial orbitals of different states (occupations)[38–50]:

$$|\Psi_{\mathrm{TPS}}\rangle = \sum_{n_1 \dots n_K} \sum_{\substack{i_{12} \dots i_{1K} \\ i_{23} \dots i_{2K} \\ \vdots \\ i_{K-1\,K}}} (M_1)^{n_1}_{i_{12} \dots i_{1K}} (M_2)^{n_2}_{i_{12} i_{23} \dots i_{2K}} \dots (M_K)^{n_K}_{i_{1K} \dots i_{K-1\,K}} |n_1 \dots n_K\rangle \tag{2.5}$$

where n_j is the occupation of the j^{th} spatial orbital, i_{jk} is an auxiliary index that represents the correlation of j^{th} orbital with the k^{th} orbital, and M_j is a tensor that describes the correlation between the j^{th} orbital and the other orbitals. Each tensor is connected to others by at least one auxiliary index, meaning that the correlation between orbitals is represented by tensor contraction on the auxiliary indices. The specific auxiliary indices used in the tensor-contraction control the correlations that the wavefunction explicitly captures and thereby the complexity of the wavefunction. The Matrix Product State (MPS) wavefunction simplifies the TPS wavefunction by only correlating the orbitals that are adjacent to each other in an ordered list [13, 51–54]:

$$|\Psi_{\text{MPS}}\rangle = \sum_{\substack{n_1,n_2,\dots,n_{K-1},n_K \\ i_{12},i_{23},\dots,i_{K-1\,K}}} (M_1)^{n_1}_{i_{12}} (M_2)^{n_2}_{i_{12}i_{23}} \dots (M_K)^{n_K}_{i_{K-1\,K}} |n_1 n_2 \dots n_{K-1} n_K\rangle \tag{2.6}$$

MPS wavefunctions are usually optimized using the Density Matrix Renormalization Group (DMRG) algorithm[9, 55–65].

While MPS and TPS wavefunctions describe, in essence, the contribution of each orbital to the wavefunction, the Antisymmetrized Product of Geminals (APG) wavefunction describes the contribution of each electron pair (geminal) to the wavefunction[52, 53, 66–74]:

$$\begin{aligned} |\Psi_{\text{APG}}\rangle &= \prod_{p=1}^{N/2} G_p^\dagger |0\rangle \\ G_p^\dagger |0\rangle &= \sum_{ij}^{2K} C_{p;ij} a_i^\dagger a_j^\dagger |0\rangle \end{aligned} \tag{2.7}$$

where $G_p^\dagger$ is the creation operator of the p^{th} geminal and $C_{p;ij}$ is the contribution of the i^{th} and j^{th} spin-orbitals to the p^{th} geminal. Again, the complexity and the accuracy of this wavefunction can be controlled by limiting the number of terms in the wavefunction. For example, we can choose a set of orbital pairs that contribute to the wavefunction. The Antisymmetrized Product of Interacting Geminals (APIG) wavefunctions and its variants[54, 75–102], such as Antisymmetrized Product of 1-Reference Orbital Geminals (AP1roG)[103] and Antisymmetrized Product of Rank-2 Geminals (APr2G) wavefunctions[104] only use spin-orbital pairs from the same spatial orbital:

$$\begin{aligned}
|\Psi_{\text{APIG}}\rangle &= \prod_{p}^{N/2} G_p^\dagger |0\rangle \\
&= \prod_{p}^{N/2} \sum_{i}^{K} C_{p;i} a_i^\dagger a_{\bar{i}}^\dagger |0\rangle
\end{aligned} \tag{2.8}$$

where $a_i^\dagger$ and $a_{\bar{i}}^\dagger$ are the creation operators of the alpha and beta spin-orbitals corresponding to the i^{th} spatial orbital.

Each of these wavefunction ansätze seems to be fundamentally different in its nomenclature, structure, and computation. Yet every wavefunction approximates the FCI wavefunction, and through this common goal, they are intrinsically connected to one another. There are known mathematical connections between certain ansätze and these are occasionally exploited to derive new flavours of these methods. For example, many geminal methods can be rewritten as special CC wavefunctions[81–83, 85, 103]. However, these insights are seldom transferred between ansätze and the development of new ansätze seems even rarer. If the goal within electronic structure theory is to find the ansatz that strikes the best balance between the cost and accuracy for a given system, then do we not limit ourselves by committing to a particular ansatz and its assumptions? In this article, we present a *general* wavefunction structure in which new ansätze can be easily developed and relations between existing ansätze can be elucidated. We express popular existing multideterminant ansätze (CI, CC, TPS, and APG wavefunctions) within this framework and develop new structures by combining their features. We hope to demonstrate that wavefunction design can be reduced to unique combinations of modular structures, indicating that an incredible number of "new" ansätze can be trivially developed.

2.2 Flexible Ansatz for N-particle Configuration Interaction (FANCI)

The proposed wavefunction structure is quite simple and resembles the CI wavefunction (Equation 2.3):

$$|\Psi_{\text{FANCI}}\rangle = \sum_{\mathbf{m} \in S_{\mathbf{m}}} f(\mathbf{m}, \vec{P}) \, |\mathbf{m}\rangle \tag{2.9}$$

where $S_{\mathbf{m}}$ is a set of allowed Slater determinants and f is a function that controls the weight of each Slater determinant, $\mathbf{m}$, using the parameters, $\vec{P}$. Since Slater determinants can be uniquely represented with an excitation operator and a reference, Equation 2.9 can be rewritten with respect to excitation operators, $\hat{E}$.

$$|\Psi_{\text{FANCI}}\rangle = \sum_{\hat{E} \in S_{\hat{E}}} f(\hat{E}, \vec{P}) \hat{E} \, |\Phi_{\text{ref}}\rangle \tag{2.10}$$

where $S_{\hat{E}}$ is a set of allowed excitation operators and f is a function that maps the weight of the excitation operator from $\hat{E}$ and $\vec{P}$. The $S_{\mathbf{m}}$ and $S_{\hat{E}}$ are equivalent representations of a set of Slater determinants and can be used interchangeably.

In this paper, we aim to demonstrate the generality, utility, and flexibility of this framework. In the next section, we show that the framework is general by expressing popular ansätze as FANCI wavefunctions. Then, in Section 2.4, we discuss how the choices of S, $\vec{P}$, and f affect the accuracy, cost, and size-consistency of the wavefunction. Finally, we demonstrate the flexibility of this framework by constructing novel wavefunction structures.

2.3 Examples

2.3.1 Hartree-Fock

The ground-state Hartree-Fock (HF) wavefunction is the Slater determinant of orthonormal orbitals that provides the lowest energy[105, 106]. Starting from an arbitrary set of orthonormal orbitals, created by $\{a_j^\dagger\}$, the HF wavefunction can be obtained by optimizing the unitary transformation that provides the lowest energy.

$$\begin{aligned} |\Psi_{\text{HF}}\rangle &= \prod_{i=1}^{N} \left(\sum_{j=1}^{2K} a_j^\dagger U_{ji} \right) |0\rangle \\ &= \sum_{\mathbf{m}} |U(\mathbf{m})|^- \, |\mathbf{m}\rangle \end{aligned} \tag{2.11}$$

where U is a unitary matrix, and $U(\mathbf{m})$ is a submatrix of U obtained by selecting the rows that correspond to the spin-orbitals in $\mathbf{m}$. The derivation is given in the Appendix 2.9.1. If only N orthonormal orbitals are rotated, or alternatively, if there is no mixing of the occupied and virtual orbitals, then the HF wavefunction is obtained trivially:

$$\begin{aligned} |\Psi_{\text{HF}}\rangle &= \prod_{i=1}^{N} \left(\sum_{j=1}^{N} a_j^\dagger U_{ji} \right) |0\rangle \\ &= |U(\mathbf{m})|^- \, |\mathbf{m}\rangle \end{aligned} \tag{2.12}$$

where $\mathbf{m}$ is the set of the occupied orbitals. With normalization, $|U(\mathbf{m})|^-$ becomes 1. In other words, the HF wavefunction is invariant to rotation of the occupied orbitals if there is no mixing between occupied and virtual orbitals [107].

2.3.2 Truncated Configuration Interaction

The truncated CI wavefunction (Equation 2.3) is a linear combination of selected Slater determinants[108]. Such wavefunctions can be trivially described in the proposed framework: the set of allowed Slater determinants, S, is the same; the parameters, $\vec{P}$, are the coefficients of the Slater determinants, $\vec{C}$; and the parameterizing function, f, simply selects the appropriate coefficient, $C_{\mathbf{m}}$, given the Slater determinant, $\mathbf{m}$.

$$f(\mathbf{m}, \vec{C}) = \vec{e}_{\mathbf{m}} \cdot \vec{C}$$

where $\vec{e}_{\mathbf{m}}$ is a vector that gives 1 in the position of $\mathbf{m}$ and 0 elsewhere. Altogether, the CI wavefunction is

$$|\Psi_{\text{CI}}\rangle = \sum_{\mathbf{m} \in S_{\mathbf{m}}} \left(\vec{e}_{\mathbf{m}} \cdot \vec{C} \right) |\mathbf{m}\rangle \tag{2.13}$$

If the CI wavefunction is expressed with respect to excitations on a reference, we get

$$|\Psi_{\mathrm{CI}}\rangle = \sum_{\hat{E}_{\mathbf{i}}^{\mathbf{a}} \in S_{\hat{E}}} \left(\vec{e}_{\hat{E}_{\mathbf{i}}^{\mathbf{a}}} \cdot \vec{C}\right) \hat{E}_{\mathbf{i}}^{\mathbf{a}} |\Phi_{ref}\rangle \tag{2.14}$$

2.3.3 Coupled-Cluster

The CC wavefunction (Equation 2.4) uses the exponential operator to approximate high-order excitations as a product of lower-order excitations [109].

$$\begin{aligned} |\Psi_{\mathrm{CC}}\rangle &= \exp(\hat{T}) |\Phi_{\mathrm{HF}}\rangle \\ &= \sum_{n=0}^{\infty} \frac{1}{n!} \hat{T}^n |\Phi_{\mathrm{HF}}\rangle \end{aligned} \tag{2.15}$$

where

$$\hat{T} = \sum_{\hat{E}_{\mathbf{i}}^{\mathbf{a}} \in \tilde{S}_{\hat{E}}} t_{\mathbf{i}}^{\mathbf{a}} \hat{E}_{\mathbf{i}}^{\mathbf{a}} \tag{2.16}$$

and $\tilde{S}_{\hat{E}}$ is a set of excitation operators. The Maclaurin series in Equation 2.15 lets one express the CI coefficients in terms of CC cluster amplitudes $t_{\mathbf{i}}^{\mathbf{a}}$. Specifically, the cluster amplitudes are cumulants of the CI coefficients [110–114]. The powers of $\hat{T}$ (cluster operator) give the wavefunction access to excitations beyond those allowed ($\tilde{S}_{\hat{E}}$) by generating all (product-wise) combinations of the allowed excitation operators. However, an excitation can be described with different combinations of excitation operators, and the cumulant can be simplified by grouping together terms that correspond to the same excitation (or Slater determinant). Each combination corresponds to a subset of $\tilde{S}_{\hat{E}}$, such that the set of all Slater determinants in the CC wavefunction can be described in terms of all possible subsets of $\tilde{S}_{\hat{E}}$:

$$S_{\hat{E}} = \left\{ \prod_{\hat{E}_k \in T} \hat{E}_k \middle| T \subseteq \tilde{S}_{\hat{E}} \right\} \tag{2.17}$$

Then, the wavefunction can be written as a sum over all possible Slater determinants and a sum over all possible combinations of excitation operators that produce the given Slater determinant.

$$\begin{aligned} f(\hat{E}_{\mathbf{i}}^{\mathbf{a}}, \mathbf{t}) &= \sum_{\substack{\{\hat{E}_{\mathbf{i}_1}^{\mathbf{a}_1} \dots \hat{E}_{\mathbf{i}_n}^{\mathbf{a}_n}\} \subseteq \tilde{S}_{\hat{E}} \\ \operatorname{sgn} \prod_{k=1}^{n} \hat{E}_{\mathbf{i}_k}^{\mathbf{a}_k} = \hat{E}_{\mathbf{i}}^{\mathbf{a}}}} \operatorname{sgn}(\sigma_{\hat{E}_{\mathbf{i}_1}^{\mathbf{a}_1} \dots \hat{E}_{\mathbf{i}_n}^{\mathbf{a}_n}}) \frac{1}{n!} \begin{vmatrix} t_{\mathbf{i}_1}^{\mathbf{a}_1} & \dots & t_{\mathbf{i}_n}^{\mathbf{a}_n} \\ \vdots & \ddots & \vdots \\ t_{\mathbf{i}_1}^{\mathbf{a}_1} & \dots & t_{\mathbf{i}_n}^{\mathbf{a}_n} \end{vmatrix}^{+} \\ &= \sum_{\substack{\{\hat{E}_{\mathbf{i}_1}^{\mathbf{a}_1} \dots \hat{E}_{\mathbf{i}_n}^{\mathbf{a}_n}\} \subseteq \tilde{S}_{\hat{E}} \\ \operatorname{sgn} \prod_{k=1}^{n} \hat{E}_{\mathbf{i}_k}^{\mathbf{a}_k} = \hat{E}_{\mathbf{i}}^{\mathbf{a}}}} \operatorname{sgn}(\sigma_{\hat{E}_{\mathbf{i}_1}^{\mathbf{a}_1} \dots \hat{E}_{\mathbf{i}_n}^{\mathbf{a}_n}}) \prod_{k=1}^{n} t_{\mathbf{i}_k}^{\mathbf{a}_k} \end{aligned} \tag{2.18}$$

where n is the dimension of the subset $\{\hat{E}_{\mathbf{i}_1}^{\mathbf{a}_1} \dots \hat{E}_{\mathbf{i}_n}^{\mathbf{a}_n}\}$. The sum can be interpreted as a sum over all possible partitions of a given excitation operator, $\hat{E}_{\mathbf{i}}^{\mathbf{a}}$, into excitations from the given set. The signature of the permutation, $\operatorname{sgn}(\sigma_{\hat{E}_{\mathbf{i}_1}^{\mathbf{a}_1} \dots \hat{E}_{\mathbf{i}_n}^{\mathbf{a}_n}})$, results from reordering the creation and annihilation operators of the lower-order excitations to the same order as the given excitation operator:

$$\hat{E}_{\mathbf{i}}^{\mathbf{a}} = \operatorname{sgn}(\sigma_{\hat{E}_{\mathbf{i}_1}^{\mathbf{a}_1} \dots \hat{E}_{\mathbf{i}_n}^{\mathbf{a}_n}}) \hat{E}_{\mathbf{i}_1}^{\mathbf{a}_1} \cdots \hat{E}_{\mathbf{i}_n}^{\mathbf{a}_n} \tag{2.19}$$

The permanent, $|A|^+$, accounts for all possible orderings within a given set of excitation operators. Altogether, the CC wavefunction can be reformulated as

$$|\Psi_{\mathrm{CC}}\rangle = \sum_{\hat{E}_{\mathbf{i}}^{\mathbf{a}} \in S_{\hat{E}}} \left(\sum_{\substack{\{\hat{E}_{\mathbf{i}_1}^{\mathbf{a}_1} \dots \hat{E}_{\mathbf{i}_n}^{\mathbf{a}_n}\} \subseteq \tilde{S}_{\hat{E}} \\ \operatorname{sgn} \prod_{k=1}^{n} \hat{E}_{\mathbf{i}_k}^{\mathbf{a}_k} = \hat{E}_{\mathbf{i}}^{\mathbf{a}}}} \operatorname{sgn}(\sigma_{\hat{E}_{\mathbf{i}_1}^{\mathbf{a}_1} \dots \hat{E}_{\mathbf{i}_n}^{\mathbf{a}_n}}) \prod_{k=1}^{n} t_{\mathbf{i}_k}^{\mathbf{a}_k} \right) \hat{E}_{\mathbf{i}}^{\mathbf{a}} |\Phi_{\mathrm{HF}}\rangle \tag{2.20}$$

More details are provided in the Appendix 2.9.4.

2.3.4 Tensor Product State

The TPS (and MPS) wavefunction (Equation 2.5) determines the weight of a Slater determinant by tensor (and matrix) contractions, where each shared index corresponds to a correlation between orbitals[9, 42]:

$$|\Psi_{\mathrm{TPS}}\rangle = \sum_{n_1 \dots n_K} \sum_{\substack{i_{12} \dots i_{1K} \\ i_{23} \dots i_{2K} \\ \vdots \\ i_{K-1\,K}}} (M_1)_{i_{12} \dots i_{1K}}^{n_1} (M_2)_{i_{12} i_{23} \dots i_{2K}}^{n_2} \dots (M_K)_{i_{1K} \dots i_{K-1\,K}}^{n_K} |n_1 \dots n_K\rangle$$

Each spatial orbital, k, is associated with a tensor, M_k, and each tensor is associated with the occupation of its spatial orbital (i.e. its state), n_k, and with other tensors using its auxiliary indices, $\{i_{1k} \dots i_{k-1\,k}\ i_{k\,k+1} \dots i_{kK}\}$. Then, the coefficient associated with the Slater determinant, represented by $\{n_1 \dots n_K\}$, is approximated by tensor-contraction. Many variants of TPS, including MPS, impose some structure on the tensor product so that the evaluation and optimization of the wavefunction are computationally tractable.

Since $\{n_1 \dots n_K\}$ is yet another representation of the Slater determinant, we can describe the wavefunction with respect to $\mathbf{n}$:

$$\mathbf{n} = \{n_1 \dots n_K\}$$

Therefore, the TPS wavefunction can be rewritten as

$$|\Psi_{\text{TPS}}\rangle = \sum_{\mathbf{n} \in S_{\text{FCI}}} \bigodot_{k=1}^{K} (M_k)^{n_k} |\mathbf{n}\rangle \tag{2.21}$$

where K is the number of spatial orbitals, and $\bigodot$ describes the specific tensor-contraction used in the wavefunction. While it is not common to do so, the TPS wavefunctions can be equivalently expressed with respect to spin-orbitals. If each state of the TPS wavefunction corresponds to the occupation of a spin-orbital, m_k, then the same notation can be used as in Equation 2.9

$$|\Psi_{\text{TPS}}\rangle = \sum_{\mathbf{m} \in S_{\text{FCI}}} \bigodot_{k=1}^{2K} (M_k)^{m_k} |\mathbf{m}\rangle \tag{2.22}$$

where M_k is the tensor associated with spin-orbital k.

2.3.5 Antisymmetrized Product of Geminals

Just as the HF wavefunction is constructed as an antisymmetrized product of one-electron wavefunctions (orbitals), the APG wavefunction is constructed as an antisymmetrized product of two-electron wavefunctions (geminals)[66, 68, 69, 74]:

$$\begin{aligned}
|\Psi_{\text{APG}}\rangle &= \prod_{p=1}^{N/2} G_p^\dagger |0\rangle \\
&= \prod_{p=1}^{N/2} \sum_{ij}^{2K} C_{p;ij} a_i^\dagger a_j^\dagger |0\rangle \\
&= \prod_{p=1}^{N/2} \sum_{\mathbf{m}_k} C_{p;\mathbf{m}_k} A_{\mathbf{m}_k}^\dagger |0\rangle
\end{aligned}$$

where the $\mathbf{m}_k$ denotes a set of a pair of indices and $A_{\mathbf{m}_k}^\dagger$ denotes the creation operator that corresponds to $\mathbf{m}_k$. Similar to the CC wavefunction, the product of sums can be expanded out as a sum over each Slater determinant and a sum over the different combinations of electron pairs that create the Slater determinant. In the HF wavefunction, the product of sums results in a determinant due to the antisymmetry with respect to the interchange of electrons. In the APG wavefunction, however, the interchange of electron pairs is symmetric, and the product of sums results in a permanent. Given the set of all possible two-electron creation operators, $\tilde{S}$, a subset of exactly $\frac{N}{2}$ two-electron creators, $\{A_{\mathbf{m}_1}^\dagger \dots A_{\mathbf{m}_{N/2}}^\dagger\}$, is needed to construct a given Slater determinant, $\mathbf{m}$, where the number of electrons, N, is even. Since an orbital cannot be occupied more than once and all the orbitals are necessary to construct a given Slater determinant, any selection of orbital pairs, $\{\mathbf{m}_1 \dots \mathbf{m}_{N/2}\}$, must be disjoint and exhaustive.

$$
\begin{aligned}
f(\mathbf{m}, \mathbf{C}) &= \sum_{\substack{\{\mathbf{m}_1 \dots \mathbf{m}_{N/2}\} \subseteq \tilde{S} \\ \bigcup_{k=1}^{N/2} \mathbf{m}_k = \mathbf{m} \\ \mathbf{m}_p \cap \mathbf{m}_q = \emptyset \; \forall p \neq q}} \operatorname{sgn}(\sigma_{\mathbf{m}_1 \dots \mathbf{m}_{N/2}}) |C(\mathbf{m}_1, \dots, \mathbf{m}_{N/2})|^+ \\
&= \sum_{\substack{\{\mathbf{m}_1 \dots \mathbf{m}_{N/2}\} \subseteq \tilde{S} \\ \operatorname{sgn} A^\dagger_{\mathbf{m}_1} \dots A^\dagger_{\mathbf{m}_{N/2}} |0\rangle = |\mathbf{m}\rangle}} \operatorname{sgn}(\sigma_{\mathbf{m}_1 \dots \mathbf{m}_{N/2}}) |C(\mathbf{m}_1, \dots, \mathbf{m}_{N/2})|^+
\end{aligned}
\tag{2.23}
$$

where the orbital pairs, $\{\mathbf{m}_1 \dots \mathbf{m}_{N/2}\}$, are selected such that they result in the given Slater determinant, $\mathbf{m}$, without duplicate orbitals. Similar to the CC wavefunction (Equation 2.18), the sum can be interpreted as a sum over all allowed partitions of the given Slater determinant into the electron pairs. The signature of the permutation, $\operatorname{sgn}(\sigma_{\mathbf{m}_1 \dots \mathbf{m}_{N/2}})$, results from reordering the creation operators in the electron pairs to the same order as in the given Slater determinant:

$$
\prod_{i \in \mathbf{m}} a^\dagger_i = \operatorname{sgn}(\sigma_{\mathbf{m}_1 \dots \mathbf{m}_{N/2}}) \prod_{p=1}^{N/2} A^\dagger_{\mathbf{m}_p} \tag{2.24}
$$

Altogether, the APG wavefunction is reformulated as

$$
|\Psi_{\text{APG}}\rangle = \sum_{\mathbf{m} \in S^{\text{FCI}}_{\mathbf{m}}} \left(\sum_{\substack{\{\mathbf{m}_1 \dots \mathbf{m}_{N/2}\} \subseteq \tilde{S} \\ \operatorname{sgn} A^\dagger_{\mathbf{m}_1} \dots A^\dagger_{\mathbf{m}_{N/2}} |0\rangle = |\mathbf{m}\rangle}} \operatorname{sgn}(\sigma_{\mathbf{m}_1 \dots \mathbf{m}_{N/2}}) |C(\mathbf{m}_1, \dots, \mathbf{m}_{N/2})|^+ \right) |\mathbf{m}\rangle \tag{2.25}
$$

where $S^{\text{FCI}}_{\mathbf{m}}$ is a set of all possible Slater determinants (i.e. Slater determinants of a FCI wavefunction). The derivation is given in the Appendix 2.9.2.

The Antisymmetrized Product of Interacting Geminals (APIG) is a special case of the APG wavefunction such that only the electron pairs within the same spatial orbital, i.e. doubly occupied spatial orbitals, are used to build the wavefunction[102]. The sum over the partitions reduces to a single element because there is only one way to construct a given (seniority-zero) Slater determinant from electron pairs of doubly occupied orbitals.

$$|\Psi_{\mathrm{APIG}}\rangle = \sum_{\mathbf{m} \in S_{\mathbf{m}}^{\mathrm{DOCI}}} |C(\mathbf{m})|^{+} |\mathbf{m}\rangle \tag{2.26}$$

where $S_{\mathbf{m}}^{\mathrm{DOCI}}$ is the set of all seniority-zero (no unpaired electrons) Slater determinants, and $|C(\mathbf{m})|^{+}$ is a permanent of the parameters that correspond to the spatial orbitals used to construct $\mathbf{m}$. The APIG wavefunction can be further simplified by imposing structures onto the permanent: the Antisymmetrized Product of 1-reference Orbitals Geminals (AP1roG) wavefunction assumes that a large portion of the coefficient matrix is an identity matrix [103]; and the Antisymmetrized Product of rank-2 Geminals (APr2G) wavefunction assumes that the coefficient matrix is a Cauchy matrix[104]. APr2G reduces the cost of evaluating a permanent ($\mathcal{O}(n!)$) to that of a determinant ($\mathcal{O}(n^3)$). AP1roG has the cost of $\mathcal{O}(m!)$ where m is the order of excitation with respect to the reference Slater determinant. It is cheap to evaluate the overlap of the AP1roG wavefunction with low-order excitations of the reference determinant.

2.3.6 Universality of FANCI

The CI, CC, TPS, and APG wavefunctions and their variants can be expressed within the FANCI framework using different S, $\vec{P}$, and f. We can define a multideterminant wavefunction as a function that has a well-defined overlap with a set of orthonormal

Slater determinants. Provided that the wavefunction exists within the space spanned by the Slater determinants, the wavefunction can be re-expressed as a linear combination of Slater determinants via a projection:

$$\begin{aligned} \left|\Psi(\vec{P})\right\rangle &= \sum_{\mathbf{m} \in S_{\mathbf{m}}} \left|\mathbf{m}\right\rangle \left\langle \mathbf{m} \middle| \Psi(\vec{P}) \right\rangle \\ &= \sum_{\mathbf{m} \in S_{\mathbf{m}}} f(\mathbf{m}, \vec{P}) \left|\mathbf{m}\right\rangle \end{aligned} \tag{2.27}$$

where

$$f(\mathbf{m}, \vec{P}) = \left\langle \mathbf{m} \middle| \Psi(\vec{P}) \right\rangle \tag{2.28}$$

Therefore, all multideterminant wavefunctions, as defined above, can be expressed within the framework of Equation 2.9: S is the minimal set of Slater determinants required to fully describe the wavefunction; $\vec{P}$ is the parameters of the wavefunction; and f is the overlap of the wavefunction with the Slater determinant, $\mathbf{m}$.

2.4 Characteristics

In the formulation of Equation 2.9, a multideterminant wavefunction is defined using only a specified (sub)set of Slater determinants, S, wavefunction parameters, $\vec{P}$, and function f. Since the characteristics of a wavefunction ansatz depend on its structure, all characteristics of a multideterminant wavefunction can be deduced from the specified S, $\vec{P}$, and f. Designing a wavefunction with desirable characteristics, therefore, merely requires selecting S, $\vec{P}$, and f. We propose to approach method development in electronic structure theory as a search for S, $\vec{P}$, and f that induce the desired wavefunction features. While many features can be considered, and we shall consider additional features in future work, here we shall address just three important characteristics: accuracy, cost, and size-consistency.

2.4.1 Accuracy

Ultimately, the FANCI wavefunction models the FCI wavefunction by parameterizing the weights of each Slater determinant. If there are Slater determinants absent from the FANCI wavefunction, i.e. $S \subset S_{\text{FCI}}$, then the omitted Slater determinants are assumed to have no contributions to the FCI wavefunction. The effects of S can be viewed as a modification of the parameterizing function.

$$|\Psi_{\text{FANCI}}\rangle = \sum_{\mathbf{m} \in S_{\text{FCI}}} g(\mathbf{m}, \vec{P}) \, |\mathbf{m}\rangle$$

where

$$g(\mathbf{m}, \vec{P}) = \begin{cases} f(\mathbf{m}, \vec{P}) & ; \mathbf{m} \in S \\ 0 & ; \mathbf{m} \notin S \end{cases}$$

Alternatively, the FANCI wavefunction can be viewed as a model for the CI wavefunction built using the same restricted set of Slater determinants. In either case, preventing Slater determinants from contributing to the wavefunction will cause deviations from the FCI wavefunction.

As with any parameterization (or fitting) problem, it becomes easier to find a function that accurately describes each weight as the number of parameters increase. FANCI wavefunctions cannot be exact, in general, unless the number of parameters is greater than or equal to the number of parameters in the Hamiltonian which, in the case of electronic structure theory, means there should be at least as many parameters as there are two-electron integrals[115, 116]. Methods with many fewer parameters are, typically, static correlation methods. On the other hand, appropriately constructed FANCI ansätze should approach the FCI limit as the number of parameters approaches the number of Slater determinants. However, the cost associated with optimizing the wavefunction typically increases superlinearly as the number of parameters increases.

2.4.2 Cost

The cost associated with a wavefunction can be divided into the cost of its storage, evaluation, and optimization, all of which are intricately linked. The cost of storage is associated with the number of parameters needed to describe the wavefunction. The cost of evaluating the wavefunction depends on the cost of evaluating f and on the number of times f needs to be evaluated. For example, in order to evaluate the norm of a wavefunction, f must be evaluated for every Slater determinant in S.

$$\begin{aligned}\langle\Psi|\Psi\rangle &= \sum_{\mathbf{m}\in S}\sum_{\mathbf{n}\in S} f^*(\mathbf{m}) \langle\mathbf{m}|\mathbf{n}\rangle f(\mathbf{n})\\ &= \sum_{\mathbf{m}\in S} f^*(\mathbf{m}) f(\mathbf{m})\end{aligned}$$

Upon optimization, a new set of parameters are found such that the wavefunction satisfies the Schrödinger equation.

$$\hat{H} \left|\Psi\right\rangle = E \left|\Psi\right\rangle \tag{2.29}$$

Equation 2.29 is often rewritten in its variational form or its projected form to make it easier to solve numerically. The optimization procedure and the associated costs depend on the equations that are being solved.

The variational Schrödinger equation involves integrating both sides of Equation 2.29 with the wavefunction[1].

$$\begin{gathered} \langle\Psi|\hat{H}|\Psi\rangle = E \left\langle\Psi|\Psi\right\rangle \\ \sum_{\mathbf{m},\mathbf{n}\in S} f^*(\mathbf{m},\vec{P}) \left\langle\mathbf{m}|\hat{H}|\mathbf{n}\right\rangle f(\mathbf{n},\vec{P}) = E \sum_{\mathbf{m}\in S} f^*(\mathbf{m},\vec{P}) f(\mathbf{m},\vec{P}) \end{gathered} \tag{2.30}$$

If the number of Slater determinants in S is comparable to those in the FCI wavefunction, even setting up Equation 2.30 will require far too many evaluations of f to be computationally tractable. During the optimization, all terms need to be evaluated at each step, where the number of steps needed for convergence varies depending on the system and the optimization algorithm.

In the projected Schrödinger equation, Equation 2.29 is integrated against an arbitrary function, Φ[85].

$$\langle\Phi|\hat{H}|\Psi\rangle = E \left\langle\Phi|\Psi\right\rangle \tag{2.31}$$

If Φ is not Ψ, then certain components of Ψ may be projected out, imposing additional structure on the wavefunction through the optimization process. We can express this projection explicitly with a projection operator onto some set of basis functions. When projected onto a complete set of Slater determinants, the objective function in the variational Schrödinger equation can be re-expressed as:

$$\begin{aligned}
\langle\Psi|\hat{H}|\Psi\rangle &= E\,\langle\Psi|\Psi\rangle \\
\langle\Psi|\left(\sum_{\mathbf{m}\in S_{\text{FCI}}}|\mathbf{m}\rangle\,\langle\mathbf{m}|\right)\hat{H}\,|\Psi\rangle &= E\,\langle\Psi|\left(\sum_{\mathbf{m}\in S_{\text{FCI}}}|\mathbf{m}\rangle\,\langle\mathbf{m}|\right)|\Psi\rangle \\
\sum_{\mathbf{m}\in S_{\text{FCI}}}\langle\Psi|\mathbf{m}\rangle\,\langle\mathbf{m}|\hat{H}|\Psi\rangle &= E\sum_{\mathbf{m}\in S_{\text{FCI}}}\langle\Psi|\mathbf{m}\rangle\,\langle\mathbf{m}|\Psi\rangle
\end{aligned} \tag{2.32}$$

Then, this equation can be separated out as a system of equations

$$\begin{aligned}
\langle\Psi|\mathbf{m}_1\rangle\,\langle\mathbf{m}_1|\hat{H}|\Psi\rangle &= E\,\langle\Psi|\mathbf{m}_1\rangle\,\langle\mathbf{m}_1|\Psi\rangle \\
&\vdots \\
\langle\Psi|\mathbf{m}_M\rangle\,\langle\mathbf{m}_M|\hat{H}|\Psi\rangle &= E\,\langle\Psi|\mathbf{m}_M\rangle\,\langle\mathbf{m}_M|\Psi\rangle
\end{aligned} \tag{2.33}$$

or equivalently,

$$\begin{aligned}
\langle\mathbf{m}_1|\hat{H}|\Psi\rangle &= E\,\langle\mathbf{m}_1|\Psi\rangle \\
&\vdots \\
\langle\mathbf{m}_M|\hat{H}|\Psi\rangle &= E\,\langle\mathbf{m}_M|\Psi\rangle
\end{aligned} \tag{2.34}$$

Therefore, we can approximate the variational solution by solving the projected Schrödinger equation using a set of functions that capture the important characteristics of the wavefunction. The cost of evaluating Equation 2.31 and 2.34 depends on the functions onto which the Schrödinger equation is projected. Some wavefunction structures have special functions such that Equation 2.31 or 2.34 can be evaluated cheaply. For example, the CC wavefunctions are often projected against $\langle \Phi_{\mathrm{HF}} | \exp(-\hat{T})$ because $\langle \Phi_{\mathrm{HF}} | \exp(-\hat{T}) \hat{H} \exp(\hat{T}) | \Phi_{\mathrm{HF}} \rangle$, can be simplified using the Baker-Campbell-Hausdorff expansion[117]. For a general FANCI wavefunction, however, it is convenient to project onto a set of Slater determinants, $\{\mathbf{m}_1 \dots \mathbf{m}_M\}$, obtaining a system of (generally nonlinear) equations to solve:

$$\begin{aligned} \sum_{\mathbf{m} \in S} f(\mathbf{m}, \vec{P}) \langle \mathbf{m}_1 | \hat{H} | \mathbf{m} \rangle &= E f(\mathbf{m}_1, \vec{P}) \\ &\vdots \\ \sum_{\mathbf{m} \in S} f(\mathbf{m}, \vec{P}) \langle \mathbf{m}_M | \hat{H} | \mathbf{m} \rangle &= E f(\mathbf{m}_M, \vec{P}) \end{aligned} \tag{2.35}$$

In order to find a solution, the number of equations in the system of equations must be greater than the number of unknowns. Since the number of possible Slater determinants grows exponentially, there will not be a shortage of equations (Slater determinants), and the number of equations will almost always be greater than the number of unknowns. If the least-squares solution of the nonlinear equations is found, then the residual can be used to measure the error associated with the optimized wavefunction (and energy).

We can derive the projected Schrödinger equation (Equation 2.34) directly from Equation 2.29 without integrating.

$$\begin{aligned} \hat{H} |\Psi\rangle &= E |\Psi\rangle \\ \left(\sum_{\mathbf{m} \in S_{\mathrm{FCI}}} |\mathbf{m}\rangle \langle \mathbf{m}| \right) \hat{H} |\Psi\rangle &= E \left(\sum_{\mathbf{m} \in S_{\mathrm{FCI}}} |\mathbf{m}\rangle \langle \mathbf{m}| \right) |\Psi\rangle \\ \sum_{\mathbf{m} \in S_{\mathrm{FCI}}} |\mathbf{m}\rangle \langle \mathbf{m} | \hat{H} | \Psi \rangle &= E \sum_{\mathbf{m} \in S_{\mathrm{FCI}}} |\mathbf{m}\rangle \langle \mathbf{m} | \Psi \rangle \end{aligned} \tag{2.36}$$

where both sides of the equations are projected onto a complete set of Slater determinants. This equation can be separated for each Slater determinant to produce Equation 2.34

$$\begin{aligned} \langle \mathbf{m}_1 | \hat{H} | \Psi \rangle &= E \langle \mathbf{m}_1 | \Psi \rangle \\ \langle \mathbf{m}_2 | \hat{H} | \Psi \rangle &= E \langle \mathbf{m}_2 | \Psi \rangle \\ &\vdots \end{aligned} \tag{2.37}$$

Essentially, the Schrödinger equation (Equation 2.29) is broken apart into separate equations for each contributing Slater determinant. If the projection operator is not complete (i.e. contributions from certain Slater determinants are discarded) then the equation (or system of equations) will be an approximation of the original Equation 2.29.

Unless there is a special algorithm that limits the number of evaluated terms in the variational Schrödinger equation (Equation 2.30) or a function Φ that allows cheap integration of the Schrödinger equation (Equation 2.31), a wavefunction should be evaluated using the projected Schrödinger equation (Equation 2.33 and 2.34) to control the optimization process. Both the cost and accuracy of the wavefunction can be controlled; as the number of projections increases, both accuracy and cost increases. In addition, we can impose symmetry on the wavefunction by projecting the Schrödinger equation onto a space that satisfies a particular symmetry [8, 99, 118–120]. For example, we can reintroduce particle number symmetry onto a number-symmetry broken wavefunction by projecting it onto Slater determinants with the selected particle number.

2.4.3 Size-Consistency

Let there be a system, AB, composed of two non-interacting subsystems, A and B. Then, a wavefunction is size-consistent if the energy of the wavefunction for AB is the sum of the energies of the wavefunctions for A and B, [117] i.e.

$$\begin{aligned} H_{AB} \left|\Psi_{AB}\right\rangle &= E_{AB} \left|\Psi_{AB}\right\rangle \\ &= (E_A + E_B) \left|\Psi_{AB}\right\rangle \end{aligned} \tag{2.38}$$

where

$$\begin{aligned} H_A \left|\Psi_A\right\rangle &= E_A \left|\Psi_A\right\rangle \\ H_B \left|\Psi_B\right\rangle &= E_B \left|\Psi_B\right\rangle \end{aligned} \tag{2.39}$$

Since subsystems A and B are non-interacting, there are no nonzero terms in the Hamiltonian that couple A and B, i.e. $H_{AB} = H_A + H_B$. Then, the (partially symmetry broken) wavefunction Ψ_{AB} can be written as a product of the (orthogonal) subsystem wavefunctions, Ψ_A and Ψ_B.

$$\left|\Psi_{AB}\right\rangle = \left|\Psi_A\right\rangle \left|\Psi_B\right\rangle$$

Similarly, a FANCI wavefunction is size-consistent if the weight function is multiplicatively separable:

$$\begin{aligned}
|\Psi_{AB}\rangle &= \sum_{\mathbf{m}\in S_{AB}} f_{AB}(\mathbf{m}, \vec{P})\,|\mathbf{m}\rangle \\
&= \sum_{\mathbf{m}_A\in S_A} \sum_{\mathbf{m}_B\in S_B} f_A(\mathbf{m}_A, \vec{P}_A) f_B(\mathbf{m}_B, \vec{P}_B)\,|\mathbf{m}_A\rangle\,|\mathbf{m}_B\rangle \\
&= \sum_{\mathbf{m}_A\in S_A} f_A(\mathbf{m}_A, \vec{P}_A)\,|\mathbf{m}_A\rangle \sum_{\mathbf{m}_B\in S_B} f_B(\mathbf{m}_B, \vec{P}_B)\,|\mathbf{m}_B\rangle \\
&= |\Psi_A\rangle\,|\Psi_B\rangle
\end{aligned} \tag{2.40}$$

where subscripts A and B designate that the quantity belongs only to subsystem A and B, respectively. This not only requires that f be multiplicatively separable f, i.e. $f_{AB} = f_A f_B$, but also that the Slater determinants, $\mathbf{m}$, and the parameters, $\vec{P}$, must be divisible into two disjoint parts, $\{\mathbf{m}_A, \mathbf{m}_B\}$ and $\{\vec{P}_A, \vec{P}_B\}$ respectively. To separate each Slater determinant into the subsystems, $\mathbf{m}$ must be expressed using orbitals localized to each subsystem. Since each $\mathbf{m}$ can have varying contribution from the subsystems A and B, S_A and S_B contain Slater determinants with different numbers of electrons. However, we can impose the particle number symmetry during the optimization process. Similarly, the parameters must represent quantities that are specific to each subsystem. Notice that $f_{AB} = f_A f_B$ is true when f is a determinant (Hartree-Fock), exponential (Coupled-Cluster), or permanent (geminals).

2.5 Ansätze

In the formulation of Equation 2.9, a multideterminant wavefunction can be entirely expressed with a set of Slater determinants, S, parameters, $\vec{P}$, and a weight function, f. Altering the S, $\vec{P}$, or f of a given ansatz will effectively result in a new ansatz. Additionally, the optimization method can be modified to produce an "ansatz" with a different accuracy and cost. For example, DMRG is an algorithm for optimizing MPS[121]. Here, we modify the FANCI forms of the CC (Equation 2.20), TPS (Equation 2.21), and APG (Equation 2.25) wavefunctions to construct several new wavefunction structures.

2.5.1 CC with Creators

Just as the TPS and APG wavefunctions are expressed with respect to creation operators, we can replace the excitation operators in the CC wavefunction with creation operators.

$$|\Psi_{\mathrm{CC}}\rangle = \exp\left(\sum_{\mathbf{b}_i} C_{\mathbf{b}_i} \hat{A}^\dagger_{\mathbf{b}_i} + \sum_{\mathbf{f}_i} C_{\mathbf{f}_i} \hat{A}^\dagger_{\mathbf{f}_i}\right) |0\rangle \tag{2.41}$$

where $A^\dagger_{\mathbf{b}_i}$ is a creation operator of even number of electrons (denoted as even-electron), $A^\dagger_{\mathbf{f}_i}$ is a creation operator of odd number of electrons (denoted as odd-electron), $\mathbf{b}_i$ is the set of orbitals created by $A^\dagger_{\mathbf{b}_i}$, and $\mathbf{f}_i$ is the set of orbitals created by $A^\dagger_{\mathbf{f}_i}$. For a consistent notation, we define $\tilde{S}_\mathbf{b}$ and $\tilde{S}_\mathbf{f}$ as the set of allowed creation operators. Since each creation operator can create a different number of electrons, the total number of operators needed for $\mathbf{m}$ can vary depending on the selection of creation operators. For a given Slater determinant, let n_b be the number of even-electron creators and n_f be the number of odd-electron creators. Similar to the APG wavefunction, there can be multiple combinations of creation operators that give the same Slater determinant. We represent these combinations as a sum over $\{\mathbf{b}_1 \dots \mathbf{b}_{n_b} \mathbf{f}_1 \dots \mathbf{f}_{n_f}\}$ such that the product of the associated creators results in the given Slater determinant:

$$\prod_{i \in \mathbf{m}} a_i^\dagger = \mathrm{sgn}\left(\sigma(\hat{A}^\dagger_{\mathbf{b}_1} \dots \hat{A}^\dagger_{\mathbf{b}_{n_b}} \hat{A}^\dagger_{\mathbf{f}_1} \dots \hat{A}^\dagger_{\mathbf{f}_{n_f}})\right) \hat{A}^\dagger_{\mathbf{b}_1} \dots \hat{A}^\dagger_{\mathbf{b}_{n_b}} \hat{A}^\dagger_{\mathbf{f}_1} \dots \hat{A}^\dagger_{\mathbf{f}_{n_f}} \tag{2.42}$$

Similar to the signature in the CC wavefunction, $\mathrm{sgn}\left(\sigma(\hat{A}^\dagger_{\mathbf{b}_1} \dots \hat{A}^\dagger_{\mathbf{b}_{n_b}} \hat{A}^\dagger_{\mathbf{f}_1} \dots \hat{A}^\dagger_{\mathbf{f}_{n_f}})\right)$ is the signature resulting from reordering the one-electron creators into the same order as the given Slater determinant.

In the CC wavefunction, all of the excitation operators commute with one another. Accounting for all possible orderings of the operators results in a permanent of the parameters with identical rows, i.e.

$$\begin{vmatrix} t_{\mathbf{i}_1}^{\mathbf{a}_1} & \dots & t_{\mathbf{i}_N}^{\mathbf{a}_N} \\ \vdots & \ddots & \vdots \\ t_{\mathbf{i}_1}^{\mathbf{a}_1} & \dots & t_{\mathbf{i}_N}^{\mathbf{a}_N} \end{vmatrix}^+$$

Similarly, if all of the creation operators commute with one another, i.e. they are all even-electron creators, then

$$|\Psi\rangle = \sum_{\mathbf{m} \in S_\mathbf{m}} \left(\sum_{\substack{\{\mathbf{b}_1 \dots \mathbf{b}_{n_b}\} \subseteq \tilde{S}_\mathbf{b} \\ \mathrm{sgn} \hat{A}^\dagger_{\mathbf{b}_1} \dots \hat{A}^\dagger_{\mathbf{b}_{n_b}} |0\rangle = |\mathbf{m}\rangle}} \mathrm{sgn}\left(\sigma(\hat{A}^\dagger_{\mathbf{b}_1} \dots \hat{A}^\dagger_{\mathbf{b}_{n_b}})\right) \prod_{i=1}^{n_b} C_{\mathbf{b}_i} \right) |\mathbf{m}\rangle \tag{2.43}$$

where

$$S_{\mathbf{m}} = \left\{ \prod_{\hat{A}_k^\dagger \in T} \hat{A}_k^\dagger \left|0\right\rangle \middle| T \subseteq \tilde{S}_{\mathbf{b}} \right\}$$

For systems with an odd number of electrons, there must be at least one odd-electron creator. The anticommutation between these creators results in a determinant. Unlike the permanent, the determinant of a matrix with identical rows is zero. Therefore, there must be one odd-electron creator within a set of creation operators that construct **m**.

$$\left|\Psi\right\rangle = \sum_{\mathbf{m}\in S_{\mathbf{m}}} \left(\sum_{\substack{\{\mathbf{b}_1 \dots \mathbf{b}_{n_b}\} \subseteq \tilde{S}_{\mathbf{b}},\, \mathbf{f}\in\tilde{S}_{\mathbf{f}} \\ \mathrm{sgn}\hat{A}_{\mathbf{b}_1}^\dagger \dots \hat{A}_{\mathbf{b}_{n_b}}^\dagger \hat{A}_{\mathbf{f}}^\dagger \left|0\right\rangle = \left|\mathbf{m}\right\rangle}} \mathrm{sgn}\left(\sigma(\hat{A}_{\mathbf{b}_1}^\dagger \dots \hat{A}_{\mathbf{b}_{n_b}}^\dagger \hat{A}_{\mathbf{f}}^\dagger)\right) \left(\prod_{i=1}^{n_b} C_{\mathbf{b}_i}\right) C_{\mathbf{f}} \right) \left|\mathbf{m}\right\rangle \tag{2.44}$$

The derivation is provided in the Appendix 2.9.5.

In the case where only two-electron creation operators are used, this wavefunction reduces to the Antisymmetrized Geminal Power (AGP)[91], HF-Bogoliubov[122], or the BCS superconducting[123] wavefunction (Equation 2.45).

$$\begin{aligned} \left|\Psi\right\rangle &= \exp\left(\sum_{ij} c_{ij} a_i^\dagger a_j^\dagger\right) \left|0\right\rangle \\ &= \sum_{\mathbf{m}\in S_{\mathbf{m}}} \left(\sum_{\substack{\{\mathbf{m}_1 \dots \mathbf{m}_{N/2}\} \subseteq \tilde{S} \\ \mathrm{sgn} A_{\mathbf{m}_1}^\dagger \dots A_{\mathbf{m}_{N/2}}^\dagger \left|0\right\rangle = \left|\mathbf{m}\right\rangle}} \mathrm{sgn}(\sigma_{\mathbf{m}_1 \dots \mathbf{m}_{N/2}}) \prod_{k=1}^{N/2} C_{\mathbf{m}_k} \right) \left|\mathbf{m}\right\rangle \end{aligned} \tag{2.45}$$

Notice that this type of coupled-cluster wavefunction is *not* size consistent and that it breaks particle number symmetry (unless it is restored with a projection onto correct particle number).

2.5.2 TPS Variants

TPS with Quasiparticles

In the TPS wavefunction (Equation 2.22), each parameter in $(M_k)^{n_k}$ describes the correlation between a spatial orbital, k, and all of the other orbitals. In the APG wavefunction (Equation 2.7) and CC-motivated quasiparticle wavefunction (Equation 2.44), each parameter is associated with a cluster of spatial orbitals (quasiparticle). Then, we should be able to build a TPS-like wavefunction using creation operators of an arbitrary number of electrons, rather than the one-electron creation operators.

$$|\Psi\rangle = \sum_{\mathbf{m}\in S_{\mathbf{m}}} \left(\sum_{\substack{\{\mathbf{m}_1 \dots \mathbf{m}_n\} \subseteq \tilde{S} \\ \operatorname{sgn} A^\dagger_{\mathbf{m}_1} \dots A^\dagger_{\mathbf{m}_n} |0\rangle = |\mathbf{m}\rangle}} \operatorname{sgn}(\sigma_{\mathbf{m}_1 \dots \mathbf{m}_n}) \bigodot_{\mathbf{k}\in\tilde{S}} (M_{\mathbf{k}})^{\delta(\mathbf{k}, \{\mathbf{m}_1 \dots \mathbf{m}_n\})} \right) |\mathbf{m}\rangle \tag{2.46}$$

where

$$S_{\mathbf{m}} = \left\{ \prod_{\hat{A}^\dagger_{\mathbf{k}} \in T} \hat{A}^\dagger_{\mathbf{k}} |0\rangle \middle| T \subseteq \tilde{S} \right\}$$

$M_{\mathbf{k}}$ is a tensor that corresponds to the creation operator $\hat{A}^\dagger_{\mathbf{k}}$, and δ is a function that checks if the creator $\hat{A}^\dagger_{\mathbf{k}}$ is in a set of creators, $\{\mathbf{m}_1 \dots \mathbf{m}_n\}$.

$$\delta(\mathbf{k}, \{\mathbf{m}_1 \dots \mathbf{m}_n\}) = \begin{cases} 1 & \text{if } \mathbf{k} \in \{\mathbf{m}_1 \dots \mathbf{m}_n\} \\ 0 & \text{if } \mathbf{k} \notin \{\mathbf{m}_1 \dots \mathbf{m}_n\} \end{cases}$$

Then, $(M_{\mathbf{k}})^0$ and $(M_{\mathbf{k}})^1$ are tensors that correspond to the absence and presence of the creator $\hat{A}^\dagger_{\mathbf{k}}$, respectively.

TPS with Excitation Operators

Just as the CC wavefunction can be rebuilt with creation operators, we can rebuild the TPS wavefunction with excitation operators. Each tensor, $t_{\hat{E}^{\mathbf{a}}_{\mathbf{i}}}$, can be associated with $\hat{E}^{\mathbf{a}}_{\mathbf{i}}$, and has auxiliary indices that describe the correlation between operators.

$$|\Psi\rangle = \sum_{\hat{E}^{\mathbf{a}}_{\mathbf{i}} \in S_{\hat{E}}} \left(\sum_{\substack{\{\hat{E}^{\mathbf{a}_1}_{\mathbf{i}_1} \dots \hat{E}^{\mathbf{a}_n}_{\mathbf{i}_n}\} \subseteq \tilde{S}_{\hat{E}} \\ \operatorname{sgn} \prod_{k=1}^{n} \hat{E}^{\mathbf{a}_k}_{\mathbf{i}_k} = \hat{E}^{\mathbf{a}}_{\mathbf{i}}}} \operatorname{sgn}(\sigma_{\hat{E}^{\mathbf{a}_1}_{\mathbf{i}_1} \dots \hat{E}^{\mathbf{a}_n}_{\mathbf{i}_n}}) \bigodot_{\hat{E}_k \in \tilde{S}_{\hat{E}}} (t_{\hat{E}_k})^{\delta(\hat{E}_k, \{\hat{E}^{\mathbf{a}_1}_{\mathbf{i}_1} \dots \hat{E}^{\mathbf{a}_n}_{\mathbf{i}_n}\})} \right) \hat{E}^{\mathbf{a}}_{\mathbf{i}} |\Phi_{\text{HF}}\rangle \quad (2.47)$$

where

$$S_{\hat{E}} = \left\{ \prod_{\hat{E}_k \in T} \hat{E}_k \middle| T \subseteq \tilde{S}_{\hat{E}} \right\}$$

and δ describes the presence of an excitation operator, $\hat{E}_k$, in the given set, $\{\hat{E}^{\mathbf{a}_1}_{\mathbf{i}_1} \dots \hat{E}^{\mathbf{a}_n}_{\mathbf{i}_n}\}$

$$\delta(\hat{E}_k, \{\hat{E}^{\mathbf{a}_1}_{\mathbf{i}_1} \dots \hat{E}^{\mathbf{a}_n}_{\mathbf{i}_n}\}) = \begin{cases} 1 & \text{if } \hat{E}_k \in \{\hat{E}^{\mathbf{a}_1}_{\mathbf{i}_1} \dots \hat{E}^{\mathbf{a}_n}_{\mathbf{i}_n}\} \\ 0 & \text{if } \hat{E}_k \notin \{\hat{E}^{\mathbf{a}_1}_{\mathbf{i}_1} \dots \hat{E}^{\mathbf{a}_n}_{\mathbf{i}_n}\} \end{cases} \quad (2.48)$$

Comparing Equation 2.47 with Equation 2.20, the CC wavefunction can be considered a special case of this wavefunction, where the tensor $(t_{\hat{E}_k})^{\delta_k}$ is 1 if $\delta_k = 0$ and a variable scalar value if $\delta_k = 1$.

2.5.3 APG Generalized to Excitation Operators

Just as the TPS wavefunction can be built with excitation operators (Equation 2.47), we can rewrite the APG wavefunction with excitation operators.

$$|\Psi\rangle = \prod_{p=1}^{n} \left(\sum_{\hat{E}_k \in \tilde{S}_{\hat{E}}} t_{p;\hat{E}_k} \hat{E}_k \right) |\Phi_{\mathrm{HF}}\rangle \tag{2.49}$$

where n is the number of excitation operators that will be multiplied together. Unlike the CC wavefunction, which can generate all possible combinations of the excitation operators, this wavefunction can only account for the combinations of n excitation operators. The corresponding weight function in the FANCI notation is

$$|\Psi\rangle = \sum_{\hat{E}^{\mathbf{a}}_{\mathbf{i}} \in S_{\hat{E}}} \left(\sum_{\substack{\{\hat{E}^{\mathbf{a}_1}_{\mathbf{i}_1} \dots \hat{E}^{\mathbf{a}_n}_{\mathbf{i}_n}\} \subseteq \tilde{S}_{\hat{E}} \\ \mathrm{sgn} \prod_{k=1}^{n} \hat{E}_k = \hat{E}^{\mathbf{a}}_{\mathbf{i}}}} \mathrm{sgn}(\sigma_{\hat{E}^{\mathbf{a}_1}_{\mathbf{i}_1} \dots \hat{E}^{\mathbf{a}_n}_{\mathbf{i}_n}}) \begin{vmatrix} t_{1;\hat{E}^{\mathbf{a}_1}_{\mathbf{i}_1}} & \dots & t_{1;\hat{E}^{\mathbf{a}_n}_{\mathbf{i}_n}} \\ \vdots & \ddots & \vdots \\ t_{n;\hat{E}^{\mathbf{a}_1}_{\mathbf{i}_1}} & \dots & t_{n;\hat{E}^{\mathbf{a}_n}_{\mathbf{i}_n}} \end{vmatrix}^{+} \right) \hat{E}^{\mathbf{a}}_{\mathbf{i}} |\Phi_{\mathrm{HF}}\rangle \tag{2.50}$$

where

$$S_{\hat{E}} = \left\{ \prod_{\hat{E}_k \in T} \hat{E}_k \middle| T \subseteq \tilde{S}_{\hat{E}} \right\}$$

and $\mathrm{sgn}(\sigma_{\hat{E}^{\mathbf{a}_1}_{\mathbf{i}_1} \dots \hat{E}^{\mathbf{a}_n}_{\mathbf{i}_n}})$ is the signature of the permutation of the one-electron creation and annihilation operators from the product of excitation operators, $\hat{E}^{\mathbf{a}_1}_{\mathbf{i}_1} \cdots \hat{E}^{\mathbf{a}_n}_{\mathbf{i}_n}$, to the given excitation operator, i.e.

$$\hat{E}^{\mathbf{a}}_{\mathbf{i}} = \mathrm{sgn}(\sigma_{\hat{E}^{\mathbf{a}_1}_{\mathbf{i}_1} \dots \hat{E}^{\mathbf{a}_n}_{\mathbf{i}_n}}) \hat{E}^{\mathbf{a}_1}_{\mathbf{i}_1} \cdots \hat{E}^{\mathbf{a}_n}_{\mathbf{i}_n} \tag{2.51}$$

Other combinations of excitation operators can be included into Equation 2.49 via the summation.

$$|\Psi\rangle = \sum_{n \in P} \frac{1}{n!} \prod_{p=1}^{n} \left(\sum_{\hat{E}_k \in \tilde{S}_{\hat{E}}} t_{p;\hat{E}_k} \hat{E}_k \right) |\Phi_{\text{HF}}\rangle \tag{2.52}$$

where P is the allowed number of excitation operators that can be combined. Recall that removing an index (so that all the geminals are identical) in the traditional APG wavefunction form leads to the AGP wavefunction. Similarly, when one removes the index, p, from the excitation-based APG wavefunction form and allows all possible combinations of excitation operators, one obtains the CC wavefunction:

$$\begin{aligned} |\Psi_{\text{CC}}\rangle &= \sum_{n=0}^{\infty} \frac{1}{n!} \prod_{p=1}^{n} \left(\sum_{\hat{E}_k \in \tilde{S}_{\hat{E}}} t_{\hat{E}_k} \hat{E}_k \right) |\Phi_{\text{HF}}\rangle \\ &= \sum_{n=0}^{\infty} \frac{1}{n!} \left(\sum_{\hat{E}_k \in \tilde{S}_{\hat{E}}} t_{\hat{E}_k} \hat{E}_k \right)^n |\Phi_{\text{HF}}\rangle \\ &= \exp \left(\sum_{\hat{E}_k \in \tilde{S}_{\hat{E}}} t_{\hat{E}_k} \hat{E}_k \right) |\Phi_{\text{HF}}\rangle \end{aligned} \tag{2.53}$$

2.5.4 General Quasiparticle Wavefunctions

In the Equation 2.44, any creation operator can be used (within the sets $\tilde{S}_{\mathbf{b}}$ and $\tilde{S}_{\mathbf{f}}$) to construct a Slater determinant. Similarly, we can generalize the APG wavefunction to include all even-electron creation operators.

$$|\Psi\rangle = \prod_{p=1}^{n} \left(\sum_{\hat{A}_k^\dagger \in \tilde{S}_{\mathbf{b}}} C_{p;\mathbf{m}_k} \hat{A}_k^\dagger \right) |0\rangle \tag{2.54}$$

where n is the number of quasiparticles in the wavefunction (i.e. number of operators used to construct a Slater determinant), and $\tilde{S}_{\mathbf{b}}$ is a set of even-electron creation operators. In the FANCI formulation,

$$|\Psi\rangle = \sum_{\mathbf{m} \in S_{\mathbf{m}}} \left(\sum_{\substack{\{\mathbf{m}_1 \dots \mathbf{m}_n\} \subseteq \tilde{S}_{\mathbf{b}} \\ \mathrm{sgn} A^\dagger_{\mathbf{m}_1} \dots A^\dagger_{\mathbf{m}_n} |0\rangle = |\mathbf{m}\rangle}} \mathrm{sgn}(\sigma_{\mathbf{m}_1 \dots \mathbf{m}_n}) \begin{vmatrix} C_{1;\mathbf{m}_1} & \dots & C_{1;\mathbf{m}_n} \\ \vdots & \ddots & \vdots \\ C_{n;\mathbf{m}_1} & \dots & C_{n;\mathbf{m}_n} \end{vmatrix}^+ \right) |\mathbf{m}\rangle \tag{2.55}$$

where $\tilde{S}_{\mathbf{b}}$ is a set of allowed even-electron creation operators,

$$S_{\mathbf{m}} = \left\{ \prod_{\hat{A}^\dagger_k \in T} \hat{A}^\dagger_k |0\rangle \middle| T \subseteq \tilde{S}_{\mathbf{b}} \right\}$$

and $\mathrm{sgn}(\sigma_{\mathbf{m}_1 \dots \mathbf{m}_n})$ is the signature of the permutation needed to reorder the one-electron creation operators from the product of quasiparticle creation operators to the given Slater determinant.

$$\prod_{i \in \mathbf{m}} a^\dagger_i = \mathrm{sgn}(\sigma_{\mathbf{m}_1 \dots \mathbf{m}_n}) \prod_{k=1}^{n} \prod_{i \in \mathbf{m}_k} a^\dagger_i \tag{2.56}$$

Note that the zero-electron creation operator can be considered as an even-electron creation operator.

Similarly, the wavefunction constructed using only odd-electron creation operator can be expressed using a determinant.

$$|\Psi\rangle = \sum_{\mathbf{m} \in S_{\mathbf{m}}} \left(\sum_{\substack{\{\mathbf{m}_1 \dots \mathbf{m}_n\} \subseteq \tilde{S}_{\mathbf{f}} \\ \mathrm{sgn} A^\dagger_{\mathbf{m}_1} \dots A^\dagger_{\mathbf{m}_n} |0\rangle = |\mathbf{m}\rangle}} \mathrm{sgn}(\sigma_{\mathbf{m}_1 \dots \mathbf{m}_n}) \begin{vmatrix} C_{1;\mathbf{m}_1} & \dots & C_{1;\mathbf{m}_n} \\ \vdots & \ddots & \vdots \\ C_{n;\mathbf{m}_1} & \dots & C_{n;\mathbf{m}_n} \end{vmatrix}^- \right) |\mathbf{m}\rangle \tag{2.57}$$

where

$$S_{\mathbf{m}} = \left\{ \prod_{\hat{A}_k^\dagger \in T} \hat{A}_k^\dagger \left|0\right\rangle \middle| T \subseteq \tilde{S}_{\mathbf{f}} \right\}$$

and $\tilde{S}_{\mathbf{f}}$ is a set of selected odd-electron creation operators.

When both even and odd-electron creation operators are present in $\tilde{S}$, the interchange of creators commutes or anticommutes depending on the pair of creators, and additional structure is necessary to account for this behaviour. First, we distinguish between the set of even and odd orbitals with $\mathbf{b}_i$ and $\mathbf{f}_j$, respectively. A given Slater determinant $\mathbf{m}$ is constructed with $n = n_b + n_f$ creation operators, where n_b is the number of even-electron creators and n_f is the number of odd-electron creators. A permanent is needed to account for all possible ordering of the even-electron creators and the resulting commutations. A determinant is needed to account for all possible ordering of the odd-electron creators and the resulting anticommutations. Finally, the commutation between an even-electron and an odd-electron creator can be accounted for by a sum over all possible selections of the n_b even-electron creators from n positions, i.e. $\{i_1^b \dots i_{n_b}^b\} \subseteq \{\mathbf{b}_1 \dots \mathbf{b}_{n_b} \mathbf{f}_1 \dots \mathbf{f}_{n_f}\}$. The positions of the odd-electron creators are the remaining n_f positions, i.e. $\{i_1^f \dots i_{n_f}^f\} = \{\mathbf{b}_1 \dots \mathbf{b}_{n_b} \mathbf{f}_1 \dots \mathbf{f}_{n_f}\} \backslash \{i_1^b \dots i_{n_b}^b\}$. Then, we can construct the generalized quasiparticle wavefunction (Equation 2.54) in which any set of creators can be used.

$$|\Psi\rangle = \sum_{\mathbf{m}} \left(\sum_{\substack{\{\mathbf{b}_1 \dots \mathbf{b}_{n_b}\} \subseteq \tilde{S}_{\mathbf{b}},\, \{\mathbf{f}_1 \dots \mathbf{f}_{n_f}\} \subseteq \tilde{S}_{\mathbf{f}} \\ \operatorname{sgn} \hat{A}_{\mathbf{b}_1}^\dagger \dots \hat{A}_{\mathbf{b}_{n_b}}^\dagger \hat{A}_{\mathbf{f}_1}^\dagger \dots \hat{A}_{\mathbf{f}_{n_f}}^\dagger |0\rangle = |\mathbf{m}\rangle}} \operatorname{sgn}(\sigma_{\hat{A}_{\mathbf{b}_1}^\dagger \dots \hat{A}_{\mathbf{b}_{n_b}}^\dagger \hat{A}_{\mathbf{f}_1}^\dagger \dots \hat{A}_{\mathbf{f}_{n_f}}^\dagger}) \sum_{\substack{\{i_1^b \dots i_{n_b}^b\} \subseteq \{\mathbf{b}_1 \dots \mathbf{b}_{n_b}, \mathbf{f}_1 \dots \mathbf{f}_{n_f}\} \\ \{i_1^f \dots i_{n_f}^f\} = \{\mathbf{b}_1 \dots \mathbf{b}_{n_b}, \mathbf{f}_1 \dots \mathbf{f}_{n_f}\} \backslash \{i_1^b \dots i_{n_b}^b\}}} \begin{vmatrix} C_{1 i_1^b}^b & \dots & C_{1 i_{n_b}^b}^b \\ \vdots & \ddots & \vdots \\ C_{n_b i_1^b}^b & \dots & C_{n_b i_{n_b}^b}^b \end{vmatrix}^+ \begin{vmatrix} C_{1 i_1^f}^f & \dots & C_{1 i_{n_f}^f}^f \\ \vdots & \ddots & \vdots \\ C_{n_f i_1^f}^f & \dots & C_{n_f i_{n_f}^f}^f \end{vmatrix}^- \right) |\mathbf{m}\rangle \tag{2.58}$$

where

$$S_{\mathbf{m}} = \left\{ \prod_{\hat{A}_k^\dagger \in T} \hat{A}_k^\dagger \left|0\right\rangle \middle| T \subseteq \tilde{S}_{\mathbf{b}} \cup \tilde{S}_{\mathbf{f}} \right\}$$

$$\prod_{i \in \mathbf{m}} a_i^\dagger = \operatorname{sgn}(\sigma_{\hat{A}_{\mathbf{b}_1}^\dagger \dots \hat{A}_{\mathbf{b}_{n_b}}^\dagger \hat{A}_{\mathbf{f}_1}^\dagger \dots \hat{A}_{\mathbf{f}_{n_f}}^\dagger}) \prod_{k=1}^{n_b} A_{\mathbf{b}_k}^\dagger \prod_{l=1}^{n_f} A_{\mathbf{f}_l}^\dagger$$

and C is a $n \times \dim(\tilde{S}_{\mathbf{b}} \cup \tilde{S}_{\mathbf{f}})$ matrix. Note that the set of orbitals, $\mathbf{b}_i$ and $\mathbf{f}_j$, correspond to a column in C and thus are used as a column index. Here, the column indices are made explicit using the index i.

The derivation and more details are in the Appendix 2.9.6.

2.5.5 Changing Solvers

Often times, the algorithm for optimizing the parameters is synonymous with the wavefuncton ansatz. For example, DMRG is often associated with MPS wavefunctions, and Quantum Monte-Carlo (QMC) is often associated with FCI wavefunctions. Using different algorithms to optimize the parameters will change the cost and the reliability of the wavefunction. Certain algorithms can be applied to a wide range of wavefunction structures and systems; whereas specialized algorithms are cheaper, but are often limited to specific wavefunction structures and systems.

For example, the DMRG algorithm with the MPS wavefunction provides variational results and is effective in describing linear systems. The algorithm is specific to the MPS wavefunction, wherein the orbitals are ordered such that adjacent orbitals are more correlated than the rest. As a result, DMRG must be extended beyond MPS to more general TPS forms, but these algorithms do not seem to be as elegant or as computationally efficient. Solving the TPS wavefunction as a projected Schrödinger equation mitigates certain complications that are present in DMRG, such as ordering of the orbitals and generalizing a one-dimensional algorithm to multiple dimensions. It is true, however, that the tensor structures that are most efficiently optimized in a variational ansatz are, at least in all cases we have considered, the same as those that are most efficiently optimized using the projected Schrödinger equation.

2.5.6 Generalization

Above, we constructed new wavefunction structures by constructing the CC wavefunction with creation operators and the TPS and APG wavefunctions using excitation operators. Effectively, the parameters are changed from the contributions of creation operators to those of excitation operators, and vice versa. Since both operators originate from orbitals, the wavefunctions are size-consistent if f is selected appropriately, provided the orbitals are localized. In Equation 2.44, 2.55, 2.57, and 2.58, particle number symmetry is not conserved, which is not a problem since the wavefunction can be optimized and projected onto the appropriate particle number. Then, just as excitation operators change the state of a reference Slater determinant, products of an arbitrary number of creation and annihilation operators can be used to explore different particle numbers with respect to the reference. For example, the product of creators and annihilators, where the number of creators exceeds the number of annihilators, will ionize a Slater determinant. Expressing the wavefunction with respect to different operators will result in different sets of parameters, even if these operators make no reference to the Slater determinants or creation operators. For example, grid-based wavefunctions will have parameters for each point in space.

So far, we have seen weight functions, f, in various forms: the CC wavefunction (Equation 2.20) uses a cumulant; the TPS wavefunction (Equation 2.21) uses a tensor product; the even-electron quasiparticle wavefunction (Equation 2.57) uses a permanent; the odd-electron quasiparticle wavefunction (Equation 2.55) uses a determinant; and the generalized quasiparticle wavefunction (Equation 2.58) uses some mix of a permanent and determinant (an immanent). In Equations 2.25, 2.44, 2.46, and 2.58, creators of arbitrary number of electrons are combined in fairly complicated manner. Since creators have distinct commutative (and anticommutative) relations with one another, the corresponding f must be symmetric (and antisymmetric) with respect to interchange of different creation operators (or parameters). Grouping together all the terms that correspond to the same set of creation operators results in a sum over all combinations of products of parameters. These combinatorial variants of product functions seem to be useful for size-consistency since they are multiplicatively separable by construction. Therefore, we can construct novel wavefunctions with quasiparticle origins using different symmetric (or antisymmetric) polynomial functions, including but not limited to determinant, permanent, immanent[124], pfaffian [125], hafnian[126, 127], hyperdeterminant [128], multidimensional permanent[129], hyperpfaffian[130], hyperhafnian[131], and mixed discriminant[132]. Unfortunately, among all the aforementioned size-consistent combinatoric forms, only the determinant can be evaluated in polynomial time. If we disregard size-consistency and do not require any quasiparticle structure, then any function can be chosen. One alternative, however, is to use a ratio of a product of determinants as the overlap; this form retains size-consistency and computational feasibility, and generalizes a single Slater determinant (one determinant in the numerator, none in the denominator) and the APr2G wavefunction (a special case with one structure determinant in the denominator and the determinant of its element-wise square in the numerator). If the total number of determinants is odd, this wavefunction describes a normal fermionic wavefunction, while if the number of determinants is even, this wavefunction represents a bosonic seniority-zero structure.

A great deal of flexibility is available within the proposed wavefunction framework: the set of Slater determinants, S, can be any set of orthonormal (for convenience) Slater determinants; the parameters, $\vec{P}$, can be any set of numbers that describe the wavefunction or the operators with which it is built; the parameterizing function, f, can be any function that maps the parameters to a coefficient for a given Slater determinant. We hope to find the combination that effectively models the optimized coefficients of the FCI wavefunction (for accuracy) using the minimal number of parameters (for cost) for as many systems as possible (for generality).

2.6 Conclusion

In the proposed framework, any multideterminant wavefunction can be expressed with respect to the components S, $\vec{P}$, and f. A wavefunction can be characterized using only these components and different combinations will result in different wavefunction characteristics. Then, we can systematically develop new structures by simply finding novel combinations. While the proposed wavefunctions are not necessarily cheap to compute, the general ansatz does not need to be cheap to be effective. For example, the CC wavefunction with all orders of excitations, the TPS wavefunction with infinitely large tensors, and the quasiparticle wavefunctions with N electron quasiparticles are no less expensive than the FCI wavefunction. However, different wavefunction structures inspire different approximations and new algorithms that reduce the computational cost to a tractable level. The greatest advantage to this perspective is that it is pragmatic: approximations and algorithms that were restricted to one wavefunction can be generalized to others using the FANCI framework; and many different methods can be implemented computationally using a common framework. Though we will defer detailed discussions on dynamical correlation corrections and orbital optimization to future papers, these elaborations to FANCI are clearly possible and are performed using similar techniques to what one would use in traditional selected CI and CC methods.

2.7 Acknowledgements

The authors thank NSERC, Compute Canada, McMaster University, and the Canada Research Chairs for funding. RAMQ acknowledges financial support from the University of Florida in the form of a start-up grant.

2.8 References

(1) Boys, S. *Proceedings of the Royal Society of London A* **1950**, *200*, 542–554.

(2) Helgaker, T.; Jørgensen, P.; Olsen, J., *Modern electronic structure theory*; Wiley: Chichester, 2000.

(3) Helgaker, T.; Jørgensen, P.; Olsen, J. *Journal of Physical Chemistry* **1996**, *100*, 13213–13225.

(4) Raghavachari, K.; Anderson, J. *Journal of Physical Chemistry* **1996**, *100*, 12960–12973.

(5) Marti, K.; Reiher, M. *Physical Chemistry Chemical Physics* **2011**, *13*, 6750–6759.

(6) Helgaker, T.; Coriani, S.; Jørgensen, P.; Kristensen, K.; Olsen, J.; Ruud, K. *Chemical Reviews* **2012**, *112*, 543–631.

(7) Sherrill, C. *Journal of Chemical Physics* **2010**, *132*, 110902.

(8) Scuseria, G.; Jiménez-Hoyos, C.; Henderson, T.; Samanta, K.; Ellis, J. *The Journal of Chemical Physics* **2011**, *134*, 124108.

(9) Chan, G.; Sharma, S. *Annu. Rev. Phys. Chem.* **2011**, *62*, 465–81.

(10) Small, D.; Head-Gordon, M. *Physical Chemistry Chemical Physics* **2011**, *13*, 19285–19297.

(11) Bytautas, L.; Henderson, T.; Jiménez-Hoyos, C.; Ellis, J.; Scuseria, G. *Journal of Chemical Physics* **2011**, *135*.

(12) Bytautas, L.; Scuseria, G.; Ruedenberg, K. *Journal of Chemical Physics* **2015**, *143*, 094105.

(13) Van Raemdonck, M.; Alcoba, D.; Poelmans, W.; De Baerdemacker, S.; Torre, A.; Lain, L.; Massaccesi, G.; Van Neck, D.; Bultinck, P. *Journal of Chemical Physics* **2015**, *143*, 104106.

(14) Alcoba, D.; Torre, A.; Luis, L.; Oña, O.; Capuzzi, P.; Van Raemdonck, M.; Bultinck, P.; Van Neck, D. *Journal of Chemical Physics* **2014**, *141*.

(15) Alcoba, D.; Torre, A.; Luis, L.; Massaccesi, G.; Oña, O.; Ayers, P.; Van Raemdonck, M.; Bultinck, P.; Van Neck, D. *Theoretical Chemistry Accounts* **2016**, *135*, 153.

(16) Carbo, R.; Hernandez, J. *Chemical Physics Letters* **1977**, *47*, 85–91.

(17) Kollmar, C.; Hess, B. *Journal of Chemical Physics* **2003**, *119*, 4655–4661.

(18) Kollmar, C. *Journal of Chemical Physics* **2006**, *125*.

(19) Cook, D. *Molecular Physics* **1975**, *30*, 733–743.

(20) Veillard, A.; Clementi, E. *Theoretica Chimica Acta* **1967**, *7*, 134–143.

(21) Roothaan, C.; Detrich, J.; Hopper, D. *International Journal of Quantum Chemistry* **1979**, *S13*, 93–101.

(22) Weinhold, F.; Wilson Jr, E. *The Journal of Chemical Physics* **1967**, *46*, 2752–58.

(23) Pople, J.; Seeger, R.; Krishnan, R. *International Journal of Quantum Chemistry: Quantum Chemistry Symposium* **1977**, *11*, 149–163.

(24) Roos, B.; Taylor, P.; Siegbahn, P. *Chemical Physics* **1980**, *48*, 157–173.

(25) Olsen, J.; Roos, B. *The Journal of Chemical Physics* **1988**, *89*, 2185–92.

(26) Schmidt, M.; Gordon, M. *Annual Review of Physical Chemistry* **1998**, *49*, 233–266.

(27) Gallup, G., *Valence bond methods: Theory and applications*; Cambridge UP: Cambridge, 2002.

(28) Shaik, S.; Hiberty, P., *A chemist's guide to valence bond theory*; Wiley: Hoboken, 2008.

(29) Wu, W.; Su, P.; Shaik, S.; Hiberty, P. *Chemical Reviews* **2011**, *111*, 7557–7593.

(30) Goddard III, W.; Dunning Jr., T.; Hunt, W.; Hay, P. *Accounts of Chemical Research* **1973**, *6*, 368–376.

(31) McWeeny, R. *Proceedings of Royal Society A* **1954**, *223*, 63–79.

(32) Paldus, J.; Li, X. In *Advances in Chemical Physics*, Prigogine, I., Rice, S., Eds., 1999; Vol. 110, pp 1–175.

(33) Cizek, J. *Journal of Chemical Physics* **1966**, *45*, 4256–4266.

(34) Shavitt, I.; Bartlett, R., *Many-body methods in chemistry and physics: MBPT and coupled-cluster theory*; Cambridge: Cambridge, 2009.

(35) Bartlett, R.; Musiał, M. *Reviews of Modern Physics* **2007**, *79*, 291–352.

(36) Evangelista, F. A.; Chan, G. K. L.; Scuseria, G. E. *Journal of Chemical Physics* **2019**, *151*, 244112.

(37) Evangelista, F. A. *Journal of Chemical Physics* **2011**, *134*, 224102.

(38) Murg, V.; Verstraete, F.; Schneider, R.; Nagy, P.; Legeza, Ö. *Journal of Chemical Theory and Computation* **2015**, *11*, 1027–1036.

(39) Buerschaper, O.; Mombelli, J.; Christandl, M.; Aguado, M. *Journal of Mathematical Physics* **2013**, *54*, 012201.

(40) Nakatani, N.; Chan, G. *Journal of Chemical Physics* **2013**, *138*, 134113.

(41) Changlani, H.; Kinder, J.; Umrigar, C.; Chan, G. *Physical Review B* **2009**, *80*, 245116.

(42) Ornus, R. *Annals of Physics* **2014**, *349*, 117–58.

(43) Gunst, K.; Verstraete, F.; Wouter, S.; Legeza, Ö.; Van Neck, D. *Journal of Chemical Theory and Computation* **2018**, *14*, 2026–2033.

(44) Verstraete, F.; Cirac, J. I. **2004**.

(45) Vidal, G. *Physical Review Letters* **2007**, *99*, 220405.

(46) Murg, V.; Verstraete, F.; Legeza, Ö.; Noack, R. M. *Physical Review B* **2010**, *82*, 205105.

(47) Nakatani, N.; Chan, G. K. L. *Journal of Chemical Physics* **2013**, *138*, 134113–134113.

(48) Murg, V.; Verstraete, F.; Schneider, R.; Nagy, P.; Legeza, Ö. *Journal of Chemical Theory and Computation* **2015**, *11*, 1027–1036.

(49) Marti, K. H.; Bauer, B.; Reiher, M.; Troyer, M.; Verstraete, F. *New Journal of Physics* **2010**, *12*, 103008.

(50) Kovyrshin, A.; Reiher, M. *Journal of Chemical Physics* **2017**, *147*, 214111.

(51) Schollwöck, U. *Annals of Physics* **2011**, *326*, 96–192.

(52) Parks, J.; Parr, R. *Journal of Chemical Physics* **1958**, *28*, 335–345.

(53) Allen, T.; Shull, H. *Journal of Physical Chemistry* **1962**, *66*, 2281–2283.

(54) Surjan, P.; Szabados, Á.; Jeszenszki, P.; Zoboki, T. *Journal of Mathematical Chemistry* **2012**, *50*, 534–551.

(55) Chan, G.; Keselman, A.; Nakatani, N.; Li, Z.; White, S. *Journal of Chemical Physics* **2016**, *145*, 014102.

(56) Daul, S.; Ciofini, I.; Daul, C.; White, S. *International Journal of Quantum Chemistry* **2000**, *79*, 331–342.

(57) White, S.; Martin, R. *Journal of Chemical Physics* **1999**, *110*, 4127–4130.

(58) White, S. *Physical Review B* **1993**, *48*, 10345–10356.

(59) Marti, K.; Reiher, M. *Zeitschrift Fur Physikalische Chemie-International Journal of Research in Physical Chemistry & Chemical Physics* **2010**, *224*, 583–599.

(60) Chan, G. *Journal of Chemical Physics* **2004**, *120*, 3172–3178.

(61) Chan, G.; Head-Gordon, M. *Journal of Chemical Physics* **2002**, *116*, 4462–4476.

(62) Wouters, S.; Poelmans, W.; De Baerdemacker, S.; Ayers, P.; Van Neck, D. *Computer Physics Communications* **2015**, *191*, 235–237.

(63) Wouters, S.; Poelmans, W.; Ayers, P.; Van Neck, D. *Computer Physics Communications* **2014**, *185*, 1501–1514.

(64) Zgid, D.; Nooijen, M. *Journal of Chemical Physics* **2008**, *128*, 114116.

(65) Chan, G. K. L.; Dorando, J. J.; Hachmann, J.; Neuscamman, E.; Wang, H.; Yanai, T. In *Frontiers in Quantum Systems in Chemistry and Physics*, Wilson, S., Glout, P. J., Delgado-Barrio, G., Piecuch, P., Eds.; Progress in Theoretical Chemistry and Physics, Vol. 18; Springer: Dordrecht, 2008, pp 49–65.

(66) Hurley, A.; Lennard-Jones, J.; Pople, J. *A theory of paired-electrons in polyatomic molecules Proceedings of the Royal Society of London Series A* **1953**, *220*, 446–455.

(67) Parr, R.; Ellison, F.; Lykos, P. *Journal of Chemical Physics* **1956**, *24*, 1106.

(68) McWeeny, R.; Sutcliffe, B. *Proceedings of the Royal Society of London Series A* **1963**, *273*, 103–116.

(69) Surjan, P. In *Correlation and Localization*, Surjan, P., Ed., 1999, pp 63–88.

(70) Tecmer, P.; Boguslawski, K.; Johnson, P.; Limacher, P.; Chan, M.; Verstraelen, T.; Ayers, P. *Journal of Physical Chemistry A* **2014**, *118*, 9058–9068.

(71) Paldus, J.; Cizek, J.; Sengupta, S. *Journal of Chemical Physics* **1971**, *55*, 2452–2462.

(72) Paldus, J.; Sengupta, S.; Cizek, J. *Journal of Chemical Physics* **1972**, *57*, 652–666.

(73) Shull, H. *The Journal of Chemical Physics* **1959**, *30*, 1405–13.

(74) Kutzelnigg, W. *The Journal of Chemical Physics* **1964**, *40*, 3640–47.

(75) Rassolov, V. *Journal of Chemical Physics* **2002**, *117*, 5978–5987.

(76) Rassolov, V.; Xu, F.; Garashchuk, S. *Journal of Chemical Physics* **2004**, *120*, 10385–10394.

(77) Rassolov, V.; Xu, F. *Journal of Chemical Physics* **2007**, *126*, 234112.

(78) Cassam-Chenaï, P. *Journal of Chemical Physics* **2006**, *124*, 194109.

(79) Cassam-Chenaï, P.; Rassolov, V. *Chemical Physics Letters* **2010**, *487*, 147–152.

(80) Cassam-Chenaï, P.; Ilmane, A. *Journal of Mathematical Chemistry* **2012**, *50*, 652–667.

(81) Stein, T.; Henderson, T.; Scuseria, G. *Journal of Chemical Physics* **2014**, *140*, 214113.

(82) Henderson, T.; Scuseria, G.; Dukelsky, J.; Signoracci, A.; Duguet, T. *Physical Review C* **2014**, *89*, 054305.

(83) Henderson, T.; Bulik, I.; Stein, T.; Scuseria, G. *Journal of Chemical Physics* **2014**, *141*, 244104.

(84) Bulik, I.; Henderson, T.; Scuseria, G. *Journal of Chemical Theory and Computation* **2015**, *11*, 3171–3179.

(85) Cullen, J. *Chemical Physics* **1996**, *202*, 217–229.

(86) Miller, K.; Ruedenberg, K. *Journal of Chemical Physics* **1965**, *43*, S88–S90.

(87) Miller, K.; Ruedenberg, K. *Journal of Chemical Physics* **1968**, *48*, 3414–3443.

(88) Silver, D.; Mehler, E.; Ruedenberg, K. *Journal of Chemical Physics* **1970**, *52*, 1174–1180.

(89) Mehler, E.; Ruedenberg, K.; Silver, D. *Journal of Chemical Physics* **1970**, *52*, 1181–1205.

(90) Silver, D.; Ruedenberg, K.; Mehler, E. *Journal of Chemical Physics* **1970**, *52*, 1206–1227.

(91) Coleman, A. *Journal of Mathematical Physics* **1965**, *6*, 1425–1431.

(92) Coleman, A. *International Journal of Quantum Chemistry* **1997**, *63*, 23–30.

(93) Bajdich, M.; Drobný, G.; Wagner, L.; Schmidt, K. *Physical Review Letters* **2006**, *96*, 130201.

(94) Bajdich, M.; Mitas, L.; Wagner, L.; Schmidt, K. *Physical Review B* **2008**, *77*, 115112.

(95) Pernal, K. *Journal of Chemical Theory and Computation* **2014**, *10*, 4332–4341.

(96) Pastorczak, E.; Pernal, K. *Physical Chemistry Chemical Physics* **2015**, *17*, 8622–8626.

(97) Limacher, P.; Kim, T.; Ayers, P.; Johnson, P.; De Baerdemacker, S.; Van Neck, D. *Molecular Physics* **2014**, *112*, 853–862.

(98) Limacher, P. *Journal of Chemical Physics* **2016**, *145*, 194102.

(99) Johnson, P.; Limacher, P.; Kim, T.; Richer, M.; Miranda-Quintana, R.; Heidar-Zadeh, F.; Ayers, P.; Bultinck, P.; De Baerdemacker, S.; Van Neck, D. *Computational and Theoretical Chemistry* **2017**, *1116*, 207–219.

(100) Boguslawski, K.; Tecmer, P.; Ayers, P.; Bultinck, P.; De Baerdemacker, S.; Van Neck, D. *Physical Review B* **2014**, *89*, 201106.

(101) Boguslawski, K.; Tecmer, P.; Bultinck, P.; De Baerdemacker, S.; Van Neck, D.; Ayers, P. *Journal of Chemical Theory and Computation* **2014**, *10*, 4873–4882.

(102) Silver, D. *The Journal of Chemical Physics* **1969**, *50*, 5108–16.

(103) Limacher, P.; Ayers, P.; Johnson, P.; De Baerdemacker, S.; Van Neck, D.; Bultinck, P. *Journal of Chemical Theory and Computation* **2013**, *9*, 1394–1401.

(104) Johnson, P.; Ayers, P.; Limacher, P.; De Baerdemacker, S.; Van Neck, D.; Bultinck, P. *Computational and Theoretical Chemistry* **2013**, *1003*, 101–13.

(105) Slater, J. *Physical Review* **1928**, *32*, 339–348.

(106) Fock, V. *Zeitschrift für Physik* **1930**, *61*, 126–148.

(107) Löwdin, P. *Physical Review* **1955**, *97*, 1490–1508.

(108) Nesbet, R. *Proceedings of the Royal Society A* **1955**, *230*, 312–321.

(109) Purvis, G.; R.J., B. *The Journal of Chemical Physics* **1982**, *76*, 1910–18.

(110) Kutzelnigg, W.; Mukherjee, D. *Journal of Chemical Physics* **1997**, *107*, 432–449.

(111) Kutzelnigg, W. *Journal of Chemical Physics* **2006**, *125*.

(112) Sinha, D.; Maitra, R.; Mukherjee, D. *Computational and Theoretical Chemistry* **2013**, *1003*, 62–70.

(113) Crawford, T.; Schaefer, H. In *Reviews in Computational Chemistry*, Lipkowitz, K., Boyd, D., Eds., 2000; Vol. 14, pp 33–136.

(114) Becker, K.; Vojta, M. *Molecular Physics* **1998**, *94*, 217–223.

(115) Kutzelnigg, W. *Chemical Physics* **2012**, *401*, 119–124.

(116) Kutzelnigg, W.; Mukherjee, D. *Physical Review A* **2005**, *71*, 022502.

(117) Nooijen, M.; Shamasundar, K.; Mukherjee, D. *Molecular Physics* **2005**, *103*, 2277–2298.

(118) Qiu, Y.; Henderson, T. M.; Scuseria, G. E. *Journal of Chemical Physics* **2017**, *146*, 184105.

(119) Qiu, Y.; Henderson, T. M.; Scuseria, G. E. *Journal of Chemical Physics* **2016**, *145*, 111102.

(120) Jiménez-Hoyos, C. A.; Henderson, T. M.; Tsuchimochi, T.; Scuseria, G. E. *Journal of Chemical Physics* **2012**, *136*, 164109.

(121) Ostlund, S.; Rommer, S. *Physical Review Letters* **1995**, *75*, 3537–3540.

(122) Valatin, J. *Physical Review* **1961**, *122*, 1012.

(123) Piris, M.; Cruz, R. *International Journal of Quantum Chemistry* **1995**, *53*, 353–359.

(124) Littlewood, D.; A.R., R. *Philosophical Transactions of the Royal Society A* **1934**, *233*, 99–141.

(125) Halton, J. *Journal of Combinatorial Theory* **1966**, *1*, 224–232.

(126) Rudelson, M.; Samorodnitsky, A.; Zeitouni, O. *The Annals of Probability* **2016**, *44*, 2858–2888.

(127) Ishikawa, M.; Kawamuko, H.; Okada, S. *The Electronic Journal of Combintorics* **2005**, *12*, N9.

(128) Gelfand, I.; Kapranov, M.; Zelevinsky, A. *Advances in Mathematics* **1992**, *96*, 226–263.

(129) Taranenko, A. *Journal of Combinatorial Design* **2015**, *23*, 305–320.

(130) Barvinok, A. *Mathematical Programming* **1995**, *69*, 449–470.

(131) Redelmeier, D. Hyperpfaffian in Algebraic Combinatorics., Ph.D. Thesis, University of Waterloo, 2006.

(132) Gurvits, L.; Samorodnitsky, A. *Discrete Computational Geometry* **2002**, *27*, 531–550.

2.9 Appendix

2.9.1 HF

$$\begin{aligned}
|\Psi_{\mathrm{HF}}\rangle &= \prod_{i=1}^{N}\left(\sum_{j=1}^{2K} a_j^\dagger U_{ji}\right)|0\rangle \\
&= \left(\sum_{j_1=1}^{2K} a_{j_1}^\dagger U_{j_1 1}\right)\left(\sum_{j_2=1}^{2K} a_{j_2}^\dagger U_{j_2 2}\right)\dots\left(\sum_{j_N=1}^{2K} a_{j_N}^\dagger U_{j_N N}\right)|0\rangle \\
&= \sum_{j_1=1}^{2K}\sum_{j_2=1}^{2K}\cdots\sum_{j_N=1}^{2K} a_{j_1}^\dagger a_{j_2}^\dagger \dots a_{j_N}^\dagger U_{j_1 1} U_{j_2 2}\dots U_{j_N N}|0\rangle
\end{aligned} \tag{2.59}$$

We can group together the terms that result in the same Slater determinant, defined here by the set of creation operators, $\{a_{j_1}^\dagger a_{j_2}^\dagger \dots a_{j_N}^\dagger\}$. Since the order of operators only affects the sign of the Slater determinant, we can split the sum over all indices into a sum over the set of creation operators and a sum over the different orderings of the given set of creation operators.

$$\begin{aligned}
|\Psi_{\mathrm{HF}}\rangle &= \sum_{j_1<j_2\dots<j_N}^{2K}\sum_{\sigma\in S_N} U_{j_{\sigma(1)}1}U_{j_{\sigma(2)}2}\dots U_{j_{\sigma(N)}N} a_{j_{\sigma(1)}}^\dagger a_{j_{\sigma(2)}}^\dagger \dots a_{j_{\sigma(N)}}^\dagger |0\rangle \\
&= \sum_{j_1<j_2\dots<j_N}^{2K}\sum_{\sigma\in S_N} \mathrm{sgn}(\sigma) U_{j_{\sigma(1)}1}U_{j_{\sigma(2)}2}\dots U_{j_{\sigma(N)}N} a_{j_1}^\dagger a_{j_2}^\dagger \dots a_{j_N}^\dagger |0\rangle
\end{aligned} \tag{2.60}$$

where sgn(σ) is the signature of the permutation σ.

$$\mathrm{sgn}(\sigma) = \begin{cases} 1 & \text{if } \sigma \text{ is even} \\ -1 & \text{if } \sigma \text{ is odd} \end{cases} \tag{2.61}$$

Using the definition of the determinant,

$$\begin{aligned} |A|^- &= \sum_{\sigma \in S_N} \operatorname{sgn}(\sigma) \prod_{i=1}^{N} A_{i\sigma(i)} \\ &= \sum_{\sigma \in S_N} \operatorname{sgn}(\sigma) \prod_{i=1}^{N} A_{\sigma(i)i} \end{aligned} \tag{2.62}$$

we find

$$\begin{aligned} |\Psi_{\text{HF}}\rangle &= \sum_{j_1 < j_2 \cdots < j_N}^{K} \begin{vmatrix} U_{j_1 1} & U_{j_1 2} & \dots & U_{j_1 N} \\ U_{j_2 1} & U_{j_2 2} & \dots & U_{j_2 N} \\ \vdots & \vdots & \ddots & \vdots \\ U_{j_N 1} & U_{j_N 2} & \dots & U_{j_N N} \end{vmatrix}^- a^\dagger_{j_1} a^\dagger_{j_2} \dots a^\dagger_{j_N} |0\rangle \\ &= \sum_{\mathbf{m}} |U(\mathbf{m})|^- |\mathbf{m}\rangle \end{aligned} \tag{2.63}$$

where $\mathbf{m} = \{j_1, j_2, \dots, j_N\}$, $|\mathbf{m}\rangle = a^\dagger_{j_1} a^\dagger_{j_2} \dots a^\dagger_{j_N} |0\rangle$ where $j_1 < j_2 < \cdots < j_N$, and $U(\mathbf{m})$ is a submatrix of U composed of rows that correspond to $\mathbf{m}$.

2.9.2 APG

For convenience, let $P = \frac{N}{2}$.

$$\begin{aligned} |\Psi_{\text{APG}}\rangle &= \prod_{p=1}^{P} \sum_{ij}^{2K} C_{p;ij} a^\dagger_i a^\dagger_j |0\rangle \\ &= \left(\sum_{i_1 j_1}^{2K} a^\dagger_{i_1} a^\dagger_{j_1} C_{1;i_1 j_1} \right) \left(\sum_{i_2 j_2}^{2K} a^\dagger_{i_2} a^\dagger_{j_2} C_{2;i_2 j_2} \right) \dots \left(\sum_{i_P j_P}^{2K} a^\dagger_{i_P} a^\dagger_{j_P} C_{P;i_P j_P} \right) |0\rangle \\ &= \sum_{i_1 j_1}^{2K} \sum_{i_2 j_2}^{2K} \cdots \sum_{i_P j_P}^{2K} a^\dagger_{i_1} a^\dagger_{j_1} a^\dagger_{i_2} a^\dagger_{j_2} \dots a^\dagger_{i_P} a^\dagger_{j_P} C_{1;i_1 j_1} C_{2;i_2 j_2} \dots C_{P;i_P j_P} |0\rangle \end{aligned}$$

Just as in the HF derivation, we can group together the creation operators, $\{a^\dagger_{i_1}, a^\dagger_{j_1}, \dots, a^\dagger_{i_P}, a^\dagger_{j_P}\}$, that corresponds to the same Slater determinant. However, there are multiple ways to construct a Slater determinant from a set of *pairs* of creators, $\{a^\dagger_{i_1} a^\dagger_{j_1}, a^\dagger_{i_2} a^\dagger_{j_2}, \dots, a^\dagger_{i_P} a^\dagger_{j_P}\}$ (Note the placement of commas). Since the Slater determinant is expressed with respect to a specific ordering of the one-electron creation operators, the sign resulting from ordering the product of creator pairs to this specific ordering must be taken into account. Then, the sum over all indices can be split into a sum over each Slater determinant, a sum over different orbital pairs that construct the given Slater determinant, and a sum over the permutation of the orbital pair creation operators.

$$\begin{aligned}
|\Psi_{\mathrm{APG}}\rangle &= \sum_{\mathbf{m} \in S} \sum_{\{i_1 j_1, \dots, i_P j_P\} = \mathbf{m}} \sum_{\tau \in S_P} C_{1; i_{\tau(1)} j_{\tau(1)}} \dots C_{P; i_{\tau(P)} j_{\tau(P)}} a^\dagger_{i_{\tau(1)}} a^\dagger_{j_{\tau(1)}} \dots a^\dagger_{i_{\tau(P)}} a^\dagger_{j_{\tau(P)}} |0\rangle \\
&= \sum_{\mathbf{m} \in S} \sum_{\{i_1 j_1, \dots, i_P j_P\} = \mathbf{m}} \mathrm{sgn}\Big(\sigma(i_1 j_1, \dots, i_P j_P)\Big) \sum_{\tau \in S_P} C_{1; i_{\tau(1)} j_{\tau(1)}} \dots C_{P; i_{\tau(P)} j_{\tau(P)}} a^\dagger_{i_1} a^\dagger_{j_1} \dots a^\dagger_{i_P} a^\dagger_{j_P} |0\rangle
\end{aligned} \tag{2.64}$$

where σ is the permutation for ordering the creation operators to that of the Slater determinant and τ is the permutation of the given pairs of creation operators. Note that the pair of creation operators commute with one another. From the definition of a permanent,

$$\begin{aligned}
|A|^+ &= \sum_{\tau \in S_N} \prod_{i=1}^{N} A_{i\tau(i)} \\
&= \sum_{\tau \in S_N} \prod_{i=1}^{N} A_{\tau(i)i}
\end{aligned} \tag{2.65}$$

we find

$$\begin{aligned}
|\Psi_{\text{APG}}\rangle &= \sum_{\mathbf{m}\in S} \sum_{\{i_1,j_1,\dots,i_P,j_P\}=\mathbf{m}} \operatorname{sgn}\Big(\sigma(i_1,j_1,\dots,i_P,j_P)\Big) \begin{vmatrix} C_{1;i_1 j_1} & \dots & C_{1;i_P j_P} \\ \vdots & \ddots & \vdots \\ C_{P;i_1 j_1} & \dots & C_{P;i_P j_P} \end{vmatrix}^{+} a^\dagger_{i_1} a^\dagger_{j_1} \dots a^\dagger_{i_P} a^\dagger_{j_P} |0\rangle \\
&= \sum_{\mathbf{m}\in S} \sum_{\substack{\{\mathbf{m}_1 \dots \mathbf{m}_P\} \\ \operatorname{sgn} A^\dagger_{\mathbf{m}_1} \dots A^\dagger_{\mathbf{m}_P} |0\rangle = |\mathbf{m}\rangle}} \operatorname{sgn}\Big(\sigma(\mathbf{m}_1 \dots \mathbf{m}_P)\Big) |C(\mathbf{m}_1,\dots,\mathbf{m}_P)|^{+} |\mathbf{m}\rangle
\end{aligned} \tag{2.66}$$

where $\mathbf{m}_p = \{i_p, j_p\}$, $A^\dagger_{\mathbf{m}_p} = a^\dagger_{i_p} a^\dagger_{j_p}$, $|\mathbf{m}\rangle = a^\dagger_{i_1} a^\dagger_{j_1} \dots a^\dagger_{i_P} a^\dagger_{j_P} |0\rangle$, and $C(\mathbf{m}_1,\dots,\mathbf{m}_P)$ is a submatrix of C composed of columns that correspond to the orbital pairs, $\{\mathbf{m}_1 \dots \mathbf{m}_P\}$.

2.9.3 Product of Linear Combinations of Operators

Using the two examples above, we can generalize the approach to reformulating wavefunctions of the form

$$|\Psi\rangle = \prod_{i=1}^{n} \sum_j C_{ij} \hat{Q}_j |\theta\rangle \tag{2.67}$$

where $\hat{Q}_j$ is an operator, C_{ij} is a coefficient that corresponds to the index i and operator $\hat{Q}_j$, and θ is some vacuum/reference. Taking the same approach as above,

$$\begin{aligned}
|\Psi\rangle &= \prod_{i=1}^{n} \sum_j C_{ij} \hat{Q}_j |\theta\rangle \\
&= \left(\sum_{j_1} C_{1j_1} \hat{Q}_{j_1} \right) \dots \left(\sum_{j_n} C_{nj_n} \hat{Q}_{j_n} \right) |\theta\rangle \\
&= \sum_{j_1} \cdots \sum_{j_n} C_{1j_1} \dots C_{nj_n} \hat{Q}_{j_1} \dots \hat{Q}_{j_n} |\theta\rangle \\
&= \underbrace{\sum_{\mathbf{m}}}_{\substack{\text{sum over} \\ \text{Slater deter-} \\ \text{minants}}} \underbrace{\sum_{\{\hat{Q}_{j_1} \dots \hat{Q}_{j_n}\} \mapsto \mathbf{m}}}_{\substack{\text{sum over all} \\ \{\hat{Q}_{j_1} \dots \hat{Q}_{j_n}\} \\ \text{that produce} \\ \text{the given Slater} \\ \text{determinant}}} \underbrace{\operatorname{sgn}\Big(\sigma(\hat{Q}_{j_1} \dots \hat{Q}_{j_n})\Big)}_{\substack{\text{signature for ordering} \\ \text{the one-electron} \\ \text{operators}}} \underbrace{\sum_{\tau \in S_n} \operatorname{sgn}(\tau) C_{1j_{\tau(1)}} \dots C_{nj_{\tau(n)}}}_{\substack{\text{sum over the permutation} \\ \text{of the given operators}}} |\mathbf{m}\rangle
\end{aligned} \tag{2.68}$$

Depending on the commutation/anticommutation relationships between the elements of $\{\hat{Q}_{j_1} \dots \hat{Q}_{j_n}\}$, the $\operatorname{sgn}(\tau)$ is different.

If all operators are expressed with an even number of one-electron operators (which we will denote as even-electron operators), then

$$\operatorname{sgn}(\tau) = 1$$

and

$$\sum_{\tau \in S_n} \operatorname{sgn}(\tau) C_{1j_{\tau(1)}} \dots C_{nj_{\tau(n)}} = \begin{vmatrix} C_{1j_1} & \dots & C_{1j_n} \\ \vdots & \ddots & \vdots \\ C_{nj_1} & \dots & C_{nj_n} \end{vmatrix}^{+} \tag{2.69}$$

If all operators are expressed with an odd number of one-electron operators (which we will denote as odd-electron operators), then

$$\operatorname{sgn}(\tau) = \begin{cases} 1 & \text{if even} \\ -1 & \text{if odd} \end{cases}$$

and

$$\sum_{\tau \in S_n} \operatorname{sgn}(\tau) C_{1j_{\tau(1)}} \dots C_{nj_{\tau(n)}} = \begin{vmatrix} C_{1j_1} & \dots & C_{1j_n} \\ \vdots & \ddots & \vdots \\ C_{nj_1} & \dots & C_{nj_n} \end{vmatrix}^{-} \tag{2.70}$$

If the operators are a mix of even and odd-electron operators, then τ must be split into three components: τ_b for permutation of the positions of even-electron operators, τ_f for permutation of the positions of odd-electron operators, and τ_{bf} for remaining permutations that mix the positions of even-electron operators with the odd-electron operators.

$$\begin{aligned}
\tau &= \tau_b \tau_f \tau_{bf} \\
\mathrm{sgn}(\tau) &= \mathrm{sgn}(\tau_b)\mathrm{sgn}(\tau_f)\mathrm{sgn}(\tau_{bf}) \\
\mathrm{sgn}(\tau_b) &= 1 \\
\mathrm{sgn}(\tau_f) &= \begin{cases} 1 & \text{if even} \\ -1 & \text{if odd} \end{cases} \\
\mathrm{sgn}(\tau_{bf}) &= 1
\end{aligned} \tag{2.71}$$

Let there be n_b even-electron operators and n_f odd-electron operators.

$$n = n_b + n_f$$

We will assume (with no loss of generality) that the first n_b columns of the coefficients correspond to the even-electron and the rest to the odd-electron operators. Then τ_{bf} would correspond to swapping the first n_b columns with the rest of the columns. After the swapped columns are obtained, the τ_b will permute the first n_b columns and τ_f will permute the others.

$$\begin{aligned}
&\sum_{\tau \in S_n} \mathrm{sgn}(\tau) C_{1 j_{\tau(1)}} \dots C_{n j_{\tau(n)}} \\
&= \sum_{\tau_f \in S_{n_f}} \sum_{\tau_b \in S_{n_b}} \sum_{\tau_{bf} \in S_n \setminus S_{n_b} \setminus S_{n_f}} \mathrm{sgn}(\tau_b) \mathrm{sgn}(\tau_f) \mathrm{sgn}(\tau_{bf}) C_{1 j_{\tau_b \tau_{bf}(1)}} \dots C_{n_b j_{\tau_b \tau_{bf}(n_b)}} C_{(n_b+1) j_{\tau_f \tau_{bf}(n_b+1)}} \dots C_{n j_{\tau_f \tau_{bf}(n)}} \\
&= \sum_{\tau_{bf} \in S_n \setminus S_{n_b} \setminus S_{n_f}} \mathrm{sgn}(\tau_{bf}) \sum_{\tau_b \in S_{n_b}} \mathrm{sgn}(\tau_b) C_{1 j_{\tau_b \tau_{bf}(1)}} \dots C_{n_b j_{\tau_b \tau_{bf}(n_b)}} \sum_{\tau_f \in S_{n_f}} \mathrm{sgn}(\tau_f) C_{(n_b+1) j_{\tau_f \tau_{bf}(n_b+1)}} \dots C_{n j_{\tau_f \tau_{bf}(n)}} \\
&= \sum_{\tau_{bf} \in S_n \setminus S_{n_b} \setminus S_{n_f}} \begin{vmatrix} C_{1 j_{\tau_{bf}(1)}} & \dots & C_{1 j_{\tau_{bf}(n_b)}} \\ \vdots & \ddots & \vdots \\ C_{n_b j_{\tau_{bf}(1)}} & \dots & C_{n_b j_{\tau_{bf}(n_b)}} \end{vmatrix}^{+} \begin{vmatrix} C_{(n_b+1) j_{\tau_{bf}(n_b+1)}} & \dots & C_{(n_b+1) j_{\tau_{bf}(n)}} \\ \vdots & \ddots & \vdots \\ C_{n j_{\tau_{bf}(n_b+1)}} & \dots & C_{n j_{\tau_{bf}(n)}} \end{vmatrix}^{-}
\end{aligned} \tag{2.72}$$

where S_n, S_{n_b}, and S_{n_f} are sets of permutations of all indices, first n_b indices, and the remaining n_f indices, respectively. Then $S_n \setminus S_{n_b} \setminus S_{n_f}$ is the set of permutations of the first n_b columns with the rest.

In other words, τ_{bf} accounts for the different ways in which the columns of the coefficient matrix can be split into two, one for τ_b and other for τ_f. Since only the first n_b rows are needed for the sum over τ_b and the remaining rows for the sum over τ_f, we will denote C^b as the submatrix composed of the first n_b rows of C, and C^f as the submatrix composed of the remaining rows. Note that C^b has dimensions $n_b \times n$ and C^f has dimensions $n_f \times n$. Then, we can rewrite the sum over τ_{bf} to be somewhat more transparent:

$$\sum_{\tau \in S_n} \mathrm{sgn}(\tau) C_{1 j_{\tau(1)}} \dots C_{n j_{\tau(n)}} = \sum_{\substack{\{i^b_1 \dots i^b_{n_f}\} \subseteq \{j_1 \dots j_n\} \\ \{i^f_1 \dots i^f_{n_b}\} = \{j_1 \dots j_n\} \setminus \{i^b_1 \dots i^b_{n_f}\}}} \begin{vmatrix} C^b_{1 i^b_1} & \dots & C^b_{1 i^b_{n_b}} \\ \vdots & \ddots & \vdots \\ C^b_{n_b i^b_1} & \dots & C^b_{n_b i^b_{n_b}} \end{vmatrix}^+ \begin{vmatrix} C^f_{1 i^f_1} & \dots & C^f_{1 i^f_{n_f}} \\ \vdots & \ddots & \vdots \\ C^f_{n_f i^f_1} & \dots & C^f_{n_f i^f_{n_f}} \end{vmatrix}^- \tag{2.73}$$

If the given operators do not commute or anticommute with one another, then an explicit sum through every permutation might be necessary, along with the signature of each permutation, unless there is a specialized structure that simplifies this sum.

2.9.4 CC

$$\begin{aligned} |\Psi_{\mathrm{CC}}\rangle &= \exp\left(\sum_{\mathbf{ia}} t^{\mathbf{a}}_{\mathbf{i}} \hat{E}^{\mathbf{a}}_{\mathbf{i}} \right) |\Phi_{\mathrm{HF}}\rangle \\ &= \sum_{n=0}^{\infty} \frac{1}{n!} \left(\sum_{\mathbf{ia}} t^{\mathbf{a}}_{\mathbf{i}} \hat{E}^{\mathbf{a}}_{\mathbf{i}} \right)^n |\Phi_{\mathrm{HF}}\rangle \\ &= \sum_{n=0}^{\infty} \frac{1}{n!} \prod_{k=1}^{n} \sum_{\mathbf{ia}} t^{\mathbf{a}}_{\mathbf{i}} \hat{E}^{\mathbf{a}}_{\mathbf{i}} |\Phi_{\mathrm{HF}}\rangle \end{aligned} \tag{2.74}$$

For each n, we can treat $\prod_{k=1}^{n} \sum_{\mathbf{ia}} t^{\mathbf{a}}_{\mathbf{i}} \hat{E}^{\mathbf{a}}_{\mathbf{i}}$ with the approach from Equation 2.68. Then, the operators, $\{\hat{Q}_{j_1} \dots \hat{Q}_{j_n}\}$, are excitation operators, $\{\hat{E}^{\mathbf{a}_1}_{\mathbf{i}_1} \dots \hat{E}^{\mathbf{a}_n}_{\mathbf{i}_n}\}$, and n controls the number of operators used in the second summation of Equation 2.68. Since they all commute with one another, $\mathrm{sgn}(\tau) = 1$ which means that the final sum of Equation 2.68

will be a permanent (Equation 2.18).

$$
\begin{aligned}
\prod_{k=1}^{n} \sum_{\mathbf{ia}} t_{\mathbf{i}}^{\mathbf{a}} \hat{E}_{\mathbf{i}}^{\mathbf{a}} &= \sum_{\mathbf{m}} \sum_{\substack{\{\hat{E}_{\mathbf{i}_1}^{\mathbf{a}_1} \dots \hat{E}_{\mathbf{i}_n}^{\mathbf{a}_n}\} \subseteq \tilde{S}_{\hat{E}} \\ \operatorname{sgn} \prod_{k=1}^{n} \hat{E}_{\mathbf{i}_k}^{\mathbf{a}_k} |\Phi_{\mathrm{HF}}\rangle = |\mathbf{m}\rangle}} \operatorname{sgn}\Big(\sigma(\hat{E}_{\mathbf{i}_1}^{\mathbf{a}_1} \dots \hat{E}_{\mathbf{i}_n}^{\mathbf{a}_n})\Big) \sum_{\tau \in S_n} t_{\mathbf{i}_{\tau(1)}}^{\mathbf{a}_{\tau(1)}} \dots t_{\mathbf{i}_{\tau(n)}}^{\mathbf{a}_{\tau(n)}} |\mathbf{m}\rangle \\
&= \sum_{\mathbf{m}} \sum_{\substack{\{\hat{E}_{\mathbf{i}_1}^{\mathbf{a}_1} \dots \hat{E}_{\mathbf{i}_n}^{\mathbf{a}_n}\} \subseteq \tilde{S}_{\hat{E}} \\ \operatorname{sgn} \prod_{k=1}^{n} \hat{E}_{\mathbf{i}_k}^{\mathbf{a}_k} |\Phi_{\mathrm{HF}}\rangle = |\mathbf{m}\rangle}} \operatorname{sgn}\Big(\sigma(\hat{E}_{\mathbf{i}_1}^{\mathbf{a}_1} \dots \hat{E}_{\mathbf{i}_n}^{\mathbf{a}_n})\Big) n! \prod_{k=1}^{n} t_{\mathbf{i}_k}^{\mathbf{a}_k} |\mathbf{m}\rangle
\end{aligned}
\tag{2.75}
$$

where $\tilde{S}_{\hat{E}}$ is the set of all excitation operators used in the wavefunction. Substituting into the Equation 2.74, we get

$$
\begin{aligned}
|\Psi_{\mathrm{CC}}\rangle &= \sum_{n=0}^{\infty} \frac{1}{n!} \sum_{\mathbf{m}} \sum_{\substack{\{\hat{E}_{\mathbf{i}_1}^{\mathbf{a}_1} \dots \hat{E}_{\mathbf{i}_n}^{\mathbf{a}_n}\} \subseteq \tilde{S}_{\hat{E}} \\ \operatorname{sgn} \prod_{k=1}^{n} \hat{E}_{\mathbf{i}_k}^{\mathbf{a}_k} |\Phi_{\mathrm{HF}}\rangle = |\mathbf{m}\rangle}} \operatorname{sgn}\Big(\sigma(\hat{E}_{\mathbf{i}_1}^{\mathbf{a}_1} \dots \hat{E}_{\mathbf{i}_n}^{\mathbf{a}_n})\Big) n! \prod_{k=1}^{n} t_{\mathbf{i}_k}^{\mathbf{a}_k} |\mathbf{m}\rangle \\
&= \sum_{\mathbf{m}} \sum_{n=0}^{\infty} \sum_{\substack{\{\hat{E}_{\mathbf{i}_1}^{\mathbf{a}_1} \dots \hat{E}_{\mathbf{i}_n}^{\mathbf{a}_n}\} \subseteq \tilde{S}_{\hat{E}} \\ \operatorname{sgn} \prod_{k=1}^{n} \hat{E}_{\mathbf{i}_k}^{\mathbf{a}_k} |\Phi_{\mathrm{HF}}\rangle = |\mathbf{m}\rangle}} \operatorname{sgn}\Big(\sigma(\hat{E}_{\mathbf{i}_1}^{\mathbf{a}_1} \dots \hat{E}_{\mathbf{i}_n}^{\mathbf{a}_n})\Big) \prod_{k=1}^{n} t_{\mathbf{i}_k}^{\mathbf{a}_k} |\mathbf{m}\rangle \\
&= \sum_{\mathbf{m}} \sum_{\substack{\{\hat{E}_{\mathbf{i}_1}^{\mathbf{a}_1} \dots \hat{E}_{\mathbf{i}_n}^{\mathbf{a}_n} | n \in \mathbb{N}\} \subseteq \tilde{S}_{\hat{E}} \\ \operatorname{sgn} \prod_{k=1}^{n} \hat{E}_{\mathbf{i}_k}^{\mathbf{a}_k} |\Phi_{\mathrm{HF}}\rangle = |\mathbf{m}\rangle}} \operatorname{sgn}\Big(\sigma(\hat{E}_{\mathbf{i}_1}^{\mathbf{a}_1} \dots \hat{E}_{\mathbf{i}_n}^{\mathbf{a}_n})\Big) \prod_{k=1}^{n} t_{\mathbf{i}_k}^{\mathbf{a}_k} |\mathbf{m}\rangle
\end{aligned}
\tag{2.76}
$$

where the second sum is the sum over all possible combinations of excitations that would produce the given Slater determinant from the reference. Though it is unnecessary, $n \in \mathbb{N}$ was included to specify that the subset is no longer constrained to be of a certain size by an external variable, n. In the rest of the article, this notation will be dropped.

2.9.5 CC with Creation Operators

Let $\mathbf{b}_i$ be a set of an even number of orbitals, $\mathbf{f}_i$ be a set of an odd number of orbitals, $\hat{A}^\dagger_{\mathbf{b}_i}$ and $\hat{A}^\dagger_{\mathbf{f}_i}$ be the creation operators that create these orbitals, and $C_{\mathbf{b}_i}$ and $C_{\mathbf{f}_i}$ are the coefficient for the associated creation operators. Then, the CC wavefunction using creation operators will be

$$|\Psi_{\mathrm{CC}}\rangle = \exp\left(\sum_{\mathbf{b}_i} C_{\mathbf{b}_i}\hat{A}^\dagger_{\mathbf{b}_i} + \sum_{\mathbf{f}_i} C_{\mathbf{f}_i}\hat{A}^\dagger_{\mathbf{f}_i}\right)|0\rangle \tag{2.77}$$

Taking the same approach as Equation 2.68, we have operators, $\{\hat{Q}_{j_1}\ldots\hat{Q}_{j_n}\}$, that are creation operators, $\{\hat{A}_{\mathbf{b}_1}\ldots\hat{A}_{\mathbf{b}_{n_b}}\hat{A}_{\mathbf{f}_1}\ldots\hat{A}_{\mathbf{f}_{n_f}}\}$, and the sum over the permutation is given by Equation 2.73. Following Equation 2.68 and 2.73,

$$\begin{aligned}
|\Psi\rangle &= \sum_{\mathbf{m}}\sum_{n=0}^{\infty}\frac{1}{n!}\sum_{\substack{\{\mathbf{b}_1\ldots\mathbf{b}_{n_b}\}\subseteq\tilde{S}_{\mathbf{b}},\,\{\mathbf{f}_1\ldots\mathbf{f}_{n_f}\}\subseteq\tilde{S}_{\mathbf{f}}\\ \mathrm{sgn}\hat{A}^\dagger_{\mathbf{b}_1}\ldots\hat{A}^\dagger_{\mathbf{b}_{n_b}}\hat{A}^\dagger_{\mathbf{f}_1}\ldots\hat{A}^\dagger_{\mathbf{f}_{n_f}}|0\rangle=|\mathbf{m}\rangle}} \mathrm{sgn}\left(\sigma(\hat{A}^\dagger_{\mathbf{b}_1}\ldots\hat{A}^\dagger_{\mathbf{b}_{n_b}}\hat{A}^\dagger_{\mathbf{f}_1}\ldots\hat{A}^\dagger_{\mathbf{f}_{n_f}})\right)\sum_{\substack{\{i^b_1\ldots i^b_{n_b}\}\subseteq\{1\ldots n\}\\ \{i^f_1\ldots i^f_{n_f}\}=\{1\ldots n\}\backslash\{i^b_1\ldots i^b_{n_b}\}}}\begin{vmatrix}C_{\mathbf{b}_1}&\ldots&C_{\mathbf{b}_{n_b}}\\ \vdots&\ddots&\vdots\\ C_{\mathbf{b}_1}&\ldots&C_{\mathbf{b}_{n_b}}\end{vmatrix}^+\begin{vmatrix}C_{\mathbf{f}_1}&\ldots&C_{\mathbf{f}_{n_f}}\\ \vdots&\ddots&\vdots\\ C_{\mathbf{f}_1}&\ldots&C_{\mathbf{f}_{n_f}}\end{vmatrix}^-|\mathbf{m}\rangle\\
&= \sum_{\mathbf{m}}\sum_{\substack{\{\mathbf{b}_1\ldots\mathbf{b}_{n_b}\}\subseteq\tilde{S}_{\mathbf{b}},\,\{\mathbf{f}_1\ldots\mathbf{f}_{n_f}\}\subseteq\tilde{S}_{\mathbf{f}}\\ \mathrm{sgn}\hat{A}^\dagger_{\mathbf{b}_1}\ldots\hat{A}^\dagger_{\mathbf{b}_{n_b}}\hat{A}^\dagger_{\mathbf{f}_1}\ldots\hat{A}^\dagger_{\mathbf{f}_{n_f}}|0\rangle=|\mathbf{m}\rangle}} \mathrm{sgn}\left(\sigma(\hat{A}^\dagger_{\mathbf{b}_1}\ldots\hat{A}^\dagger_{\mathbf{b}_{n_b}}\hat{A}^\dagger_{\mathbf{f}_1}\ldots\hat{A}^\dagger_{\mathbf{f}_{n_f}})\right)\frac{1}{n!}\binom{n}{n_b}\begin{vmatrix}C_{\mathbf{b}_1}&\ldots&C_{\mathbf{b}_{n_b}}\\ \vdots&\ddots&\vdots\\ C_{\mathbf{b}_1}&\ldots&C_{\mathbf{b}_{n_b}}\end{vmatrix}^+\begin{vmatrix}C_{\mathbf{f}_1}&\ldots&C_{\mathbf{f}_{n_f}}\\ \vdots&\ddots&\vdots\\ C_{\mathbf{f}_1}&\ldots&C_{\mathbf{f}_{n_f}}\end{vmatrix}^-|\mathbf{m}\rangle\\
&= \sum_{\mathbf{m}}\sum_{\substack{\{\mathbf{b}_1\ldots\mathbf{b}_{n_b}\}\subseteq\tilde{S}_{\mathbf{b}},\,\{\mathbf{f}_1\ldots\mathbf{f}_{n_f}\}\subseteq\tilde{S}_{\mathbf{f}}\\ \mathrm{sgn}\hat{A}^\dagger_{\mathbf{b}_1}\ldots\hat{A}^\dagger_{\mathbf{b}_{n_b}}\hat{A}^\dagger_{\mathbf{f}_1}\ldots\hat{A}^\dagger_{\mathbf{f}_{n_f}}|0\rangle=|\mathbf{m}\rangle}} \mathrm{sgn}\left(\sigma(\hat{A}^\dagger_{\mathbf{b}_1}\ldots\hat{A}^\dagger_{\mathbf{b}_{n_b}}\hat{A}^\dagger_{\mathbf{f}_1}\ldots\hat{A}^\dagger_{\mathbf{f}_{n_f}})\right)\frac{1}{n_b!n_f!}\begin{vmatrix}C_{\mathbf{b}_1}&\ldots&C_{\mathbf{b}_{n_b}}\\ \vdots&\ddots&\vdots\\ C_{\mathbf{b}_1}&\ldots&C_{\mathbf{b}_{n_b}}\end{vmatrix}^+\begin{vmatrix}C_{\mathbf{f}_1}&\ldots&C_{\mathbf{f}_{n_f}}\\ \vdots&\ddots&\vdots\\ C_{\mathbf{f}_1}&\ldots&C_{\mathbf{f}_{n_f}}\end{vmatrix}^-|\mathbf{m}\rangle
\end{aligned} \tag{2.78}$$

Note that the $n \in \mathbb{N}$ was omitted and it is implied that the size of the subset is not constrained by an external variable.

Since the determinant is zero when any two rows are equivalent, $n_f \leq 1$. In contrast, we can take the permanent of a matrix with repeating rows:

$$\begin{aligned} \begin{vmatrix} a_1 & \dots & a_n \\ \vdots & \ddots & \vdots \\ a_1 & \dots & a_n \end{vmatrix}^+ &= \sum_{\sigma \in S_n} \prod_{i=1} a_i \\ &= n! \prod_{i=1} a_i \end{aligned} \tag{2.79}$$

Therefore, we get

$$|\Psi\rangle = \sum_{\mathbf{m}} \left(\begin{array}{c} \sum\limits_{\substack{\{\mathbf{b}_1 \dots \mathbf{b}_{n_b}\} \subseteq \tilde{S}_{\mathbf{b}},\, \mathbf{f} \in \tilde{S}_{\mathbf{f}} \\ \mathrm{sgn} \hat{A}^\dagger_{\mathbf{b}_1} \dots \hat{A}^\dagger_{\mathbf{b}_{n_b}} \hat{A}^\dagger_{\mathbf{f}} |0\rangle = |\mathbf{m}\rangle}} \mathrm{sgn}\left(\sigma(\hat{A}^\dagger_{\mathbf{b}_1} \dots \hat{A}^\dagger_{\mathbf{b}_{n_b}} \hat{A}^\dagger_{\mathbf{f}}) \right) \left(\prod_{i=1}^{n_b} C_{\mathbf{b}_i} \right) C_{\mathbf{f}} \\ + \sum\limits_{\substack{\{\mathbf{b}_1 \dots \mathbf{b}_{n_b}\} \subseteq \tilde{S}_{\mathbf{b}} \\ \mathrm{sgn} \hat{A}^\dagger_{\mathbf{b}_1} \dots \hat{A}^\dagger_{\mathbf{b}_{n_b}} |0\rangle = |\mathbf{m}\rangle}} \mathrm{sgn}\left(\sigma(\hat{A}^\dagger_{\mathbf{b}_1} \dots \hat{A}^\dagger_{\mathbf{b}_{n_b}}) \right) \prod_{i=1}^{n_b} C_{\mathbf{b}_i} \end{array} \right) |\mathbf{m}\rangle \tag{2.80}$$

Since the sum over $\{\mathbf{b}_1 \dots \mathbf{b}_{n_b} \mathbf{f}\}$ only occurs when $\mathbf{m}$ has an odd number of electrons and the sum over $\{\mathbf{b}_1 \dots \mathbf{b}_{n_b}\}$ only occurs when $\mathbf{m}$ has an even number of electrons, we can separate these two cases if the desired number of electrons in the system, N, is either even or odd.

$$|\Psi\rangle = \begin{cases} \sum_{\mathbf{m}} \sum_{\substack{\{\mathbf{b}_1 \dots \mathbf{b}_{n_b}\} \subseteq \tilde{S}_{\mathbf{b}},\, \mathbf{f} \in \tilde{S}_{\mathbf{f}} \\ \mathrm{sgn} \hat{A}^\dagger_{\mathbf{b}_1} \dots \hat{A}^\dagger_{\mathbf{b}_{n_b}} \hat{A}^\dagger_{\mathbf{f}} |0\rangle = |\mathbf{m}\rangle}} \mathrm{sgn}\left(\sigma(\hat{A}^\dagger_{\mathbf{b}_1} \dots \hat{A}^\dagger_{\mathbf{b}_{n_b}} \hat{A}^\dagger_{\mathbf{f}})\right) \left(\prod_{i=1}^{n_b} C_{\mathbf{b}_i}\right) C_{\mathbf{f}} |\mathbf{m}\rangle & \text{if } N \text{ is odd} \\ \sum_{\mathbf{m}} \sum_{\substack{\{\mathbf{b}_1 \dots \mathbf{b}_{n_b}\} \subseteq \tilde{S}_{\mathbf{b}} \\ \mathrm{sgn} \hat{A}^\dagger_{\mathbf{b}_1} \dots \hat{A}^\dagger_{\mathbf{b}_{n_b}} |0\rangle = |\mathbf{m}\rangle}} \mathrm{sgn}\left(\sigma(\hat{A}^\dagger_{\mathbf{b}_1} \dots \hat{A}^\dagger_{\mathbf{b}_{n_b}})\right) \prod_{i=1}^{n_b} C_{\mathbf{b}_i} |\mathbf{m}\rangle & \text{if } N \text{ is even} \end{cases} \tag{2.81}$$

2.9.6 Generalized Quasiparticle

Just as in Section 2.9.5, let $\mathbf{b}_i$ be a set of an even number of orbitals, $\mathbf{f}_i$ be a set of an odd number of orbitals, $\hat{A}^\dagger_{\mathbf{b}_i}$ and $\hat{A}^\dagger_{\mathbf{f}_i}$ be the creation operators that create the associated orbitals. We can construct a quasiparticle as a linear combination of these creation operators. The desired wavefunction is a product of these quasiparticles

$$|\Psi\rangle = \prod_{p=1}^{n} \left(\sum_{\mathbf{b}_i} C_{p;\mathbf{b}_i} \hat{A}^\dagger_{\mathbf{b}_i} + \sum_{\mathbf{f}_i} C_{p;\mathbf{f}_i} \hat{A}^\dagger_{\mathbf{f}_i} \right) |0\rangle \tag{2.82}$$

where $C_{p;\mathbf{b}_i}$ and $C_{p;\mathbf{f}_i}$ are coefficients of the creation operators $\hat{A}^\dagger_{\mathbf{b}_i}$ and $\hat{A}^\dagger_{\mathbf{f}_i}$ in the construction of the p^{th} quasiparticle.

Taking the same approach as Equation 2.68, we have operators, $\{\hat{Q}_{j_1} \dots \hat{Q}_{j_n}\}$, that are creation operators, $\{\hat{A}_{\mathbf{b}_1} \dots \hat{A}_{\mathbf{b}_{n_b}} \hat{A}_{\mathbf{f}_1} \dots \hat{A}_{\mathbf{f}_{n_f}}\}$, and the sum over the permutation is given by Equation 2.73.

$$|\Psi\rangle = \sum_{\mathbf{m}} \sum_{\substack{\{\mathbf{b}_1 \dots \mathbf{b}_{n_b}\} \subseteq \tilde{S}_{\mathbf{b}},\, \{\mathbf{f}_1 \dots \mathbf{f}_{n_f}\} \subseteq \tilde{S}_{\mathbf{f}} \\ \mathrm{sgn} \hat{A}^\dagger_{\mathbf{b}_1} \dots \hat{A}^\dagger_{\mathbf{b}_{n_b}} \hat{A}^\dagger_{\mathbf{f}_1} \dots \hat{A}^\dagger_{\mathbf{f}_{n_f}} |0\rangle = |\mathbf{m}\rangle}} \mathrm{sgn}\left(\sigma(\hat{A}^\dagger_{\mathbf{b}_1} \dots \hat{A}^\dagger_{\mathbf{b}_{n_b}} \hat{A}^\dagger_{\mathbf{f}_1} \dots \hat{A}^\dagger_{\mathbf{f}_{n_f}})\right) \sum_{\substack{\{i^b_1 \dots i^b_{n_b}\} \subseteq \{\mathbf{b}_1 \dots \mathbf{b}_{n_b} \mathbf{f}_1 \dots \mathbf{f}_{n_f}\} \\ \{i^f_1 \dots i^f_{n_f}\} = \{\mathbf{b}_1 \dots \mathbf{b}_{n_b} \mathbf{f}_1 \dots \mathbf{f}_{n_f}\} \setminus \{i^b_1 \dots i^b_{n_b}\}}} \begin{vmatrix} C^b_{1 i^b_1} & \dots & C^b_{1 i^b_{n_b}} \\ \vdots & \ddots & \vdots \\ C^b_{n_b i^b_1} & \dots & C^b_{n_b i^b_{n_b}} \end{vmatrix}^+ \begin{vmatrix} C^f_{1 i^f_1} & \dots & C^f_{1 i^f_{n_f}} \\ \vdots & \ddots & \vdots \\ C^f_{n_f i^f_1} & \dots & C^f_{n_f i^f_{n_f}} \end{vmatrix}^- |\mathbf{m}\rangle \tag{2.83}$$

where C^b is the submatrix of C composed of the first n_b rows and C^f is composed of the remaining rows. It was assumed that the first n_b columns belong to the even-electron operators and the remaining columns to the odd-electron operators.

Chapter 3

Fanpy

`Fanpy`: A Python Library for Prototyping Multideterminant Methods in *Ab Initio* Quantum Chemistry

Taewon D. Kim[1] Michael Richer[1] Gabriela Sánchez-Díaz[1]

Farnaz Heidar-Zadeh[2] Toon Verstraelen[3]

Ramón Alain Miranda-Quintana[4] Paul W. Ayers[1]

September 26, 2020

[1]Department of Chemistry and Chemical Biology, McMaster University, Hamilton, Ontario, L8S-4L8, Canada

[2]Department of Chemistry, Queen's University, Kingston, Ontario, K7L-3N6, Canada

[3]Center for Molecular Modeling (CMM), Ghent University, Technologiepark 46, B-9052, Zwijnaarde, Belgium

[4]Department of Chemistry, University of Florida, Gainesville, FL 32603, USA

Abstract

`Fanpy` is a free and open-source Python library for developing and testing

multideterminant wavefunctions and related *ab initio* methods in electronic structure theory. The main use of `Fanpy` is to quickly prototype new methods by decreasing the barrier required to translate the mathematical conception of a new wavefunction ansätze to a working implementation. `Fanpy` uses the framework of our recently introduced Flexible Ansatz for N-electron Configuration Interaction (FANCI), where multideterminant wavefunctions are represented by their overlaps with Slater determinants of orthonormal spin-orbitals. In the simplest case, then, a new wavefunction ansatz can be implemented by simply writing a function for evaluating its overlap with an arbitrary Slater determinant. `Fanpy` is modular in both implementation and theory: the wavefunction model, the system's Hamiltonian, and the objective function to be optimized are all independent modules. This modular structure makes it easy for users to mix and match different methods and for developers to quickly try new ideas. `Fanpy` is written purely in Python with standard dependencies, making it accessible for most operating systems; it adheres to principles of modern software development, including comprehensive documentation, extensive testing, and continuous integration and delivery protocols. This article is considered to be the official release notes for the `Fanpy` library.

3.1 What is `Fanpy`?

`Fanpy` is a free and open-source Python 3 library for *ab initio* electronic structure calculations. The key innovation is the adoption of the Flexible Ansatz for N-electron Configuration Interaction (FANCI) mathematical framework[1]. By adopting this framework, *ab initio* electronic structure methods are represented as a collection of four parts, each of which is represented by an independent module of `Fanpy`: (a) the (multideterminant) wavefunction model (b) the system Hamiltonian, as represented by its one- and two-electron integrals (c) an equation (or system of equations) to solve that is equivalent to the Schrödinger equation, (d) an algorithm for optimizing the objective function(s). Section 3.4 details the features of each module, though the main advantage of `Fanpy` is that new methods can be implemented easily. For more information on the FANCI framework, consult FANCI article[1].

Although the FANCI framework allows overlaps with any convenient set of reference states (e.g., Richardson eigenfunctions) to be used, in `Fanpy` the aforementioned components are expressed explicitly in terms of Slater determinants and the wavefunction ansätze of interest are multideterminant wavefunctions with parameterized coefficients,

$$|\Psi_{\text{FANCI}}\rangle = \sum_{\mathbf{m}\in S_{\mathbf{m}}} f(\mathbf{m}, \vec{P})\,|\mathbf{m}\rangle \tag{3.1}$$

where

$$|\mathbf{m}\rangle = |m_1 m_2 \dots m_{N-1} m_N\rangle = a^\dagger_{m_1} a^\dagger_{m_2} \dots a^\dagger_{m_{N-1}} a^\dagger_{m_N}\,|\rangle \tag{3.2}$$

denotes a Slater determinant, $S_{\mathbf{m}}$ is the set of Slater determinants included in the wavefunction, and f is a function that determines the coefficient of each Slater determinant, $\mathbf{m}$, using the parameters, $\vec{P}$. Note that f is simply the overlap of the parameterized wavefunction with the Slater determinant,

$$f(\mathbf{m}, \vec{P}) = \langle \mathbf{m} | \Psi_{\text{FANCI}} \rangle \tag{3.3}$$

Similarly, the Hamiltonian is expressed in terms of its integrals with Slater determinants (CI matrix elements),

$$\langle \mathbf{m} | \hat{H} | \mathbf{n} \rangle = \langle \mathbf{m} | \sum_{ij} h_{ij} a_i^\dagger a_j + \frac{1}{2} \sum_{ijkl} g_{ijkl} a_i^\dagger a_j^\dagger a_l a_k | \mathbf{n} \rangle \tag{3.4}$$

The objective functions supported by `Fanpy` combine the overlaps and the CI matrix elements to approximate the Schrödinger equation.

3.2 About `Fanpy`

`Fanpy` source code is maintained on GitHub; see `https://github.com/quantumelephant/fanpy`, and its documentation is hosted on Read the Docs; see `https://fanpy.readthedocs.io/en/latest/index.html`. We strive to ensure that the `Fanpy` source code and its associated website are comprehensively documented, including useful tests, scripts, and examples. As that documentation is maintained with the software, providing detailed (and eventually outdated) release notes here seems unwise. Instead, we will briefly list the distinguishing features and key capabilities of `Fanpy` in section 3.4 to establish its philosophy and framework, and exemplify them in section 3.5.

3.3 Why Fanpy?

Many quantum chemistry packages enable computations using multideterminant methods. Most (e.g. Gaussian[2] and MolPro[3]) are closed-source, making it nearly impossible to develop new methods without special permission. Even for packages whose source code is available, the code is often monolithic, making it difficult to implement new fundamental methods without thoroughly understanding nearly the entire code base, from the top down. Moreover, the low-level code is often highly optimized, abstracts away critical components of *ab initio* methods, and not intended to be subsequently read or modified. Such code can, and does, remain unchanged for decades, has little documentation, and rarely follows modern software development principles. Though some packages try to address this issue, the development of post-HF methods remains difficult. For example, Psi4Numpy[4] is a collection of Python scripts and Jupyter notebooks that implement several post-HF methods using Psi4 to generate necessary inputs, such as CI matrices and one- and two-electron integrals. Though these scripts can serve as excellent tools for learning about standard quantum chemistry methods from the ground up or when implementing some types of minor embellishments of standard methods, developing a novel method (e.g. a new wavefunction ansatz) requires all related processes to be implemented in Psi4 (e.g. optimization algorithm and action of the Hamiltonian on the model wavefunction), which requires a thorough understanding of the Psi4 package as a whole[5].

Due to the difficulty of developing new methods in these legacy codes, we developed our own Helpful Open-Source Research TOol for N-electron systems (HORTON)[6].

The first two versions of HORTON were monolithic, and wavefunction models that were simple on paper were difficult to implement. HORTON 3 is strictly modular, with separate modules for input/output, numerical integration, Gaussian-basis-set evaluation and integrals, geometry optimization, and self-consistent field calculations. `Fanpy` is the correlated wavefunction module of HORTON 3.

`Fanpy` was developed as a development tool for new correlated-wavefunction methods; the goal is to help researchers quickly implement and test their ideas. Towards this goal, `Fanpy` is designed to be modular and general. Its modularity helps to isolate and minimize the necessary part of the code that needs to be understood and extended upon. For example, implementing a new wavefunction ansatz uses only the wavefunction module, and does not require explicit consideration of how the Hamiltonian will act upon that wavefunction nor of how the orbitals and parameters within the wavefunction will be optimized (unless desired). The modules of `Fanpy` are designed to be as general as possible, so that features from one module are compatible with features from the other modules. The compatibility between the modules ensures that any developed method (e.g. a wavefunction ansatz) can be used in conjunction with the other methods (e.g. orbital optimization, model Hamiltonians, the projected Schrödinger equation, etc). We provide comprehensive documentation and examples to further aid the development of new methods in `Fanpy`.

3.4 Features of `Fanpy`

We display various features of `Fanpy` by discussing each module and their intended purposes:

- The wavefunction module is developed in accordance with the FANCI framework[1]. In the FANCI framework, the wavefunction is entirely represented by its overlaps with Slater determinants built from orthonormal orbitals. Similarly, each wavefunction in `Fanpy` is defined by its parameters and a function that returns an overlap for the given Slater determinant. The overlap can be defined within a class structure, templated using abstract base class or as a standalone function. The following wavefunctions have already been implemented: configuration interaction (CI) with single and double excitations (CISD)[7]; doubly-occupied configuration interaction (DOCI)[8–11]; full CI[12]; and selected CI wavefunctions with a user-specified set of Slater determinants; antisymmetrized products of geminals (APG)[13–23]; antisymmetrized products of geminals with disjoint orbital sets (APsetG)[24]; antisymmetrized product of interacting geminals (APIG)[24–52]; antisymmetric product of 1-reference-orbital interacting geminals (AP1roG; equivalent to pair-coupled-cluster doubles)[53]; antisymmetric product of rank-two interacting geminals (APr2G)[54]; determinant ratio wavefunctions[1]; antisymmetrized products of tetrets (4-electron wavefunctions)[1]; matrix product states (MPS)[55]; neural network wavefunctions; coupled-cluster (CC) with arbitrary excitations (including, but not limited to, CCSD, CCSDT, and CC with seniority-specific excitations)[1, 56–61], and geminal coupled-cluster wavefunctions[32–34, 36, 53]. We also support these wavefunctions with nonorthogonal orbitals, and linear combinations of any of the aforementioned wavefunctions.

- The Hamiltonian module contains Hamiltonians commonly used in electronic structure theory. Similar to the wavefunctions, each Hamiltonian in `Fanpy` is defined by its representation in orbital basis set (i.e. one- and two-electron integrals) and a function that returns the integral of the Hamiltonian with respect to the given Slater determinants. The following Hamiltonians have already been implemented: the electronic Hamiltonian in the restricted, unrestricted, and generalized basis; the seniority-zero electronic Hamiltonian[62]; and the Fock operator in the restricted basis. In addition, the Pariser-Parr-Pople[63–66], Hubbard[66, 67], Hückel[66, 68], Ising[66, 69, 70], Heisenberg[66, 70, 71], and Richardson[72, 73] model Hamiltonians are available as restricted electronic Hamiltonians through the ModelHamiltonian GitHub repository[74]. Orbital optimization is available if a function returning the derivative with respect to orbital rotation parameters is provided. At the moment, only restricted electronic Hamiltonians in the restricted basis support orbital optimization.

- The objective module is responsible for combining the wavefunction and the Hamiltonian to form an equation or a system of equations that represents the Schrödinger equation. In `Fanpy`, the objective function can be the variational optimization of the expectation value of the energy[75–79], the projected Schrödinger equation[24, 36, 49], or a local energy expression to be sampled (as in variational quantum Monte Carlo)[80–85].

- The solver module contains algorithms that optimize/solve the equations from the objective module. It supports optimizers from `SciPy`[86], which includes constrained/unconstrained local/global optimizers for multivariate scalar functions (i.e. energy) and algorithms for solving nonlinear least-squares problems and for finding roots of a system of nonlinear equations (i.e., projected Schrödinger equation). For CI wavefunctions, we also support a brute-force eigenvalue decomposition of the CI matrix. In addition, `Fanpy` interfaces to several algorithms for derivative-free global optimization problems including the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) algorithm from `pycma`[87] and algorithms using decision trees and Bayesian optimization from `scikit-optimize`[88]. At the moment, no in-house optimization algorithms specialized for electronic structure theory problems are included. However, `Fanpy`'s modular design makes it easier to develop sophisticated domain-specific optimization algorithms. The objective module provides high-level control over the parameters involved in the optimization (e.g., active and frozen parameters) and can be changed dynamically throughout the optimization process. These parameters can be saved as a checkpoint throughout the optimization. (The default is to checkpoint at each function evaluation.) Furthermore, the objective module provides flexibility to add additional parameters (e.g., model hyperparameters) and to add nonlinear constraints to the Projected Schrödinger equation.

- The tool module provides various utility functions used throughout the `Fanpy` package. Though some tools have specialized uses, the tools for manipulating and generating Slater determinants are used frequently throughout `Fanpy`. These tools are essential when developing methods in `Fanpy` because Slater determinants are the common language of the independent modules. The `slater` module provides functions for manipulating Slater determinants and converting them from one form to another. Within `Fanpy`, Slater determinants are represented as a binary number, where the positions of 1's are the indices of the occupied spin-orbitals (ordered alpha orbitals first then beta orbitals). The `slater` module can, for example, provide the occupied spin orbital indices from the given Slater determinant. The `sdlist` module provides easy ways to generate Slater determinants of the desired characteristics (e.g. order of excitation from ground-state, spin, seniority). This module is frequently used to construct the projection space by which the objective function is evaluated. In addition, the `tool` module provides wrappers to other modules of `HORTON` and other quantum chemistry software, including `Gaussian`[2], `PySCF`[89], and `Psi4`[5]. These programs can then be used to generate one- and two-electron integrals for `Fanpy`.

3.5 Examples

For the most updated documentation and examples on how to use `Fanpy`, please refer to the `Fanpy` website. Here, we showcase several ways `Fanpy` can be used and incorporated into various workflows. Please note that these examples are based on version 1.0 of `Fanpy`, and the user might need to modify them if using a future major release of the `Fanpy` library. Within minor and bug-fix releases, backward compatibility is guaranteed.

Running a calculation: A calculation in `Fanpy` can be run by creating and executing a Python script and by running its command-line tool, `fanpy_run_calc`. It is recommended to create and execute a Python script because it provides a transparent record of the calculation and the full range of `Fanpy`'s features. For assistance in creating a Python script, `Fanpy` provides a command-line tool, `fanpy_make_script`. This tool creates a script from the given specifications, which can then be modified if a desired feature is not available in the tool.

The following example of a Python script runs an AP1roG calculation for oxygen molecule in a double zeta basis set:

```python
import numpy as np
from fanpy.wfn.geminal.ap1rog import AP1roG
from fanpy.ham.restricted_chemical import RestrictedMolecularHamiltonian
from fanpy.eqn.projected import ProjectedSchrodinger
from fanpy.solver.system import least_squares
from fanpy.tools.sd_list import sd_list

nelec = 16

# Hamiltonian
oneint = np.load(’one_oxygen.npy’)
twoint = np.load(’two_oxygen.npy’)
ham = RestrictedMolecularHamiltonian(oneint, twoint)

# Wavefunction
wfn = AP1roG(nelec, ham.nspin)

# Projection space of first and second order excitation
pspace = sd_list(nelec, ham.nspin, exc_orders=[2], seniority=0)

# Projected Schrodinger Equation
eqns = ProjectedSchrodinger(wfn, ham, pspace=pspace)

# Solve
results = least_squares(eqns)
print(’AP1roG electronic energy (Hartree):’, results[’energy’])
```

Since `Fanpy` targets post-HF methods, the orbitals (and the corresponding system specific information) must be provided in the form of one- and two-electron integrals. The one- and two-electron integrals must be provided as two- and four-dimensional numpy arrays, respectively, whose indices are in the same order as the integrals in the physicists' notation. To generate the integrals from a single-determinant calculation, `Fanpy` provides wrappers for `HORTON`, `PySCF`, and `Psi4` via the `fanpy.tools.wrapper` module. The Gaussian `.fchk` file can be converted into `.npy` file using the HORTON wrapper.

Implementing a wavefunction: New wavefunctions can be implemented in `Fanpy` by making a subclass of the wavefunction base class or by providing the overlap function to a utility function. The subclass requires the method `get_overlap` to be defined.

Following is an example of an implementation of the nonorthogonal Slater determinant expressed in terms of orthogonal Slater determinants (Equation 3.5):

$$\begin{aligned} |\Psi\rangle &= \prod_{i=1}^{N} \sum_{j}^{2K} C_{ij} a_j^\dagger \left|\theta\right\rangle \\ &= \sum_{\mathbf{m}} |C(\mathbf{m})|^- \left|\mathbf{m}\right\rangle \end{aligned} \tag{3.5}$$

```python
import numpy as np
from fanpy.wfn.base import BaseWavefunction
from fanpy.tools.slater import occ_indices

class NonorthogonalSlaterDeterminant(BaseWavefunction):
    def get_overlap(self, sd, deriv=None):
        # get indices of the occupied spin orbitals
        occs = occ_indices(sd)
        # reshape the parameters
        # NOTE: parameters are stored as a one-dimensional array by default
        params = self.params.reshape(self.nelec, self.nspin)
        # compute the overlap
        if deriv is None:
            return np.linalg.det(params[:, occs])

        # compute the derivative of the overlap
        output = np.zeros(params.shape)
        for deriv_row in range(self.nelec):
            for j, deriv_col in enumerate(occs):
                # compute the sign associated with Laplace formula
                sign = (-1)**(deriv_row + j)
                # get rows and columns with the appropriate row/column removed
                row_inds = np.arange(self.nelec)
                row_inds = row_inds[row_inds != deriv_row]
                col_inds = occs[occs != deriv_col]
                # compute minors (determinant after removing row and column)
                minor = np.linalg.det(params[row_inds[:, None], col_inds[None, :]])
                output[deriv_row, deriv_col] = sign * minor
        # derivative is returned as a flattened array
        # deriv contains the indices of the parameters with respect to which
        # the overlap is derivatized
        return output.ravel()[deriv]
```

The method `get_overlap` returns the overlap of the given Slater determinant when `deriv=None` and returns its gradient with respect to the parameters specified by `deriv` otherwise (`deriv` is a one-dimensional numpy array of parameter indices). For details on the API of `get_overlap`, consult the documentation. Unlike the wavefunctions already implemented in `Fanpy`, this wavefunction does not have default initial parameters, which means that it must be supplied when instantiating the wavefunction.

```python
from fanpy.tools.slater import ground, occ_indices

# get indices of the HF ground state
ground_indices = occ_indices(ground(16, 36))
# initial parameters (only contain the occupied orbitals in HF ground state)
hf_params = np.zeros((16, 36))
hf_params[np.arange(16), ground_indices] = 1

wfn = NonorthogonalSlaterDeterminant(16, 36)
```

```
# assign parameters
wfn.assign_params(hf_params)
```

Here, the initial parameters are set to the ground-state (orthogonal) Slater determinant.

Alternatively, the wavefunction can be constructed using the utility function: `wfn_factory`.

```
import numpy as np
from fanpy.wfn.utils import wfn_factory

def olp(sd, params):
    occs = occ_indices(sd)
    # NOTE: Since the only information available come from the arguments sd and
    # params, additional information that would otherwise be stored as
    # instance's attributes and properties must be explicitly defined
    nelc = 16
    nspin = 36
    # reshape the parameters
    params = params.reshape(nelec, nspin)
    return np.linalg.det(params[:, occs])

def olp_deriv(sd, params):
    occs = occ_indices(sd)
    # Hardcode essential information
    nelc = 16
    nspin = 36
    # reshape the parameters
    params = params.reshape(nelec, nspin)
    # same as above except replace self.nelec with nelec
    # ...
    # NOTE: the overlap is derivatized with respect to all wavefunction
    # parameters unlike above
    return output.ravel()


wfn = wfn_factory(olp, olp_deriv, 16, 36, hf_params)
# third argument is the number of electrons
# fourth argument is the number of spin orbitals
# fifth argument is the initial parameters
```

It is recommended to implement wavefunctions using the class structure because it helps make the code cleaner by limiting repetitions and makes the code easier to unit test by breaking the code into smaller pieces. For a quick and dirty implementation, however, the utility function may be easier.

Implementing a Hamiltonian: Similar to the wavefunction, new Hamiltonians can be implemented in `Fanpy` by making a subclass of the Hamiltonian base class or by passing a function that evaluates the integrals to a utility function. In addition to the general Hamiltonian base class, `Fanpy` provides base classes according to the type of orbitals used in the Hamiltonian: restricted, unrestricted, and generalized. The subclass to the orbital specific Hamiltonian base class requires the method `integrate_sd_sd` to be defined.

Following is an example of an implementation of the Hückel Hamiltonian[66] (Equation 3.6) in restricted orbitals:

$$\begin{aligned}
\hat{H} &= \sum_{ij} \sum_{\sigma} h_{ij} a^{\dagger}_{i\sigma} a_{j\sigma} \\
h_{ij} &= \begin{cases} \alpha_i & \text{if } i = j \\ \beta_{ij} & \text{if spatial orbitals } i \text{ and } j \text{ belong to atoms that participate in a bond} \\ 0 & \text{else} \end{cases}
\end{aligned} \tag{3.6}$$

```python
from fanpy.ham.base import BaseHamiltonian
from fanpy.tools import slater

class HuckelHamiltonian(BaseHamiltonian):
    def __init__(self, one_int):
        # NOTE: provided integrals correspond to spatial orbitals
        self.one_int = one_int

    @property
    def nspin(self):
        return self.one_int.shape[0] * 2

    def integrate_sd_wfn(self, wfn, sd, wfn_deriv=None, ham_deriv=None):
        # use the default method except only the first order excitations are used
        return super().integrate_sd_wfn(
            wfn, sd, wfn_deriv=wfn_deriv, ham_deriv=ham_deriv, orders=(1,)
        )

    def integrate_sd_sd(self, sd1, sd2, deriv=None, components=False):
        # get the difference of the Slater determinants (i.e. which orbitals are
        # occupied in one determinant but not in the other)
        diff_sd1, diff_sd2 = slater.diff_orbs(sd1, sd2)
        # derivative not supported here
        if deriv:
            raise NotImplementedError
        # If order of excitation between the two Slater determinants is two or greater
        if len(diff_sd1) >= 2 or len(diff_sd2) >= 2:
            return 0.0
        # If two Slater determinants do not have the same number of electrons
        if len(diff_sd1) != len(diff_sd2):
            return 0.0
        # If two Slater determinants are the same
        if len(diff_sd1) == 0:
            # get the indices of the spatial orbitals that correspond to the
            # occupied spin orbitals
            shared_alpha_sd, shared_beta_sd = slater.split_spin(
                slater.shared_sd(sd1, sd2), self.nspatial
            )
            shared_alpha = slater.occ_indices(shared_alpha_sd)
            shared_beta = slater.occ_indices(shared_beta_sd)
            # sum over the occupied orbitals
            output = np.sum(self.one_int[shared_alpha, shared_alpha])
            output += np.sum(self.one_int[shared_beta, shared_beta])
            return output
        # If two Slater determinants are different by one-electron excitation
        # get indices of the spatial orbitals
        spatial_ind1 = slater.spatial_index(diff_sd1[0], self.nspatial)
        spatial_ind2 = slater.spatial_index(diff_sd2[0], self.nspatial)
        return self.one_int[spatial_ind1, spatial_ind2]
```

Though it is not necessary, the subclass defines `integrate_sd_wfn` to specify that the Hamiltonian only contains one-body operators. By default, `integrate_sd_wfn` assumes that the Hamiltonian contains one- and two-body operators.

Alternatively, the hamiltonian can be constructed using the utility function

```python
from fanpy.ham.utils.factory import ham_factory
from fanpy.tools import slater

def integrate_sd_sd(sd1, sd2, one_int):
    diff_sd1, diff_sd2 = slater.diff_orbs(sd1, sd2)
    nspatial = one_int.shape[0]
    # same as above except replace self.one_int with one_int
    # and self.nspatial with nspatial
    # ...

ham = ham_factory(integrate_sd_sd, oneint, 36, orders=(1,))
# third argument is the number of electrons
```

Again, using the class structure is encouraged because its structure can be cleaner and more transparent and because it provides finer control over the Hamiltonian. For example, if `integrate_sd_wfn` is directly implemented rather than `integrate_sd_sd`, then `integrate_sd_wfn` can be vectorized over the given Slater determinant and its excitations associated with the application of the Hamiltonian. When the derivative of the integral is not provided, orbital optimization is only available through gradient-free optimization algorithms, such as CMA-ES.

Since the Hückel Hamiltonian is defined by its one-electron integrals, this class can be used to describe any Hamiltonian with only one-body operators. The integrals for the Hückel Hamiltonian (and other model Hamiltonians) can be generated using the ModelHamiltonian GitHub repository[74].

Implementing an Objective: New objectives can be implemented in `Fanpy` by making a subclass of the objective base class. The subclass requires the method `objective` to be defined. To use the gradient (or Jacobian) in the optimization algorithm, the subclass must also contain the method `gradient` (or `jacobian`).

Here is an example of an implementation of the local energy used in the orbital space variational Quantum Monte Carlo[82] (Equation 3.7).

$$E_L = \sum_i \frac{\langle \Phi_i | \hat{H} | \Psi \rangle}{\langle \Phi_i | \Psi \rangle} \tag{3.7}$$

where the Slater determinant, Φ_i, is sampled according to the distribution $p(\Phi_i) = \frac{\langle \Psi | \Phi_i \rangle^2}{\sum_k \langle \Psi | \Phi_j \rangle^2}$.

```python
from fanpy.eqn.base import BaseSchrodinger

class LocalEnergy(BaseSchrodinger):
    def __init__(self, wfn, ham, pspace, param_selection=None):
        super().__init__(wfn, ham, param_selection=param_selection)
        # param_selection is used to select the parameters that are active
        # throughout the optimization
        self.pspace = pspace
        # pspace is the list of Slater determinants from which local energy is
        # computed

    @property
    def num_eqns(self):
        # number of equations is used to differentiate objectives in the solver
        return 1

    def objective(self, params):
        # assign (active) parameters to the respective wavefunction and
        # Hamiltonian
        # note that params is always flattened (1-dimensional) for compatibility
        # with solvers
        self.assign_params(params)
        output = 0.0
        for sd in self.pspace:
            output += self.ham.integrate_sd_wfn(sd, self.wfn) / self.wfn.get_overlap(sd)
        return output

    def gradient(self, params):
        self.assign_params(params)
        # note that gradient of the objective is also flattened (1-dimensional)
        output = np.zeros(params.size)
        for sd in self.pspace:
            # indices of the wavefunction parameters that are active
            wfn_inds_component = self.indices_component_params[self.wfn]
            if wfn_inds_component.size > 0:
                # indices of the objective parameters that correspond to the
                # wavefunction
                wfn_inds_objective = self.indices_objective_params[self.wfn]

                # differentiate local energy with respect to wavefunction parameters
                output[wfn_inds_objective] += (
                    self.ham.integrate_sd_wfn(sd, self.wfn, wfn_deriv=wfn_inds_component)
                    / self.wfn.get_overlap(sd)
                )
                output[wfn_inds_objective] -= (
                    self.ham.integrate_sd_wfn(sd, self.wfn)
                    * self.wfn.get_overlap(sd, deriv=wfn_inds_component)
                    / self.wfn.get_overlap(sd) ** 2
                )
            # Indices of the Hamiltonian parameters that are active
            # Used when hamiltonian has parameters to optimize (e.g. orbital
            # optimization)
            ham_inds_component = self.indices_component_params[self.ham]
            if ham_inds_component.size > 0:
                # indices of the objective parameters that correspond to the
                # hamiltonian
                ham_inds_objective = self.indices_objective_params[self.ham]

                # differentiate local energy with respect to Hamiltonian parameters
                output[ham_inds_objective] += (
                    self.ham.integrate_sd_wfn(sd, self.wfn, ham_deriv=ham_inds_component)
                    / self.wfn.get_overlap(sd)
                )
        return output
```

Though it is not required, providing the indices in the gradient ensures that users can specify the parameters that are active during the optimization via the attribute `param_selection`.

3.6 Frequently Asked Questions

Who is `Fanpy` for? `Fanpy` was designed to be used by developers of post-HF methods, especially those interested in new multireference wavefunction ansätze. Extensive programming experience is not necessary: `Fanpy`'s modular design and extensive documentation make it easy to understand and extend the existing methods and base classes. The base classes serve as templates to help ensure that the developed method fits together with the rest of `Fanpy` seamlessly. Developers with programming experience but a limited background in post-HF methods should have an easier time understanding the code because the methods are documented with the corresponding equations (and their derivations) and are implemented in a simple and straight-forward fashion.

What is the mission of `Fanpy`? Our goal is to develop a platform where developers of new *ab initio* methods can quickly implement and test their ideas. We hope to make it easier for researchers—whether they are seasoned professors or new graduate students—to test their ideas without being burdened by undocumented code conventions, mysterious equations, or cumbersome installation processes.

What does `Fanpy` do? As elaborated in Sections 3.4 and 3.5, `Fanpy` provides independent modules that facilitate the development of new multideterminant wavefunctions, Hamiltonians, objectives for the Schrödinger equations, and optimization algorithms. We designed these modules to be compatible with one another so that researchers can easily customize their calculations and experiment with different combinations of methods and algorithms.

What are the limits of `Fanpy`? At present, `Fanpy` is not designed for high performance. In fact, its performance was often deliberately compromised to ensure ease of use and development. For accessibility, `Fanpy` was written in pure Python even though Python performs poorly for large scale numerical computation. Moreover, while `Fanpy`'s modular design is important for its extendibility and customizability, it prevents some types of algorithmic improvements. Since a method in its early stages of development is often intractably expensive, calculations in `Fanpy` are often limited to small model systems with small basis sets. Some of the more efficient methods, e.g. AP1roG, which could be extended to thousands of electrons in an efficient implementation, are limited to about 100 electrons in `Fanpy`. Consistent with the overall mission of HORTON 3, therefore, `Fanpy` should be viewed as a research tool that allows developers to quickly implement and test their ideas, rather than a comprehensive quantum chemistry suite that can simulate large chemical systems. The intention is that once researchers establish that a method is of practical utility, then it can be implemented more efficiently.

How do I install `Fanpy`? The `Fanpy` library can be installed directly from its source code available on GitHub or through the *pip* and *conda* package-management systems. Since `Fanpy` is purely Python and depends mainly on common Python libraries (NumPy and SciPy), it can be installed by simply copying the source code onto the desired directory (though this is not recommended). For the most updated instructions on how to install `Fanpy`, please refer to the `Fanpy` website.

What is the future direction of `Fanpy`? In addition to developing additional methods relevant to our scientific interests, the next iteration of `Fanpy` will focus on improving its performance. Its computationally critical components will be outsourced to highly optimized libraries, such as our in-house CI software, PyCI. Some of the performance bottlenecks will be removed by reimplementing some features in Cython or C++. These improvements may cause problems for some users in terms of ease of use and installation, the pure Python implementation of `Fanpy` will continue to be available.

In terms of features, modules for (arbitrary-order) perturbation theory, equations-of-motion, and quantum-mechanical embedding are in various stages of development.

3.7 Summary

This brief paper introduces `Fanpy` as a library for developing new post-Hartree-Fock *ab initio* methods. `Fanpy`'s goal is to help researchers quickly test ideas for new correlated electronic structure theory methods and, to achieve this goal, `Fanpy` contains many methods that can be used in countless combinations with one another. These methods are of intrinsic interest but, moreover, they serve as examples to be extended upon. Base classes are available as templates to help users develop a structure that is compatible with the rest of `Fanpy`.

Acknowledgements

We wish to acknowledge various refinements to `Fanpy` library from Matthew Chan, Cristina E. González-Espinoza, Xiaotian D. Yang, Stijn Fias, Caitlin Lanssens, Farnaz Heidar-Zadeh and the HORTON development team.

P.W.A. acknowledges Natural Sciences and Engineering Research Council (NSERC) of Canada, the Canada Research Chairs, Compute Canada, and CANARIE for financial and computational support. R.A.M.Q. acknowledges financial support from the University of Florida in the form of a start-up grant.

3.8 References

(1) Kim, T. D.; Miranda-Quintana, R. A.; Richer, M.; Ayers, P. W. Flexible Ansatz for N-body Configuration Interaction., Unpublished Manuscript, 2020.

(2) Frisch, M. J. et al. Gaussian 16 Revision C.01., Gaussian Inc. Wallingford CT, 2016.

(3) Werner, H.-J.; Knowles, P. J.; Knizia, G.; Manby, F. R.; Schütz, M. *Wiley Interdisciplinary Reviews: Computational Molecular Science* **2012**, *2*, 242–253.

(4) Smith, D. G.; Burns, L. A.; Sirianni, D. A.; Nascimento, D. R.; Kumar, A.; James, A. M.; Schriber, J. B.; Zhang, T.; Zhang, B.; Abbott, A. S., et al. *Journal of chemical theory and computation* **2018**, *14*, 3504–3511.

(5) Smith, D. G.; Burns, L. A.; Simmonett, A. C.; Parrish, R. M.; Schieber, M. C.; Galvelis, R.; Kraus, P.; Kruse, H.; Di Remigio, R.; Alenaizan, A., et al. *The Journal of Chemical Physics* **2020**, *152*, 184108.

(6) Verstraelen, T.; Tecmer, P.; Heidar-Zadeh, F.; Boguslawski, K.; Chan, M.; Zhao, Y.; Kim, T. D.; Vandenbrande, S.; Yang, D.; González-Espinoza, C. E.; Fias, S.; Limacher, P. A.; Berrocal, D.; Malek, A.; Ayers, P. W. HORTON., version 2.0.1, 2015.

(7) Pople, J.; Seeger, R.; Krishnan, R. *International Journal of Quantum Chemistry: Quantum Chemistry Symposium* **1977**, *11*, 149–163.

(8) Bytautas, L.; Henderson, T.; Jiménez-Hoyos, C.; Ellis, J.; Scuseria, G. *Journal of Chemical Physics* **2011**, *135*.

(9) Alcoba, D.; Torre, A.; Luis, L.; Oña, O.; Capuzzi, P.; Van Raemdonck, M.; Bultinck, P.; Van Neck, D. *Journal of Chemical Physics* **2014**, *141*.

(10) Cook, D. *Molecular Physics* **1975**, *30*, 733–743.

(11) Weinhold, F.; Wilson Jr, E. *The Journal of Chemical Physics* **1967**, *46*, 2752–58.

(12) Boys, S. *Proceedings of the Royal Society of London A* **1950**, *200*, 542–554.

(13) Hurley, A.; Lennard-Jones, J.; Pople, J. *A theory of paired-electrons in polyatomic molecules Proceedings of the Royal Society of London Series A* **1953**, *220*, 446–455.

(14) Parr, R.; Ellison, F.; Lykos, P. *Journal of Chemical Physics* **1956**, *24*, 1106.

(15) Parks, J.; Parr, R. *Journal of Chemical Physics* **1958**, *28*, 335–345.

(16) McWeeny, R.; Sutcliffe, B. *Proceedings of the Royal Society of London Series A* **1963**, *273*, 103–116.

(17) Surjan, P. In *Correlation and Localization*, Surjan, P., Ed., 1999, pp 63–88.

(18) Allen, T.; Shull, H. *Journal of Physical Chemistry* **1962**, *66*, 2281–2283.

(19) Tecmer, P.; Boguslawski, K.; Johnson, P.; Limacher, P.; Chan, M.; Verstraelen, T.; Ayers, P. *Journal of Physical Chemistry A* **2014**, *118*, 9058–9068.

(20) Paldus, J.; Cizek, J.; Sengupta, S. *Journal of Chemical Physics* **1971**, *55*, 2452–2462.

(21) Paldus, J.; Sengupta, S.; Cizek, J. *Journal of Chemical Physics* **1972**, *57*, 652–666.

(22) Shull, H. *The Journal of Chemical Physics* **1959**, *30*, 1405–13.

(23) Kutzelnigg, W. *The Journal of Chemical Physics* **1964**, *40*, 3640–47.

(24) Johnson, P.; Limacher, P.; Kim, T.; Richer, M.; Miranda-Quintana, R.; Heidar-Zadeh, F.; Ayers, P.; Bultinck, P.; De Baerdemacker, S.; Van Neck, D. *Computational and Theoretical Chemistry* **2017**, *1116*, 207–219.

(25) Surjan, P.; Szabados, Á.; Jeszenszki, P.; Zoboki, T. *Journal of Mathematical Chemistry* **2012**, *50*, 534–551.

(26) Rassolov, V. *Journal of Chemical Physics* **2002**, *117*, 5978–5987.

(27) Rassolov, V.; Xu, F.; Garashchuk, S. *Journal of Chemical Physics* **2004**, *120*, 10385–10394.

(28) Rassolov, V.; Xu, F. *Journal of Chemical Physics* **2007**, *126*, 234112.

(29) Cassam-Chenaï, P. *Journal of Chemical Physics* **2006**, *124*, 194109.

(30) Cassam-Chenaï, P.; Rassolov, V. *Chemical Physics Letters* **2010**, *487*, 147–152.

(31) Cassam-Chenaï, P.; Ilmane, A. *Journal of Mathematical Chemistry* **2012**, *50*, 652–667.

(32) Stein, T.; Henderson, T.; Scuseria, G. *Journal of Chemical Physics* **2014**, *140*, 214113.

(33) Henderson, T.; Scuseria, G.; Dukelsky, J.; Signoracci, A.; Duguet, T. *Physical Review C* **2014**, *89*, 054305.

(34) Henderson, T.; Bulik, I.; Stein, T.; Scuseria, G. *Journal of Chemical Physics* **2014**, *141*, 244104.

(35) Bulik, I.; Henderson, T.; Scuseria, G. *Journal of Chemical Theory and Computation* **2015**, *11*, 3171–3179.

(36) Cullen, J. *Chemical Physics* **1996**, *202*, 217–229.

(37) Miller, K.; Ruedenberg, K. *Journal of Chemical Physics* **1965**, *43*, S88–S90.

(38) Miller, K.; Ruedenberg, K. *Journal of Chemical Physics* **1968**, *48*, 3414–3443.

(39) Silver, D.; Mehler, E.; Ruedenberg, K. *Journal of Chemical Physics* **1970**, *52*, 1174–1180.

(40) Mehler, E.; Ruedenberg, K.; Silver, D. *Journal of Chemical Physics* **1970**, *52*, 1181–1205.

(41) Silver, D.; Ruedenberg, K.; Mehler, E. *Journal of Chemical Physics* **1970**, *52*, 1206–1227.

(42) Coleman, A. *Journal of Mathematical Physics* **1965**, *6*, 1425–1431.

(43) Coleman, A. *International Journal of Quantum Chemistry* **1997**, *63*, 23–30.

(44) Bajdich, M.; Drobný, G.; Wagner, L.; Schmidt, K. *Physical Review Letters* **2006**, *96*, 130201.

(45) Bajdich, M.; Mitas, L.; Wagner, L.; Schmidt, K. *Physical Review B* **2008**, *77*, 115112.

(46) Pernal, K. *Journal of Chemical Theory and Computation* **2014**, *10*, 4332–4341.

(47) Pastorczak, E.; Pernal, K. *Physical Chemistry Chemical Physics* **2015**, *17*, 8622–8626.

(48) Limacher, P.; Kim, T.; Ayers, P.; Johnson, P.; De Baerdemacker, S.; Van Neck, D. *Molecular Physics* **2014**, *112*, 853–862.

(49) Limacher, P. *Journal of Chemical Physics* **2016**, *145*, 194102.

(50) Boguslawski, K.; Tecmer, P.; Ayers, P.; Bultinck, P.; De Baerdemacker, S.; Van Neck, D. *Physical Review B* **2014**, *89*, 201106.

(51) Boguslawski, K.; Tecmer, P.; Bultinck, P.; De Baerdemacker, S.; Van Neck, D.; Ayers, P. *Journal of Chemical Theory and Computation* **2014**, *10*, 4873–4882.

(52) Silver, D. *The Journal of Chemical Physics* **1969**, *50*, 5108–16.

(53) Limacher, P.; Ayers, P.; Johnson, P.; De Baerdemacker, S.; Van Neck, D.; Bultinck, P. *Journal of Chemical Theory and Computation* **2013**, *9*, 1394–1401.

(54) Johnson, P.; Ayers, P.; Limacher, P.; De Baerdemacker, S.; Van Neck, D.; Bultinck, P. *Computational and Theoretical Chemistry* **2013**, *1003*, 101–13.

(55) Schollwöck, U. *Annals of Physics* **2011**, *326*, 96–192.

(56) Paldus, J.; Li, X. In *Advances in Chemical Physics*, Prigogine, I., Rice, S., Eds., 1999; Vol. 110, pp 1–175.

(57) Cizek, J. *Journal of Chemical Physics* **1966**, *45*, 4256–4266.

(58) Shavitt, I.; Bartlett, R., *Many-body methods in chemistry and physics: MBPT and coupled-cluster theory*; Cambridge: Cambridge, 2009.

(59) Bartlett, R.; Musiał, M. *Reviews of Modern Physics* **2007**, *79*, 291–352.

(60) Evangelista, F. A.; Chan, G. K. L.; Scuseria, G. E. *Journal of Chemical Physics* **2019**, *151*, 244112.

(61) Evangelista, F. A. *Journal of Chemical Physics* **2011**, *134*, 224102.

(62) Richardson, R. *Physical Review* **1967**, *159*, 792–805.

(63) Pariser, R.; Parr, R. G. *The Journal of Chemical Physics* **1953**, *21*, 466–471.

(64) Pariser, R.; Parr, R. G. *The Journal of Chemical Physics* **1953**, *21*, 767–776.

(65) Pople, J. A. *Transactions of the Faraday Society* **1953**, *49*, 1375–1385.

(66) Surján, P. R., *Second quantized approach to quantum chemistry: an elementary introduction*; Springer Science & Business Media: 2012.

(67) Hubbard, J. *Proceedings of the Royal Society of London A* **1963**, *276*, 238–257.

(68) Planelles, J.; Zicovich-Wilson, C.; Jaskolski, W.; Corma, A. *International journal of quantum chemistry* **1996**, *60*, 971–981.

(69) Ising, E. *Zeitschrift für Physik* **1925**, *31*, 253–258.

(70) Capelle, K.; Campo Jr, V. L. *Physics Reports* **2013**, *528*, 91–159.

(71) Heisenberg, W. In *Original Scientific Papers Wissenschaftliche Originalarbeiten*; Springer: 1985, pp 580–597.

(72) Richardson, R. **1963**.

(73) Dukelsky, J.; Pittel, S.; Sierra, G. *Reviews of modern physics* **2004**, *76*, 643.

(74) Adams, W.; Zhao, Y.; Ayers, P. W. ModelHamiltonian., version 0.0.0, 2020.

(75) Piela, L., *Ideas of quantum chemistry*; Elsevier: 2013.

(76) Jensen, F., *Introduction to computational chemistry*; John wiley & sons: 2017.

(77) Cramer, C. J., *Essentials of computational chemistry: theories and models*; John Wiley & Sons: 2013.

(78) Helgaker, T.; Jørgensen, P.; Olsen, J., *Modern electronic structure theory*; Wiley: Chichester, 2000.

(79) Szabo, A.; Ostlund, N., *Modern Quantum Chemistry - Introduction to Advanced Electronic Structure Theory*; McGraw-Hill Inc.: 1989, pp 43–107.

(80) Nightingale, M. P.; Umrigar, C. J., *Quantum Monte Carlo methods in physics and chemistry*; 525; Springer Science & Business Media: 1998.

(81) Umrigar, C. *The Journal of chemical physics* **2015**, *143*, 164105.

(82) Sabzevari, I.; Sharma, S. *Journal of chemical theory and computation* **2018**, *14*, 6276–6286.

(83) Neuscamman, E. *The Journal of chemical physics* **2013**, *139*, 194105.

(84) Neuscamman, E. *Journal of chemical theory and computation* **2016**, *12*, 3149–3159.

(85) Kurita, M.; Yamaji, Y.; Morita, S.; Imada, M. *Physical Review B* **2015**, *92*, 035122.

(86) Virtanen, P.; Gommers, R.; Oliphant, T. E.; Haberland, M.; Reddy, T.; Cournapeau, D.; Burovski, E.; Peterson, P.; Weckesser, W.; Bright, J., et al. *Nature methods* **2020**, *17*, 261–272.

(87) Hansen, N.; Akimoto, Y.; Baudis, P. CMA-ES/pycma on Github., Zenodo, DOI:10.5281/zenodo.2559634, 2019.

(88) Head, T. et al. scikit-optimize/scikit-optimize: v0.5.2., version v0.5.2, 2018.

(89) Sun, Q.; Berkelbach, T. C.; Blunt, N. S.; Booth, G. H.; Guo, S.; Li, Z.; Liu, J.; McClain, J. D.; Sayfutyarova, E. R.; Sharma, S., et al. *Wiley Interdisciplinary Reviews: Computational Molecular Science* **2018**, *8*, e1340.

Chapter 4

Graphical Interpretation of Geminals

Graphical Interpretation of Geminals

Taewon D. Kim[†] Michael Richer[†] Paul W. Ayers[†]

September 26, 2020

[†]Department of Chemistry and Chemical Biology, McMaster University, Hamilton, Ontario, L8S-4L8, Canada

Abstract

The Antisymmetrized Product of Geminals (APG) wavefunction is a generalization of the nonorthogonal Slater determinant where the wavefunction is constructed as an antisymmetrized product of two-electron wavefunctions (geminals), rather than as an antisymmetrized product of one-electron wavefunctions (orbitals). Since much of chemistry is expressed in terms of electron pairs, expressing the wavefunction as a product of geminals provides an intuitive link to many traditional chemical concepts, such as Lewis structures. However, extensive use of geminals in wavefunctions has been limited by their high cost stemming from the many combinations of the two-electron basis functions (orbital pairs) used to build the geminals. When evaluating the overlap of the APG wavefunction with an orthogonal Slater determinant, this cost can be interpreted as the cost of evaluating the permanent, resulting from the symmetry with respect to

the interchange of orbital pairs, and the cost of assigning the occupied orbitals to the orbital pairs of the wavefunction. Focusing on the latter, we present a graphical interpretation of the Slater determinant and utilize the maximum weighted matching algorithm to estimate the combination of orbital pairs with the largest contribution to the overlap. Then, the cost due to partitioning the occupied orbitals in the overlap is reduced from $\mathcal{O}((N-1)!!)$ to $\mathcal{O}(N^3 \log N)$. Computational results show that many of these combinations are not be necessary to obtain an accurate solution to the wavefunction. Because the APG wavefunction is the most general of the geminal wavefunctions, this approach can be applied to any of the simpler geminal wavefunction ansätze. In fact, this approach may even be extended to generalized quasiparticle wavefunctions, opening the door to tractable wavefunctions built using components of arbitrary numbers of electrons, not just two electrons.

4.1 From Orbital-Based To Geminal-Based Wavefunctions

The simplest practical N-electron wavefunction that satisfies the Pauli exclusion principle is a Slater determinant of spin-orbitals. In most practical calculations, the spin-orbitals, which are merely one-electron wavefunctions, are expanded as a linear combination of one-electron basis functions, which are typically chosen to approximate atomic orbitals. To satisfy the Pauli exclusion principle, an antisymmetric product (Slater determinant) of N spin-orbitals is constructed. It is common to then optimize the molecular orbitals by minimizing the energy, resulting in the venerable Hartree-Fock wavefunction,

$$|\Phi_{\mathrm{HF}}\rangle = \prod_{k=1}^{N} \sum_{i=1}^{2K} C_{ki} a_i^{\dagger} |0\rangle \tag{4.1}$$

where N is the number of electrons, K is the number of spatial basis functions ($2K$ is the number of spin-basis functions), and C is constrained so that the orbitals are orthogonal and normalized[1–9].

The Slater determinant wavefunction represents a system of independent fermions, and intrinsically omits electron correlation. The typical approach to electron correlation is to correct the Hartree-Fock method by computing contributions from additional Slater determinants, but this approach can be very inefficient for strongly-correlated systems, where the Hartree-Fock wavefunction can be a very poor starting point (e.g., it may have a very small overlap with the exact wavefunction) and/or myriad Slater determinants may be required. This suggests an alternative approach, philosophically similar to the Hartree-Fock method, wherein the wavefunction is constructed as an antisymmetric product of 2-electron wavefunctions, called geminals. (Obviously, the method can be extended to composite particles composed of three (ternions), four (quartets), or more electrons[10, 11].)

In analogy to the basis-set expansion for orbitals, in practical calculations, geminals are usually expanded as a linear combination of two-electron basis functions. In analogy to the Slater determinant, the simplest N-electron geminal-based wavefunction is an antisymmetrized product of $P = N/2$ geminals[12–15]. (In this paper we will assume that there are an even number of electrons in the system, but there are standard approaches for treating odd-electron systems with geminals.) It is convenient to choose the two-electron basis functions to be *orbital pairs*: antisymmetrized products of one-electron orthonormal basis functions (i.e., orbitals).

In its most general form, the Antisymmetrized Product of Geminals (APG) uses all possible orbital pairs and imposes no restrictions on their coefficients,

$$|\Psi_{\text{APG}}\rangle = \prod_{p=1}^{P} \sum_{ij}^{2K} C_{p;ij} a_i^\dagger a_j^\dagger |0\rangle$$

In the APG wavefunction, all creation operators $a_i^\dagger$ and $a_j^\dagger$ can be paired with one another, regardless of their spin[10, 12–21]. For both conceptual and computational purposes, it is convenient to rewrite the APG wavefunction as a linear combination of Slater determinants, each of which has a coefficient which is a sum of permanents[11],

$$|\Psi_{\text{APG}}\rangle = \sum_{\mathbf{m} \in \text{dets}} \sum_{\{i_1, j_1, \dots, i_P, j_P\} = \mathbf{m}} \text{sgn}\Big(\sigma(i_1, j_1, \dots, i_P, j_P)\Big) \begin{vmatrix} C_{1;i_1 j_1} & \dots & C_{1;i_P j_P} \\ \vdots & \ddots & \vdots \\ C_{P;i_1 j_1} & \dots & C_{P;i_P j_P} \end{vmatrix}^{+} |\mathbf{m}\rangle \tag{4.2}$$

where $\mathbf{m}$ is a set of spin-orbitals occupied in the Slater determinant, dets is the set of all Slater determinants, C is the geminal coefficient matrix whose columns correspond to the creation operator pairs,

$$\sum_{\{i_1, j_1, \dots, i_P, j_P\} = \mathbf{m}} \tag{4.3}$$

denotes the sum over all possible combinations of creation operator pairs that construct the given Slater determinant, sgn(σ) is the signature of the permutation required to reorder $a_{i_1}^\dagger a_{j_1}^\dagger \dots a_{i_P}^\dagger a_{j_P}^\dagger$ such that the indices of the creation operators are increasing, and

$$\begin{vmatrix} C_{1;i_1j_1} & \dots & C_{1;i_Pj_P} \\ \vdots & \ddots & \vdots \\ C_{P;i_1j_1} & \dots & C_{P;i_Pj_P} \end{vmatrix}^+ = \sum_{\tau \in S_P} C_{1;i_{\tau(1)}j_{\tau(1)}} \dots C_{P;i_{\tau(P)}j_{\tau(P)}} \tag{4.4}$$

is a permanent. In Equation 4.4, $\tau \in S_P$ denotes the sum over all permutations τ. We will refer to the creation operators as occupied spin orbitals, creation operator pairs as orbital pairs, and the combinations of creation operator pairs as pairing schemes.

The preceding equation makes it easy to place APG within the more general framework of Flexible Ansatze for N-electron Configuration Interaction (FANCI) wavefunctions[11]. Specifically, the overlap of the APG wavefunction with a Slater determinant, $\mathbf{m}$, is

$$\langle \mathbf{m} | \Psi_{\text{APG}} \rangle = \sum_{\{i_1, j_1, \dots, i_P, j_P\} = \mathbf{m}} \text{sgn}\Big(\sigma(i_1, j_1, \dots, i_P, j_P)\Big) \begin{vmatrix} C_{1;i_1j_1} & \dots & C_{1;i_Pj_P} \\ \vdots & \ddots & \vdots \\ C_{P;i_1j_1} & \dots & C_{P;i_Pj_P} \end{vmatrix}^+ \tag{4.5}$$

Each term in the overlap expression corresponds to a combination of the orbital pairs (pairing scheme) to form the given Slater determinant and each term in the permanent expression corresponds to an ordering of the orbital pairs from the given pairing scheme. These two components of the APG make it prohibitively expensive: 1) the number of terms in the overlap (Equation 4.5) increases double factorially with the number of electrons, $(2P-1)!!$, and 2) the cost of evaluating the permanent increases factorially with the number of electron pairs, $P!$.

Different geminal wavefunctions try to offset this computation cost by truncating the two-electron (orbital pair) basis set and by constraining the geminal coefficients[22–50]. By removing orbital pairs from the geminals, the number of terms in the sum (Equation 4.3) decreases. In the APsetG wavefunction, only the creation operators of one set, S_1 can be paired with the creation operators of a second disjoint set, S_2[47].

$$|\Psi_{\text{APsetG}}\rangle = \prod_{p}^{P} \sum_{i \in S_1} \sum_{j \in S_2} C_{p;ij} a_i^\dagger a_j^\dagger |0\rangle \tag{4.6}$$

In the spin-restricted Antisymmetrized Product of Interacting Geminals (APIG) wavefunction, only the creation operators that correspond to the same spatial orbital can be paired with one another[50].

$$|\Psi_{\text{APIG}}\rangle = \prod_{p}^{P} \sum_{i}^{K} C_{pi} a_i^\dagger a_{\bar{i}}^\dagger |0\rangle \tag{4.7}$$

Here i is the index for the spatial orbital with spin alpha and $\bar{\imath}$ is the index for the same spatial orbital with spin beta. Both APsetG and APIG wavefunctions limit the number of terms in the overlap (Equation 4.5) by truncating the orbital pairs: fewer orbital pairs results in fewer pairing schemes. Specifically, APsetG reduces the terms in the sum (Equation 4.3) to $P!$; in APIG, there is only a single term in the sum.

In addition to truncating the orbital pairs, we can apply constraints to the geminal coefficients. For example, the Antisymmtrized Product of 1-reference orbital Geminals (AP1roG) wavefunction is an approximation to the APIG wavefunction where much of the coefficient matrix is fixed to be an identity matrix[51].

$$|\Psi_{\text{AP1roG}}\rangle = \prod_{p}^{P} \left(a_p^\dagger a_{\bar{p}}^\dagger + \sum_{i=P+1}^{K} C_{pi} a_i^\dagger a_{\bar{i}}^\dagger \right) |0\rangle \tag{4.8}$$

In this wavefunction, each geminal is dominated by a spatial orbital occupied in the ground-state HF wavefunction. The cost of its overlap with a Slater determinant is $\mathcal{O}(m!)$ where m is the order of excitation of the given Slater determinant with respect to the HF ground-state. Unlike the AP1roG wavefunction, which approximates a portion of the APIG coefficient matrix, the Antisymmetrized Product of rank-2 Geminals (APr2G) approximates the entire matrix as a Cauchy matrix, whose permanent can be reduced to a quotient of two determinants[52]. Then, the cost of the overlap reduces to that of computing the determinants, which is $\mathcal{O}(P^3)$. Since these approximations apply only to the geminal coefficients, they can be applied to any geminal wavefunctions. In this article, however, we focus on developing new methodologies for selecting orbital pairs for the APG wavefunction and keep the coefficients unconstrained.

4.2 From Geminal-Based Wavefunctions to Graphs

Unlike the APIG wavefunction, the APsetG and APG wavefunctions can represent the same Slater determinant with different pairing schemes. For example, suppose we have the Slater determinant with occupied spin orbitals $(1, 2, \bar{1}, \bar{2})$, where spin orbitals i and $\bar{\imath}$ correspond to the alpha and beta spin parts of the i^{th} spatial orbital. In the APG wavefunction, there are three different pairing schemes: $(1, \bar{1})$ and $(2, \bar{2})$; $(1, \bar{2})$ and $(2, \bar{1})$; and $(1, 2)$ and $\bar{1}\bar{2}$. The APsetG wavfunction has two pairing schemes: $(1, \bar{1})$ and $(2, \bar{2})$; and $(1, \bar{2})$ and $(2, \bar{1})$. The APIG wavefunction has just one pairing scheme: $(1, \bar{1})$ and $(2, \bar{2})$. Each pairing scheme corresponds to a term in the overlap, each term requiring the evaluation of a permanent.

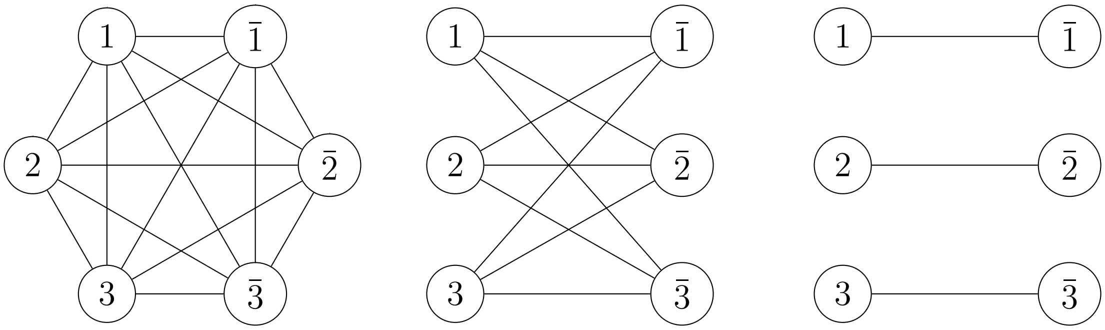

FIGURE 4.1: Graphs to describe the overlap of Slater determinant with occupied spin orbitals $(1, \bar{1}, 2, \bar{2}, 3, \bar{3})$: APG (left), APsetG (center), and APIG (right) wavefunctions

Identifying the pairing schemes associated with a given Slater determinant is analogous to dividing its occupied spin orbitals into disjoint subsets containing two elements. In other words, a pairing scheme is a set of subsets of two spin orbitals. Consistent with the Pauli exclusion principle, the subsets are all disjoint; the union of the subsets equals the set of occupied spin orbitals in the given Slater determinant.

The process of finding the pairing schemes can be simplified by interpreting the occupied spin-orbitals and subsets/orbital-pairs as vertices and edges of a graph, respectively. For example, the graph for the APG wavefunction is a complete graph since all orbital pairs can be used. The graph for the APsetG wavefunction, on the other hand, is a complete bipartite graph because only the orbitals from one set can be paired with the orbitals from a disjoint set. The graph for the APIG wavefunction is a perfect matching. The graphs that correspond to the overlaps of the APG, APsetG, and APIG wavefunctions with the Slater determinant with occupied spin-orbitals labeled by $\{1, \bar{1}, 2, \bar{2}, 3, \bar{3}\}$ are shown in Figure 4.1, and the corresponding mathematical expressions are shown in the Appendix.

If the occupied spin orbitals and the allowed orbital pairs are represented as a graph, then each pairing scheme is a perfect matching of this graph. A perfect matching is a set of edges of a graph with no shared end points that contains all of the vertices of a graph; a pairing scheme is a set of orbital pairs with no shared orbitals that contains all of the occupied spin orbitals of a Slater determinant. Then, the sum over the pairing schemes within the overlap is equivalent to the sum over the perfect matchings of the corresponding graph. In the case of the APG wavefunction, there are $(2P-1)!! = \frac{(2P)!}{P!2^P}$ perfect matchings of the corresponding complete graph of $2P$ vertices. The number of pairing schemes grows explosively as the number of vertices/electrons increases, and the overlap becomes intractable to compute. In contrast, at least for systems of chemical relevance, it seems the APIG wavefunction is a nearly exact approximation of the Doubly Occupied Configuration Interaction (DOCI) wavefunction[53–64], despite using only one pairing scheme per Slater determinant[18, 45, 51]. This suggests that not all pairing schemes are needed and that some pairing schemes are more important than others; many terms in the overlap may be negligibly small and thus can be ignored.

The goal of this work is to develop an algorithm that systematically selects the most important pairing schemes (discarding the rest) for any given Slater determinant. Since the number of pairing schemes increases exponentially with the number of electrons, this algorithm cannot consider each pairing scheme. Of course, the orbital pairs can be truncated from the geminal wavefunction, as in the APIG wavefunction, where orbital pairs are truncated according to chemical intuition. However, the methods based on heuristics often break down when the assumptions used are no longer valid for the system under study. For instance, the spin-restricted APIG wavefunction assumes that alpha-beta spin orbitals associated with a common spatial orbital are the most important pair and that other orbital pairs are not significant. When the system requires significant contributions from non-seniority zero Slater determinants, the truncated orbital pairs become more important and the APIG wavefunction is unable to accurately describe the system.

Fortunately, there already exist algorithms that find the perfect matching with the largest weights from a graph in polynomial time. In this article, we use the maximum weighted matching algorithm based on the Edmonds' Blossom algorithm[65–67]. Though the specifics of this maximum weighted matching algorithm are deferred to the references[65–67], this algorithm uses the adjacency matrix, a matrix that embodies the connectivity of the graph and the weights of the edges, to systematically contracts the graph and iteratively find the (augmenting) path whose sum of weights is maximized. The cost of the Blossom algorithm is $\mathcal{O}\Big(\binom{N}{2} N \log N\Big) = \mathcal{O}(N^3 \log N)$ for a complete graph with N vertices.

By assigning the weights of the edges so that they correspond to the contributions of the orbital pairs to a pairing scheme, then using the Blossom algorithm, we can find the pairing scheme with the largest contribution to the overlap. In the next section, we discuss how to assign the edge-weights so that they correspond to the importance of the orbital pairs. Then, in Section 4.4 we present various strategies for predicting the next most significant pairing schemes using only the weighted maximum matching algorithm. After presenting our test systems in Section 4.5, we present our numerical results in Section 4.6.

4.3 Orbital Pair Contribution

Recall that the Blossom algorithm finds the set of edges (orbital pairs) that maximizes the sum of the edge-weights. If we can define the weights of the orbital pairs, $\{w_{i_1 j_1} \dots w_{i_P j_P}\}$, such that their sum has the same ordering as the size the corresponding permanents,

$$\begin{vmatrix} C_{1;i_1 j_1} & \dots & C_{1;i_P j_P} \\ \vdots & \ddots & \vdots \\ C_{P;i_1 j_1} & \dots & C_{P;i_P j_P} \end{vmatrix}^+$$

then the maximum weighted matching will produce the pairing schemes with the largest permanents. Because the permanent is expensive to compute, we need to approximate it with an upper bound. One possibility is the Hadamard-like inequality for permanents[68]:

$$\mathrm{abs}(|A|^+) \leq \frac{N!}{N^{N/2}} \prod_{j=1}^{N} \mathrm{norm}(A^j) \tag{4.9}$$

where N is the number of columns, A^j is the jth column of A, and the norm is the Euclidian norm. However, the scalar factor $\frac{N!}{N^{N/2}}$ increases very quickly with the size of the matrix, so this inequality is too loose to be helpful for larger matrices. In particular, because the geminal wavefunctions are normalized and are optimized starting from bounded initial guesses, their permanents will likely be bounded from above by a small positive integer. For example, the initial guess for the APIG wavefunction is often the coefficient matrix where the only nonzero coefficients are those associated with the Hartree-Fock ground state. This suggests that the largest contributors to the coefficient matrix are often identity elements along the diagonal (corresponding to intermediate normalization). Therefore, most permanents will likely be less than one. Since the wavefunction is normalized, all of the coefficients will likely remain less than or equal to one during the course of the optimization. Moreover, if this were not true, one could always adjust the normalization accordingly.

Here, we propose the following inequality:

$$\mathrm{abs}(|A|^{+}) \leq \prod_{j=1}^{N} \sum_{i=1}^{N} |a_{ij}| \tag{4.10}$$

When the L1-norm, $\sum_{i=1}^{N} |a_{ij}|$, is less than or equal to one for all columns, this inequality should provide a tighter bound than Inequality 4.9. The proof of this inequality is given in the Appendix.

When the wavefunction is normalized, most of the coefficients should be less than one and the others close to one. Then, most orbital pairs will have L1-norms that are less than one, the product of which results in an even smaller upper bound for the corresponding permanent. This upper bound on small permanents means that when evaluating the overlap (cf. Equation 4.5) these terms can be safely eliminated from the sum without significant loss of accuracy.

Using Inequality 4.10, the importance of the pairing schemes, i.e. values of the permanents, can be efficiently compared without evaluating the permanents. According to Equation 4.2, the orbital pair is represented by a column of the geminal coefficient matrix and the weight of the orbital pair is the L1-norm of the corresponding column. However, the weight of the matching used in the matching algorithm is the sum of the weights of the edges, rather than the product. To directly utilize the Inequality 4.10, we use the log of the L1-norms as the weights. Numerical issues arise when the L1-norm of a column is very small, so weights below a threshold are set to zero and, for consistency, the remaining weights are shifted by the same threshold:

$$w_{ij} = \begin{cases} 0 & \text{if } \sum_{p=1}^{P} |C_{p;ij}| \leq \text{threshold} \\ \log\left(\sum_{p=1}^{P} |C_{p;ij}|\right) - \log(\text{threshold}) & \text{else} \end{cases} \tag{4.11}$$

Here w_{ij} denotes the weight of the edge that corresponds to pairing spin-orbital i with spin-orbital j. Using these weights, the maximum weighted matching algorithm will produce a pairing scheme that has the largest upper bound to the corresponding permanent. We propose that the ordering of the pairing schemes by the presented upper bound (Equation 4.10) is similar to the ordering by the permanent values but this is less important, because we can evaluate all the contributions from permanents larger than a certain threshold using the upper bound in Equation 4.10.

4.4 Algorithm

When the sum is truncated to one pairing scheme, we can find the pairing scheme with the largest upper bound of the permanent using the Blossom algorithm. To find the pairing schemes with the next largest weights, the algorithm for finding K-best perfect matchings can be used[69]. The K best perfect matchings are the K perfect matchings with the largest weights. This algorithm involves systematically removing one or more edges of the matching from the graph and applies the weighted matching algorithm to the pruned graph. An arbitrary number of pairing schemes can be obtained in the order of decreasing weight using the K-best perfect matching algorithm. The cost of this algorithm scales linearly with the number of selected pairing schemes.

When using the K-best perfect matching algorithm to evaluate Slater determinant overlaps with a geminal wavefunction, only the orbital pairs whose coefficients have an L1-norm greater than the threshold are used. If the L1-norm drops below the given threshold, then this orbital pair will no longer contribute to the wavefunction for the remainder of the optimization. In addition, it is possible that an orbital pair does not contribute to the wavefunction despite having an L1-norm above the threshold because all the pairing schemes to which it belongs have been truncated. To avoid such situations, noise can be added to the coefficients during the optimization process. The coefficients to which the noise is added can be determined according to some heuristic or probability distribution. However, these more sophisticated algorithms have not been explored in this initial study.

4.5 Computational Protocol

To test the accuracy of the aforementioned method for selecting the most important perfect matchings, we studied the symmetric dissociation of a H_8 chain (Figure 4.2)[51, 70] and symmetric stretch of a ring of four Hydrogen molecules (Figure 4.3)[71], using the ANO-1s basis set. We chose these systems because accurate full configuration interaction (FCI) reference data is available and because, for these small systems, the exact evaluation of the overlap (Equation 4.5) is feasible. Though these systems have few electrons and basis functions, they are challenging to model with traditional electronic

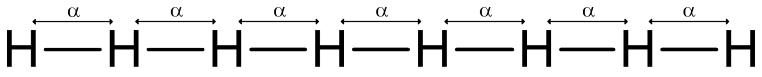

FIGURE 4.2: Linear H_8 chain: $\alpha \in \{0.6, 0.7, 0.8, 0.9, 1, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2, 2.25, 2.5, 3, 4\}$ Angstroms

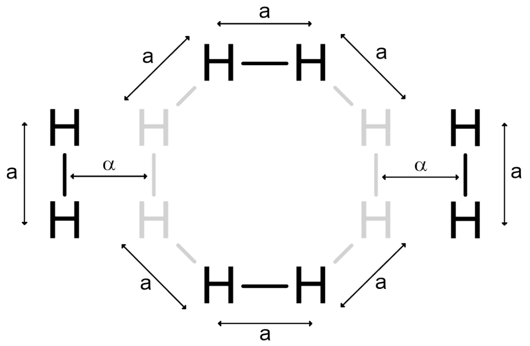

FIGURE 4.3: Octagonal H_8: a = 2 a.u., $\alpha \in \{0, 0.0001, 0.001, 0.003, 0.006, 0.01, 0.03, 0.06, 0.1, 0.5, 1\}$ a.u.

structure methods because they are very strongly multiconfigurational, as many bonds (7 in the H_8 chain; 4 in the H_8 ring) are being broken simultaneously and there are many nearly degenerate orbitals near the Fermi level.

The Restricted HF wavefunction was obtained from Gaussian 2016[72] and the 1- and 2-electron integrals were obtained from the `gbasis` module of HORTON[73]. The FCI wavefunction was computed using PySCF[74]. All other wavefunctions were implemented and obtained using our in-house Fanpy package[75]. The Blossom algorithm was obtained from the networkx package[76]. However, rather than using the efficient K-best perfect matching algorithm, in this initial study we obtained the K-best pairing schemes in a brute force fashion, by explicitly ordering weights of the pairing schemes. The threshold of 10^{-4} was used for computing the weights of the orbital pairs. With the exception of the FCI wavefunction, the orbitals of all wavefunctions were optimized.

4.6 Results and Discussion

We considered the following wavefunctions: Ap1roG, APIG, DOCI, FCI, and APG; each wavefunction was computed both with and without truncation. The truncated APG wavefunction will be denoted as -Kps, where K denotes that only the terms in the sum (Equation 4.3) corresponding to the K-best perfect matchings are included.

The energies of these wavefunctions and the relative energies of the truncated APG-Kps geminal wavefunctions with respect to the APG wavefunction are shown in Figures 4.4 and 4.5. The Hartree-Fock energy is very inaccurate in all cases, confirming the multiconfigurational nature of these test systems. The seniority-zero wavefunctions (AP1roG, APIG, and DOCI) are all qualitatively correct, and visually indistinguishable from one another, confirming once again that the AP1roG wavefunction is a very good approximation to the APIG wavefunction for systems with repulsive interparticle interactions and that the APIG wavefunction is, in turn, an excellent approximation to the DOCI wavefunction[51]. However, the seniority-zero methods are systematically above the FCI curves due to the absence of dynamic correlation, which is nonnegligible even in this small basis set, especially for small internuclear distances. Dynamic correlation within electron pairs (but not the more challenging intergeminal correlation between electron pairs) is included in the APG wavefunction, which is systematically more accurate to AP1roG and APIG. Note that unlike seniority-zero methods, which are qualitatively incorrect near $\alpha = 0$ for the H_8 ring because they predict a conical intersection, APG apparently models the avoided crossing at $\alpha = 0$ qualitatively correctly.

APG is very expensive, so it is reassuring that the energies from the APG-Kps methods quickly approach the energy from APG as K increases. It is remarkable that even though APG-1ps uses a single pairing scheme like APIG, its energy error (relative to exact APG) is about three times smaller. This is because, unlike APIG, where the pairing scheme is preassigned, the pairing scheme in APG-1ps is dynamically selected based on APG coefficients and the occupied orbitals (different pairing schemes will be used for the projection on different Slater determinants). Indeed, the energy-error of APG-1ps relative to APG is comparable to the energy-error of APG relative to FCI. The computational cost of APG-Kps grows linearly with K, so allowing including additional pairing schemes is not prohibitive.

The disadvantage of APG-Kps is that there are derivative discontinuities in the energy when the elements of the set of K-best pairing schemes changes. This is most visible in Figure 4.4b, where the errors in APG-Kps are a little erratic, as compared to the smoother errors in APIG. These discontinuities can be decreased as much as desired by increasing the number of pairing schemes.

APG-Kps is also useful for optimizing the geminal coefficients in APG. For example, one can start by loosely converging the APG-Kps coefficients for a small number of pairing schemes, and then systematically increase K as the convergence threshold is tightened, using the previous solution as an accurate initial guess. Similarly, the threshold for the weights of the edges (Equation 4.11) can be systematically tightened during the convergence procedure. We have not, however, explored these more complicated algorithms here, as their primary utility would be for larger systems where brute-force APG calculations are intractable.

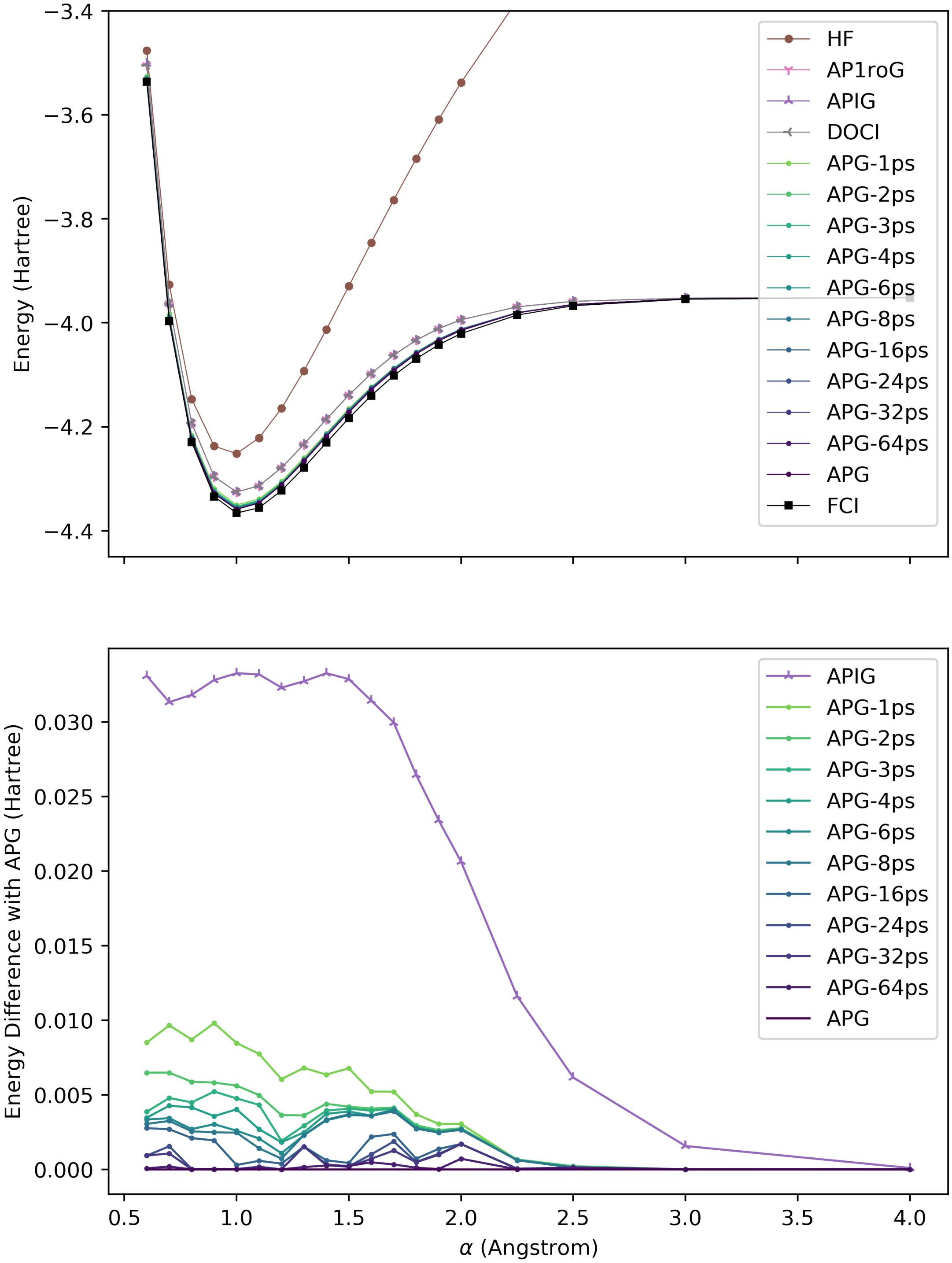

FIGURE 4.4: (a) Energies and (b) energy differences relative to APG in the H_8 chain; see Figure 4.2

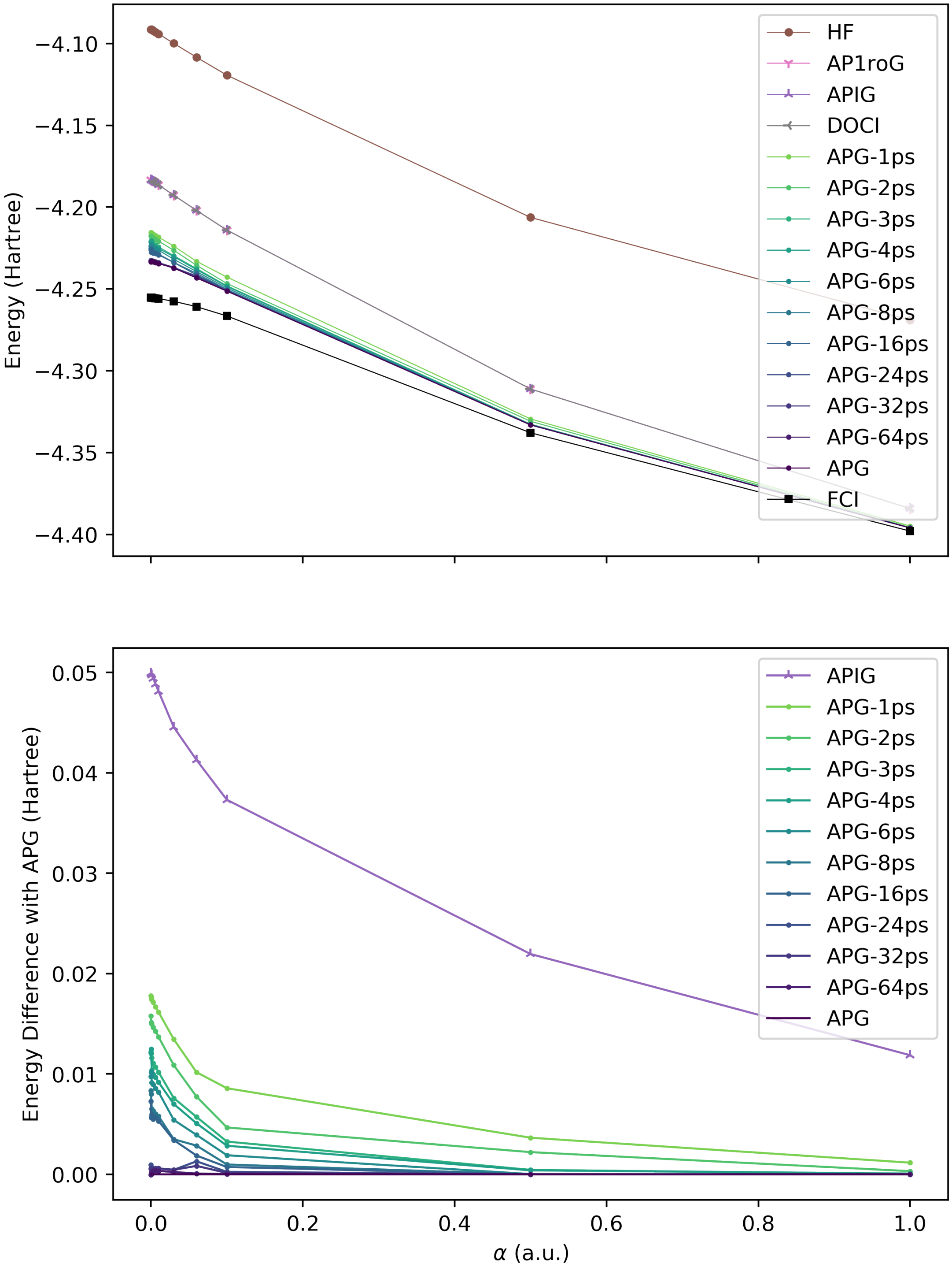

FIGURE 4.5: (a) Energies and (b) energy differences relative to APG in the H_8 ring; see Figure 4.3

Even greater gains can be obtained by making an analogy between selecting pairing schemes in APG-Kps and choosing determinants in selected configuration interaction (CI) methods. In this analogy, the APG wavefunction is analogous to FCI. One could remove pairing schemes from the APG based on certain system-independent rules; this gives approaches like APIG and APsetG, which are analogous to systematically-truncated CI wavefunctions like CISD, CASSCF, DOCI, *etc.*[77–81]. Though these wavefunctions are much more affordable, they lack critical correlations. APG-Kps is analogous to selected-CI algorithms like CIPSI[82] and HCI[83], where one selects configurations based on the estimated size of their coefficient[82–94]. To further extend APG-Kps, one may be inspired by the FCI-QMC approach[95–108], and selecting the pairing schemes by sampling from a probability distribution instead of by using a deterministic algorithm as we have done here.

Another possible improvement is to use faster methods for selecting the K-best pairing schemes. State-of-the-art weighted matching algorithms have cost of $\mathcal{O}(N^2\sqrt{N})$ for the exact algorithm[109, 110] and $\mathcal{O}(N^2\epsilon^{-1}\log\epsilon^{-1})$ for the $(1-\epsilon)$ approximate algorithm[111]. The approximate algorithm returns a pairing scheme with a weight that is within a factor of $(1-\epsilon)$ of optimal. These matching algorithms are not limited to complete graphs so they can be used for any geminal wavefunction, including APsetG.

These algorithmic improvements, however, are limited by the APG wavefunction. The APG wavefunction is not exact for systems with more than two electrons because it neglects the correlations between geminals. One way to incorporate these intergeminal correlations is to add basis functions with different numbers of electrons to form a generalized quasiparticle wavefunction[10, 11]. Just as the pairing schemes correspond to the perfect matches, each term in the overlap of the generalized quasiparticle wavefunction corresponds to a partition of the electrons into subsets. In fact, a geminal wavefunction is a special case of the generalized quasiparticle wavefunctions and a perfect matching is a special case of partitions. Adding, for example, 3- and 4-electron basis functions results in an exponential increase in the number of terms in the overlap because there are many more ways to partition the given occupied orbitals into the available components. This wavefunction will be significantly more expensive to compute than the APG wavefunction, which was already intractable. However, dynamically selecting the pairing schemes in the APG wavefunction produced results comparable to those using all pairing schemes, despite being much cheaper. Similarly, it may be possible to omit many partitions within the overlap of the generalized quasiparticle wavefunction without a significant impact on the wavefunction. This approach allows the basis functions to be built up systematically, starting from the optimization of the wavefunction with the smaller body terms, such as 1- and 2-electron basis functions, and subsequently adding the larger body terms.

Although this article focused on reducing the number of pairing schemes, reducing the cost of permanent evaluation is arguably even more critical for producing tractable geminal (and quasiparticle) wavefunctions. The 1-reference-orbital and rank-2 approximations used in AP1roG and APr2G are clearly generalizable to the APG wavefunction. Since these approximations pertain to the evaluation of permanents, they can also apply to the generalized quasiparticle wavefunction. These important practical embellishments on the general strategy we present here are an important target for future work. To give an idea of the types of performance that can be achieved, if the 1-reference-orbital structure for evaluating the permanent (analogous to AP1roG), then APG-Kps-1ro would have a computational cost that was about K times that of AP1roG, times the cost of determining the perfect matchings that need to be computed. The computational scaling of AP-Kps-1ro is thus the same as CCSD, but unlike CCSD, AP-Kps-1ro is applicable to strongly correlated systems.

4.7 Summary

As the number of electrons, N in a system increases, evaluating the APG wavefunctions becomes increasing intractable due to the cost of summing over all $(N-1)!!$ possible pairing schemes (the outer sum in Equation 4.5) and the $O((N/2)!)$ cost of evaluating the permanent. Our computational results suggest that only a small number of the pairing schemes are quantitatively important. To exploit this, we derived an efficient upper bound on the contribution of any given permanent, and then used the link between optimal pairing schemes and algorithms for selecting the K-best perfect matching of weighted graphs to select the K most important pairing schemes. (Alternatively, we could neglect the pairing schemes whose contributions were guaranteed to be less than a given threshold.) The APG-1ps method, where only a single pairing scheme is chosen, has essentially the computational cost as the venerable APIG method but recovers about 2/3 of the gap in correlation energy between the APIG and APG methods. As additional pairing schemes are included, the APG-Kps methods quickly converge to the exact APG-$(N-1)!!$ps results. The APG-Kps method opens the possibility for sophisticated optimization algorithms that select the most appropriate pairing schemes and the corresponding wavefunction parameters, as well as extensions to more general quasiparticle wavefunctions. When coupled with efficient parameterizations of the geminal coefficients that reduce the cost of evaluating the permanent, the APG-Kps strategy opens up new possibilities for wavefunctions that are not only reliable for strongly-correlated molecules, but also computationally tractable, chemically interpretable, and numerically robust.

4.8 Acknowledgements

The authors thank NSERC, Compute Canada, McMaster University, and the Canada Research Chairs for funding.

4.9 References

(1) Szabo, A.; Ostlund, N., *Modern Quantum Chemistry - Introduction to Advanced Electronic Structure Theory*; McGraw-Hill Inc.: 1989, pp 43–107.

(2) Helgaker, T.; Jørgensen, P.; Olsen, J., *Modern electronic structure theory*; Wiley: Chichester, 2000.

(3) Helgaker, T.; Jørgensen, P.; Olsen, J. *Journal of Physical Chemistry* **1996**, *100*, 13213–13225.

(4) Raghavachari, K.; Anderson, J. *Journal of Physical Chemistry* **1996**, *100*, 12960–12973.

(5) Marti, K.; Reiher, M. *Physical Chemistry Chemical Physics* **2011**, *13*, 6750–6759.

(6) Helgaker, T.; Coriani, S.; Jørgensen, P.; Kristensen, K.; Olsen, J.; Ruud, K. *Chemical Reviews* **2012**, *112*, 543–631.

(7) Sherrill, C. *Journal of Chemical Physics* **2010**, *132*, 110902.

(8) Slater, J. *Physical Review* **1928**, *32*, 339–348.

(9) Fock, V. *Zeitschrift für Physik* **1930**, *61*, 126–148.

(10) Parr, R.; Ellison, F.; Lykos, P. *Journal of Chemical Physics* **1956**, *24*, 1106.

(11) Kim, T. D.; Miranda-Quintana, R. A.; Richer, M.; Ayers, P. W. Flexible Ansatz for N-body Configuration Interaction., Unpublished Manuscript, 2020.

(12) Hurley, A.; Lennard-Jones, J.; Pople, J. *A theory of paired-electrons in polyatomic molecules Proceedings of the Royal Society of London Series A* **1953**, *220*, 446–455.

(13) McWeeny, R.; Sutcliffe, B. *Proceedings of the Royal Society of London Series A* **1963**, *273*, 103–116.

(14) Surjan, P. In *Correlation and Localization*, Surjan, P., Ed., 1999, pp 63–88.

(15) Kutzelnigg, W. *The Journal of Chemical Physics* **1964**, *40*, 3640–47.

(16) Parks, J.; Parr, R. *Journal of Chemical Physics* **1958**, *28*, 335–345.

(17) Allen, T.; Shull, H. *Journal of Physical Chemistry* **1962**, *66*, 2281–2283.

(18) Tecmer, P.; Boguslawski, K.; Johnson, P.; Limacher, P.; Chan, M.; Verstraelen, T.; Ayers, P. *Journal of Physical Chemistry A* **2014**, *118*, 9058–9068.

(19) Paldus, J.; Cizek, J.; Sengupta, S. *Journal of Chemical Physics* **1971**, *55*, 2452–2462.

(20) Paldus, J.; Sengupta, S.; Cizek, J. *Journal of Chemical Physics* **1972**, *57*, 652–666.

(21) Shull, H. *The Journal of Chemical Physics* **1959**, *30*, 1405–13.

(22) Surjan, P.; Szabados, Á.; Jeszenszki, P.; Zoboki, T. *Journal of Mathematical Chemistry* **2012**, *50*, 534–551.

(23) Rassolov, V. *Journal of Chemical Physics* **2002**, *117*, 5978–5987.

(24) Rassolov, V.; Xu, F.; Garashchuk, S. *Journal of Chemical Physics* **2004**, *120*, 10385–10394.

(25) Rassolov, V.; Xu, F. *Journal of Chemical Physics* **2007**, *126*, 234112.

(26) Cassam-Chenaï, P. *Journal of Chemical Physics* **2006**, *124*, 194109.

(27) Cassam-Chenaï, P.; Rassolov, V. *Chemical Physics Letters* **2010**, *487*, 147–152.

(28) Cassam-Chenaï, P.; Ilmane, A. *Journal of Mathematical Chemistry* **2012**, *50*, 652–667.

(29) Stein, T.; Henderson, T.; Scuseria, G. *Journal of Chemical Physics* **2014**, *140*, 214113.

(30) Henderson, T.; Scuseria, G.; Dukelsky, J.; Signoracci, A.; Duguet, T. *Physical Review C* **2014**, *89*, 054305.

(31) Henderson, T.; Bulik, I.; Stein, T.; Scuseria, G. *Journal of Chemical Physics* **2014**, *141*, 244104.

(32) Bulik, I.; Henderson, T.; Scuseria, G. *Journal of Chemical Theory and Computation* **2015**, *11*, 3171–3179.

(33) Cullen, J. *Chemical Physics* **1996**, *202*, 217–229.

(34) Miller, K.; Ruedenberg, K. *Journal of Chemical Physics* **1965**, *43*, S88–S90.

(35) Miller, K.; Ruedenberg, K. *Journal of Chemical Physics* **1968**, *48*, 3414–3443.

(36) Silver, D.; Mehler, E.; Ruedenberg, K. *Journal of Chemical Physics* **1970**, *52*, 1174–1180.

(37) Mehler, E.; Ruedenberg, K.; Silver, D. *Journal of Chemical Physics* **1970**, *52*, 1181–1205.

(38) Silver, D.; Ruedenberg, K.; Mehler, E. *Journal of Chemical Physics* **1970**, *52*, 1206–1227.

(39) Coleman, A. *Journal of Mathematical Physics* **1965**, *6*, 1425–1431.

(40) Coleman, A. *International Journal of Quantum Chemistry* **1997**, *63*, 23–30.

(41) Bajdich, M.; Drobný, G.; Wagner, L.; Schmidt, K. *Physical Review Letters* **2006**, *96*, 130201.

(42) Bajdich, M.; Mitas, L.; Wagner, L.; Schmidt, K. *Physical Review B* **2008**, *77*, 115112.

(43) Pernal, K. *Journal of Chemical Theory and Computation* **2014**, *10*, 4332–4341.

(44) Pastorczak, E.; Pernal, K. *Physical Chemistry Chemical Physics* **2015**, *17*, 8622–8626.

(45) Limacher, P.; Kim, T.; Ayers, P.; Johnson, P.; De Baerdemacker, S.; Van Neck, D. *Molecular Physics* **2014**, *112*, 853–862.

(46) Limacher, P. *Journal of Chemical Physics* **2016**, *145*, 194102.

(47) Johnson, P.; Limacher, P.; Kim, T.; Richer, M.; Miranda-Quintana, R.; Heidar-Zadeh, F.; Ayers, P.; Bultinck, P.; De Baerdemacker, S.; Van Neck, D. *Computational and Theoretical Chemistry* **2017**, *1116*, 207–219.

(48) Boguslawski, K.; Tecmer, P.; Ayers, P.; Bultinck, P.; De Baerdemacker, S.; Van Neck, D. *Physical Review B* **2014**, *89*, 201106.

(49) Boguslawski, K.; Tecmer, P.; Bultinck, P.; De Baerdemacker, S.; Van Neck, D.; Ayers, P. *Journal of Chemical Theory and Computation* **2014**, *10*, 4873–4882.

(50) Silver, D. *The Journal of Chemical Physics* **1969**, *50*, 5108–16.

(51) Limacher, P.; Ayers, P.; Johnson, P.; De Baerdemacker, S.; Van Neck, D.; Bultinck, P. *Journal of Chemical Theory and Computation* **2013**, *9*, 1394–1401.

(52) Johnson, P.; Ayers, P.; Limacher, P.; De Baerdemacker, S.; Van Neck, D.; Bultinck, P. *Computational and Theoretical Chemistry* **2013**, *1003*, 101–13.

(53) Bytautas, L.; Henderson, T.; Jiménez-Hoyos, C.; Ellis, J.; Scuseria, G. *Journal of Chemical Physics* **2011**, *135*.

(54) Bytautas, L.; Scuseria, G.; Ruedenberg, K. *Journal of Chemical Physics* **2015**, *143*, 094105.

(55) Van Raemdonck, M.; Alcoba, D.; Poelmans, W.; De Baerdemacker, S.; Torre, A.; Lain, L.; Massaccesi, G.; Van Neck, D.; Bultinck, P. *Journal of Chemical Physics* **2015**, *143*, 104106.

(56) Alcoba, D.; Torre, A.; Luis, L.; Oña, O.; Capuzzi, P.; Van Raemdonck, M.; Bultinck, P.; Van Neck, D. *Journal of Chemical Physics* **2014**, *141*.

(57) Alcoba, D.; Torre, A.; Luis, L.; Massaccesi, G.; Oña, O.; Ayers, P.; Van Raemdonck, M.; Bultinck, P.; Van Neck, D. *Theoretical Chemistry Accounts* **2016**, *135*, 153.

(58) Carbo, R.; Hernandez, J. *Chemical Physics Letters* **1977**, *47*, 85–91.

(59) Kollmar, C.; Hess, B. *Journal of Chemical Physics* **2003**, *119*, 4655–4661.

(60) Kollmar, C. *Journal of Chemical Physics* **2006**, *125*.

(61) Cook, D. *Molecular Physics* **1975**, *30*, 733–743.

(62) Veillard, A.; Clementi, E. *Theoretica Chimica Acta* **1967**, *7*, 134–143.

(63) Roothaan, C.; Detrich, J.; Hopper, D. *International Journal of Quantum Chemistry* **1979**, *S13*, 93–101.

(64) Weinhold, F.; Wilson Jr, E. *The Journal of Chemical Physics* **1967**, *46*, 2752–58.

(65) Edmonds, J. *Can. J. Math.* **1965**, *17*, 449–467.

(66) Edmonds, J. *J. Res. N. B. S. B.* **1965**, *69*, 125–130.

(67) Galil, Z. *Computing Serveys* **1986**, *18*, 23–38.

(68) Carlen, E.; Lieb, E. H.; Loss, M. *Methods and Applications of Analysis* **2006**, *13*, 1–18.

(69) Chegireddy, C. R.; Hamacher, H. W. *Discrete applied mathematics* **1987**, *18*, 155–165.

(70) Gebauer, R.; Cohen, M. H.; Car, R. *PNAS* **2016**, *113*, 12913–12918.

(71) Xiangzhu, L.; Paldus, J. *The Journal of Chemical Physics* **1995**, *103*, 1024–1034.

(72) Frisch, M. J. et al. Gaussian 16 Revision C.01., Gaussian Inc. Wallingford CT, 2016.

(73) Verstraelen, T.; Tecmer, P.; Heidar-Zadeh, F.; Boguslawski, K.; Chan, M.; Zhao, Y.; Kim, T. D.; Vandenbrande, S.; Yang, D.; González-Espinoza, C. E.; Fias, S.; Limacher, P. A.; Berrocal, D.; Malek, A.; Ayers, P. W. HORTON., version 2.0.1, 2015.

(74) Sun, Q.; Berkelbach, T. C.; Blunt, N. S.; Booth, G. H.; Guo, S.; Li, Z.; Liu, J.; McClain, J. D.; Sayfutyarova, E. R.; Sharma, S., et al. *Wiley Interdisciplinary Reviews: Computational Molecular Science* **2018**, *8*, e1340.

(75) Kim, T. D.; Richer, M.; Sánchez-Díaz, G.; Heidar-Zadeh, F.; Verstraelen, T.; Miranda-Quintana, R. A.; Ayers, P. W. Fanpy., Unpublished Manuscript, 2020.

(76) Hagberg, A. A.; Schult, D. A.; Swart, P. J. In *Proceedings of the 7th Python in Science Conference*, ed. by Varoquaux, G.; Vaught, T.; Millman, J., Pasadena, CA USA, 2008, pp 11–15.

(77) Pople, J.; Seeger, R.; Krishnan, R. *International Journal of Quantum Chemistry: Quantum Chemistry Symposium* **1977**, *11*, 149–163.

(78) Roos, B.; Taylor, P.; Siegbahn, P. *Chemical Physics* **1980**, *48*, 157–173.

(79) Olsen, J.; Roos, B. *The Journal of Chemical Physics* **1988**, *89*, 2185–92.

(80) Schmidt, M.; Gordon, M. *Annual Review of Physical Chemistry* **1998**, *49*, 233–266.

(81) Nesbet, R. *Proceedings of the Royal Society A* **1955**, *230*, 312–321.

(82) Huron, B.; Malrieu, J.; Rancurel, P. *The Journal of Chemical Physics* **1973**, *58*, 5745–5759.

(83) Holmes, A. A.; Tubman, N. M.; Umrigar, C. *Journal of chemical theory and computation* **2016**, *12*, 3674–3680.

(84) Zgid, D.; Gull, E.; Chan, G. *Physical Review B* **2012**, *86*, 165128.

(85) Knowles, P. J.; Werner, H.-J. *Chemical physics letters* **1988**, *145*, 514–522.

(86) Giner, E.; Scemama, A.; Caffarel, M. *Canadian Journal of Chemistry* **2013**, *91*, 879–885.

(87) Evangelista, F. A. *The Journal of Chemical Physics* **2014**, *140*, 124114.

(88) Tubman, N. M.; Lee, J.; Takeshita, T. Y.; Head-Gordon, M.; Whaley, K. B. *The Journal of chemical physics* **2016**, *145*, 044112.

(89) Schriber, J. B.; Evangelista, F. A. Communication: An adaptive configuration interaction approach for strongly correlated electrons with tunable accuracy., 2016.

(90) Liu, W.; Hoffmann, M. R. *Journal of chemical theory and computation* **2016**, *12*, 1169–1178.

(91) Zimmerman, P. M. *The Journal of Chemical Physics* **2017**, *146*, 104102.

(92) Garniron, Y.; Scemama, A.; Loos, P.-F.; Caffarel, M. *The Journal of Chemical Physics* **2017**, *147*, 034101.

(93) Mejuto-Zaera, C.; Tubman, N. M.; Whaley, K. B. *Physical Review B* **2019**, *100*, 125165.

(94) Tubman, N. M.; Freeman, C. D.; Levine, D. S.; Hait, D.; Head-Gordon, M.; Whaley, K. B. *Journal of chemical theory and computation* **2020**, *16*, 2139–2159.

(95) Nightingale, M. P.; Umrigar, C. J., *Quantum Monte Carlo methods in physics and chemistry*; 525; Springer Science & Business Media: 1998.

(96) Umrigar, C. *The Journal of chemical physics* **2015**, *143*, 164105.

(97) Booth, G. H.; Thom, A. J.; Alavi, A. *The Journal of chemical physics* **2009**, *131*, 054106.

(98) Booth, G. H.; Grüneis, A.; Kresse, G.; Alavi, A. *Nature* **2013**, *493*, 365–370.

(99) Li Manni, G.; Smart, S. D.; Alavi, A. *Journal of chemical theory and computation* **2016**, *12*, 1245–1258.

(100) Holmes, A. A.; Changlani, H. J.; Umrigar, C. *Journal of Chemical Theory and Computation* **2016**, *12*, 1561–1571.

(101) Zhao, H.-H.; Ido, K.; Morita, S.; Imada, M. *Physical Review B* **2017**, *96*, 085103.

(102) Tahara, D.; Imada, M. *Journal of the Physical Society of Japan* **2008**, *77*, 114701.

(103) Neuscamman, E. *The Journal of chemical physics* **2013**, *139*, 194105.

(104) Schwarz, L. R.; Alavi, A.; Booth, G. H. *Physical review letters* **2017**, *118*, 176403.

(105) Sorella, S.; Casula, M.; Rocca, D. *The Journal of chemical physics* **2007**, *127*, 014105.

(106) Neuscamman, E. *Journal of chemical theory and computation* **2016**, *12*, 3149–3159.

(107) Kurita, M.; Yamaji, Y.; Morita, S.; Imada, M. *Physical Review B* **2015**, *92*, 035122.

(108) Sabzevari, I.; Sharma, S. *Journal of chemical theory and computation* **2018**, *14*, 6276–6286.

(109) Gabow, H. N.; Tarjan, R. E. *Journal of the Association for Computing Machinery* **1991**, *38*, 815–853.

(110) Duan, R.; Pettie, S.; Su, H. H. *ACM Transactions on Algorithms* **2018**, *14*, 8.

(111) Duan, R.; Pettie, S. *Journal of the ACM* **2014**, *61*, 1.

4.10 Appendix

4.10.1 Explicit Equations corresponding to Figure 4.1

Suppose a Slater determinant has occupied spin orbitals $(1, 2, 3, \bar{1}, \bar{2}, \bar{3})$. The explicit equation for the overlap of the APG wavefunction with this Slater determinant is:

$$
\begin{aligned}
\langle (1,2,3,\bar{1},\bar{2},\bar{3}) | \Psi_{\text{APG}} \rangle &= \sum_{\{i_1,j_1,\ldots,i_3,j_3\} = \{1,2,3,\bar{1},\bar{2},\bar{3}\}} \operatorname{sgn}\Big(\sigma(i_1,j_1,\ldots,i_3,j_3)\Big) \begin{vmatrix} C_{1;i_1j_1} & C_{1;i_2j_2} & C_{1;i_3j_3} \\ C_{2;i_1j_1} & C_{2;i_2j_2} & C_{2;i_3j_3} \\ C_{3;i_1j_1} & C_{3;i_2j_2} & C_{3;i_3j_3} \end{vmatrix}^+ \\
&= \begin{vmatrix} C_{1;12} & C_{1;3\bar{1}} & C_{1;\bar{2}\bar{3}} \\ C_{2;12} & C_{2;3\bar{1}} & C_{2;\bar{2}\bar{3}} \\ C_{3;12} & C_{3;3\bar{1}} & C_{3;\bar{2}\bar{3}} \end{vmatrix}^+
- \begin{vmatrix} C_{1;12} & C_{1;3\bar{2}} & C_{1;\bar{1}\bar{3}} \\ C_{2;12} & C_{2;3\bar{2}} & C_{2;\bar{1}\bar{3}} \\ C_{3;12} & C_{3;3\bar{2}} & C_{3;\bar{1}\bar{3}} \end{vmatrix}^+
+ \begin{vmatrix} C_{1;12} & C_{1;3\bar{3}} & C_{1;\bar{1}\bar{2}} \\ C_{2;12} & C_{2;3\bar{3}} & C_{2;\bar{1}\bar{2}} \\ C_{3;12} & C_{3;3\bar{3}} & C_{3;\bar{1}\bar{2}} \end{vmatrix}^+
- \begin{vmatrix} C_{1;13} & C_{1;2\bar{1}} & C_{1;\bar{2}\bar{3}} \\ C_{2;13} & C_{2;2\bar{1}} & C_{2;\bar{2}\bar{3}} \\ C_{3;13} & C_{3;2\bar{1}} & C_{3;\bar{2}\bar{3}} \end{vmatrix}^+ \\
&+ \begin{vmatrix} C_{1;13} & C_{1;2\bar{2}} & C_{1;\bar{1}\bar{3}} \\ C_{2;13} & C_{2;2\bar{2}} & C_{2;\bar{1}\bar{3}} \\ C_{3;13} & C_{3;2\bar{2}} & C_{3;\bar{1}\bar{3}} \end{vmatrix}^+
- \begin{vmatrix} C_{1;13} & C_{1;2\bar{3}} & C_{1;\bar{1}\bar{2}} \\ C_{2;13} & C_{2;2\bar{3}} & C_{2;\bar{1}\bar{2}} \\ C_{3;13} & C_{3;2\bar{3}} & C_{3;\bar{1}\bar{2}} \end{vmatrix}^+
+ \begin{vmatrix} C_{1;1\bar{1}} & C_{1;23} & C_{1;\bar{2}\bar{3}} \\ C_{2;1\bar{1}} & C_{2;23} & C_{2;\bar{2}\bar{3}} \\ C_{3;1\bar{1}} & C_{3;23} & C_{3;\bar{2}\bar{3}} \end{vmatrix}^+
- \begin{vmatrix} C_{1;1\bar{1}} & C_{1;2\bar{2}} & C_{1;3\bar{3}} \\ C_{2;1\bar{1}} & C_{2;2\bar{2}} & C_{2;3\bar{3}} \\ C_{3;1\bar{1}} & C_{3;2\bar{2}} & C_{3;3\bar{3}} \end{vmatrix}^+ \\
&+ \begin{vmatrix} C_{1;1\bar{1}} & C_{1;2\bar{3}} & C_{1;3\bar{2}} \\ C_{2;1\bar{1}} & C_{2;2\bar{3}} & C_{2;3\bar{2}} \\ C_{3;1\bar{1}} & C_{3;2\bar{3}} & C_{3;3\bar{2}} \end{vmatrix}^+
- \begin{vmatrix} C_{1;1\bar{2}} & C_{1;23} & C_{1;\bar{1}\bar{3}} \\ C_{2;1\bar{2}} & C_{2;23} & C_{2;\bar{1}\bar{3}} \\ C_{3;1\bar{2}} & C_{3;23} & C_{3;\bar{1}\bar{3}} \end{vmatrix}^+
+ \begin{vmatrix} C_{1;1\bar{2}} & C_{1;2\bar{1}} & C_{1;3\bar{3}} \\ C_{2;1\bar{2}} & C_{2;2\bar{1}} & C_{2;3\bar{3}} \\ C_{3;1\bar{2}} & C_{3;2\bar{1}} & C_{3;3\bar{3}} \end{vmatrix}^+
- \begin{vmatrix} C_{1;1\bar{2}} & C_{1;2\bar{3}} & C_{1;3\bar{1}} \\ C_{2;1\bar{2}} & C_{2;2\bar{3}} & C_{2;3\bar{1}} \\ C_{3;1\bar{2}} & C_{3;2\bar{3}} & C_{3;3\bar{1}} \end{vmatrix}^+ \\
&+ \begin{vmatrix} C_{1;1\bar{3}} & C_{1;23} & C_{1;\bar{1}\bar{2}} \\ C_{2;1\bar{3}} & C_{2;23} & C_{2;\bar{1}\bar{2}} \\ C_{3;1\bar{3}} & C_{3;23} & C_{3;\bar{1}\bar{2}} \end{vmatrix}^+
- \begin{vmatrix} C_{1;1\bar{3}} & C_{1;2\bar{1}} & C_{1;3\bar{2}} \\ C_{2;1\bar{3}} & C_{2;2\bar{1}} & C_{2;3\bar{2}} \\ C_{3;1\bar{3}} & C_{3;2\bar{1}} & C_{3;3\bar{2}} \end{vmatrix}^+
+ \begin{vmatrix} C_{1;1\bar{3}} & C_{1;2\bar{2}} & C_{1;3\bar{1}} \\ C_{2;1\bar{3}} & C_{2;2\bar{2}} & C_{2;3\bar{1}} \\ C_{3;1\bar{3}} & C_{3;2\bar{2}} & C_{3;3\bar{1}} \end{vmatrix}^+
+ \begin{vmatrix} C_{1;1\bar{3}} & C_{1;23} & C_{1;\bar{1}\bar{2}} \\ C_{2;1\bar{3}} & C_{2;23} & C_{2;\bar{1}\bar{2}} \\ C_{3;1\bar{3}} & C_{3;23} & C_{3;\bar{1}\bar{2}} \end{vmatrix}^+ \\
&- \begin{vmatrix} C_{1;1\bar{3}} & C_{1;2\bar{1}} & C_{1;3\bar{2}} \\ C_{2;1\bar{3}} & C_{2;2\bar{1}} & C_{2;3\bar{2}} \\ C_{3;1\bar{3}} & C_{3;2\bar{1}} & C_{3;3\bar{2}} \end{vmatrix}^+
+ \begin{vmatrix} C_{1;1\bar{3}} & C_{1;2\bar{2}} & C_{1;3\bar{1}} \\ C_{2;1\bar{3}} & C_{2;2\bar{2}} & C_{2;3\bar{1}} \\ C_{3;1\bar{3}} & C_{3;2\bar{2}} & C_{3;3\bar{1}} \end{vmatrix}^+
\end{aligned}
\tag{4.12}
$$

Similarly, the explicit equation for the overlap of the APsetG wavefunction with this Slater determinant is:

$$
\begin{aligned}
\langle(1,2,3,\bar{1},\bar{2},\bar{3})|\Psi_{\text{APsetG}}\rangle &= \sum_{\substack{\{i_1,i_2,i_3\}=\{1,2,3\}\\ \{j_1,j_2,j_3\}=\{\bar{1},\bar{2},\bar{3}\}}} \operatorname{sgn}\Big(\sigma(i_1,j_1,\ldots,i_3,j_3)\Big) \begin{vmatrix} C_{1;i_1j_1} & C_{1;i_2j_2} & C_{1;i_3j_3} \\ C_{2;i_1j_1} & C_{2;i_2j_2} & C_{2;i_3j_3} \\ C_{3;i_1j_1} & C_{3;i_2j_2} & C_{3;i_3j_3} \end{vmatrix}^{+} \\
&= - \begin{vmatrix} C_{1;1\bar{1}} & C_{1;2\bar{2}} & C_{1;3\bar{3}} \\ C_{2;1\bar{1}} & C_{2;2\bar{2}} & C_{2;3\bar{3}} \\ C_{3;1\bar{1}} & C_{3;2\bar{2}} & C_{3;3\bar{3}} \end{vmatrix}^{+} + \begin{vmatrix} C_{1;1\bar{1}} & C_{1;2\bar{3}} & C_{1;3\bar{2}} \\ C_{2;1\bar{1}} & C_{2;2\bar{3}} & C_{2;3\bar{2}} \\ C_{3;1\bar{1}} & C_{3;2\bar{3}} & C_{3;3\bar{2}} \end{vmatrix}^{+} + \begin{vmatrix} C_{1;1\bar{2}} & C_{1;2\bar{1}} & C_{1;3\bar{3}} \\ C_{2;1\bar{2}} & C_{2;2\bar{1}} & C_{2;3\bar{3}} \\ C_{3;1\bar{2}} & C_{3;2\bar{1}} & C_{3;3\bar{3}} \end{vmatrix}^{+} - \begin{vmatrix} C_{1;1\bar{2}} & C_{1;2\bar{3}} & C_{1;3\bar{1}} \\ C_{2;1\bar{2}} & C_{2;2\bar{3}} & C_{2;3\bar{1}} \\ C_{3;1\bar{2}} & C_{3;2\bar{3}} & C_{3;3\bar{1}} \end{vmatrix}^{+} \\
&\quad - \begin{vmatrix} C_{1;1\bar{3}} & C_{1;2\bar{1}} & C_{1;3\bar{2}} \\ C_{2;1\bar{3}} & C_{2;2\bar{1}} & C_{2;3\bar{2}} \\ C_{3;1\bar{3}} & C_{3;2\bar{1}} & C_{3;3\bar{2}} \end{vmatrix}^{+} + \begin{vmatrix} C_{1;1\bar{3}} & C_{1;2\bar{2}} & C_{1;3\bar{1}} \\ C_{2;1\bar{3}} & C_{2;2\bar{2}} & C_{2;3\bar{1}} \\ C_{3;1\bar{3}} & C_{3;2\bar{2}} & C_{3;3\bar{1}} \end{vmatrix}^{+} - \begin{vmatrix} C_{1;1\bar{3}} & C_{1;2\bar{1}} & C_{1;3\bar{2}} \\ C_{2;1\bar{3}} & C_{2;2\bar{1}} & C_{2;3\bar{2}} \\ C_{3;1\bar{3}} & C_{3;2\bar{1}} & C_{3;3\bar{2}} \end{vmatrix}^{+} + \begin{vmatrix} C_{1;1\bar{3}} & C_{1;2\bar{2}} & C_{1;3\bar{1}} \\ C_{2;1\bar{3}} & C_{2;2\bar{2}} & C_{2;3\bar{1}} \\ C_{3;1\bar{3}} & C_{3;2\bar{2}} & C_{3;3\bar{1}} \end{vmatrix}^{+}
\end{aligned}
\tag{4.13}
$$

Finally, the explicit equation for the overlap of the APIG wavefunction with this Slater determinant is:

$$
\langle(1,2,3,\bar{1},\bar{2},\bar{3})|\Psi_{\text{APIG}}\rangle = - \begin{vmatrix} C_{1;1\bar{1}} & C_{1;2\bar{2}} & C_{1;3\bar{3}} \\ C_{2;1\bar{1}} & C_{2;2\bar{2}} & C_{2;3\bar{3}} \\ C_{3;1\bar{1}} & C_{3;2\bar{2}} & C_{3;3\bar{3}} \end{vmatrix}^{+} \tag{4.14}
$$

4.10.2 Proof for Equation 4.10

The proof is by induction. Starting with the 2 × 2 permanent.

$$\text{abs}\left(\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}^{+}\right) = |a_{11}a_{22}+a_{12}a_{21}| \leq (|a_{11}|+|a_{21}|)(|a_{12}|+|a_{22}|) = |a_{11}a_{22}|+|a_{12}a_{21}|+|a_{11}a_{12}|+|a_{21}a_{22}| \tag{4.15}$$

$$\begin{aligned}
&\text{abs}\left(\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}^{+}\right) = |a_{11}|\begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix}^{+} + |a_{21}|\begin{vmatrix} a_{12} & a_{13} \\ a_{32} & a_{33} \end{vmatrix}^{+} + |a_{31}|\begin{vmatrix} a_{12} & a_{13} \\ a_{22} & a_{23} \end{vmatrix}^{+} \\
&\leq (|a_{11}|+|a_{21}|+|a_{31}|)\max\left(\text{abs}\left(\begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix}^{+}\right), \text{abs}\left(\begin{vmatrix} a_{12} & a_{13} \\ a_{32} & a_{33} \end{vmatrix}^{+}\right), \text{abs}\left(\begin{vmatrix} a_{12} & a_{13} \\ a_{22} & a_{23} \end{vmatrix}^{+}\right)\right) \\
&\leq (|a_{11}|+|a_{21}|+|a_{31}|)\max\Big((|a_{22}|+|a_{32}|)(|a_{23}|+|a_{33}|), (|a_{12}|+|a_{32}|)(|a_{13}|+|a_{33}|), (|a_{12}|+|a_{22}|)(|a_{13}|+|a_{23}|)\Big) \\
&\leq (|a_{11}|+|a_{21}|+|a_{31}|)(|a_{12}|+|a_{22}|+|a_{32}|)(|a_{13}|+|a_{23}|+|a_{33}|)
\end{aligned} \tag{4.16}$$

For the inductive step, we generalize to the $n \times n$ case

$$|A|^{+} = \sum_{i=1}^{N} a_{i1}M_{i1} \tag{4.17}$$

where M_{ij} is the minor of A where the row i and column 1 were removed. Then,

$$\begin{aligned}
\mathrm{abs}(|A|^{+}) &\leq \left(\sum_{i=1}^{N} |a_{i1}|\right) \max_i \Big(\mathrm{abs}(M_{i1})\Big) \\
&\leq \left(\sum_{i=1}^{N} |a_{i1}|\right) \max_i \left(\prod_{j=2}^{N} \sum_{k \neq i}^{N} |a_{kj}|\right) \\
&\leq \left(\sum_{i=1}^{N} |a_{i1}|\right) \left(\prod_{j=2}^{N} \sum_{i=1}^{N} |a_{ij}|\right) \\
&\leq \prod_{j=1}^{N} \sum_{i=1}^{N} |a_{ij}|
\end{aligned} \tag{4.18}$$

Chapter 5

Applying Concepts from Machine Learning to Solve the Schrödinger Equation

Applying Concepts from Machine Learning to Solve the Schrödinger Equation

Taewon D. Kim[†] Toon Verstraelen[‡] Paul W. Ayers[†]

September 26, 2020

[†]Department of Chemistry and Chemical Biology, McMaster University, Hamilton, Ontario, L8S-4L8, Canada

[‡]Center for Molecular Modeling (CMM), Ghent University, Technologiepark 46, B-9052, Zwijnaarde, Belgium

In traditional *ab initio* molecular electronic structure theory, a mathematical formalism is derived from fundamental physical laws, and then computational implementation of the mathematical formalism is designed. This approach requires physical and chemical insight, obtained through first-hand computational experience and reading the literature. Over the last few decades, machine learning has emerged as an alternative to this analytic approach. In machine-learning (ML) approaches, the focus is on finding a function that fits the inputs of the dataset to their corresponding outputs[1–4]. Using ML, difficult problems can be modelled using only a dataset of observations, including applications like simulating fluids[5, 6] and stock market prediction[7–9]. In these cases, the model emerges directly from the data and is sculpted with the goal of reproducing observations. These supervised learning algorithms are largely agnostic about the nature of the underlying data, and can be used for a wide range of datasets. One ML model, called a neural network, has been particularly successful in modelling difficult problems in cases where abundant data is available. Many of the recent advances in image recognition[10–13], speech recognition[14–16], and language processing[17–20] use neural networks.

Neural network models have been known since the 1950's[21, 22], but their emergence is due, in large part, to the availability of new types of computer hardware, the agglomeration of large datasets via the internet, and the development of better optimization algorithms[23–27]. In this chapter, we explore possible uses of the neural network model and a popular optimization strategy, stochastic gradient descent (SGD), for solving the Schrödinger equation. The neural network model is incorporated into the wavefunction using the framework proposed in Chapter 2 and the SGD algorithm is used to solve the projected Schrödinger equation is presented in Section 1.6. The concept of sampling data points in SGD is generalized to adaptively sample the projection space.

5.1 Neural Network

5.1.1 Overview of Neural Networks

A neural network, at its core, involves a sequence of linear transformation followed by a nonlinear transformation[1, 2]. The simplest neural network, a feed-forward neural network, has the following structure:

$$f(\mathbf{x}) = \sigma_L \left(w_0^{(L-1)} + \sum_{j_{L-1}=1}^{K_{L-1}} w_{j_{L-1}}^{(L-1)} \dots \sigma_2 \left(w_0^{(1)} + \sum_{j_1=1}^{K_1} w_{j_2 j_1}^{(1)} \sigma_1 \left(w_0^{(0)} + \sum_{i=1}^{N} w_{j_1 i}^{(0)} x_i \right) \right) \dots \right) \tag{5.1}$$

This neural network consists of L layers, each layer k consisting of linear transformation via $w_{j_{k+1} j_k}^{(k)}$ and $w_0^{(k)}$ and nonlinear transformation via the function σ_k. The input is linearly transformed into K_1 units in the 1st layer, each of which is transformed using a nonlinear function σ_1. These units are then linearly transformed into K_2 units in the 2nd layer, then nonlinear transformed with σ_2. This process is repeated until the final layer, which produces the output. The graphical representation of this neural network is given in Figure 5.1. The first layer is called the input layer, the last layer is called the output layer, and layers in between are called hidden layers. The units of each hidden layer are called hidden units and the nonlinear functions are called activation functions. One interpretation of the neural network is that it is a linear combination of adaptable basis functions where both the basis functions and the linear combinations are parameterizable and, thus, optimizable.

Normally, neural networks are optimized to minimize the residual sum of squares (Equation 5.2) in which the neural network, f, maps the input, $\mathbf{x}_i$, to the output, $f(\mathbf{x}_i)$, such that it most closely resembles the target value, t_i, for every data point, $\{\mathbf{x}_i, t_i\}$.

$$\begin{aligned} R_{\text{emp}}(\mathbf{P}) &= \frac{1}{M} \sum_{i=1}^{M} \Big(f(\mathbf{x}_i|\mathbf{P}) - t_i \Big)^2 \\ &= \frac{1}{M} \sum_{i=1}^{M} L(\mathbf{x}_i, t_i|\mathbf{P}) \end{aligned} \tag{5.2}$$

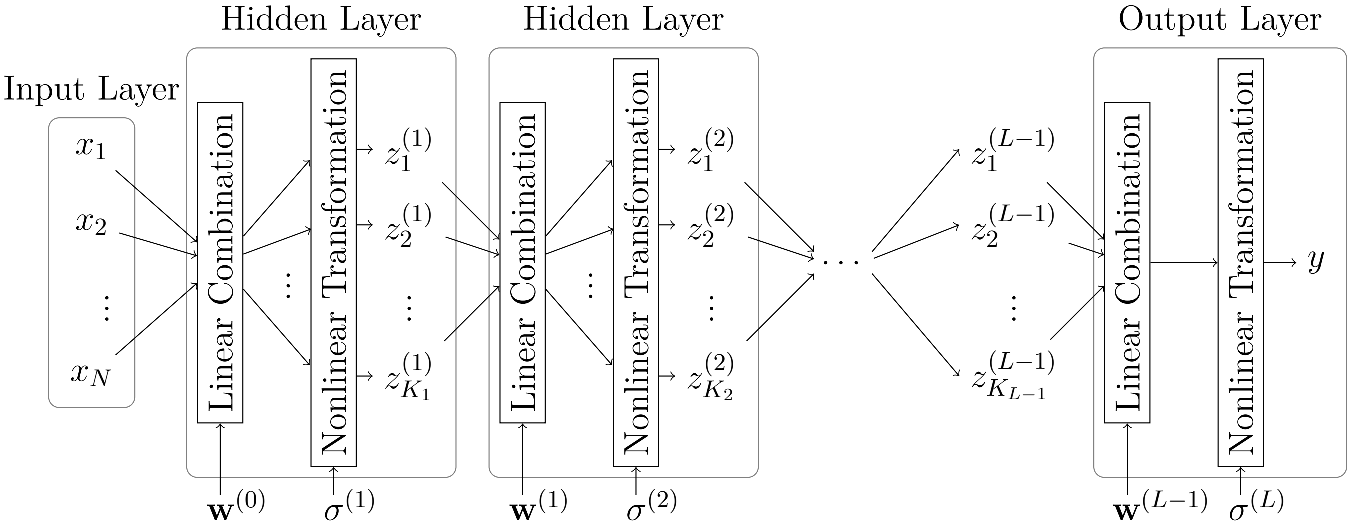

Figure 5.1: L-Layer Feed-Forward Neural Network

Optimizing the neural network in this manner is termed “training” the network. Since training a network is equivalent to finding a function that approximates the target values for each data point, the accuracy of the neural network is limited by the functions that can be represented with a neural network given its structure and parameters. However, a simple feed-forward neural network can approximate any continuous function to an arbitrary accuracy by increasing its width (number of hidden units)[28, 29] or by increasing its depth (number of layers)[30, 31]. Though the performance of the network depends on the activation functions used, computationally friendly (i.e., easy to differentiate and evaluate) activation functions are preferred since their deficiencies can be remedied with a more complex network structure (with more hidden units or layers).

The cost of evaluating a neural network (with a sufficiently large number of weights) scales as $\mathcal{O}(W)$, where W is the number of weights[1]. The evaluation of derivatives (with respect to weights and input) scales as $\mathcal{O}(W^2)$. In addition, neural networks are easily parallelized, which allows neural networks to make efficient use of computational resources for large datasets.

5.1.2 Applications to Quantum Chemistry

Can neural networks be used to solve the Schrödinger equation[32]? There is precedent for using neural networks to represent wavefunctions in real space for model systems like harmonic oscillators and hydrogen atoms[33–37]. In these cases, the input to the network is the coordinates of the wavefunction and the outputs are the values of the wavefunction at these coordinates. Though the results are promising, the number of data points (the coordinates and their corresponding wavefunction values) necessarily grows exponentially with the number of particles, rendering the network too expensive to train except for a small number of particles[38, 39]. Manzhos et al.[40, 41] partly address this issue by using the radial basis neural network and a specialized optimization algorithm that involves solving its linear coefficients as a generalized eigenvalue problem. However, solving the Schrödinger equation in real-space is not competitive to methods based on basis sets due to poor scaling in heavier atoms[38, 39]. Nonetheless, by introducing more complex structures to the neural network, the neural network can become easier to optimize and require fewer parameters for the same level of accuracy.

In another application, neural networks are given atomic information to produce molecular energies[42–44]. These networks are designed such that the features of an atom for one molecular system can be transferred to the network for other systems. Though these networks show promising results for medium-sized organic molecules in their ground electronic states, it is likely difficult to train the network for large systems with heavier elements, as the scaling seems to grow as the square of the number of distinct elements. Moreover, heavier elements tend to have a wider range of complex and nuanced behaviour than ground-state organic systems due to their range of oxidation states and electron configurations. Complex chemical behaviours may be better described by providing the network with quantum chemical information, such as orbitals in the form of Fock, Coulomb, and exchange integrals[45, 46] or one-electron reduced density matrix[47]. However, building training data that encompass all such phenomena seems implausible. Moreover, larger systems have larger energies, so to maintain the same level of chemical accuracy ($\sim$1 kcal/mol on energy differences) either the relative accuracy of the model must increase with molecule size or, as with more traditional quantum chemistry methods, the model must be designed to benefit from the cancellation of errors or transferability across different chemical structures. In particular, obtaining even a single piece of training data for a large metal complex (e.g., a biological iron-sulfur complex) is prohibitively expensive[48].

More recently, Mills et al.[49] constructed a mapping from two-dimensional potentials to energies using a convolutional neural network. Convolutional neural networks have structures that group together inputs and hidden units that are closer to one another. It has been shown to be incredibly powerful in processing visual information since it naturally compartmentalizes into spatially relevant pieces[50]. However, the proposed network is restricted by the dataset from which it is trained: the size of the grids is fixed, meaning that larger systems will either have a low resolution or the entire neural network needs to be re-trained with a larger grid. While Mills et al. exploited the locality of potentials to determine wavefunctions, Carleo & Troyer[51] build a wavefunctions in terms of its spin states using the restricted Boltzmann machine architecture.

In contrast, Han et al.[52] and Hermann et al.[53] build neural networks whose inputs are in real-space (i.e. positions of electrons and nuclei): Han et al.[52] builds a wavefunction as a deep neural network for the real-space quantum Monte Carlo algorithm; and Hermann et al.[53] builds a wavefunction via a multipart neural network architecture that creates a set of nonorthogonal Slater determinants which are then linearly combined. Like other traditional quantum chemistry methods, these networks do not require a dataset and are optimized to solve the Schrödinger equation by minimizing the energy variationally. Consquently, the networks of Carleo & Troyer, Han et al., and Hermann et al. fix the number of particles (e.g. spin variables and electron positions), and a change in the number of particles requires re-training. In a similar spirit, the network proposed by Schütt[54] and Hegde[55] predict the intermediary quantities used to build the Schrödinger equation. Specifically, it predicts features that can be used to build the one-electron Hamiltonian matrix, which can then be diagonalized to obtain the wavefunction and energy of a single-determinant wavefunction.

5.1.3 Objective Functions

Looking at these applications of neural networks to quantum chemistry, we can identify the possible problems that arise. Specifically, in order to develop a neural network that can reliably model the behaviour of complex chemical systems: (1) there must be enough data for the given training method, (2) an optimized network should be reusable for other systems or dataset, and (3) the network should be structured such that it can handle inputs of different sizes.

Solving the Schrödinger equation is an NP-hard problem and therefore intractable for larger systems[56]. If a network can successfully solve the Schrödinger equation, whether directly by finding the N-electron wavefunction or implicitly by directly predicting energies or other observables, the network is attempting to solve an NP-hard problem. Since NP-hard problems are unlikely to be solved tractably without approximations[57], the network will either be intractably expensive to optimize or fail to provide accurate results for many systems. A common compromise is to develop a method that can model a subset of the chemical space while ignoring the rest. These methods tend to be accurate for the systems that resemble the training data but fail catastrophically if misapplied outside their training domain. For example, the aforementioned networks above that were designed for small organic systems in their ground electronic state near-equilibrium geometries are likely to fail, catastrophically, for large inorganic systems near transition states. Therefore, to handle a wider range of systems, the network must have a larger dataset from which it is trained.

However, it is difficult to obtain a large high-quality dataset that contains complex chemical systems. Experimental results are often unavailable and accurate calculations are too expensive to obtain in bulk. We can use cheaper and less accurate calculations, but the network trained on these calculations will have (at best) the same level of accuracy. If there are not enough data to train the network, the network must be trained with an objective different from the traditional residual sum of squares (Equation 5.2). Therefore, we feel that if the goal of training the neural network is to solve the Schrödinger equation (Equation 5.3) by reproducing the dataset that corresponds to its solutions, the most straight-forward objective is one that directly solves the Schrödinger equation.

$$\hat{H} |\Psi\rangle = E |\Psi\rangle \tag{5.3}$$

One way is to minimize the expected energy:

$$E = \frac{\langle\Psi|\hat{H}|\Psi\rangle}{\langle\Psi|\Psi\rangle} \tag{5.4}$$

The expected energy follows the variational principle, which means that the calculated energy will be bound from below by the true energy[58–60]. Therefore, optimizing the wavefunction to minimize its energy results in an approximate solution that most closely reproduces the true energy. Another approach is to solve the projected Schrödinger equation:

$$\begin{aligned} \langle\Phi_1|\hat{H}|\Psi\rangle - E \langle\Phi_1|\Psi\rangle &= 0 \\ &\vdots \\ \langle\Phi_M|\hat{H}|\Psi\rangle - E \langle\Phi_M|\Psi\rangle &= 0 \end{aligned} \tag{5.5}$$

By integrating the Schrödinger equation against a set of states (which define the projection space), a system of nonlinear equations is obtained[61–63]. Optimizing the wavefunction to satisfy this system of equations results in an approximate solution that most closely satisfies the Schrödinger equation within the space spanned by the projection space. This approach satisfies the variational principle if the projection space is complete (contains all possible Slater determinants)[64].

In the examples above, neural networks were incorporated into the Schrödinger equation by building the wavefunction using a network architecture[51] and by producing an intermediary quantity used to solve the Schrödinger equation[54]. Similarly, a network can represent a multideterminant wavefunction by producing the overlap of the wavefunction for the given Slater determinant:

$$|\Psi\rangle = \sum_{\mathbf{m}} g(\mathbf{m}) |\mathbf{m}\rangle \tag{5.6}$$

where $\mathbf{m}$ is an occupation vector that corresponds to a Slater determinant and $g(\mathbf{m})$ is the neural network. These overlaps act as the intermediary quantities for solving the Schrödinger equation since both Equations 5.4 and 5.5 are expressed with respect to these overlaps. If this approach is taken, the neural network can be considered a wavefunction ansatz, specifically a wavefunction ansatz that fits within the Flexible Ansatz for N-electron Configurate Interaction (FANCI) that we recently presented[65]. The overlaps of the Slater determinants are available in great abundance so constructing a large dataset is not a problem; in fact, the exponential number of Slater determinants is the reason that the accurate calculations are expensive, but here this exponential number of overlaps is exploited as training data. This network can be optimized to satisfy the Schrödinger equation associated with a Hamiltonian *and* to minimize the residual sum of squares associated with a dataset.

For the moment, assume that the neural network will be optimized using only the Schrödinger equation and that the input to the neural network is only the occupation vector of the Slater determinant, without any other system-dependent information. In other words, the network needs to be optimized to solve the Schrödinger equation for each system separately. Before a network can be designed that can simultaneously solve many chemical systems, it should solve a single system. Though structural changes are necessary to incorporate system-specific information, such as geometry, nuclear charges, and the Hamiltonian, the lessons learned from designing the network that solves one chemical system can be used to design the network that can solve multiple systems. These networks can share much of their structures because they both involve mapping the occupation vector to the overlap. Furthermore, the limitations of a network can be quickly demonstrated on the network that solves a single system since its performance can serve as an upper bound to the performance of the network that solves multiple systems simultaneously.

When optimizing a wavefunction to solve the Schrödinger equation, it is important to have a good initial guess. A fully random initial guess, commonly used in neural networks, is not viable because of the second-quantized Hamiltonian has a factorial number of eigenstates, only one of which is the desired ground state. A random guess, then, is overwhelmingly likely to optimize to an excited-state solution of the Schrödinger equation. Instead, it is preferable to use another (optimized) wavefunction as a starting point to the calculation. We attempted to use the optimized ground-state Hartree-Fock (HF) wavefunction, with some random noise to help escape the local minima. Since the HF wavefunction is a single determinant method, the neural network must return 1 for the ground-state Slater determinant and 0 otherwise to reproduce its behaviour. The Hartree-Fock network then serves as a starting point for another network. If the networks have the same structure, i.e. they share the same number of layers, hidden units, and the connectivity between them, then the network can simply be transferred and optimized for the other systems. When the networks have different structures, the network needs to be re-expressed in terms of the structure of the other network, sometimes requiring transformation of its parameters. The advantage of incorporating a network of a different structure is that the complexity of the network, and consequently the difficulty of the optimization process, can be controlled. For example, a network with fewer layers and hidden units will be easier to optimize, but will not be as accurate. By representing this network (after optimization) as one with additional layers or hidden units, the network will provide better results with an easier optimization process.

Here, we build a simple feed-forward network for which we can easily obtain its HF initial guess. Producing an HF initial guess is straight forward if the network produces N values, rather than 1. To obtain the overlap, the N values are multiplied together. To ensure that the final output ranges between -1 and 1, a typical range for the overlap of a normalized wavefunction with a Slater determinant, an appropriate activation function can be applied to the product. This step can be interpreted as a post-processing step of a neural network that produces N outputs, or as a specialized layer that pools together units of the previous layer. This layer can also be loosely interpreted as sequence of logarithmic activation functions, followed by a sum of the produced units, then an exponential activation function (note that this is not technically correct due to the domain of the logarithm function and the range of the exponential function).

The HF initial guess can be reproduced with the following weights:

$$\begin{aligned} w_{ij} &= \begin{bmatrix} 1 & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0 & \dots & 0 \end{bmatrix}_{N \times 2K} \\ &= \begin{cases} 1 & \text{if } i = j \\ 0 & \text{else} \end{cases} \end{aligned} \tag{5.7}$$

Again, the first N out of $2K$ spin orbitals are assumed to be occupied in the ground-state Slater determinant. After the linear transformation, only the ground-state Slater determinant will produce a vector that does not contain a zero. Multiplying together the entries of this vector reproduces the overlaps of the HF wavefunction. After the multiplication, an activation function can be applied provided that it returns 0 with a 0 input and that it ranges between -1 and 1. If not, it can simply be scaled to satisfy these conditions. One activation function that satisfies these conditions is based on the hyperbolic tangent (Equation 5.8).

$$\begin{aligned} \sigma(x) &= \frac{\tanh(x)}{\tanh(1)} \\ &= \frac{(e^{2x} + 1)(e^2 - 1)}{(e^{2x} - 1)(e^2 + 1)} \end{aligned} \tag{5.8}$$

The scale factor $\tanh(1)^{-1}$ ensures that the HF initial guess is normalized.

The structure above can be interpreted as the output layer of the network and more layers and hidden units can be added accordingly. Each additional hidden layer can be initialized to a linear transformation by choosing the weights as an identity matrix scaled according to the activation function used in the previous layer:

$$\mathbf{w}^{(i)} = \begin{bmatrix} \frac{1}{\sigma^{(i-1)}(1)} & 0 & \cdots & 0 \\ 0 & \frac{1}{\sigma^{(i-1)}(1)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{1}{\sigma^{(i-1)}(1)} \end{bmatrix}_{2K \times 2K} \tag{5.9}$$

where $2K$ is the number of spin orbitals and the size of the occupation vector. The scaling is necessary to ensure that the output of the network are all 1 for the HF ground state input. Hidden units can be added to a given layer by adding rows of zeros to the weights of the previous layer and columns of zeros to the current layer. In both of these modifications, the behaviour of the network is unchanged, which means that they can be used to find the HF initial guess or to modify an existing network to improve its complexity. The graphical representation of the L-layer neural network wavefunction is given in Figure 5.2. Here, we use the hyperbolic tangent as the activation function for all of the layers, except for the output layer, which uses a modified hyperbolic tangent given by Equation 5.8. Each hidden layer is assumed to have $2K$ units.

The output of this neural network can be interpreted as a product-wise decomposition of the overlap into the contributions by each electron. Though uncommon in neural networks, multiplying together contributions by components of a Slater determinant is very common when constructing wavefunctions. For example, in quasiparticle wavefunctions[65, 66], a Slater determinant is expressed in terms of its contributing quasiparticles: the HF wavefunction with nonorthogonal orbitals divides a Slater determinant into contributions by each electron[58, 60], and geminal product wavefunctions

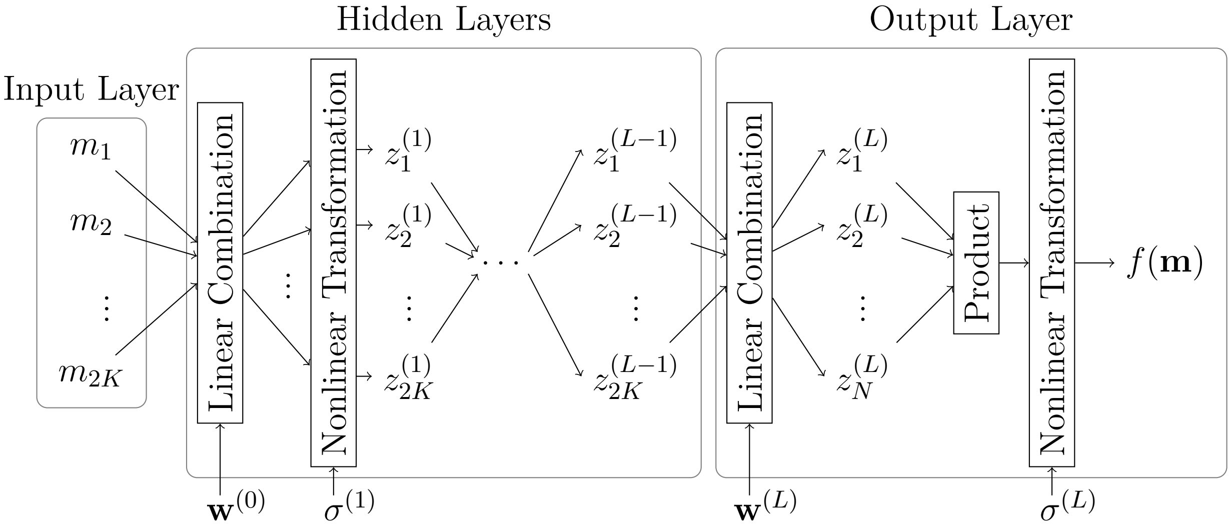

FIGURE 5.2: L-Layer Neural Network Wavefunction

divide a wavefunction into contributions by electron pairs[67–76]. In Coupled-Cluster wavefunctions, the wavefunction is a product of excitations upon a reference determinant[77–80]. However, the overlaps of these wavefunctions are more complicated than a simple product, requiring a sum over permutations arising from the antisymmetry property in the case of quasiparticles and a sum over partitions arising from the use of the exponential operator in the case of Coupled-Cluster. These complications arise from the underlying approximations from which the wavefunctions are derived. Owing to this structure involving a product, the proposed neural network wavefunction is size-consistent[81] - multiplicatively separable into two noninteracting components[65] - without the complication. Though the proposed wavefunction was designed without regard for its theoretical implications, it is, by construction, a universal approximation and may even be interpretable, especially if an appropriate initial guess was used. Note that the wavefunction constructed using this neural network is antisymmetric because the underlying basis functions, Slater determinants, are antisymmetric.

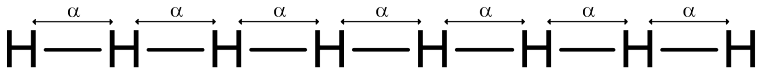

FIGURE 5.3: Linear H_8 chain: $\alpha \in \{0.6, 0.7, 0.8, 0.9, 1, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2, 2.25, 2.5, 3, 4\}$ Angstroms

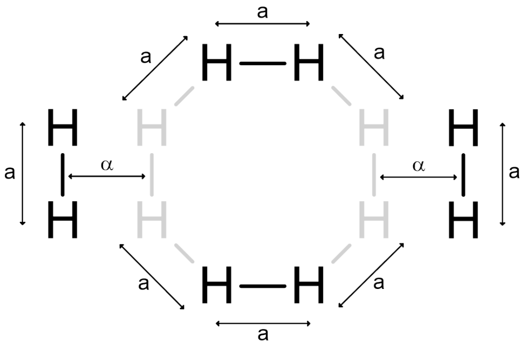

FIGURE 5.4: Octagonal H_8: a = 2 a.u., $\alpha \in \{0, 0.0001, 0.001, 0.003, 0.006, 0.01, 0.03, 0.06, 0.1, 0.5, 1\}$ a.u.

5.1.4 Results

This wavefunction was implemented using `Fanpy`[82] and was tested on two H_8 systems in the ANO-1s basis set: stretching of an H_8 chain (Figure 5.3)[75, 83] and the elongation of a ring of four H_2 molecules (Figure 5.4)[84]. The energies of these systems are given by Figure 5.5 and 5.6.

As the number of layers increases, the energies of the neural network wavefunctions approach the FCI energy. The largest improvements come from the addition of one and two hidden layers; adding additional layers gives small, but systematic, further improvements. As the number of layers increases, the wavefunction gains enough complexity to represent the FCI wavefunction, but this also increases the number of local minima and

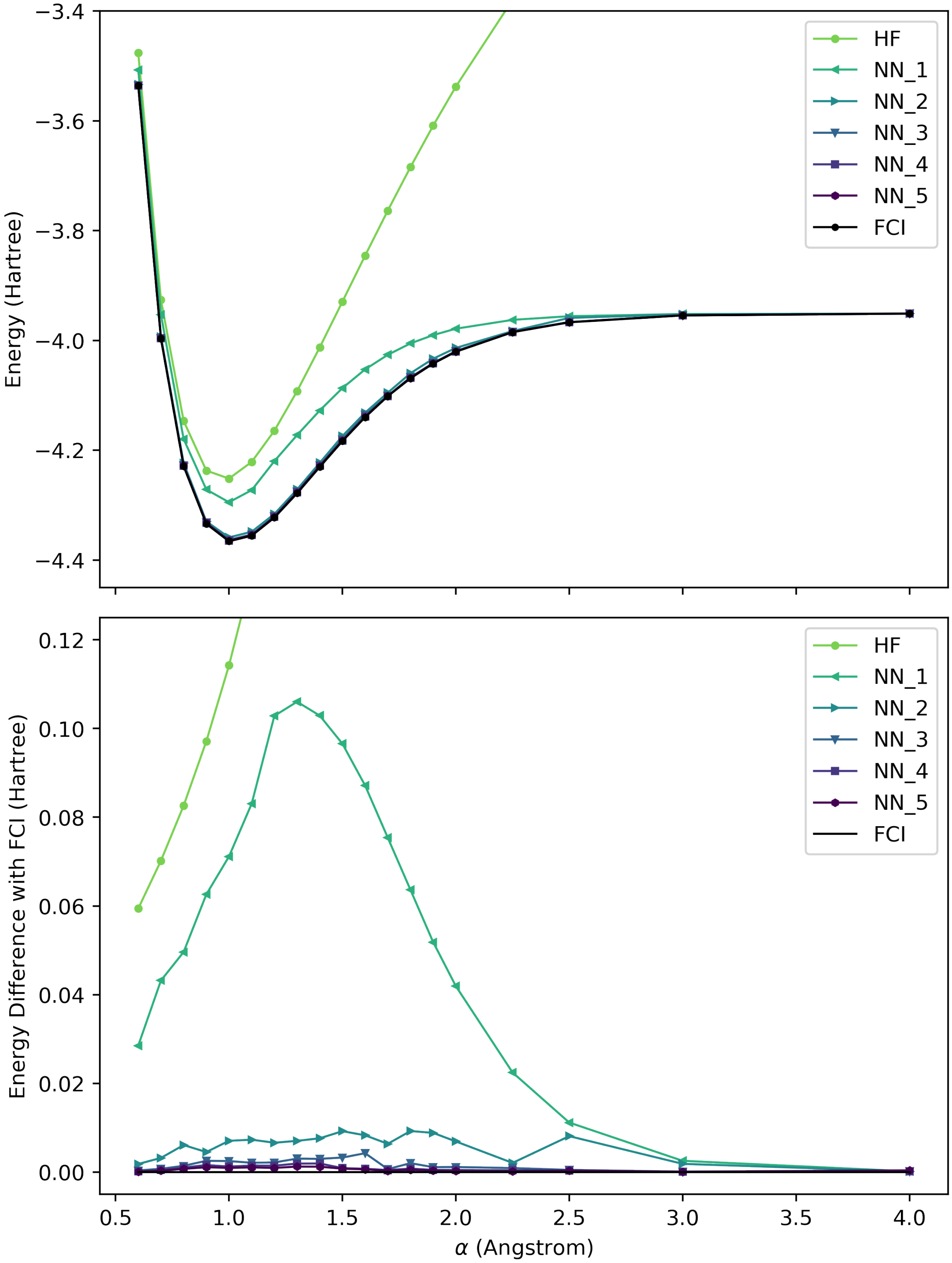

FIGURE 5.5: Energies and energy differences with APG wavefunction in linear H_8 systems

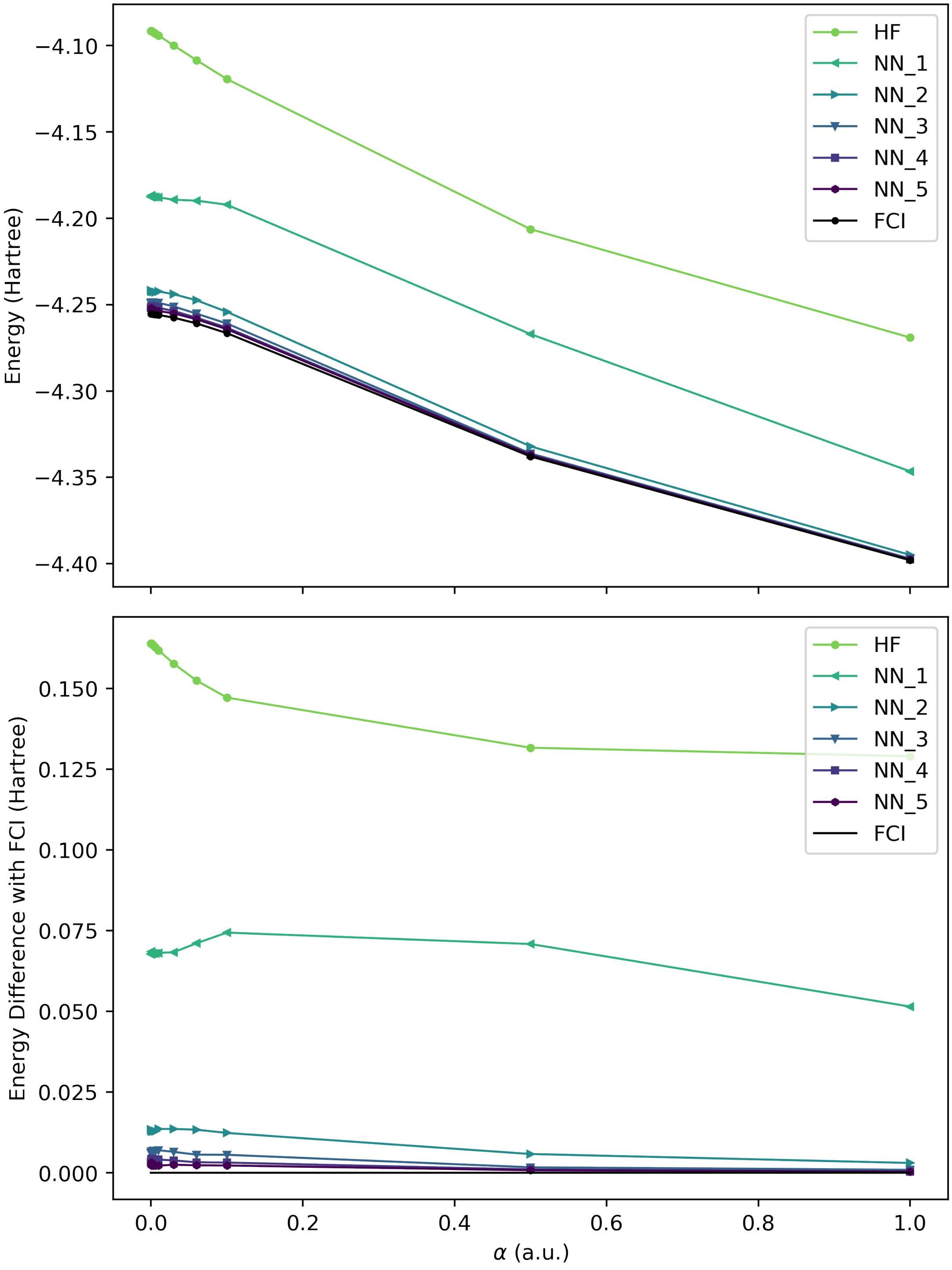

FIGURE 5.6: Energies and energy differences with APG wavefunction in H_8 ring systems

makes it difficult to find the global minimum that would reproduce FCI. In our calculations, we use the optimized result from the network with one fewer layer and, when possible, estimate initial parameters for systems from other systems with nearby geometries. Though the results are promising (neural networks provide accurate results, are cheap to evaluate, and can be parallelized efficiently), without an effective optimization algorithm, large-scale high-throughput calculations involving this wavefunction may be prohibitively difficult.

5.1.5 Generalization

Furthermore, the presented network is designed to be solved for each system independently. Though it is possible to use an optimized network as an initial guess for a similar system, this initial guess will not be too useful for the systems that are quite different, especially those with different numbers of spin-orbitals or electrons. Changing the number of spin orbitals changes the number of input variables in the network, requiring the subsequent layers to be changed. Likewise, changing the number of electrons affects the final layer of the network, affecting the previous layers. When adding layers and hidden units to improve the network's complexity, the behaviour of the network is unaffected. However, when the number of spin orbitals or electrons is decreased, it will be unclear which rows and columns need to be removed and their removal will have a significant impact on the network. Conversely, if the number of spin orbitals or electrons is increased, we can use the HF initial guess as a guideline and fill in the difference, but this approach assumes that the behaviour of a wavefunction of a higher number of spin orbitals and electrons is similar to that with fewer spin orbitals and electrons. Though wavefunctions with different numbers of spin orbitals behave similarly, especially for those of large basis sets, wavefunctions of different numbers of electrons can be qualitatively different due to, for example, redox-induced electron transfer[85–87]. Since the network was designed solely to solve the Schrödinger equation associated with the given Hamiltonian, an optimized network will not be close to a solution if the Hamiltonian changes significantly - it can be used as an initial guess only for small changes in the Hamiltonian (e.g., small changes in molecular geometry).

In the ideal case, a single neural network should simultaneously model multiple independent systems. Some questions seem to suggest that it might be very difficult or, worse, not practical to find and optimize such a network. Is it possible to solve an NP-hard problem (i.e. solving the Schrödinger equation) to the desired level of accuracy in polynomial (real world) time[56, 57, 88]? This problem becomes increasingly more difficult as the system gets larger because the problem must be solved more accurately, relatively, to obtain chemical accuracy (~ 0.001 Hartree or 1 kcal/mol) It is possible that modeling the overlaps of Slater determinants will be more forgiving with possibilities of cancellation of errors. Is solving the Schrödinger equation for the desired region of chemical space an NP-hard problem? This, of course, depends on the systems considered and becomes easier as less systems are considered. Since large systems are prohibitively expensive to solve and have greater chemical diversity and show emergent electron-correlation phenomena (e.g., superconductivity[89, 90]), it is essential for the network to generalize from small systems to large systems.

In contrast to the simple feed-forward network proposed here, a sophisticated network structure is necessary to generalize the network to multiple systems. Though the specifics of the network structure that can generalize the observations of small chemical systems to that of large chemical systems is beyond the scope of the current study, some network structures are worth noting. As was used with SchNet[44] and SchNOrb[54], structure incorporating convolution will be useful in filtering out the unimportant features allowing the useful features to be pooled together. In the case of electronic structure, the number of possible interactions between the orbitals increases exponentially with the size of the system. If these interactions can be limited to a fixed set of features for both large and small systems, convolutions may be important to support systems of different sizes. Since many properties in quantum chemistry are often described with a graph, a graph convolution neural network may be an important structure to consider. In addition, long short-term memory and other recurrent network structures have shown great promise for inputs of arbitrary sizes. Though many state-of-the-art network architectures may hint at an out-of-the-box solution for solving the Schrödinger equation, much work is likely needed to adapt these networks for the problem at hand: to figure out how these structural components and the system-specific information, such as the one- and two-electron integrals of the Hamiltonian or the nuclear coordinates and charges, fit together within the context of the Schrödinger equation.

5.2 Stochastic Gradient Descent for the Projected Schrödinger Equation

As mentioned above, one approach to solving the Schrödinger equation is to solve the system of nonlinear equations obtained by integrating the Schrödinger equation with a set of states, $\{\Phi_1, \dots, \Phi_M\}$, called projection space[61–63]:

$$\begin{aligned} \langle \Phi_1 | \hat{H} | \Psi \rangle &= E \langle \Phi_1 | \Psi \rangle \\ &\vdots \\ \langle \Phi_M | \hat{H} | \Psi \rangle &= E \langle \Phi_M | \Psi \rangle \end{aligned} \tag{5.10}$$

If the wavefunction and the energy satisfy all of these equations simultaneously, then the wavefunction is the solution to the Schrödinger equation for all possible linear combinations of the Slater determinants in the projection space. If the projection space is complete, i.e. contains all possible Slater determinants, the wavefunction (and the energy) is the exact solution to the Schrödinger equation for all possible wavefunctions within the given basis set[64]. The projected Schrödinger equation can be rearranged to equal to zero, which makes it easier to solve using a system of nonlinear equations solver.

$$\begin{aligned} f(\Phi_1 | \mathbf{P}) &= 0 \\ &\vdots \\ f(\Phi_M | \mathbf{P}) &= 0 \end{aligned} \tag{5.11}$$

where $f(\Phi_i | \mathbf{P}) = \langle \Phi_i | \hat{H} | \Psi \rangle - E \langle \Phi_i | \Psi \rangle$ and $\mathbf{P}$ are the parameters involved in solving these equations.

One way to solve a system of nonlinear equations is to use a nonlinear least-squares algorithm[91, 92]. In a nonlinear least-squares algorithm, the system of nonlinear equations is often reduced to a sum of squared residuals:

$$\sum_{i=1}^{M} \Big(f(\Phi_i|\mathbf{P}) \Big)^2 \tag{5.12}$$

As this sum is minimized, each equation becomes closer to being satisfied. All of the equations are satisfied when the sum is zero. Thus, this sum can serve as an indicator of how closely the wavefunction (and the energy) satisfies the Schrödinger equation - a metric that is unavailable when solved variationally.

Interestingly, the sum of squared residuals is the prototypical equation to be optimized in most machine learning problems. In a typical machine learning problem, the goal is to find the parameters, $\mathbf{P}$, to the function, f, that best maps a set of inputs $\{\mathbf{x}_i\}$ to their target values $\{t_i\}$:

$$\begin{aligned} f(\mathbf{x}_1|\mathbf{P}) &\approx t_1 \\ &\vdots \\ f(\mathbf{x}_M|\mathbf{P}) &\approx t_M \end{aligned} \tag{5.13}$$

The most common way to solve this problem is to minimize the mean squared error:

$$\begin{aligned} R_{\text{emp}}(\mathbf{P}) &= \frac{1}{M} \sum_{i=1}^{M} \Big(f(\mathbf{x}_i|\mathbf{P}) - t_i \Big)^2 \\ &= \frac{1}{M} \sum_{i=1}^{M} L(\mathbf{x}_i, t_i|\mathbf{P}) \end{aligned} \tag{5.14}$$

where the error associated with each data point is represented with the cost function, $L(\mathbf{x}_i, t_i|\mathbf{P}) = \Big(f(\mathbf{x}_i|\mathbf{P}) - t_i \Big)^2$. This equation is identical to the least-squares equation for the projected Schrödinger equation (Equation 5.12). In the least-squares equation, the target values are all 0 and the scalar constant $\frac{1}{M}$ is omitted. In fact, $\frac{1}{M}$ can be added to Equation 5.12 without affecting the optimization landscape (except for the scaling). For brevity, we will often refer to each equation of the projected Schrödinger equation, $f(\Phi_i|\mathbf{P})$, as a cost function.

Many applications of neural networks within machine learning involve replacing the desired function, f, with a neural network, and optimizing its parameters to minimize Equation 5.14. Since neural networks are incredibly nonlinear and, oftentimes, not smooth, Equation 5.14 becomes incredibly difficult to optimize and this difficulty was one of the main bottlenecks that prevented widespread use of neural networks. In recent years, however, the neural network model became more ubiquitous in practical applications thanks (in part) to the development of optimization algorithms that exploit available computational resources[1–3]. Since the underlying equations of the projected Schrödinger equation are also very nonlinear and difficult to optimize, the algorithms effective for optimizing neural networks may also be effective for solving the projected Schrödinger equation. Here, we explore possible benefits and ramifications of using one such algorithm, the stochastic gradient descent (SGD), to solve the projected Schrödinger equation.

The gradient of the mean squared error (Equations 5.12 and 5.14) is an average of the gradients for each equation in the projected Schrödinger equation 5.13

$$\begin{aligned}\nabla R_{\text{emp}}(\mathbf{P}) &= \nabla \frac{1}{M}\sum_{i=1}^{M} L(\mathbf{x}_i, t_i|\mathbf{P})\\ &= \frac{1}{M}\sum_{i=1}^{M} \nabla L(\mathbf{x}_i, t_i|\mathbf{P})\end{aligned} \tag{5.15}$$

In stochastic gradient descent, the gradient of the objective is approximated with the gradient of the cost function of one data point[1, 2, 93, 94]. A standard implementation of stochastic gradient descent involves randomly selecting a data point for the gradient at each iteration of the optimization process. However, in the case of the projected Schrödinger equation, using a gradient associated with a random Slater determinant within the projection space will likely be inefficient, as many of these gradients will have small contributions to the total gradient. For most ground-state wavefunctions, the lowest-energy Slater determinant makes the largest contribution to the wavefunction while excited state determinants make smaller contributions as the order of excitation increases[58, 59, 95]. The numerical consequence of the Slater determinants (and their one- and two-electron excitations) with negligibly small contributions is that the corresponding equations in the projected Schrödinger equation are nearly zero. Similarly, the gradients that correspond to these Slater determinants will be small in magnitude. However, the number of higher-order excitations grows combinatorially with the order of excitations. This means that if Slater determinants are selected randomly (according to a uniform distribution), then the Slater determinants of higher-order excitations will more likely be selected and the gradients used throughout the optimization will likely (1) be too small to be efficient and (2) favor the minimization of equations that have small errors (i.e. near zero).

To adapt the stochastic gradient descent to solve the projected Schrödinger equation, we first look into the motivation for sampling the data points. Assuming that the data points are selected according to a uniform distribution, i.e. probability of selecting each data point is $\frac{1}{M}$, the expectation value of the gradient for one data point is equal to the total gradient.

$$\begin{aligned}\mathbb{E}_{i\sim p(i)}\Big[\nabla L(\mathbf{x}_i, t_i|\mathbf{P})\Big] &= \sum_{i=1}^{M} p(\Phi_i)\nabla L(\mathbf{x}_i, t_i|\mathbf{P}) \\ &= \sum_{i=1}^{M} \frac{1}{M}\nabla L(\mathbf{x}_i, t_i|\mathbf{P}) \\ &= \nabla R_{\text{emp}}(\mathbf{P})\end{aligned} \tag{5.16}$$

Therefore, the exact gradient does not need to be computed because, over enough iterations, the total gradient is reproduced (on average).

Just as data points can be sampled to estimate the gradient, we can sample the projection space and we would expect that the stochastic gradient descent algorithm will equally be effective for the projected Schrödinger equation. Randomly selecting a Slater determinant via a uniform distribution is unsuitable for solving the projected Schrödinger equation because the gradient associated with individual Slater determinants are very different. In other words, not all Slater determinants are equally important in the projection space. However, if the distribution is not uniform, i.e. $p(\Phi_i) \neq \frac{1}{M}$, then the expected gradient does not correspond to the gradient of the projected Schrödinger equation (Equation 5.12).

The projected Schrödinger equation, as it is presented in Equations 5.11 and 5.12, each equation (i.e., Slater determinant) is weighted equally. If the Slater determinants are not all equally important, then the equations in the projected Schrödinger equation need to be weighted according to a distribution that reflects the importance of each

Slater determinant. The corresponding weighted sum of squared residuals is

$$\sum_{i=1}^{M} p(\Phi_i)\Big(f(\Phi_i|\mathbf{P})\Big)^2 \tag{5.17}$$

where $p(\Phi_i)$ is the weight for each residual. Since the weights are arbitrary, they can be constrained to be probabilities, i.e. $0 \leq p(\Phi_i) \leq 1$ and $\sum_{i=1}^{M} p(\Phi_i) = 1$. With appropriate probabilities, the SGD algorithm can more efficiently optimize the projected Schrödinger equation.

In SGD, the gradient is estimated by sampling Slater determinants from the projection space. Similarly, the exact projected Schrödinger equation can be estimated by sampling the projection space from all possible Slater determinants. Provided that each Slater determinant is independently sampled according to an identical distribution (i.i.d.), the expected average of the cost functions is equal to the projected Schrödinger equation with a complete projection space:

$$\begin{aligned}\sum_{i=1}^{N_{\text{FCI}}} \rho(\Phi_i)\Big(f(\Phi_i|\mathbf{P})\Big)^2 &= \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N}\Big(f(\Phi_i|\mathbf{P})\Big)^2\right]\\ &= \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}\left[\Big(f(\Phi_i|\mathbf{P})\Big)^2\right]\\ &= \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N_{\text{FCI}}} \rho(\Phi_j)\Big(f(\Phi_j|\mathbf{P})\Big)^2\\ &= \sum_{j=1}^{N_{\text{FCI}}} \rho(\Phi_j)\Big(f(\Phi_j|\mathbf{P})\Big)^2\end{aligned} \tag{5.18}$$

where $\rho(\Phi_i)$ is the probability of selecting Slater determinant Φ_i for the projection space and N_{FCI} is the number of all possible Slater determinants. When the i.i.d. approximation is valid, the average of the cost functions corresponds to the projected Schrödinger equation with a complete projection space. However, since the Slater determinants are independently sampled, the projection space will likely contain repetitions. Grouping the repetitions together, we can obtain a more compact set of equations:

$$\begin{aligned}\frac{1}{N}\sum_{\Phi_j \in S_{\text{proj}}} \Big(f(\Phi_j|\mathbf{P})\Big)^2 &= \frac{1}{N}\sum_{i=1}^{M} m_i \Big(f(\Phi_i|\mathbf{P})\Big)^2 \\ &= \sum_{i=1}^{M} \frac{m_i}{N}\Big(f(\Phi_i|\mathbf{P})\Big)^2\end{aligned} \tag{5.19}$$

where S_{proj} is the sampled projection space, m_i is the number of Slater determinant Φ_i in the sampled projection space, N is the size of the sampled projection space, and M is the number of unique Slater determinants sampled. If the gradient is estimated by sampling Slater determinants from the (adapted) projection space, then $p(\Phi_i) = \frac{m_i}{N}$ in accordance to Equation 5.17.

Since the probabilities are arbitrary, existing only as weights in the least-squares equation, it seems that any distribution of Slater determinants is equally valid. However, as mentioned above, some Slater determinants, such as those of high-order excitations, can accurately satisfy the projected Schrödinger equation for the ground state. Therefore, numerically at least, not all Slater determinants are equal and there may exist a set of weights that encapsulates these differences. Taking motivation from variational quantum Monte Carlo[96–101], the Slater determinants can be sampled and weighted according to the following probability distribution:

$$\rho(\Phi_i) = \frac{|\langle\Psi|\Phi_i\rangle|^2}{\sum_j |\langle\Psi|\Phi_j\rangle|^2} \tag{5.20}$$

If a Slater determinant contributes significantly to the wavefunction, its first and second order excitations will likely have non-negligible contributions to the wavefunctions and the associated cost function, $f(\Phi_i|\mathbf{P})$, will likely be more significant. Unlike variational quantum Monte Carlo, which has sound theoretical basis for this probability distribution - it helps make an unbiased estimator of the expected energy - the weights (and the probability distribution) in the projected Schrödinger equation are arbitrary. At the moment, it is unclear whether there exists a better probability distribution with a sound theoretical reasoning or whether this probability distribution is optimal for the optimization process. Nonetheless, these modifications provide a link to variational quantum Monte Carlo. For example, the Slater determinants can be sampled using algorithms typically used in variational quantum Monte Carlo like the Metropolis-Hastings algorithm[102–104] and the continuous-time Monte Carlo algorithm[98, 105, 106]. Hopefully, these modifications provide similar benefits to solving the projected Schrödinger equation as the variational quantum Monte Carlo did for minimizing the energy and as the stochastic gradient descent did for optimizing neural networks. This will be explored in future work.

5.3 Conclusion

In this chapter, the neural network model and the stochastic gradient descent algorithm are incorporated into *ab initio* methods to solve the Schrödinger equation. A new wavefunction ansatz utilizing a simple feed-forward neural network is developed and assessed. For the systems tested, this ansatz seems promising: the wavefunction is cheap to evaluate and provides accurate results. It is amenable to good initial guesses: HF initial guess can be obtained easily and the number of layers can be increased without changing its behaviour. However, the wavefunction becomes progressively more difficult to optimize as the network gets deeper. Similar to machine learning applications of neural networks, practical application will likely involve a specialized network structure and a specialized optimization algorithm. Stochastic gradient descent (SGD) is one algorithm that was explored. It estimates the gradient of the sum of squared residuals of the data set with an average of the gradients of the sampled data points. Interpreting the projected Schrödinger equation as a residual sum of squares, an unbiased estimate of its gradient can be obtained by sampling Slater determinants from the projection space. Extending upon this application, an unbiased estimate of the exact projected Schrödinger equation can be obtained by sampling the projection space, analogous to quantum Monte Carlo (QMC) algorithms that utilize orbital space. Similar to the QMC algorithms, the efficacy of this algorithm depends on the probability distribution with which Slater determinants are drawn, though it is unclear at the moment which formulation is the most practical or elegant within the context of the projected Schrödinger equation.

The presented usage of the feed-forward neural network and the stochastic gradient descent algorithm in solving the Schrödinger equation have critical flaws: the neural network wavefunction is difficult to optimize for deeper structures; the neural network lacks specialized structures that exploit quantum chemical concepts; the neural network is not designed for machine learning application; and the details of implementing adaptive sampling of the Slater determinants and projection space into an optimization algorithm are not clear. However, these applications may be useful as starting points for practical application of machine learning in *ab initio* methods in the future.

5.4 References

(1) Bishop, C. M., *Pattern recognition and machine learning*; Springer: 2006, p 738.

(2) Goodfellow, I.; Bengio, Y.; Courville, A., *Deep learning*; MIT press: 2016.

(3) LeCun, Y.; Bengio, Y.; Hinton, G. *nature* **2015**, *521*, 436–444.

(4) Schmidhuber, J. *Neural networks* **2015**, *61*, 85–117.

(5) Brunton, S. L.; Noack, B. R.; Koumoutsakos, P. *Annual Review of Fluid Mechanics* **2020**, *52*, 477–508.

(6) Kutz, J. N. *Journal of Fluid Mechanics* **2017**, *814*, 1–4.

(7) Schumaker, R. P.; Chen, H. *ACM Transactions on Information Systems (TOIS)* **2009**, *27*, 1–19.

(8) Patel, J.; Shah, S.; Thakkar, P.; Kotecha, K. *Expert systems with applications* **2015**, *42*, 259–268.

(9) Kimoto, T.; Asakawa, K.; Yoda, M.; Takeoka, M. In *1990 IJCNN International Joint Conference on Neural Networks*, IEEE: 1990, 1–6 vol.1.

(10) Krizhevsky, A.; Sutskever, I.; Hinton, G. E. In *Advances in Neural Information Processing Systems 25*, Pereira, F., Burges, C. J. C., Bottou, L., Weinberger, K. Q., Eds.; Curran Associates, Inc.: 2012, pp 1097–1105.

(11) Farabet, C.; Couprie, C.; Najman, L.; LeCun, Y. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2013**, *35*, 1915–1929.

(12) Tompson, J. J.; Jain, A.; LeCun, Y.; Bregler, C. In *Advances in Neural Information Processing Systems 27*, Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., Weinberger, K. Q., Eds.; Curran Associates, Inc.: 2014, pp 1799–1807.

(13) Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

(14) Mikolov, T.; Deoras, A.; Povey, D.; Burget, L.; Černockỳ, J. In *2011 IEEE Workshop on Automatic Speech Recognition & Understanding*, 2011, pp 196–201.

(15) Hinton, G.; Deng, L.; Yu, D.; Dahl, G. E.; Mohamed, A.-r.; Jaitly, N.; Senior, A.; Vanhoucke, V.; Nguyen, P.; Sainath, T. N., et al. *IEEE Signal processing magazine* **2012**, *29*, 82–97.

(16) Sainath, T. N.; Mohamed, A.-r.; Kingsbury, B.; Ramabhadran, B. In *2013 IEEE international conference on acoustics, speech and signal processing*, 2013, pp 8614–8618.

(17) Collobert, R.; Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K.; Kuksa, P. *Journal of machine learning research* **2011**, *12*, 2493–2537.

(18) Bordes, A.; Chopra, S.; Weston, J. *arXiv preprint arXiv:1406.3676* **2014**.

(19) Jean, S.; Cho, K.; Memisevic, R.; Bengio, Y. *arXiv preprint arXiv:1412.2007* **2014**.

(20) Sutskever, I.; Vinyals, O.; Le, Q. V. In *Advances in neural information processing systems*, 2014, pp 3104–3112.

(21) Rosenblatt, F. *Psychological review* **1958**, *65*, 386.

(22) McCulloch, W. S.; Pitts, W. *The bulletin of mathematical biophysics* **1943**, *5*, 115–133.

(23) Glorot, X.; Bordes, A.; Bengio, Y. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 2011, pp 315–323.

(24) Hinton, G. E.; Osindero, S.; Teh, Y.-W. *Neural computation* **2006**, *18*, 1527–1554.

(25) LeCun, Y.; Boser, B. E.; Denker, J. S.; Henderson, D.; Howard, R. E.; Hubbard, W. E.; Jackel, L. D. In *Advances in neural information processing systems*, 1990, pp 396–404.

(26) Bottou, L.; Bousquet, O. In *Advances in neural information processing systems*, 2008, pp 161–168.

(27) Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. *The journal of machine learning research* **2014**, *15*, 1929–1958.

(28) Cybenko, G. *Mathematics of Control, Signals, and Systems* **1989**, *2*, 303–314.

(29) Hornik, K. *Neural Networks* **1991**, *4*, 251–257.

(30) Lu, Z.; Pu, H.; Wang, F.; Hu, Z.; Wang, L. In *Advances in neural information processing systems*, 2017, pp 6231–6239.

(31) Kidger, P.; Lyons, T. In *Conference on Learning Theory*, 2020, pp 2306–2327.

(32) Dral, P. O. *The Journal of Physical Chemistry Letters* **2020**, *11*, 2336–2347.

(33) Lagaris, I. E.; Likas, A.; Fotiadis, D. I. *Computer Physics Communications* **1997**, *104*, 1–14.

(34) Caetano, C.; Reis Jr, J.; Amorim, J.; Lemes, M. R.; Pino Jr, A. D. *International Journal of Quantum Chemistry* **2011**, *111*, 2732–2740.

(35) Sugawara, M. *Computer Physics Communications* **2001**, *140*, 366–380.

(36) Nakanishi, H.; Sugawara, M. *Chemical Physics Letters* **2000**, *327*, 429–438.

(37) Shirvany, Y.; Hayati, M.; Moradian, R. *Applied Soft Computing* **2009**, *9*, 20–29.

(38) Toulouse, J.; Assaraf, R.; Umrigar, C. J. In *Advances in Quantum Chemistry*; Elsevier: 2016; Vol. 73, pp 285–314.

(39) Anderson, J. B. *The Journal of Chemical Physics* **1976**, *65*, 4121–4127.

(40) Manzhos, S.; Carrington, T. *Canadian Journal of Chemistry* **2009**, *87*, 864–871.

(41) Manzhos, S.; Yamashita, K.; Carrington Jr, T. *Chemical Physics Letters* **2009**, *474*, 217–221.

(42) Behler, J.; Parrinello, M. *Physical review letters* **2007**, *98*, 146401.

(43) Smith, J. S.; Isayev, O.; Roitberg, A. E. *Chemical science* **2017**, *8*, 3192–3203.

(44) Schütt, K. T.; Arbabzadah, F.; Chmiela, S.; Müller, K. R.; Tkatchenko, A. *Nature communications* **2017**, *8*, 1–8.

(45) Welborn, M.; Cheng, L.; Miller III, T. F. *Journal of chemical theory and computation* **2018**, *14*, 4772–4779.

(46) Cheng, L.; Welborn, M.; Christensen, A. S.; Miller III, T. F. *The Journal of chemical physics* **2019**, *150*, 131103.

(47) Chen, Y.; Zhang, L.; Wang, H., et al. *arXiv preprint arXiv:2005.00169* **2020**.

(48) Sharma, S.; Sivalingam, K.; Neese, F.; Chan, G. K.-L. *Nature chemistry* **2014**, *6*, 927–933.

(49) Mills, K.; Spanner, M.; Tamblyn, I. *Physical Review A* **2017**, *96*, 042113.

(50) Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp 1–9.

(51) Carleo, G.; Troyer, M. *Science* **2017**, *355*, 602–606.

(52) Han, J.; Zhang, L.; Weinan, E. *Journal of Computational Physics* **2019**, *399*, 108929.

(53) Hermann, J.; Schätzle, Z.; Noé, F. *arXiv preprint arXiv:1909.08423* **2019**.

(54) Schütt, K.; Gastegger, M.; Tkatchenko, A.; Müller, K.-R.; Maurer, R. J. *Nature communications* **2019**, *10*, 1–10.

(55) Hegde, G.; Bowen, R. C. *Scientific reports* **2017**, *7*, 42669.

(56) Troyer, M.; Wiese, U.-J. *Physical review letters* **2005**, *94*, 170201.

(57) Cook, S. A. In *Proceedings of the third annual ACM symposium on Theory of computing*, 1971, pp 151–158.

(58) Szabo, A.; Ostlund, N., *Modern Quantum Chemistry - Introduction to Advanced Electronic Structure Theory*; McGraw-Hill Inc.: 1989, pp 43–107.

(59) Piela, L., *Ideas of quantum chemistry*; Elsevier: 2013.

(60) Helgaker, T.; Jørgensen, P.; Olsen, J., *Modern electronic structure theory*; Wiley: Chichester, 2000.

(61) Cullen, J. *Chemical Physics* **1996**, *202*, 217–229.

(62) Limacher, P. *Journal of Chemical Physics* **2016**, *145*, 194102.

(63) Johnson, P.; Limacher, P.; Kim, T.; Richer, M.; Miranda-Quintana, R.; Heidar-Zadeh, F.; Ayers, P.; Bultinck, P.; De Baerdemacker, S.; Van Neck, D. *Computational and Theoretical Chemistry* **2017**, *1116*, 207–219.

(64) Kim, T. D.; Miranda-Quintana, R. A.; Ayers, P. W. On the link between projected Schrödinger equation and variational methods., Unpublished Manuscript, 2020.

(65) Kim, T. D.; Miranda-Quintana, R. A.; Richer, M.; Ayers, P. W. Flexible Ansatz for N-body Configuration Interaction., Unpublished Manuscript, 2020.

(66) Parr, R.; Ellison, F.; Lykos, P. *Journal of Chemical Physics* **1956**, *24*, 1106.

(67) Hurley, A.; Lennard-Jones, J.; Pople, J. *A theory of paired-electrons in polyatomic molecules Proceedings of the Royal Society of London Series A* **1953**, *220*, 446–455.

(68) McWeeny, R.; Sutcliffe, B. *Proceedings of the Royal Society of London Series A* **1963**, *273*, 103–116.

(69) Surjan, P. In *Correlation and Localization*, Surjan, P., Ed., 1999, pp 63–88.

(70) Allen, T.; Shull, H. *Journal of Physical Chemistry* **1962**, *66*, 2281–2283.

(71) Tecmer, P.; Boguslawski, K.; Johnson, P.; Limacher, P.; Chan, M.; Verstraelen, T.; Ayers, P. *Journal of Physical Chemistry A* **2014**, *118*, 9058–9068.

(72) Paldus, J.; Cizek, J.; Sengupta, S. *Journal of Chemical Physics* **1971**, *55*, 2452–2462.

(73) Paldus, J.; Sengupta, S.; Cizek, J. *Journal of Chemical Physics* **1972**, *57*, 652–666.

(74) Surjan, P.; Szabados, Á.; Jeszenszki, P.; Zoboki, T. *Journal of Mathematical Chemistry* **2012**, *50*, 534–551.

(75) Limacher, P.; Ayers, P.; Johnson, P.; De Baerdemacker, S.; Van Neck, D.; Bultinck, P. *Journal of Chemical Theory and Computation* **2013**, *9*, 1394–1401.

(76) Johnson, P.; Ayers, P.; Limacher, P.; De Baerdemacker, S.; Van Neck, D.; Bultinck, P. *Computational and Theoretical Chemistry* **2013**, *1003*, 101–13.

(77) Bartlett, R.; Musiał, M. *Reviews of Modern Physics* **2007**, *79*, 291–352.

(78) Paldus, J.; Li, X. In *Advances in Chemical Physics*, Prigogine, I., Rice, S., Eds., 1999; Vol. 110, pp 1–175.

(79) Cizek, J. *Journal of Chemical Physics* **1966**, *45*, 4256–4266.

(80) Shavitt, I.; Bartlett, R., *Many-body methods in chemistry and physics: MBPT and coupled-cluster theory*; Cambridge: Cambridge, 2009.

(81) Nooijen, M.; Shamasundar, K.; Mukherjee, D. *Molecular Physics* **2005**, *103*, 2277–2298.

(82) Kim, T. D.; Richer, M.; Sánchez-Díaz, G.; Heidar-Zadeh, F.; Verstraelen, T.; Miranda-Quintana, R. A.; Ayers, P. W. Fanpy., Unpublished Manuscript, 2020.

(83) Gebauer, R.; Cohen, M. H.; Car, R. *PNAS* **2016**, *113*, 12913–12918.

(84) Xiangzhu, L.; Paldus, J. *The Journal of Chemical Physics* **1995**, *103*, 1024–1034.

(85) Ayers, P. W. *Physical Chemistry Chemical Physics* **2006**, *8*, 3387–3390.

(86) Min, K. S.; DiPasquale, A. G.; Rheingold, A. L.; White, H. S.; Miller, J. S. *Journal of the American Chemical Society* **2009**, *131*, 6229–6236.

(87) Miller, J. S.; Min, K. S. *Angewandte Chemie International Edition* **2009**, *48*, 262–272.

(88) Aaronson, S. *ACM Sigact News* **2005**, *36*, 30–52.

(89) Lüders, M.; Marques, M.; Lathiotakis, N.; Floris, A.; Profeta, G.; Fast, L.; Continenza, A.; Massidda, S.; Gross, E. *Physical Review B* **2005**, *72*, 024545.

(90) Nakamura, K.; Arita, R.; Imada, M. *Journal of the Physical Society of Japan* **2008**, *77*, 093711.

(91) Nocedal, J.; Wright, S., *Numerical optimization*; Springer Science & Business Media: 2006.

(92) Press, W. H.; Teukolsky, S. A.; Vetterling, W. T.; Flannery, B. P., *Numerical recipes 3rd edition: The art of scientific computing*; Cambridge university press: 2007.

(93) Bottou, L. In *Neural networks: Tricks of the trade*; Springer: 2012, pp 421–436.

(94) Ruder, S. *arXiv preprint arXiv:1609.04747* **2016**.

(95) Dreuw, A.; Head-Gordon, M. *Chemical reviews* **2005**, *105*, 4009–4037.

(96) Nightingale, M. P.; Umrigar, C. J., *Quantum Monte Carlo methods in physics and chemistry*; 525; Springer Science & Business Media: 1998.

(97) Umrigar, C. *The Journal of chemical physics* **2015**, *143*, 164105.

(98) Sabzevari, I.; Sharma, S. *Journal of chemical theory and computation* **2018**, *14*, 6276–6286.

(99) Neuscamman, E. *The Journal of chemical physics* **2013**, *139*, 194105.

(100) Neuscamman, E. *Journal of chemical theory and computation* **2016**, *12*, 3149–3159.

(101) Kurita, M.; Yamaji, Y.; Morita, S.; Imada, M. *Physical Review B* **2015**, *92*, 035122.

(102) Umrigar, C. *The Journal of chemical physics* **2015**, *143*, 164105.

(103) Metropolis, N.; Rosenbluth, A. W.; Rosenbluth, M. N.; Teller, A. H.; Teller, E. *The journal of chemical physics* **1953**, *21*, 1087–1092.

(104) Hastings, W. K. **1970**.

(105) Bortz, A. B.; Kalos, M. H.; Lebowitz, J. L. *Journal of Computational Physics* **1975**, *17*, 10–18.

(106) Gillespie, D. T. *Journal of computational physics* **1976**, *22*, 403–434.

Chapter 6

Conclusion

6.1 Summary

The goal in *ab initio* method development in quantum chemistry is to understand chemical systems and to predict their properties. Brute-force solutions to the Schrödinger equation are intractable, so useful approximate *ab initio* methods maintain a delicate balance between cost and accuracy. Even when a method seems affordable and adequately accurate, close inspection often shows that it breaks down for certain "evil" systems. It is hard to say that one method is (or will be) superior to all the others when studying chemical systems in general. Therefore, it is important to explore - we must develop many methods and algorithms and systematically assess their ability to solve the Schrödinger equation for different chemical systems.

The goal throughout this thesis is to establish a systematic approach to method development in electronic structure theory. Chapter 2 introduces a general framework for multideterminant wavefunctions called the Flexible Ansatz for N-electron Configuration Interaction (FANCI). Within this framework, multideterminant wavefunctions become compatible with one another and the development of new methods becomes more transparent, making them more accessible. Moreover, it was possible to mathematically specify what makes a multideterminant method have desirable properties (e.g., size consistency), thereby providing essential guidance on how (and how not to) develop new wavefunction forms. To demonstrate the power of this idea, many existing and novel wavefunctions ansätze are presented with respect to this framework. However, the developed methods must be implemented and tested to fully evaluate their effectiveness.

Chapter 3 presents an open-source Python library that serves as a platform where researchers can quickly prototype their methods. This library supports the development of methods that pertain to the multideterminant wavefunction, Hamiltonian, formulation of the Schrödinger equation, and optimization algorithm, each of which is embodied as an independent module. Its modular design and user-friendly templates help minimize the amount of code read and written by new developers that are implementing their ideas. Each module is designed to be compatible with one another so that researchers can customize their calculations and experiment with different combinations of methods and algorithms.

Utilizing the streamlined development process established in the first two chapters, the remaining two chapters develop novel methods motivated from fields outside of quantum chemistry. Chapter 4 draws upon graph theory to interpret the geminal wavefunctions as pairing schemes in a graph. Using the weighted perfect matching algorithm, we present an approximation that significantly decreases the cost associated with geminal wavefunctions. Numerical evidence suggests that this approximation does not result in a significant decrease in accuracy and can be incrementally improved by increasing the number of pairing schemes. The presented interpretation and the corresponding approximation pave the way for more sophisticated optimization algorithms and tractable generalizations to the geminal wavefunctions.

Chapter 5 finds inspiration from neural networks to develop a wavefunction whose overlap is a neural network and a formulation of the projected Schrödinger equation analogous to the energy equation from orbital-space variational quantum Monte Carlo. The neural network wavefunction shows promise as an cheap and accurate wavefunction ansatz but, like other neural networks, becomes difficult to optimize as the network becomes deeper. Specialized network structures and optimization algorithms are likely needed for wide-spread use of this wavefunction. One optimization algorithm we explored is stochastic gradient descent algorithm, which has been found to successfully optimize many neural networks. Similar to sampling data points from a data set in standard stochastic gradient descent implementations, the Slater determinants can be sampled from the projection space to obtain an unbiased estimate of the least-squared sum of the projected Schrödinger equation. Furthermore, sampling the Slater determinants to form a projection space results in an unbiased estimate of the exact projected Schrödinger equation. Analogous to variational quantum Monte Carlo algorithms, sampling Slater determinant alludes to a stochastic optimization algorithm with adaptive projection space, though its details, such as the optimal distribution from which to draw the Slater determinants, are not yet clear.

6.2 Outlook

Using the FANCI framework presented in Chapter 2, a wide range of multideterminant wavefunctions can be built from a common structure, allowing them to share a wide range of methods involving different Hamiltonians, objectives, and optimization algorithms, as demonstrated by the extensive customizability within `Fanpy` of Chapter 3. Similarly, existing post-processing methods, such as perturbation theory, equations-of-motion, and embedding, can be expressed terms of the FANCI framework (i.e., expressed in terms of Slater determinants and possibly involving the projected Schrödinger equation) and implemented in `Fanpy`. Then, these methods can be used alongside different combinations of wavefunctions, Hamitonians, and optimization algorithms.

At the moment, `Fanpy` uses generic general-purpose optimization methods, rather than methods specialized for problems in electronic structure theory. However, as explored in Section 5.2, quantum Monte Carlo algorithms can be implemented to optimize the energy and the projected Schrödinger equation. Direct inversion of the iterative subspace (DIIS), a common optimization technique in quantum chemistry, might be useful for optimizing the system of nonlinear equations associated with the projected equation.

The performance issues in `Fanpy` can be addressed by interfacing to numerically efficient languages, such as `C++` and `FORTRAN`, or by interfacing to performance-oriented libraries, such as `Psi4`, `PySCF`, and `PyCI`. These may come at the cost of customizability and ease-of-development, cornerstone features of `Fanpy`. Nonetheless, providing the option of a better performance will be useful for those that want to test their methods in larger systems.

The APG wavefunction with adaptive selection of pairing schemes, presented in Chapter 4, is intractable due to the cost of the permanent evaluation. To address this issue, the one-reference orbital and rank-2 approximations used in AP1roG and APr2G, respectively, can be generalized to APG wavefunctions. Then, the adaptive selection algorithm and the permanent approximation can be combined to build the generalized quasiparticle wavefunction introduced in Chapter 2. This sort of extension is not difficult in `Fanpy` but thorough testing of the methods is likely intractable until improved optimizers are implemented.

The neural network wavefunctions, though cheap and accurate, are difficult to optimize. In addition to better optimization algorithms, suggested above, specialized network structures are needed to make the optimization simpler (often by decreasing the number of parameters) and to allow generalization of the network to multiple chemical systems. Fortunately, `Fanpy`'s flexibility and support for `Keras`, a popular neural network library, make it easy to explore various network structures.

More generally, when developing new *ab initio* methods, one needs a way to assess their performance. The usual metrics are usually only the accuracy of the optimized energy and the computational cost associated thereto. This clearly ignores other properties (e.g., the response of the system to external fields) and ignores the system-dependence of methods' performance. The blind pursuit of a cheap and accurate method risks overlooking its deficiencies until the final stages of development, where one might discover that the method is not viable for practical use due, for example, to technical factors related to slow convergence or the failure for special types of systems (e.g., diradicals or zwitterions). We need to develop better metrics by which to evaluate a method, and by which to assess the systems for which the methods should be used. One advantage of the projected Schrödinger equation is that the least-squares error can be used to provide (approximate) error bars on the energy and the error in the wavefunction, and these error bars can be made systematically more reliable by augmenting the projection space. This is a significant advantage over most traditional wavefunction approaches, whose accuracy can deteriorate without warning.

6.3 Perspective

This thesis presents a new approach, termed the Flexible Ansatze for N-electron Configuration Interaction (FANCI) that subsumes every multideterminant approach. At a mathematical level, FANCI provides a common framework for new methods and helps inform the search for new wavefunction ansatze with desirable properties. At the level of software development, the FANCI idea allows us to write a single software program, `Fanpy`, that supports all possible multideterminant wavefunction methods. Many existing, and new, wavefunctions have been implemented into `Fanpy` already. The utility of FANCI and `Fanpy` is mainly for theoretical exploration; once a method is found to be effective, it is probably desirable to write efficient, dedicated, software for it, including nuances related to making accurate initial guesses and robust dedicated optimization methods. Further corrections (e.g., dynamic correlation) should also be extended.