GRAPHICAL PROCESSING UNIT ACCELERATION AND DEVELOPMENT OF MULTIREFERENCE QUANTUM CHEMICAL METHODS

By

Bryan Scott Fales

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Chemistry – Doctor of Philosophy

2017

ABSTRACT

GRAPHICAL PROCESSING UNIT ACCELERATION AND DEVELOPMENT OF MULTIREFERENCE QUANTUM CHEMICAL METHODS

By

Bryan Scott Fales

Understanding the electronic structure and excited-state dynamics of photochemical systems is a challenging problem in computational chemistry. Features of the adiabatic potential energy surfaces such as avoided crossings and conical intersections play important roles in events such as non-radiative recombination of excited fluorophores and photovoltaic cells. Nuclear geometries corresponding to points of (near) degeneracy of adiabatic potential energy surfaces represent regions where the Born-Oppenheimer approximation breaks down. A consequence of an erosion of the ability to separate the nuclear and electronic degrees of freedom in the wavefunction is the need for multireference electronic structure methods.

The state-averaged complete active space self-consistent field (SA-CASSCF) and the complete active space configuration interaction (CASCI) methods are standard tools for generating multireference wavefunctions. A common component of each of these methods is a configuration interaction (CI) calculation step. The CI method is computationally demanding, however, and imposes hard limits on the dimension of the correlated region. In this work we describe our direct, graphical processing unit (GPU) vectorized full CI implementation, allowing us to calculate CI wavefunctions and energies for systems having $\mathcal{O}(10^3)$ basis functions and $\mathcal{O}(10^8)$ determinants on timescales of minutes. We apply our direct CI method to a series of molecular benchmark systems and demonstrate that our atomic orbital to molecular orbital basis integral transformation scales approximately quadratically with respect to basis set size. Analysis of the scaling behavior of the rate-limiting component of the CI iterations, the matrix-vector product $\boldsymbol{\sigma} = \mathbf{Hc}$, where $\mathbf{H}$ is the electronic Hamiltonian and $\mathbf{c}$ is the CI vector, reveals that our algorithm scales approximately linearly with respect to the number of determinants.

We have developed a GPU-based implementation of the 1- and 2-particle reduced density matrices (1- and 2-RDMs), both of which are necessary for fast evaluation of analytical nuclear energy gradients and nonadiabatic coupling vectors. Calculation of certain properties such as the dipole moment and the transition dipole moment also require the 1-RDM, and SA-CASSCF energy and gradient calculations require the 1-RDM and 2-RDMs, respectively.

Formulation and implementation of several spin-purification schemes that are useful in the context of open-shell determinantal CI are described next. These approaches counteract the numerical instabilities associated with high-accuracy open-shell CI calculations. A GPU accelerated direct $\mathbf{S^2c}$ algorithm, where $\mathbf{S}$ is the spin matrix, enables robust spin purification of trial vectors entering the (Krylov) subspace for purification approaches relying on projection or, conversely, modification of the $\boldsymbol{\sigma}$ vector to correspond to a spin-penalized Hamiltonian for penalty-based approaches. To demonstrate the utility of spin-purification methods we include a study of the multireference and multi-excitation character of plasmonic open-shell silver clusters.

The rank-reduced CI (rrCI) method is described for extremely large configuration spaces. The rrCI method allows ground state singlet and triplet calculations having configuration spaces on the order of $\mathcal{O}(10^{16})$ determinants while achieving mH accuracy relative to full CI (FCI). Single point energies of acenes having $2-5$ aromatic rings are reported using HF-CAS-rrCI and compared with density matrix renormalization group (DMRG) calculated energies as an additional verification of the accuracy of rrCI.

All methods described in this work have been implemented in the TeraChem GPU accelerated electronic structure software package. We conclude with a description of large-scale CI calculations that have been performed using methods described in this manuscript.

For Reagan.

ACKNOWLEDGEMENTS

BSF would like to thank the following individuals:

- Professor Benjamin Levine, my Ph.D. adviser, for providing me the opportunity to be a part of his science team.
- Professor Mary T. Rodgers, my undergraduate adviser, who introduced me to both scientific research and to computational chemistry.
- Professor C. David Sherrill, for developing and freely sharing his lecture notes on configuration interaction theory.
- Professor Alan Munn, for development and maintenance of the "msu-thesis" LaTeX package used for typesetting the present manuscript.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Photochemical processes involve absorption of one or more photons by a chemical system coupled with electronic excitation, often followed by nuclear and/or electronic rearrangement, before subsequent return to the electronic ground state through photon emission or nuclear vibration. The excited state potential energy landscape is highly dimensional, presenting a challenging problem in quantum chemistry. Modeling photochemical processes by exploring potential energy surfaces (PESs) using time-dependent simulations is an efficient way of sampling this high-dimensional space.

Chemical modeling is used to facilitate understanding of physical processes. Examples include simple 2- and 3-dimensional physical models used for pedagogical purposes, molecular mechanics, and fully quantum mechanical ab initio theories. The type of model used for a particular system is chosen based on several factors:

- The nature of the system's constituents (i.e. how big are the particles? Is the behavior quantum or classical? Are relativistic effects important?)
- The physical size (expanse) of the system
- The desired accuracy
- The available computational resources
- Is electron correlation important?
- Are electronic excited states desired?

Many photochemical problems require modeling strategies that push the limits of currently available methods. Nanoparticles, including those composed of low-cost, non-toxic, and abundant atoms such as silicon, have been used in solar energy conversion applications with

increasing frequency in recent years due to their reasonable efficiency and tunability. These nanoparticles may be large, sometimes on the order of tens of nanometers, and it is important to provide a high accuracy ab initio correlated treatment of both the ground and excited electronic states. Electron correlation is the explicit interaction between individual electrons. Since electron correlation, and the related correlation energy, are often defined in terms of Hartree-Fock and full configuration interaction (FCI) theory, we provide a brief description of each in the following sections.

1.1 Single Particle Basis Sets, Slater Determinants, and Hartree-Fock Theory

The spin-free Hamiltonian is

$$\hat{H} = \left[-\frac{1}{2}\sum_i \nabla_i^2 - \sum_{i,A} \frac{Z_A}{r_{iA}} - \frac{1}{2}\sum_A \nabla_A^2 + \sum_{A<B} \frac{Z_A Z_B}{r_{AB}} + \sum_{i<j} \frac{1}{r_{ij}} \right] \tag{1.1}$$

where i, j index electrons, A, B index nuclei, Z are nuclear charges, ∇^2 is the kinetic energy operator, and r is the inter-particle radius. The third and fourth terms (the nuclear kinetic energy operator and nuclear-nuclear repulsion, respectively) are independent of the electronic motion and correspond to a rigid shift of the eigenvalues in the Schrödinger equation, allowing us to define the electronic Hamiltonian as

$$\hat{H}_e = \left[-\frac{1}{2}\sum_i \nabla_i^2 - \sum_{i,A} \frac{Z_A}{r_{iA}} + \sum_{i<j} \frac{1}{r_{ij}} \right] \tag{1.2}$$

For the remainder of this work we will treat only the electronic components of the Hamiltonian, $\hat{H} \equiv \hat{H}_e$. We consider the nuclei to be fixed point charges in accordance with the Born-Oppenheimer representation, and the electronic Schrödinger equation is

$$\hat{H}\Psi = E\Psi \tag{1.3}$$

where E are the eigenvalues and Ψ is the molecular wavefunction.

Solutions to the wave equations may be given in terms of products of one-particle spin-orbitals, χ, having both space and spin coordinates. The resulting wavefunction ansatz, known as the Hartree product, fails to satisfy the antisymmetry principle for fermions, however. Slater remedied this in 1929 by suggesting the use of an antisymmetrized Hartree product. The so-called Slater determinant for an $N-$electron system is defined as

$$\Psi(x_1, x_2, ..., x_N) = \frac{1}{\sqrt{N!}} \begin{vmatrix} \chi_1(x_1) & \chi_2(x_1) & \dots & \chi_N(x_1) \\ \chi_1(x_2) & \chi_2(x_2) & \dots & \chi_N(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ \chi_1(x_N) & \chi_2(x_N) & \dots & \chi_N(x_N) \end{vmatrix} \tag{1.4}$$

where Ψ is the single determinantal wavefunction and x are the electron coordinates. Applying the variational principle to the Schrödinger equation we optimize the spin-orbital coefficients χ to minimize the energy ϵ

$$\epsilon = \frac{\langle \Psi | \hat{H} | \Psi \rangle}{\langle \Psi | \Psi \rangle} \tag{1.5}$$

The Hartree-Fock equations provide a simplification of the molecular electronic Hamiltonian described above by approximating the Coulombic electron repulsion terms. Rather than allow each electron to interact with each other electron individually, instead the electron interacts with the average position of the $N-1$ remaining electrons. While this mean-field approach reduces the computational complexity significantly, it also results in the neglect of correlation between each individual electron pair. The Hartree-Fock equations are written as

$$f(x_1)\chi_i(x_1) = \epsilon_i \chi_i(x_1) \tag{1.6}$$

where the Fock operator f is

$$f(x_1) = h(x_1) + \sum_j \mathcal{J}_j(x_1) - \mathcal{K}_j(x_1) \tag{1.7}$$

with h the one-electron operator and $\mathcal{J}$ and $\mathcal{K}$ the Coulomb and exchange operators, respectively. Both the Fock operator and the Slater determinant are dependent on the underlying orbitals, resulting in a partial differential equation that must be solved self-consistently. The Hartree-Fock equations become the Hartree-Fock-Roothaan equations upon introduction of a basis set, with atomic basis functions $\tilde{\chi}$, giving the form of the spin-orbitals as

$$\chi_i = \sum_\mu C_{\mu i} \tilde{\chi}_\mu \tag{1.8}$$

where $C_{\mu i}$ are the self-consistent field (SCF) coefficients. The Hartree-Fock-Roothan matrix equations then take the form of a generalized eigenvalue problem

$$\mathbf{FC} = \mathbf{SC}\epsilon \tag{1.9}$$

where the elements of each matrix are defined as

$$F_{\mu\nu} = \int dx_1 \tilde{\chi}_\mu^\star(x_1) f(x_1) \tilde{\chi}_\nu(x_1) \tag{1.10}$$

$$S_{\mu\nu} = \int dx_1 \tilde{\chi}_\mu^\star(x_1) \tilde{\chi}_\nu(x_1) \tag{1.11}$$

Diagonalization of $\mathbf{F}$ gives a new set of orbitals $\mathbf{C}$, which are then used to define a new Fock matrix $\mathbf{F}$. This process is repeated until the orbitals and Fock matrix are self-consistently converged. The Fock matrix can be defined in terms of three matrices,

$$\mathbf{F} = \mathbf{h} + \mathbf{J} + \mathbf{K} \tag{1.12}$$

where **h** is a matrix comprised of the one-electron integrals and **J** and **K** are the Coulomb and exchange matrices, respectively. The Coulomb matrix has a classical interpretation, where each electron "sees" the remaining $N-1$ electron charge distribution in an average sense. The exchange matrix does not have a classical analog, arising instead as a consequence of the "exchange" of electrons between spin-orbitals i and j due to the Pauli exclusion principle. As previously mentioned, the mean-field nature of the Hartree-Fock approximation results in the absence of electron correlation. Hartree-Fock is successfully used for generation of zeroth-order wavefunctions, and provides a reference for definion of the full configuration interaction method to explicitly correlate all electrons in the space of the atomic orbital basis functions.

1.2 Configuration Interaction

Beginning with the canonical Hartree-Fock optimized wavefunction, Ψ_0, we define a determinantal basis such that all possible configurations are included in a linear expansion, vide infra

$$\Psi_{CI} = c_0\Psi_0 + \sum_{i}^{a} c_i^a \Psi_i^a + \sum_{ij}^{ab} c_{ij}^{ab} \Psi_{ij}^{ab} + \sum_{ijk}^{abc} c_{ijk}^{abc} \Psi_{ijk}^{abc} + \dots \tag{1.13}$$

i, j, k index occupied orbitals, a, b, c index virtual orbitals, and **c** is the vector of CI coefficients (each corresponding to a subscript $\rightarrow$ superscript excitation) optimized by solving the eigenvalue problem

$$\hat{H}\Psi_{CI} = E\Psi_{CI} \tag{1.14}$$

Full configuration interaction (FCI) provides the exact electronic wavefunction for both ground and excited states in the given atomic orbital basis. Unfortunately, FCI scales according to

$$N_{det} = \left(\frac{N_o!}{N_\alpha!(N_o - N_\alpha)!}\right)\left(\frac{N_o!}{N_\beta!(N_o - N_\beta)!}\right) \tag{1.15}$$

where N_{det} is the number of configurations (determinants) in the CI space, N_o is the number of basis functions (orbitals) and N_α and N_β are the number of α and β electrons, respectively. The largest FCI calculations performed to date are on the order of $\mathcal{O}(10^{10})$ configurations [48], with calculations on the order of $\mathcal{O}(10^8)$ more typical of those being performed routinely. The severe scaling of FCI places strict constraints on the size of the system, limiting its use to that of primarily serving as a benchmarking tool for lower-scaling approximations to FCI.

Solution of the CI eigenvalue problem naïvely requires formation and diagonalization of the FCI Hamiltonian matrix. The N_{det}^2 storage and N_{det}^3 operations explicit matrix diagonalization requires makes the CI problem intractable for all but the smallest systems. To alleviate these issues, Roos developed a direct approach to solving the CI problem without constructing the molecular electronic Hamiltonian[141]. Instead, the matrix-vector product of the Hamiltonian matrix $\mathbf{H}$ with a trial vector $\mathbf{c}$ is directly formed and used to iteratively solve the CI equations. While first presented for the case of closed-shell restricted CI truncated to single and double excitations (CISD) using a configuration state function (CSF) expansion, the approach was soon generalized to include open-shell and unrestricted CISD[142] and multireference CISD (MRCISD)[146, 145]. Siegbahn later extended direct CI approaches to allow for large CI expansions in small orbital spaces by employing resolution of the identity to eliminate the need for evaluation and storage of the two-particle coupling coefficients[164]. This method is especially well-suited for FCI calculations in a complete active space (CAS). Shortly after this, Handy recognized that using a determinantal basis permits efficient factorization of the CI vector into α and β spin components[53]. Further, the one-particle coupling coefficients between determinants are trivially the integers $-1, 0, 1$. A subsequent collaboration between Knowles and Handy resulted in a FCI algorithm tailored for vector machines[73]. The Knowles and Handy algorithm serves as the foundation

for many of the CI algorithms developed in the present work. Other noteworthy direct CI approaches are described by Olsen[128] and by Harrison and Zarrabian[203].

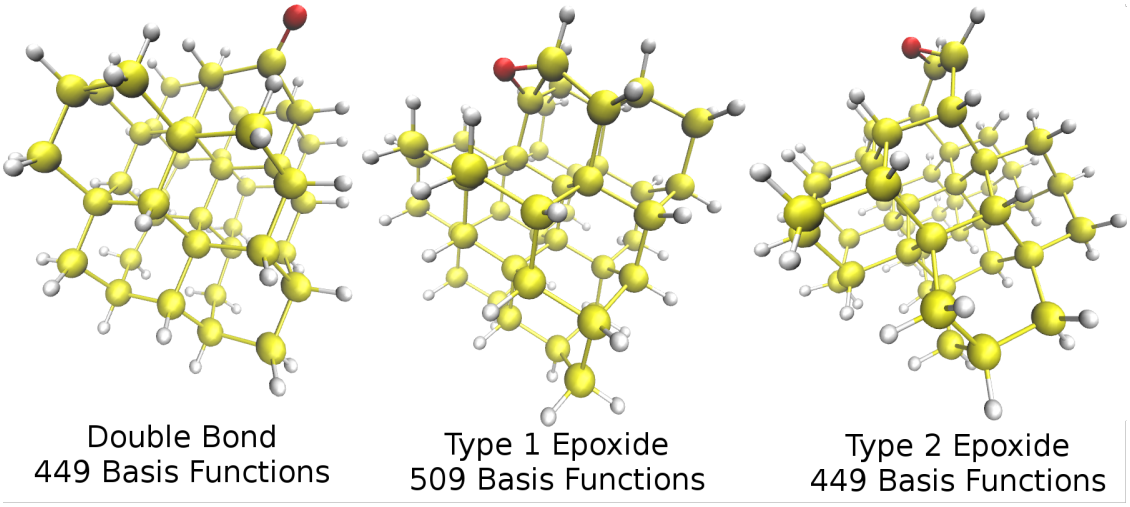

Figure 1.1: Examples of systems which require multiple Slater determinants for a qualitatively correct description. Silicon clusters with oxygen defects at their MECI geometries. From left to right, molecular formulae are $Si_{44}H_{44}O$ (with Si=O bond), $Si_{50}H_{50}O$ (with Si—O—Si epoxide), and $Si_{44}H_{44}O$ (with Si—O—Si epoxide)

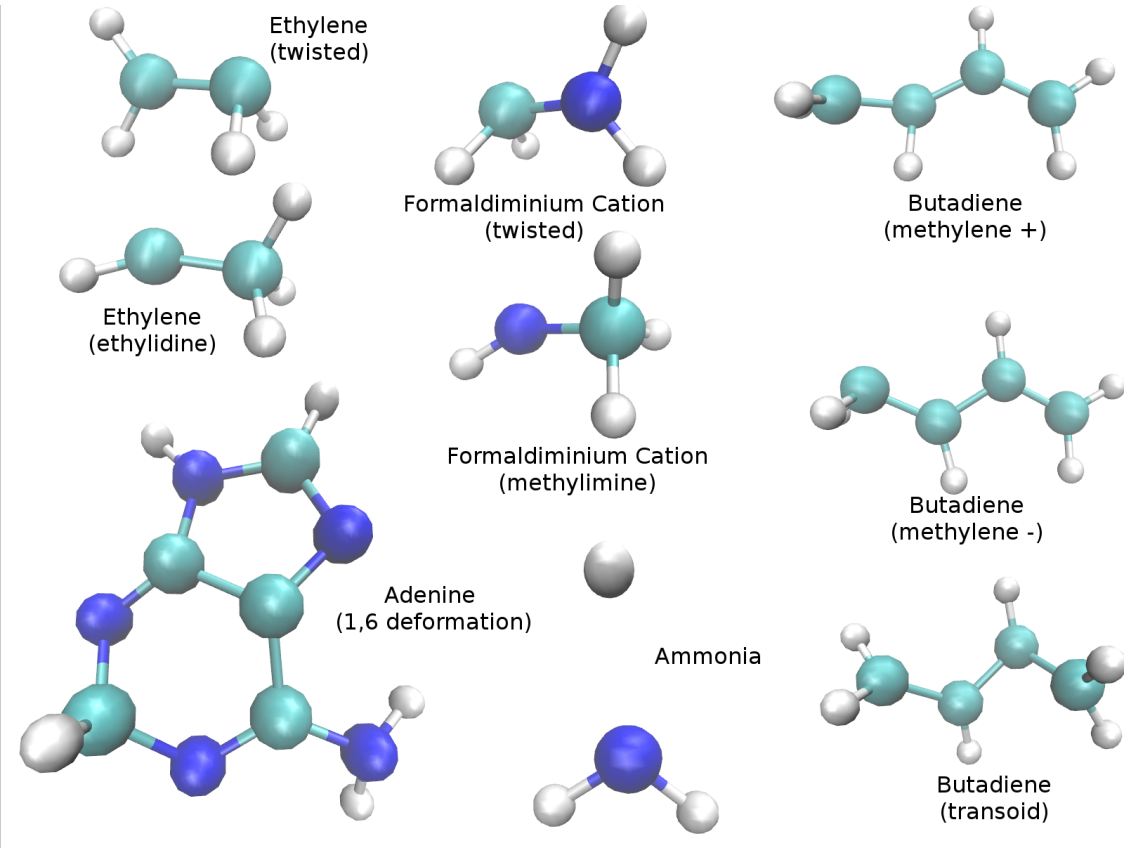

Figure 1.2: Examples of systems having multireference character. Small molecules at their MECI geometries.

The difference in energy between the Hartree-Fock and FCI energy is defined as the correlation energy. This energy may be categorized as arising from either static or dynamic electronic correlation. Dynamic correlation is the instantaneous interaction between individual electrons in motion. Static, or non-dynamical correlation, arises when the wavefunction is dominated by contributions from two or more Slater determinants having qualitatively different character.

While it is convenient to distinguish between dynamical and non-dynamical correlation, the distinction between the two becomes blurred at the limit of the exact (FCI) correlation energy. Still, electronic structure methods may tend to describe one form of correlation better than the other. For example, both coupled-cluster and perturbation theory tend to provide good descriptions of dynamic electron correlation, but fail to provide a qualitatively correct

picture of static electron correlation. Shortcomings in such methods arise due their single reference nature, i.e. the wavefunction is composed of a single electronic configuration and corrections to it. Static correlation must be accounted for explicitly in these theories, giving rise to methods such as multireference coupled cluster theory (MR-CC) and many-body perturbation theory (MR-PT). Approaches which give qualitatively correct descriptions of systems which are strongly statically correlated have been designed explicitly to treat these systems by using a multireference approach. Multireference methods are those whose zeroth order wavefunctions are composed of two (or more) electronic configurations. Examples of such systems include those which are far from equilibrium, including bond stretching and those near the intersection of multiple adiabatic potential energy surfaces. Several examples of molecular systems that require multireference methods are given in Figures 1.1 and 1.2. Regions of the potential energy surface (PES) where the energies of different electronic states are (nearly) degenerate include both avoided crossings and conical intersections. Conical intersections are $N-2$ dimensional seams of degeneracy between adiabatic electronic states, where there are N nuclear degrees of freedom. These seams may be required by symmetry, where the states are uncoupled but the presence of non-Abelian point group symmetry ensures energy degeneracy. This type of intersection is also known as a Jahn-Teller intersection. Alternatively, these seams may be allowed regardless of the point group symmetry. These types of intersections are referred to as accidental intersections. The lowest energy point on the seam is known as a minimum energy conical intersection, or MECI. The PES in the region of the twisted/pyramidalized MECI of ethylene is depicted in Figure 1.3. These PES features become critical when describing the time-dependent behavior of photochemical systems, for example, where non-radiative transitions between electronic states may occur with high probability and at high rates in regions where strong nonadiabatic coupling is present, such as near an MECI. The prototypical method for describing such systems with qualitative accuracy is the complete active space self-consistent field (CASSCF) method, which we introduce in the following section.

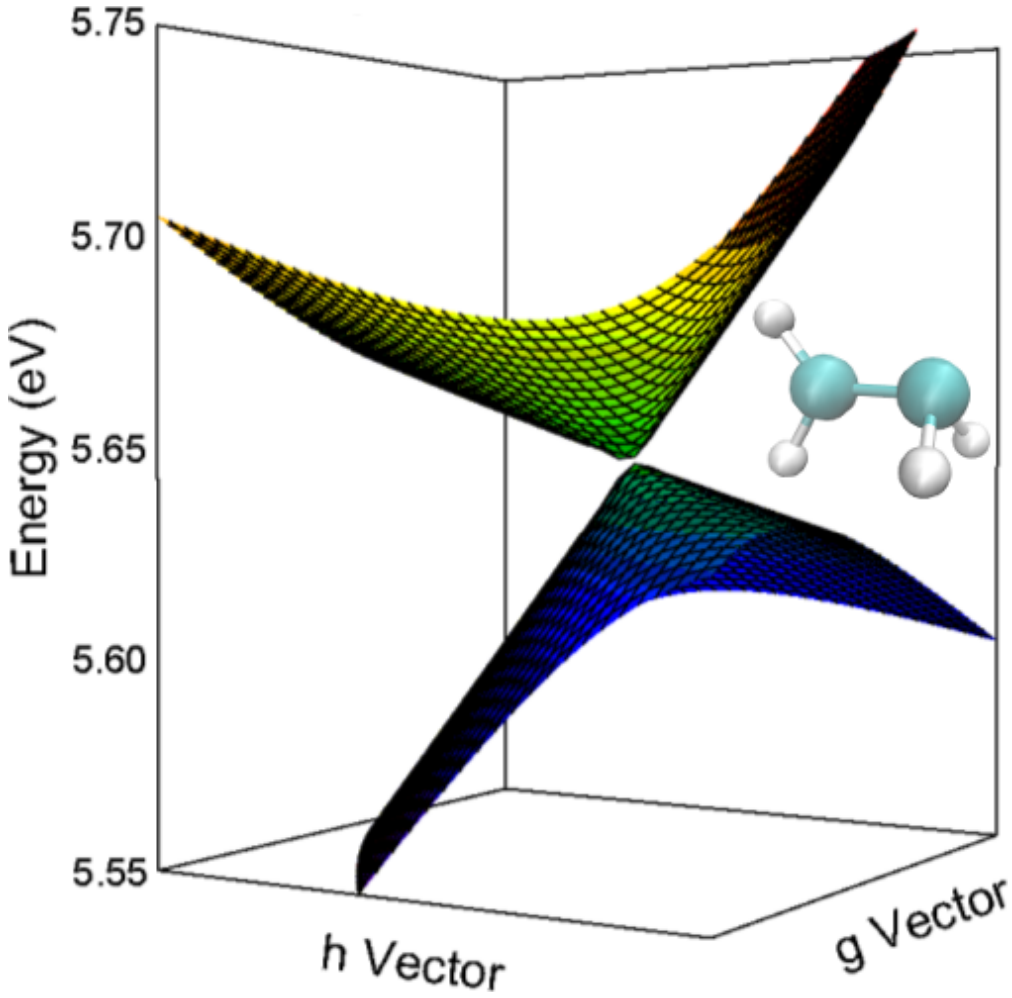

Figure 1.3: Twisted-pyramidalized minimum energy conical intersection (MECI) PES of ethylene at SA-2-CAS-(2,2)-SCF/6-31G**. Energies in eV relative to the Franck-Condon geometry S_0 energy.

1.3 CASSCF

The CASSCF method was developed to solve the problem of describing multireference systems[144]. The fully optimized reaction space(FORS)[148] method of Ruedenberg is closely related to CASSCF, and each is a specific case of the more general multiconfigurational self-consistent field (MCSCF) approach, where the reference space may be chosen to include any number of (user selected) configurations, which simplifies the selection of the reference space by instead specifying the electrons and orbitals to be included in the configuration expansion rather than explicitly selecting individual configurations. State-averaging

in the CASSCF method is important for treating each electronic state on an equal footing and for alleviating the root-flipping problem associated with excited state optimization problems[37, 198, 88, 90, 49]. The SA-CASSCF calculation proceeds by calculating both the SCF orbital and CI vector coefficients self-consistently by variationally optimizing the SA-CASSCF energy

$$E^{SA-CASSCF} = \sum_{pq} \gamma^{SA}_{pq} (p|\hat{h}|q) + \sum_{pqrs} \Gamma^{SA}_{pqrs} (pq|rs) \tag{1.16}$$

where $\boldsymbol{\gamma^{SA}}$ and $\boldsymbol{\Gamma^{SA}}$ are the state-averaged 1- and 2-particle reduced density matrices (RDMs) and $(p|\hat{h}|q)$ and $(pq|rs)$ are the 1- and 2-electron integrals indexed by orbitals p, q, r, s. The state-averaged RDMs are given in terms of the state-specific RDMs according to

$$\gamma^{SA}_{pq} = \sum_{\Theta} \omega_{\Theta} \gamma^{\Theta}_{pq} \tag{1.17}$$

$$\Gamma^{SA}_{pqrs} = \sum_{\Theta} \omega_{\Theta} \Gamma^{\Theta}_{pqrs} \tag{1.18}$$

where Θ indexes states included in the average and ω are the state weights.

SA-CASSCF may be performed using a two-step "super CI" approach[144], where the orbital and CI vector coefficients are optimized separately in succession, or using a Newton-Raphson[163] scheme where both sets of coefficients are optimized simultaneously. In either case, convergence difficulties may be encountered due to the highly non-linear nature of the coupled equations, and optimization of both the orbital and CI coefficients may be non-trivial even when convergence is straightforward. While advances in hardware technology have provided opportunities for the development of improved algorithms by our group[40, 166] and by others[60, 167] for more efficient calculation of SA-CASSCF energies and analytical gradients, the pursuit of lower cost SA-CASSCF alternatives having improved convergence characteristics remains worthwhile. One particular SA-CASSCF alternative is the complete active space

configuration interaction (CASCI) family of methods, closely related to CASSCF, where the orbital and CI vector coefficient optimization is uncoupled.

1.4 CASCI

CASCI methods aim to provide approximations to, or improve upon, the CASSCF method for describing statically correlated systems. Since no orbital re-optimization is performed, the quality of the final solution eigenvectors depends on the orbitals generated for the CI calculation. The CI step provides a description of static correlation and dynamic electron correlation within the CAS space, but fails to allow orbital relaxation in the same way as CASSCF. A variety of orbital generation schemes have been developed to compensate for this, including the use of unrestricted natural orbitals (UNO-CASCI)[22], improved virtual orbitals (IVO-CASCI)[136], natural orbitals derived from correlated wavefunctions including Möller-Plesset second-order perturbation theory (MP2), configuration interaction with single and double replacements (CISD), and coupled-cluster singles and doubles (CCSD)[1], floating occupied molecular orbitals (FOMO-CASCI)[165], high-multiplicity natural orbitals (HMNO-CASCI) in the context of multireference CI (MRCI)[96], and configuration interaction singles natural orbitals (CISNO-CASCI)[160]. Each of these methods provide an orbital basis suitable for describing statically correlated systems at a reduced computational cost relative to CASSCF.

1.5 Problem Statement

The previous sections serve to provide general background information and to motivate the work presented here. This dissertation discusses the following open challenges in computational chemistry:

- Many efficient means of describing multireference character rely on use of CI methods. Algorithms for FCI and active space CI, including CAS, restricted active space (RAS)[128], and generalized active space (GAS)[101], calculations are developed with

current hardware capabilities in mind. Previously, CPU based hardware limited the size of CI calculations to $\mathcal{O}(10^8)$ configurations for dynamics simulations or optimizations, and $\mathcal{O}(10^{10})$ configurations for benchmark type calculations. Chapter 2 deals with the problem of leveraging novel graphical processing unit (GPU) based hardware technologies to enable CI-based calculation of systems having larger atomic orbital basis sets and larger configuration spaces.

- Computational chemists benefit from the availability of analytical nuclear energy gradients and nonadiabatic and derivative coupling vectors. Reduced density matrices (RDMs) provide a convenient means of storing wavefunction information that may be used by subsequent gradient calculations. Chapter 3 describes the efficient GPU accelerated formation of 1- and 2-particle RDMs that enable SA-CASSCF and CASCI gradients and nonadiabatic and derivative coupling vectors.

- CI is often performed using a determinantal basis, rather than a configuration state function (CSF) basis, because of the convenient integer form of the 1-particle coupling coefficients. Aside from the additional computational cost associated with the use of determinants ($2-3\times$), determinants are not necessarily eigenfunctions of the total spin operator $\hat{S}^2$. This can, in certain situations, lead to catastrophic spin contamination of the eigenvectors. In Chapter 4 we demonstrate schemes to address numerical instabilities such as these encountered when solving the CI eigenvalue problem.

- While the ever expanding capabilities of new computer hardware allows us to tackle larger problems, the exponential scaling of the CI methods make brute force solutions generally unsuitable. Instead, in Chapter 5 we explore approximate methods for solving extremely large CI problems.

We conclude with a brief discussion of the work presented herein in Chapter 6 and by describing examples of studies made possible by the methods presented here.

CHAPTER 2

NANOSCALE MULTIREFERENCE QUANTUM CHEMISTRY: FULL CONFIGURATION INTERACTION ON GRAPHICAL PROCESSING UNITS

2.1 Abstract

Methods based on a full configuration interaction (FCI) expansion in an active space of orbitals are widely used for modeling chemical phenomena such as bond breaking, multiply-excited states, and conical intersections in small-to-medium-sized molecules, but these phenomena occur in systems of all sizes. To scale such calculations up to the nanoscale, we have developed an implementation of FCI in which electron repulsion integral transformation and several of the more expensive steps in $\boldsymbol{\sigma}$ vector formation are performed on graphical processing unit (GPU) hardware. When applied to a 1.7 x 1.4 x 1.4 nm silicon nanoparticle ($Si_{72}H_{64}$) described with the polarized, all-electron 6-31G** basis set, our implementation can solve for the ground state of the 16-active-electron/16-active-orbital CASCI Hamiltonian (more than 100,000,000 configurations) in 39 min on a single NVidia K40 GPU.

2.2 Introduction

Computational science has recently seen a resurgence in the vector processing paradigm due to the availability of low-cost video acceleration co-processors tailored for the gaming market known as graphical processing units (GPUs). With a large number of floating point units operating in parallel and a fast hierarchical memory subsystem, GPUs provide computational power comparable to that of a small cluster of CPU-based machines at a cost of only a few hundreds or thousands of dollars. Electronic structure calculations are

particularly amenable to GPU acceleration because many of the relevant algorithms are formulated in terms of matrix operations which can be efficiently vectorized. Applications of GPUs to electron repulsion integral evaluation and the self-consistent field procedure have proven fruitful [179, 180, 181, 9, 98, 193, 105, 195, 5, 11, 113, 122, 200, 114], and correlated, linear-scaling, and excited state methods have also been accelerated by GPUs [183, 126, 35, 65, 102, 10, 103, 36, 83, 109, 81, 201]. Production calculations which take advantage of GPUs for the study of complex nanoscale or biological systems or for automated discovery of new chemistries are now regularly reported [110, 186, 185, 199, 162].

To date most GPU-accelerated development efforts in ab initio quantum chemistry have focused on single-reference methods, though exceptions include recent implementations of the complete active space self-consistent field (CASSCF) method[60] and multireference coupled cluster theory[20]. Many chemical phenomena, such as bond breaking, multiply-excited electronic states, and non-radiative decay via conical intersections, benefit from treatment at a multireference level of theory [152, 171], as evidenced by the popularity of CASSCF, which is the standard tool used for calculating zeroth-order wave functions of such strongly correlated systems[147]. Inspired by the success of CASSCF, a class of efficient and accurate alternatives has been introduced: the *two-step* complete active space configuration interaction (CASCI) methods, in which the optimization of the orbitals is decoupled from that of the configuration interaction (CI) coefficients. The various two-step methods are differentiated by the procedure used to determine the orbitals on which the CI expansion is based. Orbital determination schemes suitable for describing bond breaking include the unrestricted natural orbital (UNO) method [22] and methods based on the natural orbitals of correlated single-reference methods [2]. Two-step methods have been gaining popularity for the description of excited states as well, and various schemes exist: improved virtual orbitals (IVO)[136, 30], floating occupation molecular orbitals (FOMO)[50, 165], high-multiplicity natural orbitals (HMNO)[95, 97], variationally optimal orbitals from restricted active space calculations (singly excited active space; SEAS)[161], and configuration interaction singles

natural orbitals (CISNO)[160]. In addition to computational efficiency, two-step methods often exhibit other desirable behaviors, such as a reduced propensity for spatial symmetry breaking [161], size intensivity [160], and simplification of the choice of active space[160, 69].

CASCI is simply full CI (FCI) in a user-defined active space of orbitals, and the last several decades have seen the development of many efficient FCI algorithms[158]. These *direct* algorithms achieve efficiency by circumventing the formation of the full Hamiltonian matrix. Though spin-adapted formulations significantly reduce the dimensionality of the FCI problem [130, 157], determinantal CI algorithms became popular in the 1980s due to the ease with which the coupling coefficients can be evaluated [53]. Though several FCI algorithms with reduced operation counts have been developed [128, 203, 18], the seminal algorithm introduced by Knowles and Handy (KH) [73] is an excellent starting point for a GPU implementation of FCI because of its amenability to vectorization and the regular memory access pattern it allows. Specifically, the performance limiting step of the KH algorithm is a large matrix multiplication, for which extremely efficient GPU algorithms exist.

Further acceleration of FCI or FCI-like calculations can be achieved in various ways. It is possible to improve performance by taking advantage of the sparsity of the data structures involved [72, 115, 140]. The density matrix renormalization group approach allows chemists to solve problems with larger active spaces than previously possible [192, 117, 28, 86, 118]. In systems with larger one-electron bases, performance may be determined not by the cost of the diagonalization itself, but by the preceding two-electron integral transformation. The formal scaling of this step has been reduced by employing approximations to the integrals [108, 175, 8, 34], and reduced effective scaling has also been observed when efficient prescreening is employed in the absence of any additional approximation[155, 60].

Herein we report the implementation of a vectorized direct determinantal FCI algorithm on GPU hardware. Section 2.3 describes our adaptations of two potentially performance-limiting steps to GPU hardware: the two-electron integral transformation and the KH al-

gorithm for formation of the $\boldsymbol{\sigma}$ vector. In Section 2.4 we report the performance of our algorithm for systems as large as a 1.7 x 1.4 x 1.4 nm silicon nanoparticle. In Section 2.5 we draw conclusions and discuss future prospects.

2.3 Methods

A typical CASCI calculation requires a) determination of a suitable set of orbitals, b) transformation of integrals from the atomic orbital (AO) to the molecular orbital (MO) basis, and c) solving for the lowest eigenvalues and eigenstates of the CASCI Hamiltonian. Each of these steps, respectively, contain a potential bottleneck: a) solution of the Hartree-Fock (HF) wave function (or perhaps configuration interaction singles (CIS), in the case of CISNO), b) two-electron integral transformation, and c) formation of the $\boldsymbol{\sigma}$ vector. GPU algorithms for HF and CIS have been well documented [179, 180, 181, 65]. Below we report the details of our GPU implementations of each of the latter two potential bottlenecks. Note that double precision floating point math was used for all calculations presented in this work, even though GPUs perform floating point math faster in single precision.

2.3.1 Integral Transformation

The transformation of the electron repulsion integrals (ERIs) is the performance-limiting step for a FCI calculation using a large single-electron basis and modest active space, so we focus on the GPU implementation of this step here. When fast integral evaluation is possible (as on the GPU), integral-direct calculation of the transformed integrals can provide good scaling without the need for storage of large matrices of intermediate sums[55, 45, 155]. In our implementation we take advantage of the existence of efficient GPU-accelerated code for computing generalized Coulomb matrices, $\mathbf{J_{ij}}$, to achieve these goals [180]. ($[J_{ij}]_{\mu\nu} = (\mu\nu|ij)$; two-electron integrals are expressed in chemists' notation, μ and ν index the AO basis, and i and j index the transformed (MO) basis.)

Our transformation proceeds as follows: The matrix $\mathbf{J_{ij}}$ is computed for a particular pair,

ij, from a density matrix created by taking the outer product of molecular orbital i with orbital j. From $\mathbf{J_{ij}}$, a two-index transformation is performed to build the fully transformed integrals

$$(ij|kl) = \sum_{\mu\nu} C_{\mu k} C_{\nu l} [J_{ij}]_{\mu\nu} \tag{2.1}$$

where k and l (like i and j) index the MO basis and $C_{\mu k}$ is the coefficient of basis function μ in MO k. Then, $\mathbf{J_{ij}}$ is discarded before computing $\mathbf{J_{ij}}$ for the next pair ij. Note that to perform a CASCI calculation i, j, k, and l need only span the space of active orbitals. Formally, the operation count of this calculation scales as $\mathcal{O}(N^4 m^2)$, where N and m are the number of AOs and active MOs, respectively, and thus $m << N$ in most cases. Only enough storage for $(ij|kl)$ (m^4) and a single instance of $\mathbf{J_{ij}}$ (N^2) is required. Further reduction of the formal scaling of the operation count to $\mathcal{O}(N^4 m)$ would be possible by taking an alternative approach: performing four successive one-index transforms rather than two two-index transforms as above. However, the observed scaling of the two-index algorithm, which will be analyzed in detail below, is considerably less than $\mathcal{O}(N^4 m^2)$ because we prescreen the two-electron integrals in the process of forming $\mathbf{J_{ij}}$, as described by Ufimtsev, et al.[180]

2.3.2 KH Direct FCI

Direct CI methods require access only to the transformed integrals, the one-electron coupling coefficients, $\boldsymbol{\gamma}$, and a matrix listing the orbital labels corresponding to the non-zero elements of $\boldsymbol{\gamma}$, $\mathbf{l}$. As noted above, we work in a determinantal basis to simplify evaluation of $\boldsymbol{\gamma}$. As a result, computational performance depends strongly on how efficiently the integrals and coupling coefficients are accessed in memory. A lexical configuration ordering scheme allows for contiguous memory access, an important consideration for many high performance computing platforms, but especially important for GPUs. By writing configurations as strings of α and β occupied orbitals and by taking advantage of their symmetry, $\boldsymbol{\gamma}$ and $\mathbf{l}$ can be accessed contiguously in all cases. The graphical method described by Duch[38]

was used to map the addresses for the α and β strings and is functionally identical to the method described by KH[73]. Below we review the KH algorithm and then detail our GPU adaptation, with particular attention to memory access.

Our code uses the iterative Davidson solver[33] for diagonalization of the Hamiltonian matrix. The most computationally expensive step in this procedure is generally the formation of the $\boldsymbol{\sigma}$ vector

$$\sigma_I = \sum_J H_{IJ} c_J \tag{2.2}$$

where c_J is an element of the CI coefficient vector, H_{IJ} is a Hamiltonian matrix element, and I and J index determinants. For problems involving a large active space and small single electron basis, it is this step rather than the integral transformation that is performance-limiting. The Hamiltonian matrix elements can be represented as[164]

$$H_{IJ} = \sum_{ij}^{n} \left[\gamma_{ij}^{IJ} (i|\hat{h}|j) - \sum_k \gamma_{ik}^{IJ} (ik|kj) \right] + \frac{1}{2} \sum_{ijkl}^{n} \sum_K \gamma_{ij}^{IK} \gamma_{kl}^{KJ} (ij|kl) \tag{2.3}$$

where K (like I and J) indexes determinants and $\hat{h}$ is the one-electron Hamiltonian. The one-electron coupling coefficients are defined

$$\gamma_{ij}^{IJ} = \langle \Psi_I \left| \hat{E}_{ij} \right| \Psi_J \rangle \tag{2.4}$$

where $\hat{E}_{ij}$ is the excitation operator that promotes an electron from orbital j to orbital i and Ψ_I is a determinant. In terms of α and β strings ($|\Psi_I\rangle = |\Psi_{\alpha\beta}\rangle$ and $|\Psi_J\rangle = |\Psi_{\alpha'\beta'}\rangle$), we could equivalently represent the coupling coefficients as

$$\gamma_{ij}^{\alpha\beta\alpha'\beta'} = \langle \Psi_{\alpha\beta} \left| \hat{E}_{ij} \right| \Psi_{\alpha'\beta'} \rangle \tag{2.5}$$

Of course E_{ij} is a one-electron operator, and therefore $\gamma_{ij}^{\alpha\beta\alpha'\beta'}$ can only be non-zero when either $\alpha = \alpha'$ or $\beta = \beta'$. If $\beta = \beta'$, for example, $\gamma_{ij}^{\alpha\beta\alpha'\beta}$ is the same for all β, and thus the dimensionality of $\boldsymbol{\gamma}$ can be reduced, with all non-zero elements described by

$$\gamma_{ij}^{\alpha\alpha'} = \langle \Psi_{\alpha\beta} \left| \hat{E}_{ij} \right| \Psi_{\alpha'\beta} \rangle \tag{2.6}$$

for all β and

$$\gamma_{ij}^{\beta\beta'} = \langle \Psi_{\alpha\beta} \left| \hat{E}_{ij} \right| \Psi_{\alpha\beta'} \rangle \tag{2.7}$$

for all α. Further, for a given pair of strings, $\alpha \neq \alpha'$, $\gamma_{ij}^{\alpha\alpha'}$ is non-zero for at most a single pair of orbitals, ij. Thus, storage is saved by storing each non-zero element, $\gamma_{ij}^{\alpha\alpha'}$, in a pair of array elements: $\mathbf{l}[\alpha, \alpha']$, which stores the value of ij for which $\gamma_{ij}^{\alpha\alpha'}$ is non-zero, and $\boldsymbol{\gamma}[\alpha, \alpha']$ which stores the value of $\gamma_{ij}^{\alpha\alpha'}$ itself.

The two-electron term on the far right of Equation 2.3 forms the bulk of the effort. In the KH algorithm, the contributions to $\boldsymbol{\sigma}$ from this term are calculated in a series of three steps [73]

$$D_{ij}^{K} = \sum_{J} \gamma_{ij}^{JK} c_J \tag{2.8a}$$

$$E_{kl}^{K} = \sum_{ij} \frac{1}{2} (ij|kl) D_{ij}^{K} \tag{2.8b}$$

$$\sigma_I = \sum_{K} \sum_{kl}^{n} \gamma_{kl}^{KI} E_{kl}^{K} \tag{2.8c}$$

Applying the definition of $\boldsymbol{\gamma}$ in Equations 2.6 and 2.7 to Equations 2.8a - 2.8c, we arrive at the serial form of the FCI algorithm originally described by KH (Algorithm 1)[73]. This algorithm has five steps: calculation of the α and β contributions to the $\mathbf{D}$ matrix, formation of the $\mathbf{E}$ matrix (the central matrix-matrix multiply), and calculation of the α and β contributions to the $\boldsymbol{\sigma}$ vector. Equation 2.8b is simply a matrix-matrix multiplication, and it is this step that dominates the operation count of the KH algorithm. This operation is trivially ported to the GPU by a call to the CUBLAS library [124], which contains a highly

Algorithm 1 Serial Knowles and Handy algorithm

for strings α **do**
  for strings α' differing from α by zero or one occupations **do**
    for strings β **do**
      $ij \leftarrow \mathbf{l}[\alpha, \alpha']$
      $\mathbf{D}[ij, \alpha, \beta] \mathrel{+}= \boldsymbol{\gamma}[\alpha, \alpha']\mathbf{C}[\alpha', \beta]$
    end for
  end for
end for
for strings β **do**
  for strings β' differing from β by zero or one occupations **do**
    for strings α **do**
      $ij \leftarrow \mathbf{l}[\beta, \beta']$
      $\mathbf{D}[ij, \alpha, \beta] \mathrel{+}= \boldsymbol{\gamma}[\beta, \beta']\mathbf{C}[\alpha, \beta']$
    end for
  end for
end for
matrix multiply: $\mathbf{E}[kl, \alpha, \beta] = (\mathbf{ij}|\mathbf{kl})\mathbf{D}[ij, \alpha, \beta]$
for strings α **do**
  for strings α' differing from α by zero or one occupations **do**
    for strings β **do**
      $ij \leftarrow \mathbf{l}[\alpha, \alpha']$
      $\boldsymbol{\sigma}[\alpha', \beta] \mathrel{+}= \boldsymbol{\gamma}[\alpha, \alpha']\mathbf{E}[ij, \alpha, \beta]$
    end for
  end for
end for
for strings β **do**
  for strings β' differing from β by zero or one occupations **do**
    for strings α **do**
      $ij \leftarrow \mathbf{l}[\beta, \beta']$
      $\boldsymbol{\sigma}[\alpha, \beta'] \mathrel{+}= \boldsymbol{\gamma}[\beta, \beta']\mathbf{E}[ij, \alpha, \beta]$
    end for
  end for
end for

optimized matrix-matrix multiplication routine. We treat this as a dense matrix operation in the present implementation. The $\mathbf{D}$ matrix can be quite sparse in the first several iterations of the CI procedure, however[164], and accounting for this sparsity to further reduce the computational effort is an area of active development. Naive implementation of the lower-scaling operations in Equations 2.8a and 2.8c results in performance that is limited by these steps. Thus, efficient implementation of these lower-scaling steps is essential for high performance, and it is the GPU implementation of these that will be presented below.

The nested loop structures that form $\mathbf{D}$ and $\boldsymbol{\sigma}$ are similar. Each comprises three nested loops over strings, with the string corresponding to the innermost loop differing in spin from

Algorithm 2 GPU vectorized calculation of the α terms in $\mathbf{D}$

GPU vectorize over strings α, β
  for all orbitals i occupied in α **do**
    $\mathbf{D}[ii, \alpha, \beta] += \mathbf{C}[\alpha, \beta]$
  end for
end GPU vectorize
GPU vectorize over strings α, α', β, where α and α' differ by exactly one occupation
  $ij \leftarrow \mathbf{l}[\alpha, \alpha']$
  $\mathbf{D}[ij, \alpha, \beta] += \boldsymbol{\gamma}[\alpha, \alpha']\mathbf{C}[\alpha', \beta]$
end GPU vectorize

those of the outer two loops. In what follows, we consider in detail the GPU implementation of the first set of nested loops in Algorithm 1: the calculation of the α contributions to $\mathbf{D}$. The GPU algorithm for this step is presented in Algorithm 2. Though we do not explicitly present the GPU algorithms for the calculations of the β contributions to $\mathbf{D}$ and both components of $\boldsymbol{\sigma}$, their implementation is analogous to that of the α contributions to $\mathbf{D}$. Key differences will be noted below.

The expressions for the diagonal ($\alpha = \alpha'$) and off-diagonal ($\alpha \neq \alpha'$) contributions to the matrix elements for $\mathbf{D}$ and $\boldsymbol{\sigma}$ differ, and as such we split the calculation of these two sets of terms into two different GPU kernels. In the computation of the diagonal terms (first block in Algorithm 2), the calculation is vectorized across α and β. Because $\alpha = \alpha'$, no loop over α' is required, and this term is low enough scaling so as to never be performance limiting. We therefore focus our attention on the efficient calculation of the off-diagonal terms (second block in Algorithm 2). Here $\alpha \neq \alpha'$ so we vectorize over a third string, α', in addition to α and β. However, while we must compute terms for all possible values of α and β, we take advantage of the sparse nature of $\boldsymbol{\gamma}$ by including only those α' for which there is some non-zero $\boldsymbol{\gamma}[\alpha, \alpha']$. Each thread loads a coupling coefficient, $\boldsymbol{\gamma}[\alpha, \alpha']$ (along with its corresponding $\mathbf{l}[\alpha, \alpha']$), and CI coefficient, $\mathbf{C}[\alpha', \beta]$, multiplies them, and adds the product to element $\mathbf{D}[ij, \alpha, \beta]$. For optimal performance, it would be desirable to access memory in one of two ways: such that contiguous memory is simultaneously accessed by neighboring threads (*coalesced* memory access, in GPU parlance) or such that blocks of threads all access exactly the same element of memory. An optimal memory access pattern is achieved through careful

ordering of array and thread indices. Figure 2.1 depicts the arrangement of data structures whose efficient access is critical to algorithm performance. Note that the $\mathbf{D}$ and $\mathbf{E}$ matrices are ordered using the same convention, as are the $\mathbf{C}$ and $\boldsymbol{\sigma}$ data structures. Access to $\mathbf{D}[ij, \alpha, \beta]$ and $\mathbf{C}[\alpha', \beta]$ are coalesced, and each thread block requires access to only a single value of $\mathbf{l}$ and $\boldsymbol{\gamma}$.

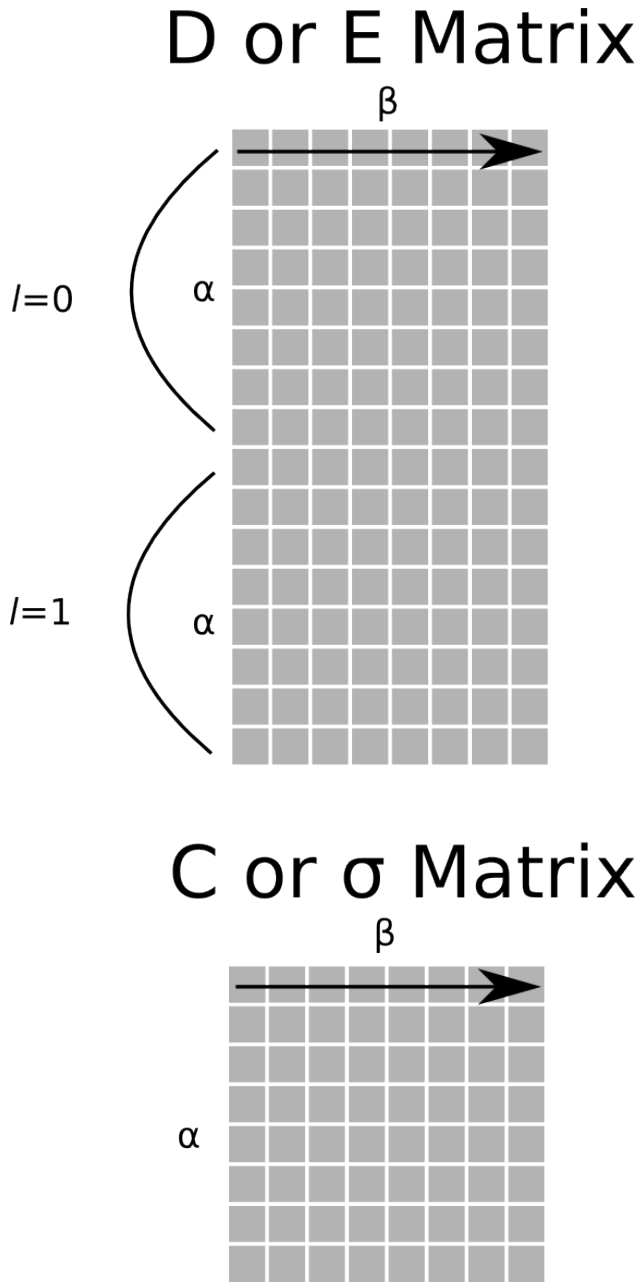

Figure 2.1: Memory layout for relevant data structures. The contiguous memory direction is defined here as being row-major, as indicated by the arrow, keeping with the C programming language convention. We depict the number of l=ij pairs here as being 2, while in practice the number of ij pairs is m^2.

The off-diagonal β contributions to $\mathbf{D}$ are calculated in a similar fashion to the α contributions. The thread indices are reorganized such that $\mathbf{C}[\alpha, \beta']$, $\mathbf{l}[\beta, \beta']$, and $\boldsymbol{\gamma}[\beta, \beta']$ are accessed in a coalesced fashion. Access to $\mathbf{D}[ij, \alpha, \beta]$ itself is not coalesced, however. In an

attempt to eliminate this non-coalesced memory access, we have implemented a second algorithm in which $\mathbf{D}$ is transposed between the calculation of the α and β terms. It was found that the prior approach provides better performance than the latter because the transpose of $\mathbf{D}$ requires an inefficient memory access pattern similar to that required by the non-coalesced algorithm. Therefore, it is the non-coalesced algorithm that was used for calculation of the β terms of $\mathbf{D}$, though utilization of a more optimized transpose algorithm may be worth further investigation [99].

The α and β contributions to the $\boldsymbol{\sigma}$ vector are computed using identical indexing patterns and vectorization schemes to their $\mathbf{D}$ counterparts, but there is one significant difference. Formation of $\boldsymbol{\sigma}$ requires several threads to sum into the same location of memory which, implemented naively, results in a race condition. The solution is to employ an atomic add operation to ensure inclusion of all contributions to the final $\boldsymbol{\sigma}$ vector. While hardware support for double precision floating point atomic addition is not currently available on NVidia GPU hardware, we were able to achieve the same functionality using a custom kernel that invokes the atomic compare-and-swap library function to ensure synchronized memory access across thread blocks[125]. A similar approach has previously been used in the implementation of radial distribution function histograms on GPU hardware [91]. Finally, we have implemented the one-electron contributions to the $\boldsymbol{\sigma}$ vector using a GPU vectorized approach very similar to that which was taken for the two-electron contributions.

2.3.3 Computational Details

The above-described direct FCI algorithm was implemented for NVidia GPU hardware in a development version of the TeraChem software package[181] using the Compute Unified Device Architecture (CUDA) API[125]. All benchmark calculations were performed on a system composed of an Intel Xeon E5-2680 2.80 GHz processor and a single NVidia K40 GPU. The accuracy of our code was established by comparison to identical calculations performed using the Molpro software package[190, 73, 189, 76, 74, 155]. Except where noted, all CASCI

Table 2.1: CASCI/6-31G** times-to-solution using (6,6), (12,12), and (16,16) active spaces. The numbers of CI iterations required for convergence are shown in parentheses. Times-to-solution for the HF step, which is required for orbital determination but is not included in the CASCI times reported here, are shown in the final column for comparison.

System	**Time-to-Solution (s)**			
	(6,6)	**(12,12)**	**(16,16)**	**HF**
Pyrazine	0.51 (5)	3.75 (7)	1834.74 (11)	1.14
Dimelamine	2.81 (5)	11.17 (6)	689.23 (6)	8.48
Dimelem	7.19 (4)	22.75 (6)	1047.50 (7)	31.36
C_{60}	63.69 (3)	163.42 (5)	807.82 (6)	258.01
$Si_{72}H_{64}$	458.42 (1)	1200.40 (2)	2343.93 (2)	1595.36

calculations in this work were performed in the basis of canonical Hartree-Fock orbitals for simplicity, but note that the orbital determination steps for CISNO, IVO, and FOMO have all been implemented either by our group or by Hohenstein, et al.[160, 60], and will be available in a future release of TeraChem. Throughout this work active spaces are abbreviated (n, m) where n is the number of active electrons and m is the number of active orbitals. Except where noted, all calculations use the 6-31G** basis and thus describe all electrons explicitly and include polarization functions on all atoms. Reported times-to-solution include the integral transformation and direct diagonalization, but exclude any orbital determination step. HF times-to-solution, which are comparable to orbital determination times-to-solution in many cases, have been reported separately for the reader's convenience.

2.4 Results and Discussion

The test set of molecules used in our benchmark calculations is shown in Figure 2.2. It includes one of the classic systems in molecular photodynamics (pyrazine), two models of graphitic carbon nitride photocatalyst (dimelamine and dimelem), and two nanoscale systems (C_{60} and $Si_{72}H_{64}$). Note that $Si_{72}H_{64}$ is a silicon nanoparticle measuring 1.7 x 1.4 x 1.4 nm.

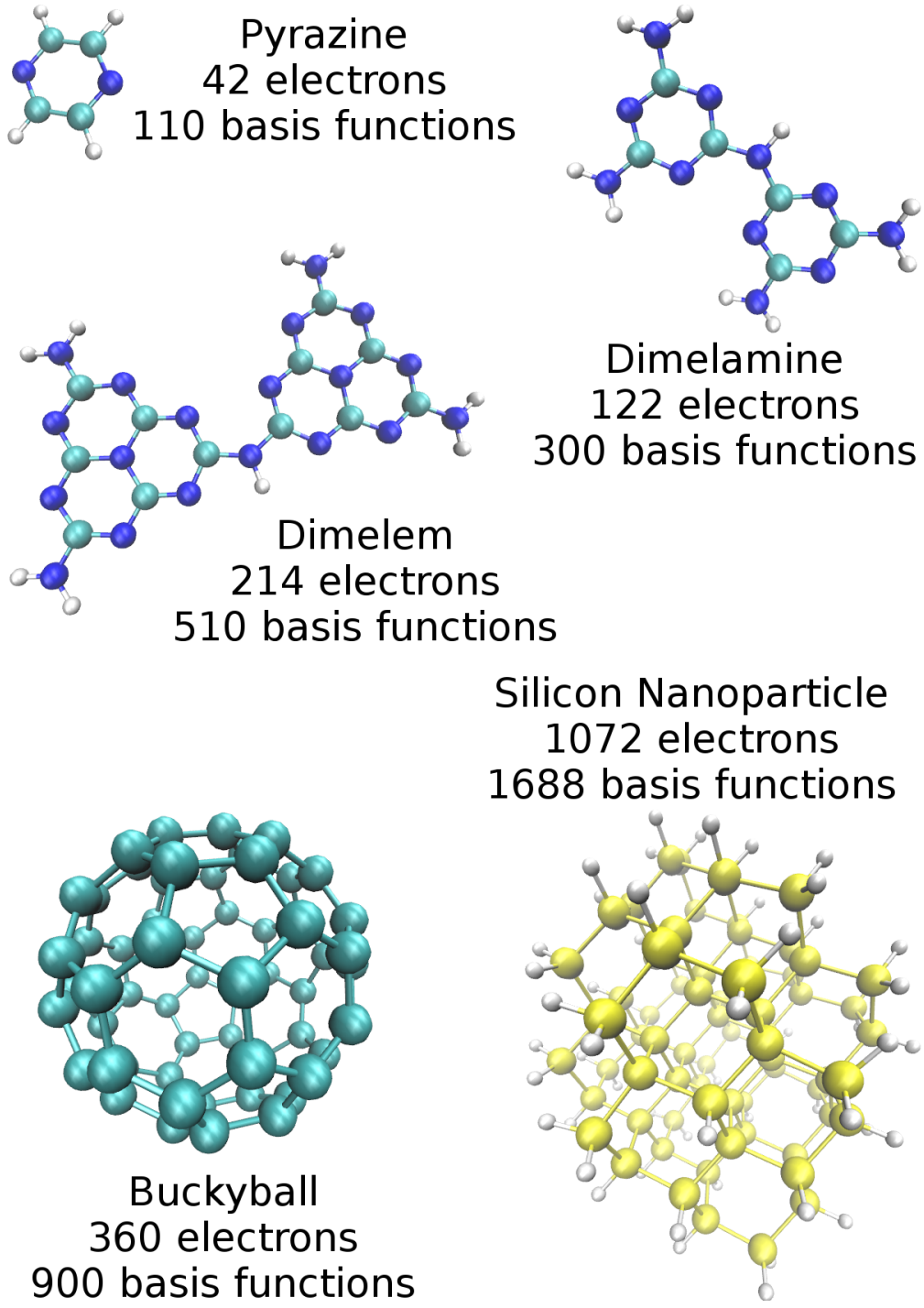

Figure 2.2: Test set molecules. Carbon, nitrogen, silicon, and hydrogen atoms are shown in teal, blue, yellow, and white, respectively. The numbers of basis functions correspond to the 6-31G** basis.

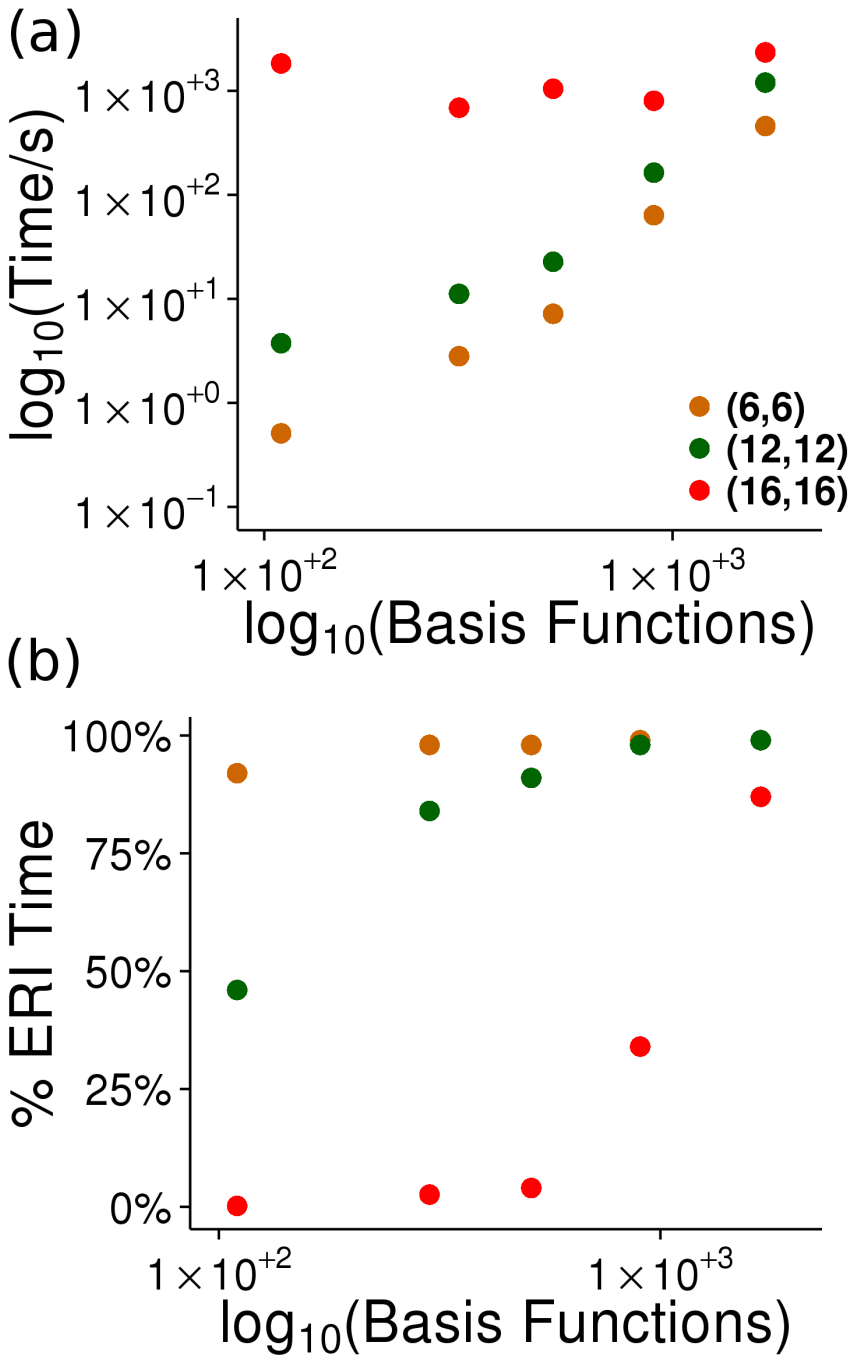

Figure 2.3: CASCI/6-31G** times-to-solution (a) and percent of total time spent performing the ERI transformation (b) for our test set molecules plotted as a function of the number of single-electron basis functions (N). Results for the (6,6), (12,12), and (16,16) active spaces are shown in orange, green, and red, respectively.

2.4.1 Overall Performance

The total times-to-solution for CASCI/6-31G** calculations of the test set molecules (excluding any orbital determination step) are reported in Figure 2.3a and Table 2.1 with active spaces ranging from (6,6) to (16,16). The times range from subsecond (0.51 s for pyrazine with a (6,6) active space) to just over half an hour (2343.93 s = 39 min for $Si_{72}H_{64}$ with a (16,16) active space). Note that for the (6,6) active space the CASCI calculation is always faster than the HF calculation required to determine the orbitals. Even for the (12,12) active space, solving the HF equations requires more time than the subsequent CI calculation for the larger systems (dimelem, C_{60}, and $Si_{72}H_{64}$).

There is a very noticeable difference in scaling behavior for the different active spaces. This is more easily understood if we consider the percentages of the total times-to-solution required to perform the ERI transformations, which are reported in Figure 2.3b and in parentheses in Table 2.2. As expected, the ERI cost dominates when large molecules are treated with small active spaces, and the CI (which consumes the remainder of the time) dominates when small molecules are treated with large active spaces. The ERI transformation requires more than 90% of the total time-to-solution for all systems when they are computed with the small (6,6) active space, and consumes more than half of the time for all but the smallest system (pyrazine) when the (12,12) space is used. The ERI transformation requires 87% of the time-to-solution for the largest reported calculation ($Si_{72}H_{64}$, (16,16)). For these systems where the performance is dominated by the ERI transformation the total time-to-solution appears to scale strongly with system size. Only when smaller molecules are treated with the (16,16) active space does the CI diagonalization dominate. In these cases there is no apparent dependence of performance on the size of the basis set, which is consistent with the fact that the CI diagonalization dominates the cost of the calculation. Instead, it is observed that the number of CI iterations (reported in parenthesis in Table 2.1) determines the relative time-to-solution.

Table 2.2: Comparison of the times required to perform ERI transformations with different sized active spaces and single-electron basis sets. The percentage of the total CASCI time-to-solution attributable to the ERI transformation is reported in parentheses. All calculations used the 6-31G** basis and were performed on a single NVidia K40 GPU.

System	**Basis Functions**	**ERI Transformation Time (s)**		
		(6,6)	**(12,12)**	**(16,16)**
Pyrazine	110	0.47 (92%)	1.73 (46%)	3.24 (0.2%)
Dimelamine	300	2.78 (98.9%)	9.34 (84%)	17.77 (2.6%)
Dimelem	510	7.06 (98.2%)	20.77 (91%)	42.22 (4.0%)
C_{60}	900	63.18 (99.2%)	161.12 (98.6%)	277.35 (34%)
$Si_{72}H_{64}$	1688	455.66 (99.4%)	1196.02 (99.6%)	2043.81 (87%)

2.4.2 ERI Transformation Performance

The times required to compute the transformed two-electron integrals for our test set molecules using three different size active spaces ((6,6), (12,12), and (16,16)) are presented in Figure 2.4 and Table 2.2. These times range from sub-second (0.47 s for pyrazine, $m = 6$) to roughly half an hour (2043.81 s = 34 min for $Si_{72}H_{64}$, $m = 16$). Though the direct ERI transformation procedure described above formally scales as $\mathcal{O}(N^4m^2)$, it is worthwhile to investigate the scaling observed in practice, as the observed scaling may be significantly more favorable than the theoretical scaling when the integrals are prescreened[155, 60]. Upon attempting to fit the data in Figure 2.4 to extract the effective scaling exponent we noted that a linear function fit the data very poorly. We hypothesized that this is because the planar graphitic carbon nitride systems exhibit different scaling behavior than denser, three-dimensional structures, like the silicon nanoparticle. We thus investigate the scaling of the integral transformation by performing benchmark calculations on two series of molecules with basis sets of consistent sparsity: basis-function-dense, three-dimensional silicon clusters and models of planar graphitic carbon nitride for which we expect the ERIs to be sparser. The silicon systems computed are $Si_{15}H_{25}$, $Si_{25}H_{30}$, $Si_{44}H_{44}$, $Si_{50}H_{50}$, and $Si_{72}H_{64}$, while the carbon nitride systems are $C_4N_2H_4$, $C_6N_{11}H_9$, $C_{12}N_{19}H_9$, $C_{18}N_{27}H_9$, and $C_{36}N_{52}H_{12}$. Both sets of clusters are treated with the 6-31G** basis. Additionally, the series of sili-

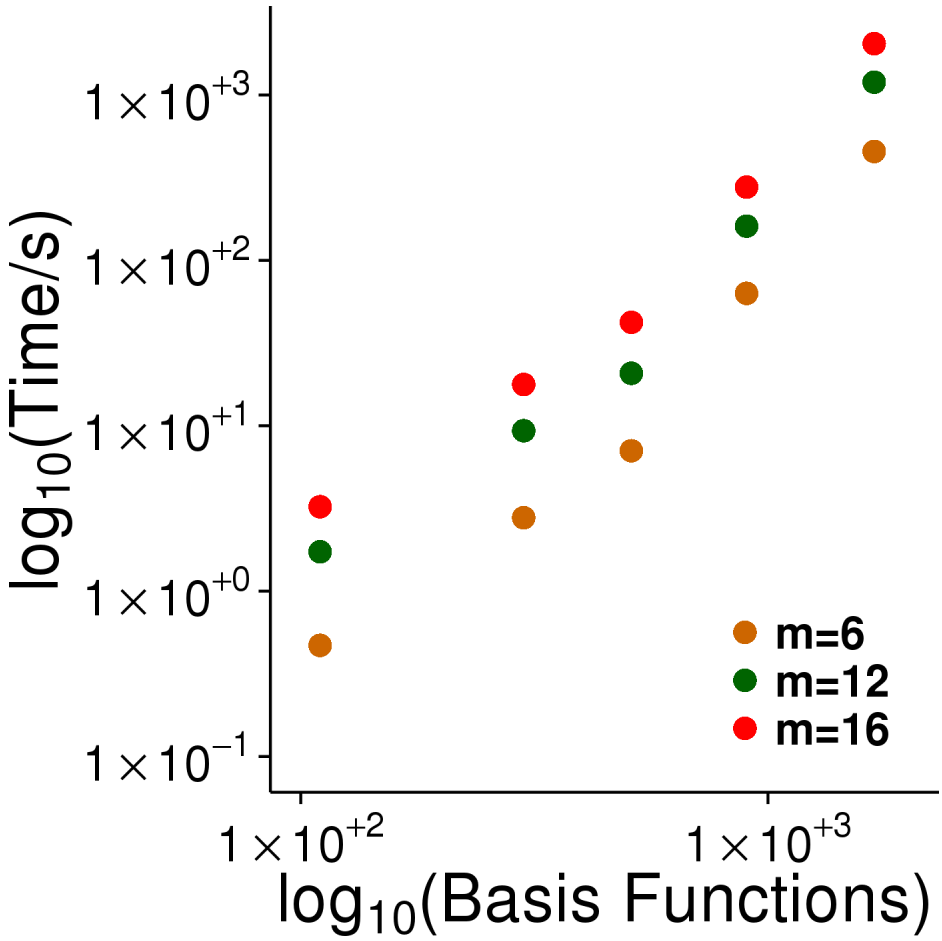

Figure 2.4: Times required to perform the ERI transformations as a function of the number of single-electron basis functions (N) for different numbers of active orbitals (m). All calculations used the 6-31G** basis and were performed on a single NVidia K40 GPU.

con clusters was computed using the much smaller LANL2DZ effective core potential basis set [184] to investigate the effect of atomic basis set size on scaling. All three series were computed with three different size active spaces ($m = 6$, $m = 12$, and $m = 16$).

The results of these scaling studies are presented in Figures 2.5 and 2.6. Geometries and tables of the timing data, including total times-to-solution and ERI transformation times, are presented in Supporting Information. In all cases, we have performed linear regression analysis of the log(ERI transform time) as a function of log(N) to determine the effective scaling exponents. For all three series the effective scaling exponent is considerably lower

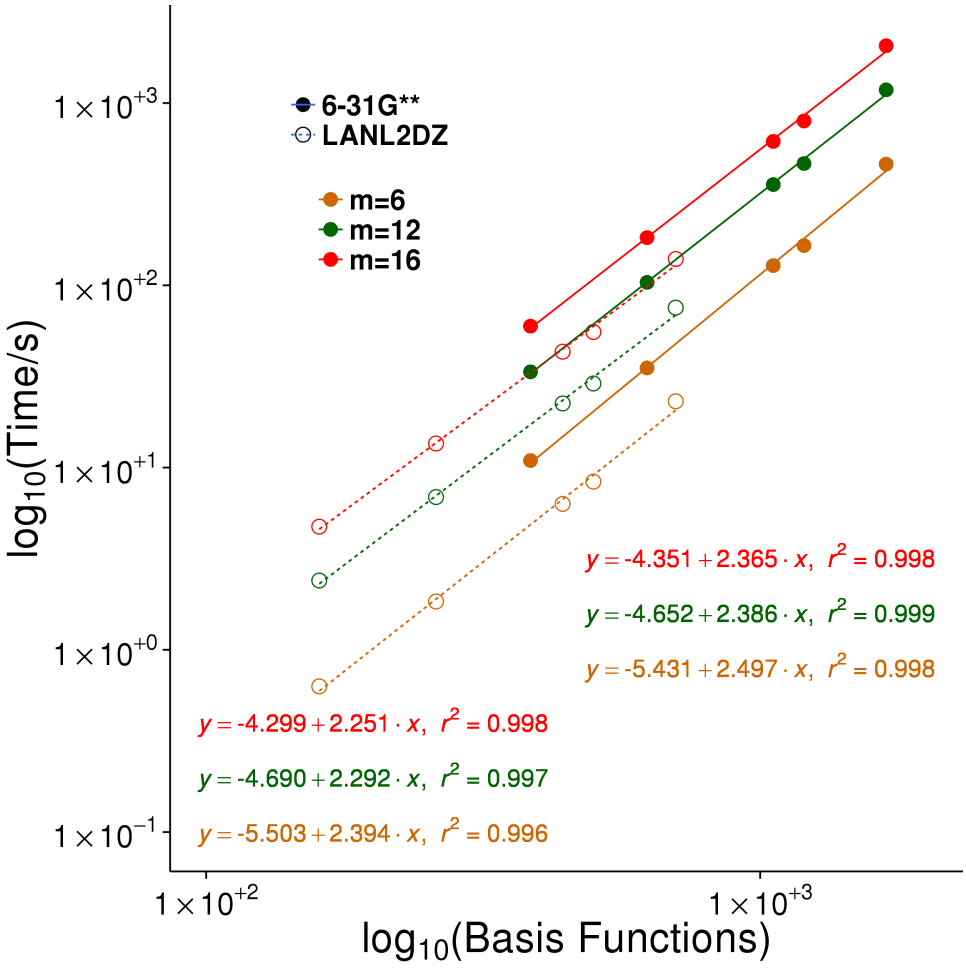

Figure 2.5: Times required to perform the ERI transformations for a series of silicon clusters as a function of the number of single-electron basis functions (N) for different numbers of active orbitals (m). Calculations using the LANL2DZ(6-31G**) basis are depicted using open(filled) circles and were performed on a single NVidia K40 GPU. Linear fits used to determine scaling exponents are shown as dashed(solid) lines.

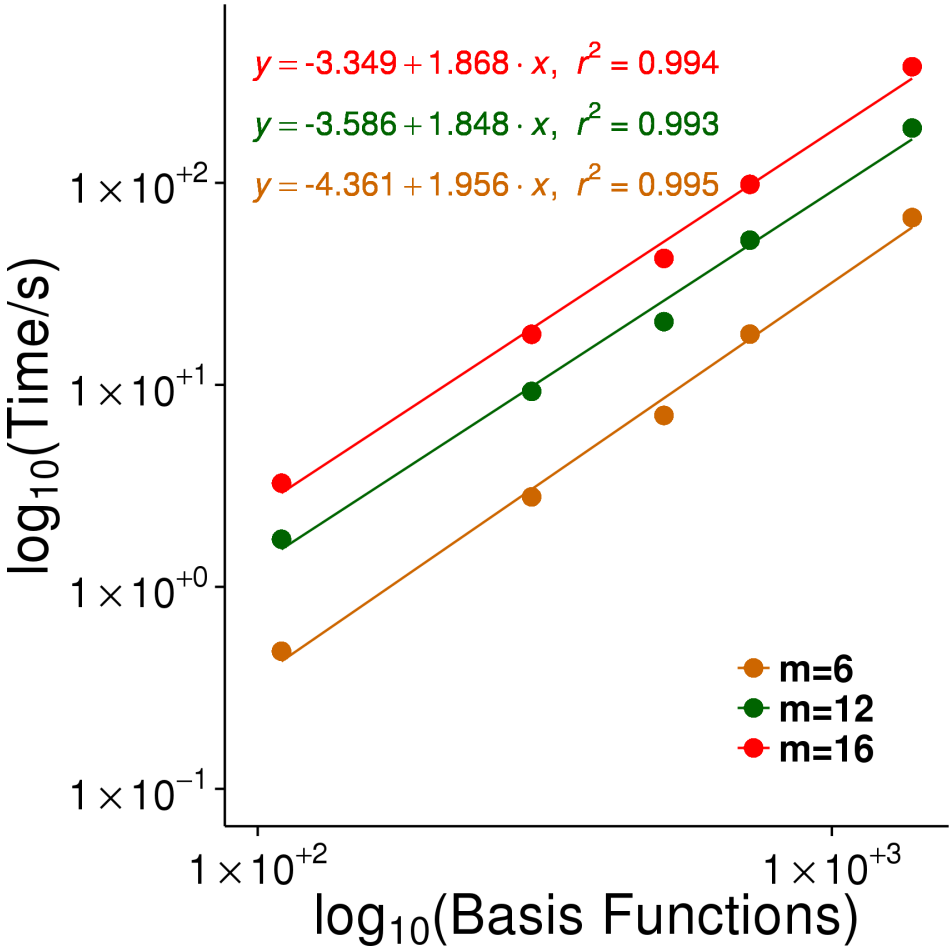

Figure 2.6: Times required to perform the ERI transformations for a series of models of graphitic carbon nitride as a function of the number of single-electron basis functions (N) for different numbers of active orbitals (m). All calculations used the 6-31G** basis and were performed on a single NVidia K40 GPU. Linear fits used to determine scaling exponents are shown as solid lines.

than the theoretical value of 4.0. When the silicon cluster series is computed with the LANL2DZ basis set (Figure 2.5) scaling exponents in the range 2.2-2.4 are observed for the various active spaces. The same series computed with the larger 6-31G** basis set (Figure 2.5) yields nearly identical scaling exponents: 2.3-2.5. Thus, the size of the atomic basis set and number of active orbitals do not affect the scaling with respect to N. On the other hand, the two-dimensional carbon nitride series (Figure 2.6) scales more favorably, with an exponent ranging from 1.8-1.9 observed, indicating that the geometry of a series of molecules does affect the scaling.

2.4.3 Direct CI Performance

When small molecules are treated with large active spaces, the direct CI step, the cost of which scales factorially with the number of active orbitals, limits performance. We have performed benchmark calculations for active spaces ranging from $10^5 - 10^8$ configurations. The times to complete a single $\boldsymbol{\sigma}$ formation are reported in Figure 2.7 and Table 2.3. Note that we report two different times in Table 2.3: the $\boldsymbol{\sigma}$ formation time and the total time per $\boldsymbol{\sigma}$ formation. The former is the time spent computing the $\boldsymbol{\sigma}$ vector itself, while the latter is the total time needed to perform the direct CI calculation divided by the total number of $\boldsymbol{\sigma}$ formations performed. Thus, the total time per $\boldsymbol{\sigma}$ formation includes overhead which is excluded from the $\boldsymbol{\sigma}$ formation time. All calculations were performed on ethylene (C_2H_4), but note that CI performance is independent of the total number of basis functions and electrons. It depends only on the number of active electrons and active orbitals, and therefore equivalent CI performance is observed for systems of any size.

The largest active space included in this study, (16,16), requires less than one minute per $\boldsymbol{\sigma}$ formation. Note that the computational cost scales approximately linearly with the size of the configuration space; fitting $\log_{10}$(time) to $\log_{10}$(number of configurations) yields a scaling exponent of 1.1. The present implementation uses only a single GPU device, and the maximum configuration space is limited by the available device memory (12 GB on the

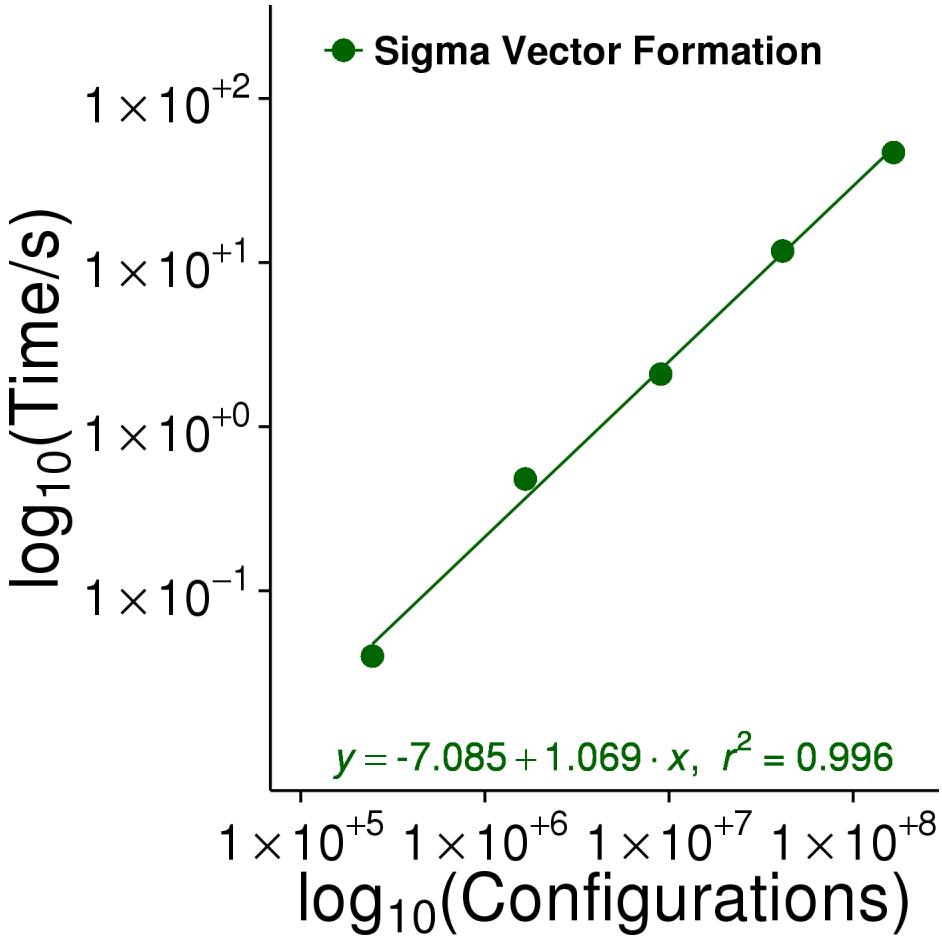

Figure 2.7: Time per $\boldsymbol{\sigma}$ formation as a function of the number of determinants for active spaces ranging from (16,12) to (16,16). All calculations were performed on a single NVidia K40 GPU.

Table 2.3: Time per $\boldsymbol{\sigma}$ formation and average iteration time for active spaces ranging from (16,12) to (16,16). All calculations were performed on a single NVidia K40 GPU.

Active Space	**# of Configurations**	**$\boldsymbol{\sigma}$ Formation (s)**	**Total Time Per $\boldsymbol{\sigma}$ Formation (s)**
(16,12)	245025	0.04	0.08
(16,13)	1656369	0.48	0.74
(16,14)	9018009	2.09	3.55
(16,15)	41409225	11.77	18.25
(16,16)	165636900	46.89	70.82

Table 2.4: Times to perform different steps in $\boldsymbol{\sigma}$ vector formation for a (16,16) active space.

Step	**Time (s)**
One-electron Contributions to $\boldsymbol{\sigma}$	6.43
α Diagonal **D**	0.12
β Diagonal **D**	0.19
α Off-Diagonal **D**	1.50
β Off-Diagonal **D**	5.30
E Formation (Matrix Multiply)	21.06
α Diagonal $\boldsymbol{\sigma}$	0.08
β Diagonal $\boldsymbol{\sigma}$	0.13
α Off-Diagonal $\boldsymbol{\sigma}$	1.74
β Off-Diagonal $\boldsymbol{\sigma}$	5.21
Total Time (GPU Time)	41.76
Total Time (CPU + GPU Time)	46.89

NVIDIA K40) rather than the time required for the calculation. We expect to be able to expand to (18,18) and possibly (20,20) active spaces by extending our implementation to take advantage of the direct inter-GPU communication and distributed GPU memory addressing features available on the most recent devices.

Note that the total time per $\boldsymbol{\sigma}$ vector formation for the (16,16) active space is 70.82 seconds, and only 46.89 seconds is attributable to the formation of $\boldsymbol{\sigma}$ itself. The remaining time (approximately 34% of the average iteration time) corresponds to overhead associated with other low-scaling steps in the Davidson algorithm, memory allocation and access, CPU-GPU communication, and other housekeeping tasks such as stack frame setup and context changes between the CPU and GPU. This again underlines the fact that for all but the simplest algorithms achieving optimal GPU performance requires optimization of several steps, and these steps often include operations that would not be performance-limiting in a serial implementation. The performance is often limited not by the step with the highest operation count, but by the step with the highest operation count that is not (or cannot be) effectively implemented on the GPU.

In Table 2.4 we present timings for each individual step of a single $\boldsymbol{\sigma}$ vector formation for a (16,16) active space. The central matrix-matrix multiplication operation consumes

Table 2.5: Absolute ground state energies of several systems computed at the HF-CASCI/6-31G** level of theory with various active spaces.

System	Active Space	TeraChem Energy (E_h)	Molpro Energy (E_h)	Δ (E_h)
Ethylene	(16,10)	-78.063304843314	-78.063304824538	1.9×10^{-8}
	(16,11)	-78.066690699464	-78.066690639571	6.0×10^{-8}
	(16,12)	-78.074505748149	-78.074505683619	6.5×10^{-8}
	(16,13)	-78.083428418532	-78.083428366856	5.2×10^{-8}
	(16,14)	-78.091109138010	-78.091109107138	3.1×10^{-8}
	(16,15)	-78.100302547285	-78.100302524072	2.3×10^{-8}
Pyrazine	(6,6)	-262.726271748762	-262.726271553812	1.9×10^{-7}
	(12,12)	-262.731145249735	-262.731145075701	1.7×10^{-7}
Dimelamine	(6,6)	-831.559377872566	-831.559377131972	7.4×10^{-7}
	(12,12)	-831.571307352380	-831.571306630668	7.2×10^{-7}
Dimelem	(6,6)	-1494.410755484148	-1494.410754288612	1.1×10^{-6}
	(12,12)	-1494.425909386718	-1494.425908295103	1.1×10^{-6}

approximately 45% of the overall computational effort for a single $\boldsymbol{\sigma}$ vector formation. The calculations of the off-diagonal terms in $\mathbf{D}$, $\boldsymbol{\sigma}$, and the one-electron part requires most of the remainder of the time, while the cost of the diagonal contributions is almost negligible. Within the off-diagonal elements, note that the β contributions, which require costly non-coalesced memory accesses, require more effort than the α contributions by a factor of 3.2. This underscores the necessity of careful management of the memory access pattern.

2.4.4 Accuracy

As noted above, the CASCI procedure is performed in double precision in this work. Additionally, no approximations are made to the Hamiltonian matrix to improve scaling. Here we demonstrate the accuracy of our implementation by comparison to a code of known quality. Single point energies for a survey of several systems and active spaces calculated on the CPU using Molpro and on the GPU using TeraChem are shown in Table 2.5. Calculated energies are in agreement to within a maximum error of $1.1 \times 10^{-6} E_h$. Note that for the largest system (dimelem) the GPU and CPU results agree to 9 decimal places, more than would be

available if the calculations were performed entirely in single precision.

2.5 Conclusions

We have presented an implementation of FCI/CASCI on GPU hardware. Even when a 1.7 x 1.4 x 1.4 nm silicon nanoparticle is calculated with a large (16,16) active space and a polarized, all-electron basis set, the cost of a CASCI calculation is only 39 minutes on a single GPU processor. This capability opens up many interesting scientific possibilities—e.g. the study of the photochemistry of large biological chromophores in complex environments or photocatalytic reactions on the surfaces of semiconducting nanoparticles, treating the entire system via a multireference level of theory capable of describing bond breaking, multiply-excited states, and conical intersections. This algorithm performs well in the limits of both large active space and large single-electron basis, but memory constrains our present implementation to configuration spaces on the order of 10^8 determinants, which corresponds to a (16,16) active space.

It is reasonable to ask whether a (16,16) or smaller active space is capable of describing chemistry at the nanoscale. Of course the answer to this question depends on the problem of interest, but there is reason for optimism in many cases. For many problems where a multireference description would be beneficial, including those described in the previous paragraph, the region where a multireference description is necessary is no larger than a typical molecule. Thus, an active space comparable to those employed in the study of molecular chemistry would likely be sufficient. Even when an electronic excitation is delocalized over more than a nanometer, it is not necessarily true that more active orbitals are required for an adequate description of the electronic structure. For example, when the low-lying electronic excitations of semiconductor nanoparticles and conjugated polymers are computed via a single reference level of theory such as time-dependent density functional theory, the lowest excited state can often be described in terms of a single natural transition orbital pair, indicating that a very small active space (possibly even (2,2)) would be sufficient [107, 12, 121].

Processes which require a larger active space than this could be approached by pairing methods which restrict the configuration space[128, 119, 66, 101, 106, 131] with carefully defined orbital determination schemes to reduce the factorial scaling of the problem.

The present FCI implementation can trivially be merged with the recently reported CASSCF orbital optimization algorithm of Hohenstein, et al. [60], which makes use of the sparsity of the AO basis in order to decrease the scaling of the full CASSCF procedure. Because the orbital gradients are computed directly from integrals in the AO basis, CASSCF does not require transformed integrals beyond those required by CASCI to be computed and stored. This combined CASSCF algorithm will allow calculations of nanoscale systems with large (16,16) active spaces to be performed in minutes or hours on a single GPU. Further acceleration would be possible by employing lower-scaling approximations to the electron repulsion integrals such as density fitting.

Our GPU FCI code also demonstrates two key points regarding the development of scientific code on GPUs. First, it is not sufficient to simply accelerate the highest scaling step or steps in a given algorithm and implement the remaining code in serial. GPU acceleration can reduce the cost of a high-scaling portion of an algorithm by two orders of magnitude, leaving lower-scaling steps as performance limiting. In our case it was necessary to accelerate all five steps in Algorithm 1 in order to achieve high performance. Second, many brilliant computational algorithms were developed for the vector processors of the 1980s, and it is fruitful to revisit them when designing quantum chemical algorithms for the GPU.

Tables of benchmark timings for series of silicon and graphitic carbon nitride clusters and geometries for all structures in Cartesian coordinates are included in the Supporting Information.

2.6 Acknowledgements

The authors thank Warren Beck, Ed Hohenstein, Todd Martínez, and Piotr Piecuch for helpful discussion. We are grateful to Paul Reed for technical assistance. Calculations in

this work were performed, in part, using computer time obtained via NSF XSEDE allocation CHE-140101. We gratefully acknowledge startup funding from Michigan State University and a generous donation of hardware from NVidia.

CHAPTER 3

GPU ACCELERATED REDUCED DENSITY MATRIX FORMATION

Reproduced in part with permission from "Complete Active Space Configuration Interaction from State-Averaged Configuration Interaction Singles Natural Orbitals: Analytic First Derivatives and Derivative Coupling Vectors", B.S. Fales, Y. Shu, B.G. Levine, and E.G. Hohenstein, *J. Chem. Phys., submitted*, 2017.

3.1 Abstract

Analytic nuclear energy gradients, nonadiabatic coupling vectors, and various one-electron properties for configuration interaction (CI) based methods rely on availability of the 1- and 2-particle reduced density matrices (1- and 2-RDM, respectively). Improved direct CI algorithms taking advantage of graphical processing unit (GPU) hardware acceleration[40] have increased the size of routinely performed CI calculations. To match the high performance of our direct CI algorithms we have developed and implemented GPU accelerated 1- and 2-RDM algorithms, the latter exhibiting similar scaling as our direct CI $\boldsymbol{\sigma}$ formation algorithm. We present details of our implementation as well as timing and scaling information for both 1- and 2-RDM formation.

3.2 Introduction

Computational chemists routinely leverage analytical nuclear energy gradients and nonadiabatic coupling vectors to great effect, allowing for efficient exploration of highly-dimensional chemical spaces in the form of geometry optimizations and transition state searches[151], ab initio molecular dynamics (AIMD) simulations[4, 138], minimum energy conical intersection (MECI) geometry optimizations[14], and ab initio multiple spawning (AIMS) simulations[16, 89, 32]. Derivation and implementation of analytical nuclear energy gradients and nonadiabatic coupling vectors for electronic structure methods where the Hellman—Feynman

theorem does not apply is simplified using Lagrangian based formalisms. In this context, generalized reduced density matrices are of fundamental importance.

Reduced density matrices provide a concise means of storing wavefunction information in terms of electron density. The CI energy for state n can be represented according to

$$E_{CI}^{(n)} = \sum_{pq} (p|\hat{h}|q)\gamma_{pq}^{(n)} + \sum_{pqrs} (pq|rs)\Gamma_{pqrs}^{(n)} \tag{3.1}$$

where p, q, r, s index molecular orbitals, $\boldsymbol{\gamma}$ is the one-particle reduced density matrix (1-RDM), and $\boldsymbol{\Gamma}$ is the two-particle reduced density matrix (2-RDM). The 1- and 2-RDMs are defined according to

$$\gamma_{pq}^{(n)} = \sum_{IJ} c_I^{(n)} c_J^{(n)} \langle \Phi_I | \hat{E}_{pq} | \Phi_J \rangle \tag{3.2}$$

$$\Gamma_{pqrs}^{(n)} = \frac{1}{2} \sum_{IJ} c_I^{(n)} c_J^{(n)} \langle \Phi_I | \hat{E}_{pq}\hat{E}_{rs} - \delta_{qr}\hat{E}_{ps} | \Phi_J \rangle, \tag{3.3}$$

where I and J index determinants, $\mathbf{c}$ is the CI vector, and $\hat{E}_{pq}$ is an excitation generator corresponding to orbitals p and q. RDMs as shown are single-state RDMs, and generalization to transition RDMs is trivial.

The 1-RDM is required for the calculation of various one-electron properties, such as the dipole moment and the transition dipole moment. Methods where the orbital and CI vector coefficient optimizations are coupled, such as the state averaged complete active space self-consistent field (SA-CASSCF) method, use the 1-RDM when calculating the orbital gradient during the orbital optimization step. Lagrangian based implementations of analytical nuclear energy gradients and nonadiabatic coupling vectors rely on solution of the coupled perturbed Hartree-Fock (CPHF) and coupled perturbed configuration interaction (CPCI) equations, respectively, each of which uses generalized single-state and transition 1- and 2-RDMs.

3.3 Computational Details

Graphical processing unit (GPU) accelerated 1-RDM and 2-RDM algorithms have been implemented in the TeraChem software package. Our SA-CASSCF energy implementation[60], as well as analytic energy gradients and nonadiabatic coupling vectors for both the SA-CASSCF[167, 166] and CASCI methods (including the configuration interaction singles natural orbitals [160, 41], or CISNO, and the floating occupation molecular orbitals [50, 165, 58, 57], or FOMO, methods) requires fast calculation of both the 1-RDM and 2-RDM. When writing high-performance GPU-accelerated code, low-scaling tasks often become bottlenecks if not implemented efficiently. This is true for 1-RDM formation in the present case, particularly when large configuration spaces are used in the context of SA-CASSCF. As such, we describe the efficient computation of both the 1- and 2-RDMs on GPU hardware.

3.3.1 1-RDM

Defining an intermediate matrix, $\mathbf{D}$, the 1-RDM (defined in Equation 3.2) can be computed:

$$D_{pq}^{J} = \sum_{I} \gamma_{pq}^{IJ} c_{I}^{(n)} \tag{3.4a}$$

$$\gamma_{pq}^{(n)} = \sum_{J} c_{J}^{(n)} D_{pq}^{J} \tag{3.4b}$$

where $\gamma_{pq}^{(n)}$ are the elements of the 1-RDM and γ_{pq}^{IJ} are the one-electron coupling coefficients. Note that the $\mathbf{D}$ matrix is identical to that used in the direct full CI algorithm of Knowles and Handy[73]. Our implementation takes advantage of the GPU-accelerated implementation of $\mathbf{D}$ matrix formation described in Chapter 2 [40]. The 1-RDM is then formed according to Equation 3.4b using the highly optimized matrix-vector multiplication library function provided by NVidia's basic linear algebra subroutine library, CUBLAS[124].

3.3.2 2-RDM

The calculation of the 2-RDM is the computational bottleneck of the gradient calculation when large configuration spaces are used. In the same spirit as for the 1-RDM, the 2-RDM (defined in Equation 3.3) can be computed from the intermediate quantity $\mathbf{D}$ and its transpose $\mathbf{E}$ (defined such that $D^I_{rs} = E^I_{sr}$).[164]

$$\Gamma^{(n)}_{pqrs} = \frac{1}{2}\sum_I E^I_{qp} D^I_{rs} - \frac{1}{2}\delta_{qr}\sum_J c^{(n)}_J D^J_{ps} \tag{3.5}$$

Formally, $\mathbf{D}$ and $\mathbf{E}$ contain the same information, but algorithmically, both are required in order to form the final 2-RDM using the standard CUBLAS matrix-matrix multiplication routine, so we differentiate them here. Algorithm 3 presents the vectorized approach we used to build the 2-RDM. Note that here α and β index the α and β strings representing the occupations of a given determinant. Each element of the $\mathbf{l}$ matrix contains the orbital address defining the nonzero excitation operator matrix element between a particular bra-ket pair in determinant space; i.e. if γ^{IJ}_{pq} is nonzero then $l_{IJ} = pq$. To improve the efficiency of our implementation we have introduced an analog to the $\mathbf{l}$ matrix, $\mathbf{l'}$, which differs only in that the orbital indices are transposed ($l'_{IJ} = qp$ when γ^{IJ}_{pq} is nonzero).

Algorithm 3 shares common elements with the $\boldsymbol{\sigma}$ vector formation algorithm used in our direct CI program[40]. It is important to highlight differences between the two procedures:

- $\boldsymbol{\sigma}$ vector formation requires use of atomic operations in the scatter operation (write to the $\boldsymbol{\sigma}$ vector), while 2-RDM matrix formation requires no blocking operations.
- The size of the matrix-matrix multiplication in $\boldsymbol{\sigma}$ vector formation is smaller than that required for building the 2-RDM.
- An additional operation, a matrix-vector multiplication, is needed when constructing the 2-RDM.

Algorithm 3 Vectorized 2-RDM algorithm

GPU vectorize over strings α, β
  for all orbitals r occupied in α **do**
    $\mathbf{D}[rr, \alpha, \beta]$ += $\mathbf{c^{(n)}}[\alpha, \beta]$
  end for
end GPU vectorize
GPU vectorize over strings α, α', β, where α and α' differ by exactly one occupation
  $rs \leftarrow \mathbf{l}[\alpha, \alpha']$
  $\mathbf{D}[rs, \alpha, \beta]$ += $\boldsymbol{\gamma}[\alpha, \alpha']\mathbf{c^{(n)}}[\alpha', \beta]$
end GPU vectorize
GPU vectorize over strings α, β
  for all orbitals r occupied in β **do**
    $\mathbf{D}[rr, \alpha, \beta]$ += $\mathbf{c^{(n)}}[\alpha, \beta]$
  end for
end GPU vectorize
GPU vectorize over strings β, β', α, where β and β' differ by exactly one occupation
  $rs \leftarrow \mathbf{l}[\beta, \beta']$
  $\mathbf{D}[rs, \alpha, \beta]$ += $\boldsymbol{\gamma}[\beta, \beta']\mathbf{c^{(n)}}[\alpha, \beta']$
end GPU vectorize
GPU vectorize over strings α, β
  for all orbitals p occupied in α **do**
    $\mathbf{E}[pp, \alpha, \beta]$ += $\mathbf{c^{(n)}}[\alpha, \beta]$
  end for
end GPU vectorize
GPU vectorize over strings α, α', β, where α and α' differ by exactly one occupation
  $qp \leftarrow \mathbf{l'}[\alpha, \alpha']$
  $\mathbf{E}[qp, \alpha, \beta]$ += $\boldsymbol{\gamma}[\alpha, \alpha']\mathbf{c^{(n)}}[\alpha', \beta]$
end GPU vectorize
GPU vectorize over strings α, β
  for all orbitals p occupied in β **do**
    $\mathbf{E}[pp, \alpha, \beta]$ += $\mathbf{c^{(n)}}[\alpha, \beta]$
  end for
end GPU vectorize
GPU vectorize over strings β, β', α, where β and β' differ by exactly one occupation
  $qp \leftarrow \mathbf{l'}[\beta, \beta']$
  $\mathbf{E}[qp, \alpha, \beta]$ += $\boldsymbol{\gamma}[\beta, \beta']\mathbf{c^{(n)}}[\alpha, \beta']$
end GPU vectorize
matrix-matrix multiply: $\boldsymbol{\Gamma}[pq, rs] \leftarrow \frac{1}{2}\mathbf{E}[qp, \alpha, \beta]\mathbf{D}[rs, \alpha, \beta]$
matrix-vector multiply: $\boldsymbol{\gamma^{(n)}}[pq] \leftarrow \mathbf{D}[pq, \alpha, \beta]\mathbf{c^{(n)}}[\alpha, \beta]$
GPU vectorize over orbitals p, q, s
  $\boldsymbol{\Gamma}[pq, qs]$ += $-\frac{1}{2}\boldsymbol{\gamma^{(n)}}[ps]$
end GPU vectorize

- Our implementation is structured in such a way as to require construction of the α and β string lists for each 2-RDM formation, while $\boldsymbol{\sigma}$ vector formation has access to pre-built lists in memory.

3.3.3 Benchmark Calculations Details

The 1- and 2-RDM calculations described above were implemented for NVidia GPU hardware in a development version of the TeraChem software package using the Compute Unified Device Architecture (CUDA) API[125]. Herein we report benchmark calculations for 2-RDM formation for various active spaces. These calculations were performed on a single NVidia K40 GPU. Those portions of the code that run on the CPU were performed on a single core of an Intel E5603 1.60 GHz processor. All calculations are performed in double precision. For comparison, the time needed for a single formation of the $\boldsymbol{\sigma}$ vector for each active space using GPU-accelerated code[40] is provided as a reference. Formation of the $\boldsymbol{\sigma} = \mathbf{Hc}$ vector is the rate limiting step in solution for the CI energy, and has the same formal computational scaling as 2-RDM formation.

3.3.4 2-RDM Benchmark Calculations

Figure 3.1 presents the performance of the 2-RDM algorithm for several active spaces, each containing 12 electrons and a number of orbitals varying from 12 to 15. The scaling of the computational cost with the number of configurations is essentially linear, with a scaling exponent of 1.054. Table 3.1 reports the time for various steps in 2-RDM formation for the (12,15) active space. In addition to the 2-RDM data, we also provide $\boldsymbol{\sigma}$ vector formation timings for the same active space for comparison.

As seen in Table 3.1, the floating point operations ($\mathbf{D}/\mathbf{E}$ matrix formation, matrix-matrix multiplication, and matrix-vector multiplication) dominate the cost of 2-RDM formation. In the case of the (12,15) active space they require 75% of the total time. At the same time, we note that the system overhead (one-electron coupling coefficient formation, memory

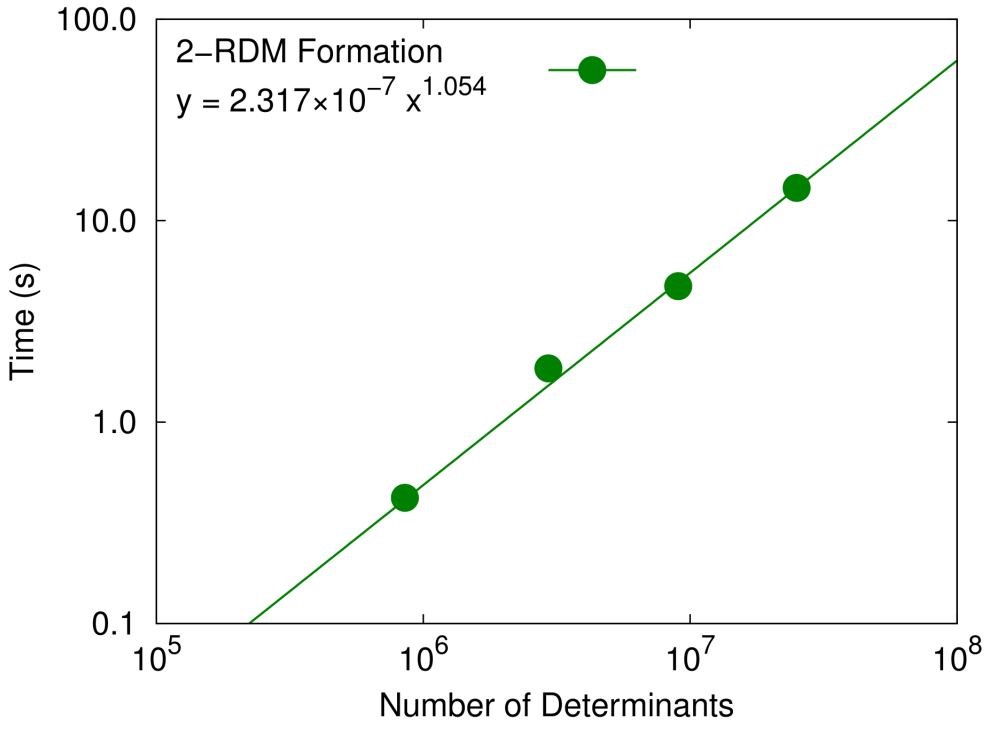

Figure 3.1: Time for 2-RDM formation as a function of the number of determinants for active spaces ranging from (12,12) to (12,15). All calculations were performed using a single NVidia K40 GPU.

allocation, and data transfer) times are non-negligible, suggesting that our algorithm takes advantage of both the high data transfer rates and the massively parallel processing power of the GPU hardware in a balanced fashion.

3.3.5 Conclusions

We have implemented fast GPU accelerated 1- and 2-RDM formation algorithms for use in conjunction with the SA-CASSCF and several CASCI programs in the TeraChem software package. Near-linear scaling with respect to configuration space size is achieved for 2-RDM formation, similar to that observed for $\boldsymbol{\sigma}$ formation in our direct CI program. Future plans include continued application of recursion relations in directly computing higher-order RDMs to be used with internally contracted multireference CI.

Table 3.1: Times to perform different steps in 2-RDM formation for a (12,15) active space. For comparison we also show $\boldsymbol{\sigma}$ vector formation time. All calculations were performed using a single NVidia K40 GPU.

Step	**Time (s)**
1-electron coupling coefficient formation	1.54
memory allocation/transfer	2.13
D/**E** matrix formation	1.93
ED multiplication	7.99
Dc multiplication	0.94
summations of $-\frac{1}{2}\boldsymbol{\gamma}$	<0.01
Total 2-RDM formation	14.53
$\boldsymbol{\sigma}$ vector formation	8.08

CHAPTER 4

ROBUST AND EFFICIENT SPIN PURIFICATION FOR DETERMINANTAL CONFIGURATION INTERACTION

4.1 Abstract

The limited precision of floating point arithmetic can lead to the qualitative and even catastrophic failure of quantum chemical algorithms, especially when high accuracy solutions are sought. For example, numerical errors accumulated while solving for determinantal configuration interaction wavefunctions via Davidson diagonalization may lead to spin contamination in the trial subspace. This spin contamination may cause the procedure to converge to roots with undesired $\langle \hat{S}^2 \rangle$ eigenvalues, wasting computer time in the best case and leading to incorrect conclusions in the worst. To remedy this, we present four purification schemes for ensuring that the eigenvectors have the desired $\langle \hat{S}^2 \rangle$. These schemes are based on projection, penalty, and iterative approaches. All of these schemes rely on a direct, graphics processing unit (GPU-) accelerated algorithm for calculating the $\mathbf{S^2c}$ matrix-vector product. We assess the computational cost and convergence behavior of these methods by application to several benchmark systems. Finally, to demonstrate the utility of these approaches we have computed the lowest several excited states of an open-shell silver cluster (Ag_{19}) using the state-averaged complete active space self-consistent field (SA-CASSCF) method where spin purification was required to ensure spin stability of the CI vector coefficients. Several low-lying states with significant multiply-excited character are predicted, suggesting the value of a multireference approach for modeling plasmonic nanomaterials.

4.2 Introduction

Configuration interaction (CI) is a conceptually straightforward method used for describing electronic correlation in molecules. Common variants of CI include full CI, truncated

CI (e.g. CI singles and doubles, CISD), and active space CI (e.g. restricted, complete, or generalized active space[101], or RAS-CI, CASCI, and GAS-CI). The desired eigenvalue-eigenvector pairs are usually obtained using iterative methods such as the Davidson-Liu approach[33, 93] to avoid the $\mathcal{O}(N^3)$ scaling of complete Hamiltonian diagonalization, where N is the number of basis functions in the configuration space. Modern CI implementations often use a determinantal basis due to the convenient integer form of the 1-particle coupling coefficients. [53, 73, 128, 203, 158] However, the use of determinants comes at a two-fold cost: first, the number of determinants is larger than the number of configuration state functions (CSFs), and second, determinants are not necessarily eigenfunctions of the $\hat{S^2}$ operator. The latter difficulty is ameliorated through use of guess vectors that are eigenfunctions of $\hat{S^2}$. Still, spin contamination can occur during diagonalization, resulting in erroneous convergence to roots with undesired $\langle \hat{S^2} \rangle$. This contamination may arise by two mechanisms. In the first case, failure to use a spin-averaged preconditioner can result in severe spin contamination. This problem is effectively resolved through use of preconditioners based on averaged[87] or maximum[73] exchange integrals within a spin-coupling set or orbital energy differences[39]. A second issue can arise, however, when the wave function must be converged to a tight tolerance, e.g. when the orbital gradient or analytical energy gradient are desired. In this situation, numerical instability can result in unphysical mixing between states having different spin symmetries, ultimately causing convergence to eigenvectors having the incorrect spin. Recent work has investigated numerical instability in several quantum chemistry algorithms using a statistical approach, demonstrating deficiencies in several commonly used methods[71]. It is well known that the associative law of addition does not hold for floating point algebra, $(a + b) + c \neq a + (b + c)$, and variations in rounding can lead to numerical instability. This instability is compounded in parallel implementations, as the floating point operations may occur in different orders from run to run, potentially giving non-deterministic solutions as higher accuracy is sought. Even widely used multi-threaded mathematical libraries cannot guarantee determinism. We wish to em-

phasize that the spin stability we discuss in this work is numerical in nature, and is distinct from spin instability of the Hartree-Fock wave function that arises in cases of form (near-)degeneracy.[176, 100, 79, 182, 46, 129, 197, 156, 112, 27, 70, 17, 68, 172, 202, 123, 56, 196]. (See Supporting Information for an illustrative example of how spin contamination arises in a real system.)

Algorithm 4 Davidson-Liu iterative method for the lowest few eigenvectors and eigenvalues of real, symmetric matrices

1: Select a set of L orthonormal guess vectors, denote as $\{\mathbf{c}_i\}$
2: **while** not converged **do**
3: Form a sigma vector: $\boldsymbol{\sigma}_j = \sum_i \mathrm{H}_{ij}\mathbf{c}_i$
4: Solve the $L \times L$ eigenvalue problem $\mathbf{G}\boldsymbol{\alpha}^k = \rho^k\boldsymbol{\alpha}^k$ where $\mathrm{G}_{ij} = \langle \mathbf{c}_i, \boldsymbol{\sigma}_j \rangle$
5: Form the residual vector $\mathbf{r}^k$ for the k^{th} root according to: $\mathbf{r}^k = \sum_{i=1}^{L} \boldsymbol{\alpha}_i^k \left(\boldsymbol{\sigma}_i - \rho^k\mathbf{c}_i\right)$
6: Test $||\mathbf{r}^k|| < convTol$
7: Form the correction vector $\boldsymbol{\delta}^k$ by: $\boldsymbol{\delta}_I^k = -\left(\mathrm{H}_{II} - \rho^k\right)^{-1}\mathbf{r}_I^k$
8: Normalize $\boldsymbol{\delta}^k$
9: Orthogonalize and append the correction vectors $\boldsymbol{\delta}^k$ to the set of trial vectors $\{\mathbf{c}_i\}$
10: **end while**

To facilitate a clear discussion we present the Davidson-Liu method (Algorithm 4) using notation partially derived from Leininger *et al.*[87], neglecting details related to subspace collapse for conciseness. If we begin with a guess vector that is an eigenfunction of $\hat{S^2}$ (line 1 of Algorithm 4) and utilize a spin-averaged preconditioner (line 7 of Algorithm 4), spin contamination can only be introduced to the subspace through the addition of trial vectors that are not spin eigenfunctions (line 9 of Algorithm 4). This is largely the result of numerical errors accumulated in lines 3 through 5.

Currently, these issues can be at least partially solved in a number of ways. For systems with equal numbers of α and β electrons, it is possible to take advantage of the symmetric or antisymmetric nature of the CI vector to eliminate one type of spin contamination, namely that which occurs between singlets and triplets, through explicit (anti)symmetrization of the trial vectors as they are added to the space. Unfortunately, higher angular momentum spin contamination can still occur, such as quintet contamination of a singlet state, when employing such a simple scheme. Furthermore, odd-electron systems lack this type of symmetry

altogether, and a more sophisticated means of ensuring spin purity of the wave function becomes necessary.

Another approach that solves these issues was suggested by Knowles and Werner [189]. Instead of solving the full dimensional eigenvalue problem, they partition the CI space into primary and secondary spaces based on the diagonal Hamiltonian matrix elements. The primary space problem is solved exactly, and the secondary space is solved perturbatively. Since the primary space problem is solved through explicit diagonalization of a Hamiltonian, the subspace eigenvectors will be spin eigenfunctions, and any contamination arising in the secondary space is easily removed using spin projection methods (such as the approach described by Löwdin [94]) after convergence is reached. An unfortunate side effect of the separation of the CI space is that it can result in certain roots not being located by the eigensolver, especially when several (> 10) roots are sought in large configuration spaces.

In this work we will describe four related approaches for eliminating spin contamination during the iterative diagonalization process. In Section 4.3 we describe these methods as well as a graphics processing unit (GPU-) accelerated direct algorithm for computing the $\mathbf{S^2c}$ matrix-vector product that makes each of the methods tractable. This is followed by an analysis of convergence behavior of and benchmark calculations for each of the purification schemes in Section 4.4. Finally, we demonstrate the utility of our approach in Section 4.5 by computing the excited states of a Ag_{19} cluster, an open-shell system where spin purification is required to efficiently obtain wave functions having the desired spin.

4.3 Methods

In this section we will present the proposed spin purification schemes. Assuming atomic units, the total spin operator is defined as

$$\hat{S}^2 = \hat{S}_z(\hat{S}_z - 1) + \hat{S}_+\hat{S}_- \tag{4.1}$$

where $\hat{S}_z$ is the spin projection, and $\hat{S}_+$ and $\hat{S}_-$ are the raising and lowering operators,

defined as

$$\hat{S}_+ = \sum_p a^\dagger_{p\alpha} a_{p\beta} \tag{4.2a}$$

$$\hat{S}_- = \sum_p a^\dagger_{p\beta} a_{p\alpha} \tag{4.2b}$$

where $a^\dagger_{p\theta}$ and $a_{p\theta}$ are the creation and annihilation operators for an electron of spin θ in spatial orbital p. The matrix elements of the total spin matrix, $\mathbf{S^2}$, in a determinantal basis are

$$S^2_{IJ} = \left\{ \begin{array}{cc} \langle I|\frac{1}{4}\left[(n_\alpha - n_\beta)^2 + 2(n_\alpha + n_\beta)\right]|J\rangle, & I = J \\ \langle I|sgn\hat{S}_+\hat{S}_-|J\rangle, & I \neq J \end{array} \right\} \tag{4.3}$$

where I and J index determinants, n_α and n_β are the numbers of unpaired α and β electrons, and sgn is -1 raised to the power of the number of permutations needed to bring the singly occupied orbitals in configurations I and J into maximum coincidence.

4.3.1 Spin Penalty

The first approach we describe for purifying the eigenvector is inspired by the folded spectrum method[104], which has been applied to quantum chemical problems to facilitate solving for roots far from the extrema[187, 188]. Here we modify the Hamiltonian matrix by addition of a "folded" $\mathbf{S^2}$ matrix that energetically penalizes those eigenvectors which do not have the desired (reference) spin, $\langle\hat{S}^2\rangle_{target}$

$$\mathbf{H'} = \mathbf{H} + \alpha(\mathbf{S^2} - \mathbf{I}\langle\hat{S}^2\rangle_{target})^2 \tag{4.4}$$

The variable α is a user-chosen positive scalar multiplier that defines the magnitude of the penalty. Because $\mathbf{H}$ and $\mathbf{S^2}$ commute, the eigenvectors of the Hamiltonian remain intact, but the energies of states whose spins are not equal to $\langle\hat{S}^2\rangle_{target}$ are penalized. By energetically

separating the states with undesired spin from those with desired spin, spin contamination can be avoided.

Since a total of two $\mathbf{S^2c}$ matrix-vector products must be formed for every $\boldsymbol{\sigma'}$ vector formation ($\boldsymbol{\sigma'} = \mathbf{H'c}$), and a typical calculation requires tens to hundreds of $\boldsymbol{\sigma}$ formations ($\boldsymbol{\sigma} = \mathbf{Hc}$), to make this approach viable we require an $\mathbf{S^2c}$ implementation that is fast relative to $\boldsymbol{\sigma}$ vector formation. Furthermore, for large configuration spaces storage of the $\mathbf{S^2}$ matrix becomes prohibitive. A direct, GPU-accelerated algorithm for addressing these difficulties will be presented in subsection 4.3.5.

One might consider modifying the preconditioner to be consistent with the penalized Hamiltonian. We tested such a modification and found that, in practice, it slowed convergence. Results and discussion on this topic are presented in Supporting Information. Below we use the unmodified preconditioners with the penalized Hamiltonian.

4.3.2 Löwdin Projection

Perhaps the most obvious approach to annihilating spin contamination during determinantal CI calculations is to use the Löwdin spin projection operator[94]. Löwdin's operator is defined

$$^{2l+1}\mathcal{O} = \prod_{k}^{k\neq l} \frac{\hat{S}^2 - k(k+1)}{l(l+1) - k(k+1)} \tag{4.5}$$

where $l = n, n-1, n-2, ..., 0$ or $\frac{1}{2}$ (depending on whether $2n$ is even or odd) and k is bounded by the minimal and maximal values of the resulting spin (k is the spin of all states in the manifold not equal to the target, l is the spin of the target state).

Here, we monitor the value of $\langle \hat{S}^2 \rangle$ for each root at each iteration of the iterative diagonalization. Should a root deviate from the target spin value by a user defined threshold, the Löwdin operator is applied. In our tests we have determined that a deviation threshold of 1.0×10^{-10} from the target $\langle \hat{S}^2 \rangle$ value produces reasonable results in most cases, given the wavefunction convergence criteria used in this work. Though formally a single application to

a given vector should be sufficient to annihilate all spin contamination, in practice we find that some spin contamination often remains. Thus, we retest for spin contamination after application of the projector to a given vector. If the deviation of $\langle \hat{S}^2 \rangle$ remains above the threshold, we reapply as necessary. There are several possible places in the Davidson-Liu algorithm in which the projection can be applied; we opt to apply it after the residual is computed.

The rationale for choice of the Löwdin projection is that it can be efficiently implemented using the fast, direct $\mathbf{S^2c}$ code described in subsection 4.3.5. An alternative spin projector developed by Pratt[137] was shown to be equivalent to Löwdin's by Berencz[19]. An interesting alternative is the novel spin projection operator developed by Jiménez-Hoyos, et al. for use in spin-projected Hartree-Fock. Closely related to the methods described by Percus and Rotenberg[135] and by Lefebvre and Prat[84, 85], an advantage of these projectors is the formal reduction in scaling from that of a 2-particle operator to that of a 1-particle operator. However, in the present context this form of spin projection operator has two potential disadvantages. 1) It requires numerical integration, which adds an additional level of complexity to the spin projection process, and 2) it requires consideration of spin-flip excitations and spin operators that are not standard data structures constructed when performing direct CI. Further exploration of the use of this spin operator would be very interesting, but is outside the scope of the present work.

4.3.3 First-Order Spin Projection

A clear disadvantage of Löwdin projection is that the projector is a high-order polynomial of $\mathbf{S^2}$, and thus is computationally expensive in many cases. Here we define a first-order projection scheme which may be capable of annihilating spin contamination at a lower cost. In this method, we define a vector, $\boldsymbol{\chi}$, that projects only onto the spin-contaminated components of the trial vector

$$\boldsymbol{\chi} = (\mathbf{S}^2 - \mathbf{I}\langle\hat{S}^2\rangle_{target})\mathbf{c} \tag{4.6}$$

and then orthogonalize the trial vector, $\mathbf{c}$, to $\boldsymbol{\chi}$

$$\mathbf{c}' = \mathbf{c} - \left(\frac{\mathbf{c}\cdot\boldsymbol{\chi}}{\boldsymbol{\chi}\cdot\boldsymbol{\chi}}\right)\boldsymbol{\chi} \tag{4.7}$$

where $\mathbf{c}'$ is the projected trial vector. In contrast to the penalty method, this projection approach need not be applied every iteration; it is only applied when spin contamination is detected in the wave function.

The algorithm for applying the first-order projection procedure is the same as for the Löwdin projection described in the previous subsection. It comprises two steps: 1) $\langle\hat{S}^2\rangle$ is determined at each iteration for each root, and 2) the projection scheme is applied if the observed $\langle\hat{S}^2\rangle$ differs from the target value by some threshold (typically 1.0×10^{-10}, as discussed above and in Supporting Information). We have elected to purify following formation of the residual vector. In many cases a single application of the first-order projection scheme does not produce vectors having spin purity within the defined threshold, and in these cases we iteratively apply the projection process until a spin-pure vector is obtained. (Note that for a given $\boldsymbol{\chi}$ our projection scheme is formally idempotent, but $\boldsymbol{\chi}$ changes at every application, allowing our approach to be applied repeatedly to systematically improve the spin purity of the trial vector.) In practice, spin purity is often accomplished after $3-5$ applications of the projector on each trial vector (though cases have been observed where many more applications are required). The overall computational cost of first-order projection is $(N_\sigma + N_p)$ total $\mathbf{S^2c}$ formations, where N_σ is the number of $\boldsymbol{\sigma}$ vector formations and N_p is the total number of purifications required.

One could imagine higher-order projection schemes (intermediate between first-order and Löwdin). We attempted a second-order scheme with poor results. These are discussed in Supporting Information.

4.3.4 Inverse Iteration

The final spin purification method is based on inverse iteration. When spin contamination is detected, the trial vector is modified according to

$$\mathbf{c}' = (\mathbf{S^2} - \mathbf{I}\langle\hat{S}^2\rangle_{target})^{-1}\mathbf{c} \tag{4.8}$$

In practice, the diagonal of $\mathbf{S^2}$ is shifted not by the exact $\langle\hat{S}^2\rangle_{target}$, but instead by $(\langle\hat{S}^2\rangle_{target} - 0.01)$ to avoid numerical issues when solving the linear system. Like the projection methods, inverse iteration can be used to systematically improve $\langle\hat{S}^2\rangle$ of the trial vector. In this case the additional stability comes at increased computational cost, however, as solution of the linear system must be performed at each iteration. For small configuration spaces (fewer than $\sim 10^5$ configurations), the linear system can be trivially solved after the matrix is explicitly formed. For large configuration spaces an iterative linear system solver must be employed (e.g. preconditioned conjugate gradient), the first iteration of which requires two $\mathbf{S^2c}$ formations per purification (one each for formation of the residual and the search direction vectors), while subsequent linear system solver iterations require an additional $\mathbf{S^2c}$ formation each. The resulting computational effort is $\big(2(N_p) + (N_i - N_p)\big)$ $\mathbf{S^2c}$ formations, where N_i is the total overall number of iterations required to solve the linear system. The total computational cost of inverse iteration is then $N_\sigma + \big(2(N_p) + (N_i - N_p)\big)$ $\mathbf{S^2c}$ formations. Even in the limit where the linear system solver converges in the first iteration (i.e. $N_i = N_p$), inverse iteration requires an additional N_p $\mathbf{S^2c}$ formations relative to first-order projection purification.

4.3.5 Direct $\mathbf{S^2c}$ Formation

Naive implementation of the $\mathbf{S^2c}$ product requires formation and storage of the $N \times N$ $\mathbf{S^2}$ matrix. For long CI expansions this is untenable, motivating us to implement a direct approach for the calculation of $\mathbf{S^2c}$. Other desirable attributes include vectorizability and

a reduction in algorithmic scaling by taking advantage of sparsity in the $\mathbf{S^2}$ matrix. While $\hat{S}^2$ is formally a 2-electron operator, in practice it involves the product of two 1-electron operators. We leverage this property to reduce the effective scaling without introducing any approximations. To better understand the scaling we consider the expression for off-diagonal elements, S^2_{IJ}, in Equation 4.3. These elements can be obtained without the explicit use of the spin operators $\hat{S}_+$ and $\hat{S}_-$ according to

$$S^2_{IJ} = \langle I| - \gamma^{\alpha}_{pq}\gamma^{\beta}_{qp}|J\rangle, I \neq J \tag{4.9}$$

where γ_{ij} are the 1-particle coupling coefficients. Defining the matrix elements in terms of the already available 1-particle coupling coefficients thus avoids the complicated calculation of the matrix elements of the raising and lowering operators. The $\mathbf{S^2c}$ product can be written as

$$\mathbf{S^2c}[\alpha,\beta] = \sum_{\alpha',\beta'} \mathbf{S^2}[\alpha,\beta][\alpha',\beta']\mathbf{c}[\alpha',\beta'] \tag{4.10}$$

where α, α', β, and β' are α and β strings, respectively. As written above, the formal scaling of this calculation is $\mathcal{O}(N^2)$. We improve the scaling significantly by taking advantage of the fact that $\mathbf{S^2}$ is only nonzero when α' and β' are singly excited with respect to α and β, respectively. Taking advantage of this sparsity reduces the number of α' and β' terms from $\binom{O}{E}$ to $E(O-E)$, where O is the number of orbitals and E is the number of electrons of a given spin in the configuration space.

Scaling is further reduced by defining a new data structure that eliminates the need to iterate over β'. A consequence of the $\hat{S}^2$ operator being a product of two 1-electron operators is that the compound orbital index pq corresponding to the $\alpha \rightarrow \alpha'$ excitation is the same as the qp index of the corresponding $\beta \rightarrow \beta'$ excitation. This allows us to form a matrix, $\mathbf{B}$, with the first dimension corresponding to the β index, the second dimension to the qp

pair, and matrix elements to the index, β'. As our configuration indexing scheme is positive semi-definite, we initialize the $\mathbf{B}$ array with an out-of-bounds value (-1) to indicate that the configurations are not coupled. Since the product of the raising and lowering operators results in unique determinant pairs, precalculation of the $\mathbf{B}$ matrix reduces the scaling of the problem from that of a product of two one-electron operators to that of a single one-electron operator.

Due to their low computational scaling and low storage requirements, the diagonal elements of $\mathbf{S^2}$ are evaluated once and stored in main memory for the duration of the calculation. The computation of the off-diagonal elements comprises the majority of the effort. We present the details of our GPU-accelerated algorithm for the off-diagonal contribution to $\mathbf{S^2c}$ in Algorithm 5. In this algorithm, $\mathbf{l}$ provides the $p \rightarrow q$ orbital excitation index relating configurations α and α'. The 1-particle coupling coefficient matrices $\boldsymbol{\gamma}^{\boldsymbol{\alpha}}$ and $\boldsymbol{\gamma}^{\boldsymbol{\beta}}$ are equivalent in cases where the numbers of α and β electrons are equal and unique otherwise. Recall that we initialized $\mathbf{B}$ with a negative value, giving the desired result of no contribution to the $\mathbf{S^2c}$ vector in the case where $pq^{\alpha} \neq qp^{\beta}$.

Algorithm 5 GPU-vectorized algorithm for computing off-diagonal contribution to $\mathbf{S^2c}$

```
GPU vectorize over strings α, β
    for strings α' differing from α by zero or one occupations do
        pq ← l[α, α']
        β' ← B[β, pq]
        if β' ≥ 0 then
            S²c[α, β] −= γ^α[α, α'] γ^β[β, β'] c[α', β']
        end if
    end for
end GPU vectorize
```

4.3.6 Guess Vector Formation

It is well known that spin-pure guess vectors are essential if one hopes to target specific spin states in determinantal CI calculations. Throughout this work the initial guess vectors are obtained using the straightforward procedure outlined in Algorithm 6, which guarantees that these vectors are spin eigenstates. A subspace Hamiltonian, $\mathbf{H_{00}}$ (line 2, Algorithm

6) is constructed by considering the lowest energy n determinants (reasonable defaults are between $400 - 1000$ determinants). When one determinant is included in the subspace, all members of its spin-coupling set are also included. The $\mathbf{S^2}$ matrix corresponding to the subspace is constructed and explicitly diagonalized. The block of diagonalized $\mathbf{S^2}$ eigenvectors corresponding to the target spin multiplicity are then used to transform the subspace Hamiltonian to a basis of configuration state functions (CSFs, line 5 Algorithm 6). The CSF basis Hamiltonian is then diagonalized (producing $\mathbf{D_{00}}$, Algorithm 6 line 6) and transformed back to the determinant basis (Algorithm 6 line 7). Finally, the subspace guess vector is projected onto the full CI space using the appropriate addressing scheme (Algorithm 6 line 8).

Algorithm 6 Procedure for forming spin-adapted guess vectors. $\mathbf{b_{tar}}$ are the $\mathbf{S^2}$ solution eigenvectors of the target spin, $\mathbf{d_{tar,00}}$ are the solution eigenvectors to the subspace Hamiltonian in the CSF basis comprising the matrix $\mathbf{D_{tar,00}}$, e are the CSF subspace eigenvalues, $\mathbf{c_{subspace}}$ as a matrix comprised of the subspace guess vectors, P projects from the subspace to the full CI space, and $\mathbf{c}$ is a matrix formed by the final guess vectors.

1: Calculate and order all H_{ii}. Select lowest 400-1000 spin-coupled determinants.
2: Form $\mathbf{H_{00}}$ and $\mathbf{S^2}$.
3: Solve $\mathbf{S^2 b} = s\mathbf{b}$
4: Select set $\mathbf{b_{tar}} = \sum_i \mathbf{b_i}$ according to target M_S
5: Calculate $\mathbf{H_{tar,00}} = \mathbf{b_{tar}}^\dagger \mathbf{H_{00} b_{tar}}$
6: Solve $\mathbf{H_{tar,00}}\ \mathbf{d_{tar,00}} = e\mathbf{d_{tar,00}}$
7: Calculate $\mathbf{c_{subspace}} = \mathbf{D_{tar,00}}\ \mathbf{b}_{\mathbf{tar}}^\dagger$
8: $\mathbf{c} = P\mathbf{c_{subspace}}$

4.3.7 Computational Details

The above algorithms were implemented for NVidia graphical processing unit (GPU) hardware in a development version of the TeraChem[179, 180, 181, 98, 177, 40, 60, 132] software package using the Compute Unified Device Architecture (CUDA) API[125] and the NVidia CUDA basic linear algebra subprograms (cuBLAS) library[124]. All benchmark calculations were performed on a single core of an Intel E5603 1.60 GHz processor and a single NVidia

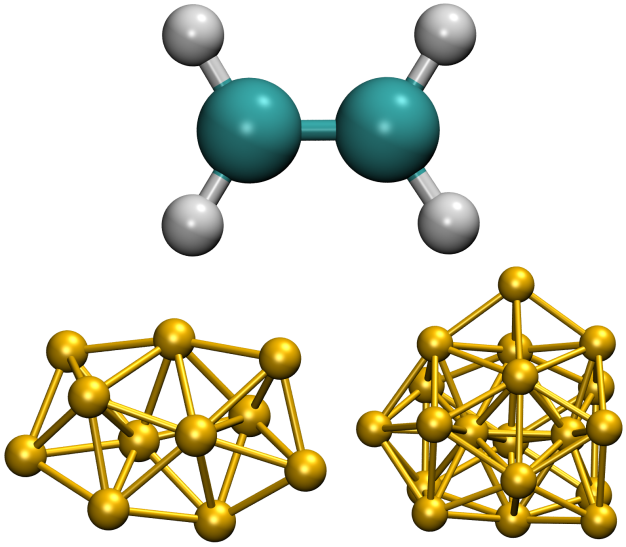

Figure 4.1: Ethylene, Ag_{11}, and Ag_{19} molecules used in the present work are shown at the top, bottom left, and bottom right, respectively. Carbon, hydrogen, and silver atoms are shown in blue, white, and yellow, respectively. Cartesian coordinates are given in the Supporting Information.

K40 GPU. A standard Davidson-Liu eigensolver was used where the maximum subspace dimension was chosen to avoid subspace collapse whenever possible, subject to memory limitations, and the initial search space was chosen to have two guess vectors for each desired root. Purified trial vectors were normalized prior to preconditioning and addition to the search space. The initial guess space for all calculations included 400 determinants. Throughout this work we abbreviate configuration spaces according to the convention (p, q), where p is the number of active electrons and q is the number of active orbitals. In the numerical tests that follow, both Hartree-Fock and state-averaged complete active space self-consistent field (SA-CASSCF) orbitals are used in the construction of CASCI wave functions. When Hartree-Fock orbitals are used, the resulting wave function will be denoted HF-CASCI.

4.4 Results

4.4.1 Direct $\mathbf{S^2c}$ Performance

Formation of the $\boldsymbol{\sigma}$ vector is the performance-limiting step in direct CI calculations, and using modern algorithms, this step scales approximately linearly with respect to the number of configurations. For efficient spin purification, calculation of the $\mathbf{S^2c}$ matrix-vector product must be fast relative to $\boldsymbol{\sigma}$ formation. Since the costs of both $\boldsymbol{\sigma}$ and $\mathbf{S^2c}$ formation depend solely on the size of the configuration space, and not on the size of the single-electron basis set, we illustrate the performance of our algorithms using a small test system: the neutral C_2H_4 molecule shown in the top panel of Figure 4.1. Figure 4.2 presents a comparison of the times to perform $\boldsymbol{\sigma}$ and $\mathbf{S^2c}$ formation for active spaces ranging from 12 electrons in 12 orbitals ((12,12), or 853,776 determinants) to 12 electrons in 15 orbitals ((12,15), or 25,050,025 determinants).

The scaling exponents for $\boldsymbol{\sigma}$ and $\mathbf{S^2c}$ formation are observed to be 1.09 and 1.12, respectively. Due to a smaller prefactor, the formation of $\mathbf{S^2c}$ requires a roughly 15-fold shorter time than $\boldsymbol{\sigma}$. It is worth mentioning that our $\boldsymbol{\sigma}$ formation algorithm does not yet take advantage of spin symmetry to reduce the computational cost, but the systems most likely to suffer spin contamination have an odd number of electrons and the performance of $\mathbf{S^2c}$ formation would not benefit from inclusion of such symmetry in these cases.

4.4.2 Wave Function Convergence

To evaluate the degree to which the purification schemes described above aid convergence, we have performed calculations of singlet and doublet wave functions of neutral and anionic ethylene, respectively. First we consider HF-CAS-(8,8)-CI/6-31G** calculations of singlet ethylene. We have varied the numbers of roots from 5 to 15 and investigated convergence both with and without spin purification. The number of iterations required for convergence and the number of converged roots of undesired spin symmetry are tabulated in Table 4.1.

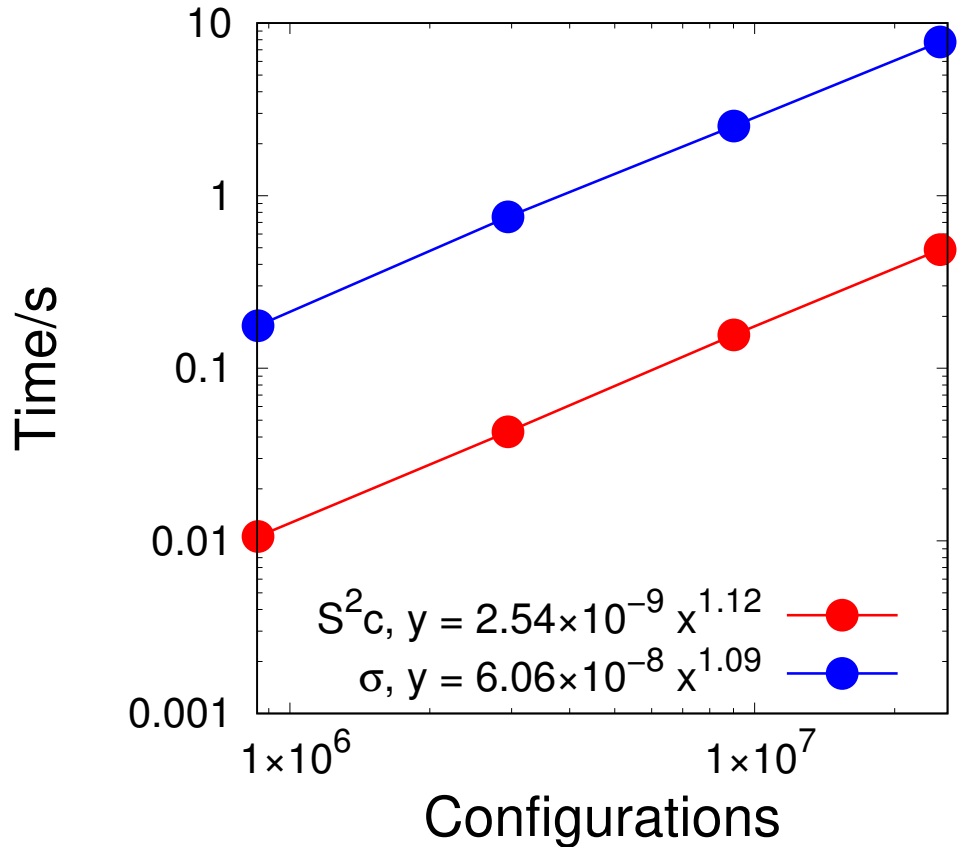

Figure 4.2: Times for $\mathbf{S^2c}$ and $\boldsymbol{\sigma}$ formation for neutral C_2H_4 at the HF-CASCI/6-31G** level with active spaces ranging from (12,12) to (12,15). Calculations were performed using a single NVidia K40 GPU.

Note that we have encouraged spin contamination in this case by tightening convergence to a residual threshold of $||r|| = 1.0 \times 10^{-7}$. (Convergence to $||r|| = 1.0 \times 10^{-6}$ is generally considered suitable for calculation of analytic energy or orbital gradients.) Penalty purification was performed with $\alpha = 0.10$. The preconditioners suggested by Davidson (exact diagonal energies) [33] and Evangelisti (reference determinant energy modified by orbital energy differences)[39] were both tested in this context. Remember that preconditioning with the exact diagonal energies is known to lead to spin contamination, while Evangelisti should not.

First consider computations with the Evangelisti preconditioner. For cases where only 5-8 states are requested all schemes (including diagonalization without purification) result

Table 4.1: The number of iterations required for convergence of varying numbers of singlet states of ethylene at the HF-CAS-(8,8)-CI/6-31G** level, with $||r|| = 1.0 \times 10^{-7}$. Results for the uncorrected Davidson-Liu method, the penalty method ($\alpha = 0.10$), the first-order and Löwdin projection methods, and inverse iteration are all reported. When some roots converge to the states of undesired spin, the number of such roots is reported in parentheses.

States	**No Purif.**	**Penalty**	**Löwdin Proj.**	**1st-Ord. Proj.**	**Inv. Iter.**
Orbital Energy Preconditioner (Evangelisti)					
5	12	14	12	12	12
6	12	13	12	12	12
7	12	15	12	12	12
8	13	15	13	13	13
9	52 (5)	16	13	13	13
10	50 (5)	20	14	14	14
11	44 (6)	22	15	15	15
12	40 (6)	21	16	16	16
13	35 (6)	18	14	14	14
14	41 (7)	18	14	14	14
15	35 (7)	19	15	15	15
Exact Diagonal Energy Preconditioner (Davidson)					
5	13	47	13	13	13
6	35 (3)	43	13	13	13
7	13	41	13	13	13
8	35 (4)	36	13	13	13
9	35 (5)	36	13	13	13
10	33 (5)	67	16	16	16
11	36 (6)	47	14	14	14
12	32 (6)	94 (1)	22	21	22
13	33 (6)	69 (1)	17	16	17
14	36 (7)	70 (1)	16	15	15
15	34 (7)	79 (1)	18	18	18

in successful convergence to singlet states, with the penalty approach requiring more iterations (13-15) than the other schemes (12-13). When 9 states are requested, unpurified Davidson-Liu diagonalization with Evangelisti preconditioning requires more iterations (52) than above, and ultimately converges to several (5) unwanted triplet states. Each of the penalty, Löwdin projection, first-order projection, and inverse iteration purification methods continue to perform well, showing modest increases in iteration count relative to the cases with fewer states, while achieving the desired spin in all states. The two projection methods and inverse iteration require 13 iterations, while the penalty algorithm requires 16. This general pattern holds as the number of states in increased from 9 to 15; unpurified calculations converge to many roots with incorrect spin, while all four spin purification schemes are effective. The convergence behaviors of the two projection methods and inverse iteration are comparable to one another and superior to that of the penalty method in all of these cases.

Exact diagonal energy preconditioning presents a greater challenge because this preconditioner introduces spin contamination into the wave function. When no purification method is used, spin contamination is first observed when only 6 states are requested. Interestingly, using the exact diagonal preconditioner often provides faster convergence in the absence of purification than Evangelisti does, albeit to eigenvectors having the incorrect spin symmetry. Both projection methods and inverse iteration continue to provide robust convergence and display similar convergence characteristics, converging to the desired spin state in 13-22 iterations in all cases. Penalty purification performs more poorly, exhibiting slow convergence for calculations of 5-11 states (requiring 67 iterations in the worst case), and a failure to completely purify all roots in calculations of 12-15 states. By increasing the penalty parameter to $\alpha = 0.15$ we are able to converge to the desired singlet states at the cost of an increase in iteration count for each calculation. The dependence of convergence on the penalty parameter (α) is discussed in more detail below.

Next we consider a more difficult case—the doublet states of the ethylene anion—examining the convergence behavior of each spin purification method. The lowest 20 doublet

Table 4.2: Performance of various spin purification methods for $C_2H_4^-$ calculated at the HF-CAS-(7,8)-CI/6-31G** level with $||r|| = 1.0 \times 10^{-6}$. Spin penalty calculations were performed with $\alpha = 0.10$.

Method	States Incorrect	Iterations	$\boldsymbol{\sigma}$ Formations
Orbital Energy Preconditioner (Evangelisti)			
No Purification	7	46	359
Spin Penalty	0	41	331
Löwdin Projection	0	27	252
1st-Order Projection	0	27	252
Inverse Iteration	0	27	252
Exact Diagonal Energy Preconditioner (Davidson)			
No Purification	7	26	270
Spin Penalty	0	51	508
Löwdin Projection	0	40	233
1st-Order Projection	0	40	233
Inverse Iteration	0	40	233

states of the ethylene anion were calculated at the HF-CAS-(7,8)-CI/6-31G** level using $||r|| = 1.0 \times 10^{-6}$, with the unpurified Davidson-Liu and the penalty, Löwdin projection, first-order projection, and inverse iteration spin purification methods. We, again, ran tests for both Evangelisti and Davidson preconditioning to investigate whether spin purification can compensate for contamination induced by exact diagonal element preconditioning. The numbers of iterations and $\boldsymbol{\sigma}$ formations required for convergence as well as the number of roots that converged to the incorrect spin are reported in Table 4.2.

Standard Davidson-Liu without spin purification requires 46 iterations for convergence with the Evangelisti preconditioner, corresponding to 359 $\boldsymbol{\sigma}$ formations, and 7 of the 20 roots converge to states having the incorrect quartet spin symmetry. Applying the penalty method ($\alpha = 0.10$) improves the results significantly, reducing the number of iterations to 41 while, most importantly, all 20 converged roots now have the desired spin symmetry. A smaller penalty parameter of $\alpha = 0.01$ results in a spin contaminated wave function. The Löwdin projection, first-order projection, and inverse iteration methods fare even better, converging to the correct solution in 27 iterations. If Davidson preconditioning is used the

trends observed are similar, with the two projection methods and inverse iteration purification requiring 51% fewer $\boldsymbol{\sigma}$ formations than penalty purification. Again, our spin purification methods are robust even when faced with a preconditioner known to introduce spin contamination into the wave function.

The ability to use a wider variety of preconditioners is of practical value. Two of the most useful applications of our spin purification approaches are in geometry optimization and ab initio molecular dynamics methods (e.g. ab initio multiple spawning[89, 16] and its recent enhancements[42, 32, 111]), where reliable wave function convergence minimizes the human-time-intensive task of troubleshooting finicky simulations. In this context convergence failures may be observed when using the orbital energy preconditioner fails due to numerical instability following preconditioning. Reverting to the exact diagonal preconditioner allows for convergence of these wave functions, and while spin contamination is systematically introduced using this preconditioner, our purification methods serve to efficiently remove all traces of contamination when detected. This additional flexibility can be used to ensure consistent solution of the CI equations.

Counting $\boldsymbol{\sigma}$ formations alone is a poor measure of computational cost, as it neglects $\langle \hat{S}^2 \rangle$ testing and purification. Calculation of $\mathbf{S^2c}$ is inexpensive compared to $\boldsymbol{\sigma}$, but not negligibly so, therefore it is important to compare the actual times-to-solution for the CI calculations using different purification methods. Here we report such timings for both penalty and projection purification. We do not include inverse iteration in these tests because it exhibits similar convergence to Löwdin and first-order projection but requires significantly more $\mathbf{S^2c}$ formations, which is the performance-limiting operation in all purification methods. The penalty purification parameter, α, is varied from 0.00 (unpurified Davidson-Liu) to 0.20. We calculate the lowest 12 singlet states of neutral ethylene at the HF-CAS-(12,12)-CI/6-31G** level with $||r|| = 1.0 \times 10^{-7}$, with the number of iterations, number of $\boldsymbol{\sigma}$ formations, and times-to-solution reported in Table 4.3. Evangelisti preconditioning was used for the results to follow.

Table 4.3: Number of iterations required and times-to-solution for convergence of 12 singlet states of neutral ethylene at the HF-CAS-(12,12)-CI/6-31G** level. Results are shown for the spin penalty method as a function of the spin penalty parameter, α, and for the first-order and Löwdin spin projection methods, with $||r|| = 1.0 \times 10^{-7}$ in each case. The number of states converging to the incorrect spin symmetry are given in parentheses.

α	Iterations	σ Formations	CI Time-to-Solution (s)
		Spin Penalty	
0.00	54 (6)	337	657.77
0.01	66 (5)	343	707.18
0.02	90 (4)	377	819.53
0.05	67 (1)	319	694.53
0.10	72	416	737.42
0.15	85	522	1071.96
0.20	102	604	1262.56
		First-Order Projection	
—	26	219	485.80
		Löwdin Projection	
—	26	219	562.23

Table 4.4: Number of iterations required and times-to-solution for convergence of 12 doublet states of anionic ethylene at the HF-CAS-(13,12)-CI/6-31G** level. Results are shown for the spin penalty method as a function of the spin penalty parameter, α, and for the first-order and Löwdin spin projection methods, with $||r|| = 1.0 \times 10^{-6}$ in each case. The number of states converging to the incorrect spin symmetry are given in parentheses.

α	Iterations	σ Formations	CI Time-to-Solution (s)
		Spin Penalty	
0.00	68 (3)	282	620.64
0.01	43	225	380.44
0.02	53	242	465.57
0.05	73	294	709.74
0.10	132	424	777.69
0.15	137	495	1017.60
0.20	159	565	1375.84
		First-Order Projection	
—	37	218	372.17
		Löwdin Projection	
—	37	218	379.09

Standard Davidson-Liu results in 6 erroneous roots (triplet states) after 54 iterations. Increasing the α parameter has the effect of reducing the number of triplet states found, until at a value of $\alpha = 0.10$ all 12 of the lowest-lying singlets are located in this particular case. Continuing to increase the value of α results in slower convergence, though the spin purity of the wave function is retained. This trend suggests that the smallest α value that provides spin-pure solutions is the optimal choice. Both Löwdin and first-order projection provide superior performance to the penalty method, however, regardless of the value of α. First-order and Löwdin projection require 485.80 s and 562.23 s, respectively, compared to 737.42 s for the spin penalty method with $\alpha = 0.10$.

We performed a similar set of tests for the lowest 12 doublet states of the ethylene anion, computed at the HF-CAS-(13,12)-CI/6-31G** level. Here a less stringent (and more typical) convergence requirement is used ($||r|| = 1.0 \times 10^{-6}$). Results are reported in Table 4.4. Here first-order projection, Löwdin projection, and spin penalty purification yield very similar performance with times-to-solution of 372.17 s, 379.09 s, and 380.44 s, respectively. For this system a more modest penalty value of $\alpha = 0.01$ eliminates spin contamination without significantly increasing the number of iterations required for convergence. The doublet calculations show a similar trend to the singlet calculations in that the smallest penalty parameter, α, that produces spin-pure eigenvectors results in the fastest convergence, and further increasing the penalty parameter slows convergence. Though optimal performance of the spin penalty method is comparable to the projection methods in this case, we note that there is no way to predict *a priori* what value of α will give optimal results. Choosing a safer value of $\alpha = 0.10$ results in a more than two-fold increase in time-to-solution. In this sense, the two projection schemes are preferable.

A similar series of calculations on a small silver cluster (doublet Ag_{11}, pictured in the bottom left panel of Figure 4.1) yield similar results, with first-order projection performing marginally better than Löwdin projection (335.98 s compared to 375.26 s). In this case, however, the fastest spin penalty calculations outperformed first-order projection (217.45

s compared to 335.98 s). Still, a similarly dramatic dependence of performance on α is observed, making it difficult to achieve optimal performance without prior knowledge of the optimal α. Thus the two projection schemes remain preferable. Detailed results are reported in Supporting Information.

Above we have only considered states with maximal m_s (e.g. doublets with $m_s = 1/2$), but our spin purification schemes also allow us to target states with lower m_s. Though computing such interior spin states is not a common practice, we have investigated the effectiveness of our methods on such states. In two cases ($m_s = 1/2$ quartet states of ethylene anion and Ag_{11}) we find spin penalty to be a robust method for solving such states. The projection methods, on the other hand, show very slow or even failed convergence in these cases. Detailed results are reported in Supporting Information.

4.5 Application to Silver Clusters

Nobel metal nanoclusters, including silver nanoclusters (AgNCs), have unique and tunable optical properties. Many of these properties arise from the existence of localized surface plasmon resonances (LSPRs)[54, 194, 67, 170]. LSPRs are collective oscillations of conduction electrons in nanoscale systems that have strong absorption and scattering cross sections and can lead to greatly enhanced local electric fields at a material's surface. Quantum chemical calculations at the linear response time-dependent density functional theory (LR-TDDFT) level suggest that LSPRs are well represented as linear combinations of configurations which are singly-excited with respect to the ground state [3, 13, 51]. However, intuitively, the collective excitation of multiple electrons may involve multiply-excited determinants that are absent from LR-TDDFT. Multiply excited states may also be involved in nonlinear optical processes in AgNCs. In fact, doubly-excited states have previously been reported in small silver clusters computed at the equation-of-motion coupled cluster level[23].

Here we apply our spin purification scheme in combination with recent algorithmic and hardware advances[60, 167, 166] to investigate the role that multiply-excited determinants

may play in the low-lying excitations of a small silver cluster. Specifically, we compute the 12 lowest doublet states and 1-electron properties of the Ag_{19} icosahedral cluster whose geometry was determined by the global minimization study of Fournier [43] (bottom right panel of Figure 4.1). These calculations are carried out at the state-averaged complete active space self-consistent field (SA-CASSCF) level of theory using an active space of 11 electrons in 11 orbitals and averaging over 12 states.

The degree of single excitation character of a given transition can be quantified using ideas inspired by natural transition orbital (NTO) analysis[107]. The NTOs of an open shell system are obtained, in part, by solving the eigenvalue equation

$$\left(\boldsymbol{\gamma^{\alpha}}\left(\boldsymbol{\gamma^{\alpha}}\right)^{\dagger}+\boldsymbol{\gamma^{\beta}}\left(\boldsymbol{\gamma^{\beta}}\right)^{\dagger}\right)u_i=\lambda_i u_i \tag{4.11}$$

with u_i defining the orbitals, λ_i the eigenvalues (occupation numbers), and $\boldsymbol{\gamma^{\alpha}}$ and $\boldsymbol{\gamma^{\beta}}$ the α- and β-spin components of the transition one-particle reduced density matrix (1-RDM) between the states of interest. In a standard TDDFT or configuration interaction singles calculation, the sum of the eigenvalues, λ_i, (or equivalently the trace of the matrix) is exactly one. This reflects the fact that all amplitudes in these single reference wave functions can be described by single excitations between NTO pairs. In higher-order CI calculations, however, the trace varies between zero and one and indicates the degree of multiple excitation character between states; a value near one indicates strongly singly-excited character while significant deviation indicates a transition with multiply-excited character. We present the single excitation character of the interstate transitions from the ground doublet state D_0 for Ag_{19} in Table 4.5 along with the magnitudes of the transition dipole moments and energies of these transitions. (The largest CI vector coefficients for each state are reported in Supporting Information.)

As can be seen, many of the lower energy transitions are predominantly singly-excited, but several of the higher energy transitions exhibit significant multiply-excited character, most notably $D_0 \rightarrow D_9$, $D_0 \rightarrow D_{10}$, and $D_0 \rightarrow D_{11}$, which all have singly-excited characters

Table 4.5: Absolute transition dipole moments ($|\boldsymbol{\mu}|$), single excitation characters, and transition energies of $D_0 - D_X$ transitions of Ag_{19} calculated at the SA-12-CAS-(11,11)-SCF/LANL2DZ level. Spin purification was performed using the penalty method with $\alpha = 0.01$.

Transition	**$\|\boldsymbol{\mu}\|$ (Debye)**	**Single Excitation Character**	**Transition Energy (eV)**
$D_0 \rightarrow D_1$	0.0150	0.9506	0.7510
$D_0 \rightarrow D_2$	0.5953	0.9161	1.2038
$D_0 \rightarrow D_3$	0.0102	0.9151	1.2837
$D_0 \rightarrow D_4$	0.4906	0.8956	1.3858
$D_0 \rightarrow D_5$	0.1342	0.8934	1.4411
$D_0 \rightarrow D_6$	0.5737	0.8476	1.5107
$D_0 \rightarrow D_7$	0.6911	0.9393	1.5953
$D_0 \rightarrow D_8$	0.5643	0.8155	1.6624
$D_0 \rightarrow D_9$	0.1728	0.6400	1.8344
$D_0 \rightarrow D_{10}$	0.1304	0.4619	1.8949
$D_0 \rightarrow D_{11}$	0.6626	0.4180	1.9065

Table 4.6: Transition dipole moments and energies between excited states of Ag_{19} calculated at the SA-12-CAS-(11,11)-SCF/LANL2DZ level. Spin purification was performed using the penalty method with $\alpha = 0.01$. Only transitions with absolute transition dipole moments ≥ 0.25 Debye are reported.

Transition	**$\|\boldsymbol{\mu}\|$ (Debye)**	**Transition Energies (eV)**
$D_1 \rightarrow D_7$	0.8309	0.8443
$D_1 \rightarrow D_8$	0.6718	0.9114
$D_1 \rightarrow D_9$	0.3988	1.0833
$D_1 \rightarrow D_{10}$	0.4001	1.1439
$D_2 \rightarrow D_3$	0.5610	0.0799
$D_2 \rightarrow D_4$	0.6235	0.1820
$D_2 \rightarrow D_5$	0.5706	0.2373
$D_2 \rightarrow D_6$	0.6731	0.3069
$D_2 \rightarrow D_{11}$	0.5091	0.7027
$D_3 \rightarrow D_4$	0.5033	0.1021
$D_3 \rightarrow D_5$	0.4451	0.1574
$D_4 \rightarrow D_6$	0.5897	0.1249
$D_6 \rightarrow D_{10}$	0.3485	0.3842
$D_6 \rightarrow D_{11}$	0.2930	0.3958
$D_8 \rightarrow D_9$	0.6423	0.1720
$D_8 \rightarrow D_{10}$	0.6113	0.2325
$D_{10} \rightarrow D_{11}$	0.6265	0.0116

below 0.7. In some cases the excitation remains quite bright despite the significant multiply-excited character. For example, the $D_0 \rightarrow D_{11}$ transition has an absolute transition dipole of 0.66 Debye despite a single excitation character of 0.42. Other transitions with low single excitation character ($D_0 \rightarrow D_9$ and $D_0 \rightarrow D_{10}$) are darker.

Interestingly, even those darker excitations may be accessible via multi-photon processes. Table 4.6 reports the transition dipoles and energies for all excitations originating from states above D_0 with absolute transition dipole moments greater than 0.25 Debye. The dark multiply-excited D_{10} state, for example, has a large transition dipole moment from the bright D_6 and D_8 states.

These results suggest that multiply-excited electronic states may play a roll in the non-linear optical processes of plasmonic materials. This suggests that in order to understand such processes quantum chemical methods capable of describing such multiply-excited states are required. We plan to further investigate this intriguing, albeit preliminary result in the future.

4.6 Conclusions

In this work we presented several approaches for systematically avoiding or removing numerical spin contamination from CI wave functions during Davidson diagonalization. In order to make these approaches viable we developed a direct, GPU-accelerated algorithm for computing the $\mathbf{S^2c}$ matrix-vector product, which we demonstrated to scale linearly with respect to the number of electronic configurations. The performance of each of the purification schemes was compared for a variety of cases, and we found that first-order projection offers the best balance of low computational cost and accurate wave function convergence. Löwdin projection showed equally robust convergence, but marginally slower performance in our test cases. Penalty purification was robust and efficient so long as the optimal penalty parameter, α, was chosen, but the optimal choice for this parameter was found to be system specific, and an *a priori* scheme for choosing it is not obvious. Spin contamination or poor

convergence may result from a sub-optimal choice. Spin penalty was, however, the only one of the four approaches that allowed us to reliably optimize states with m_s less than the maximum allowed value for a given multiplicity (e.g. an $m_s = 1/2$ quartet). The final approach, based on inverse iteration, provided robust results but was far less computationally efficient than the two projection approaches, and thus is not recommended. We have incorporated all of our purification methods into a development version of the TeraChem software package, where they will be made available in a future release.

To demonstrate the utility of these schemes, we performed a brief study of the icosahedral Ag_{19} silver cluster using the SA-CASSCF method. Our results provide evidence of low-lying states with multiply-excited character in this system, suggesting that multireference electronic structure approaches may be valuable in the study of plasmonic nanomaterials. A future study will elaborate on this preliminary result.

While the work described here focused on FCI-based methods, the purification schemes presented are applicable to any determinant-based method having a spin-free Hamiltonian. Looking towards the future, we are exploring ways to reduce the storage and computational requirements of CI through use of single precision floating point data structures, making use of a Lagrangian based approach in conjunction with purification based on our efficient $\mathbf{S^2c}$ formation algorithm.

CHAPTER 5

LARGE SCALE ELECTRONIC CORRELATION CALCULATIONS: RANK-REDUCED FULL CONFIGURATION INTERACTION

5.1 Abstract

We have developed a variational methodology suitable for the calculation of extremely large configuration interaction (CI) wavefunctions. In this report we demonstrate the ability to obtain closed shell ground state singlet and triplet energies within mH accuracy of the full CI (FCI) answer with $\sqrt{N_{det}}$ scaling, where N_{det} is the number of configurations in the CI space. Fast graphical processing unit (GPU) accelerated projected $\boldsymbol{\sigma} = \mathbf{Hc}$ matrix-vector product formation enables calculations using configuration spaces as large as 30 electrons in 30 orbitals, corresponding to an FCI calculation with over 2.4×10^{16} configurations.

5.2 Introduction

Improving methods for accurate and efficient description of electron correlation in complex systems is an ongoing effort in theoretical and computational chemistry. Full configuration interaction (FCI), a linear expansion of configurations in the vector space defined by single particle basis functions, represents the exact solution to the electronic structure problem in the limit of the defined basis set. Due to the extremely high scaling of FCI (proportional to the product of the binomial coefficients of α and β electrons and the number of basis functions (orbitals) in the configuration space), much effort has been spent developing approximations that offer a compromise between computational effort and accuracy. Besides being formally exact, FCI is also conceptually simple relative to other high accuracy methods. Despite its advantages, FCI most often serves as a benchmarking tool for quantifying the relative accuracy of lower cost approximate methods because of its extreme computational cost. In addition to serving as an evaluation tool for lower-cost methods, FCI

provides the mechanism for electron correlation in a specific orbital space, often the valence space, which when combined with iterative orbital reoptimization defines the complete active space self-consistent field (CASSCF) [144, 163, 143] or fully optimized reaction space (FORS) [148] methods. These approaches were specifically designed to describe static electron correlation, where the electronic structure is not well described by a single Slater determinant, instead requiring contributions from two or more dominant configurations. This situation often arises in polyradicaloid systems which are found in many photochemical processes, in transition metal systems, and in non-equilibrium systems such as those which experience bond stretching. A related approach to CASSCF is the complete active space configuration interaction (CASCI) method, where the salient difference is that CASCI omits the orbital reoptimization procedure.

FCI may be considered a mature electronic structure method. Numerical approaches have been developed specifically to enhance the relatively slow convergence of CI through improved diagonalization [33], and algorithmic aspects of the method have been dramatically improved, perhaps most notably including the development of direct CI methods [141]. Slater determinants are often used in lieu of configuration state functions due to the convenient factorization of the determinantal CI vector into α and β strings and because the 1-particle coupling coefficients are trivially integers [53]. Additional progress by Siegbahn [164] resulted in approaches even more suitable for modern computing by leveraging fast matrix operations, and additional refinement in the context of vector processing machines was performed by Knowles and Handy [73] and more recently by our group [40]. Even through use of cutting edge hardware and highly efficient algorithms, however, configuration spaces for high-throughput calculations are still limited to approximately 10^9 determinants. Thus, the problem of high computational scaling with respect to the configuration space size places hard limits on the dimension of problems that can be reasonably studied, both due to time and hardware (memory) constraints.

Efforts to improve scaling can be focused on either (or both) of the configuration space

dimension or the orbital space dimension. An upper bound to the formal scaling of direct CI is $N_{det}N_o^4$, where N_{det} is the number of configurations (determinants in our case) and N_o is the number of basis functions in the configuration space (algorithms having smaller operation counts have been described, for example by Olsen [128], but are less efficiently implemented for vector machines). Approaches that target the N_o^4, or orbital space, scaling include pseudospectral methods [44, 108], density fitting and Cholesky decomposition [15, 139, 78, 7, 6, 191], and tensor hypercontraction density fitting [61, 133, 59, 134, 168, 169, 154]. These approaches all work to reduce the dimension of the MO basis electron repulsion integrals, a rank 4 tensor, saving both the expense of the integral transformation as well as some potential computational cost in the $\boldsymbol{\sigma} = \mathbf{Hc}$ vector formation itself during the direct CI iterations. Unfortunately, unless the orbital basis is constrained by locality [31, 80] (which simultaneously limits the dimension of the configuration space), the savings realized during the CI step is modest, since modern direct CI algorithms exhibit approximately linear scaling with respect to the number of configurations, the cost associated with the N_o^4 scaling being amortized through a combination of sparsity leveraging as well as asynchronous memory operations to hide some of this cost. Furthermore, the size of the configuration space increases combinatorially with the number of orbitals, growing much faster than the quartic scaling of the N_o^4 component, motivating the development of methods which reduce the scaling of the configuration space dimension.

Direct CI algorithms based on Krylov subspace diagonalization rely on iteratively expanding the subspace size with search directions determined by preconditioning the residual vector from the previous iteration. This concept leads naturally to that of *selected CI*, where the search directions for the addition of trial vectors to the subspace are determined through alternative approaches including those based on perturbation theory[63, 25, 178, 153] and stochastics[62]. Selected CI methods are being actively developed as promising lower cost FCI alternatives.

Stochastic approaches to solving the FCI problem such as quantum Monte Carlo (QMC-)

FCI[24, 174] have been demonstrated to be powerful ways of describing correlated fermionic systems. Unfortunately, uncontrolled errors resulting from the constraint imposed by the antisymmetry principle (known as the Fermion sign problem) remain open challenges in QMC-FCI. Alternatively, variational methods based on tensor networks, such as density matrix renormalization group (DMRG)[192, 29], provide high-accuracy deterministic solutions for both 1-dimensional and branched/tree-like systems using matrix product states and tree tensor network states[120], respectively. Systems possessing higher orders of quantum entanglement, however, are somewhat poorly described by DMRG.

Another reasonable approach for reducing the configuration space scaling relies on taking advantage of sparsity in the CI vector. Examples include methods by Knowles [72, 75], Mitrushenkov [115, 116], and Rolik [140]. These approaches adaptively expand the linear search space during the addition of correction vectors to the CI subspace, resulting in a shorter CI vector, especially during early iterations, and a reduction in operation count.

Olsen et al. developed the low-rank CI (LR CI) method using spectral resolution of the CI coefficient matrix [127] to compress wavefunction information. A second-order Newton-Raphson scheme was used to solve the CI with single and double replacements (CISD). The LR CISD approach was applied to test systems including neon, nitrogen, and water, and systematic improvement of the energy relative to full CISD was observed as higher-rank wavefunctions were employed. Lindh et al. followed this work by implementing a low-rank multi-configuration self-consistent field (LR SCF) method [92] and showed an example where polarizabilities were better described using LR SCF than with LR CI, though wavefunction convergence was hindered by strong coupling between orbital rotations and single excitations. Inspired by this work, Koch and Dalgaard [77] formulated a variational matrix decomposition using non-linear optimization methods to solve the FCI problem. Koch modified a FCI code to assess the performance of the decomposition on a series of small benchmark systems. Preliminary results were promising, showing rapid convergence to the FCI energies with mH accuracy.

Both the LR CI and variational matrix decomposition rely on the assumption that much of the information represented by the CI coefficients is redundant. An equivalent statement is that the matrix form of the CI vector has low rank. Recently, Taylor [173] reported a study where the single value decomposition (SVD) was used to quantify the effect of solving the linear CI equations using low rank approximations to the current trial vector at each iteration with various accuracy thresholds. Several representative FCI cases were examined, including both "stout" (typical of a CASSCF or a CASCI calculation, where the number of electrons is similar to the number of orbitals) and "slim" (more characteristic of a benchmark type calculation, where the number of orbitals is much larger than the number of electrons), and in each case Taylor demonstrated that while use of a lower-rank trial vector at each iteration slowed convergence, the LR-SVD calculation ultimately converged to the same eigenvector as standard FCI (to within a given residual threshold). A significant impact of Taylor's work is demonstration of the low-rank of the CI vector, though high computational cost of the SVD operation (similar to that of a $\boldsymbol{\sigma}$ build in standard direct CI) and absence of a reduction in trial vector length are non-trivial challenges in adapting LR-SVD for use in large scale production calculations.

As described above, writing the CI vector coefficients, $\mathbf{C}$, in matrix form where the rows correspond to α strings and columns to β strings permits a reduction in scaling by using a matrix decomposition. In the present work, we extend the ideas of Koch and Dalgaard [77], where the product space vectors are determined variationally through solution of a non-linear multi-scale eigenvalue problem (a type of super-CI approach, similar to those used in two-step CASSCF implementations), in developing a rank-reduced FCI (rrFCI) formulation. The structure of the paper is as follows: in Section 5.3 we introduce the augmented eigenvalue problem for the product space vectors (Section 5.3.1), including details related to program design, 1-particle coupling coefficient formation (Section 5.3.3.1), metric formation (Section 5.3.3.3), and projected $\boldsymbol{\sigma}$ formation (Section 5.3.2). Next we demonstrate computational performance of the rrFCI method in Section 5.4.2, before applying rrFCI to acenes having

between $2-5$ polycyclic aromatic rings (Section 5.4.3) and to molecular nitrogen dissociation (Section 5.4.5). We conclude with a brief overview of the current method before discussing future directions of work.

5.3 Methods

The second-quantization electronic Hamiltonian is

$$\hat{H} = \sum_{kl}^{m} h_{kl}\hat{E}_{kl} + \frac{1}{2}\sum_{ijkl}^{m}(ij|kl)\left(\hat{E}_{ij}\hat{E}_{kl} - \delta_{jk}\hat{E}_{il}\right) \tag{5.1}$$

where i, j, k, l index molecular orbitals, h_{ij} and $(ij|kl)$ are the one- and two-electron integrals, and $\hat{E}$ is a single-particle excitation generator.

Instead of building and diagonalizing the full Hamiltonian to determine the eigenvalues and their corresponding eigenvectors, direct methods allow the iterative solution for a few of the lowest-lying eigenvalues and eigenvectors. In the present work we describe calculation of the lowest eigenvalue/eigenvector pair (ground state) - the calculation of higher-lying states (excited states) is an effort currently underway in our lab and will be presented in a future work. The rate-limiting step in the direct CI procedure is formation of the matrix-vector product $\boldsymbol{\sigma}$ defined as

$$\sigma_I = \sum_J H_{IJ}c_J \tag{5.2}$$

Formally, $\boldsymbol{\sigma}$ formation requires $\mathcal{O}(N^2)$ operations, but modern algorithms can achieve scaling approaching $\mathcal{O}(N)$. We present here a reduction in the formal scaling to $\mathcal{O}(N)$ which implies an effective scaling of $\mathcal{O}(\sqrt{N})$ for the rate-limiting operation.

5.3.1 Eigenvalue Problem

We define the CI vector coefficient matrix decomposition as

$$\mathbf{C} = \sum_i c_i \left(\mathbf{P}_i \otimes \mathbf{Q}_i\right) \tag{5.3}$$

where i is less than or equal to the full (exact) rank of $\mathbf{C}$, c_i are product term coefficients, and $\mathbf{P}$ and $\mathbf{Q}$ are product state vectors of length N_{string}. To balance the product space vectors with respect to one another we have adopted the convention of normalizing the converged $\mathbf{P}$ and $\mathbf{Q}$ vectors, giving the state vector $\boldsymbol{\Psi_i}$

$$|\boldsymbol{\Psi_i}\rangle = |\mathbf{P}_i \otimes \mathbf{Q}_i\rangle \tag{5.4}$$

$$\langle\boldsymbol{\Psi}_i|\boldsymbol{\Psi}_i\rangle = |\mathbf{P}_i|^2|\mathbf{Q}_i|^2 = 1 \tag{5.5}$$

Applying the variational principle to the full wavefunction we obtain

$$\delta\left(\langle\boldsymbol{\Psi}|\hat{H}|\boldsymbol{\Psi}\rangle - E\langle\boldsymbol{\Psi}|\boldsymbol{\Psi}\rangle\right) = 0 \tag{5.6}$$

giving the coupled equations for the first set of product terms

$$\left(\langle\boldsymbol{\mu}\otimes\mathbf{Q}|\hat{H} - E|\boldsymbol{\mu}\otimes\mathbf{Q}\rangle\right)\mathbf{P} = 0 \tag{5.7}$$

$$\left(\langle\mathbf{P}\otimes\boldsymbol{\nu}|\hat{H} - E|\mathbf{P}\otimes\boldsymbol{\nu}\rangle\right)\mathbf{Q} = 0 \tag{5.8}$$

and the coupled augmented eigenvalue problems

$$\begin{bmatrix} \langle\boldsymbol{\Psi}|\hat{H} - E|\boldsymbol{\Psi}\rangle & \langle\boldsymbol{\Psi}|\hat{H} - E|\boldsymbol{\mu}\otimes\mathbf{Q}\rangle \\ \langle\boldsymbol{\mu}\otimes\mathbf{Q}|\hat{H} - E|\boldsymbol{\Psi}\rangle & \langle\boldsymbol{\mu}\otimes\mathbf{Q}|\hat{H} - E|\boldsymbol{\mu}\otimes\mathbf{Q}\rangle \end{bmatrix} \begin{bmatrix} a_0 \\ \mathbf{P} \end{bmatrix} = 0 \tag{5.9}$$

for the $\mathbf{P}$ optimization and

$$\begin{bmatrix} \langle\boldsymbol{\Psi}|\hat{H} - E|\boldsymbol{\Psi}\rangle & \langle\boldsymbol{\Psi}|\hat{H} - E|\mathbf{P}\otimes\boldsymbol{\nu}\rangle \\ \langle\mathbf{P}\otimes\boldsymbol{\nu}|\hat{H} - E|\boldsymbol{\Psi}\rangle & \langle\mathbf{P}\otimes\boldsymbol{\nu}|\hat{H} - E|\mathbf{P}\otimes\boldsymbol{\nu}\rangle \end{bmatrix} \begin{bmatrix} a_0 \\ \mathbf{Q} \end{bmatrix} = 0 \tag{5.10}$$

for the $\mathbf{Q}$ optimization, where $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$ are unit vectors. As previously described, this approach provides fast convergence to within mH accuracy of the FCI energy. Unfortunately, this formulation is prone to spin contamination[77], and only singlet states may be described. By modifying our ansatz to be a linear combination of products of $\mathbf{P}$ and $\mathbf{Q}$ vectors we can avoid singlet-triplet spin contamination while simultaneously allowing calculation of triplet states.

$$|\mathbf{\Psi}_i\rangle = |\mathbf{P}_i \otimes \mathbf{Q}_i \pm \mathbf{Q}_i \otimes \mathbf{P}_i\rangle \tag{5.11}$$

where the singlet corresponds to the symmetric case (+) and the triplet to the antisymmetric case (−). While wavefunction symmetry/antisymmetry alone does not guarantee that the final eigenvector is an eigenfunction of $\hat{S}^2$ (quintet states can in principle contaminate the singlet states, for example), we find that in practice this is not a serious issue and we have observed no cases in our tests where spin contamination of this type occurred. The coupled equations for the initial set of product space vectors then become

$$\Big(\langle \boldsymbol{\mu} \otimes \mathbf{Q} \pm \mathbf{Q} \otimes \boldsymbol{\mu}|\hat{H} - E|\boldsymbol{\mu} \otimes \mathbf{Q} \pm \mathbf{Q} \otimes \boldsymbol{\mu}\rangle\Big)\,\mathbf{P} = 0 \tag{5.12}$$

$$\Big(\langle \mathbf{P} \otimes \boldsymbol{\nu} \pm \boldsymbol{\nu} \otimes \mathbf{P}|\hat{H} - E|\mathbf{P} \otimes \boldsymbol{\nu} \pm \boldsymbol{\nu} \otimes \mathbf{P}\rangle\Big)\,\mathbf{Q} = 0 \tag{5.13}$$

giving the eigenvalue problem

$$\begin{bmatrix} \langle \mathbf{\Psi}|\hat{H} - E|\mathbf{\Psi}\rangle & \langle \mathbf{\Psi}|\hat{H} - E|\boldsymbol{\mu} \otimes \mathbf{Q} \pm \mathbf{Q} \otimes \boldsymbol{\mu}\rangle \\ \langle \boldsymbol{\mu} \otimes \mathbf{Q} \pm \mathbf{Q} \otimes \boldsymbol{\mu}|\hat{H} - E|\mathbf{\Psi}\rangle & \langle \boldsymbol{\mu} \otimes \mathbf{Q} \pm \mathbf{Q} \otimes \boldsymbol{\mu}|\hat{H} - E|\boldsymbol{\mu} \otimes \mathbf{Q} \pm \mathbf{Q} \otimes \boldsymbol{\mu}\rangle \end{bmatrix} \begin{bmatrix} a_0 \\ \mathbf{P} \end{bmatrix} = 0 \tag{5.14}$$

for the $\mathbf{P}$ optimization and

$$\begin{bmatrix} \langle \mathbf{\Psi}|\hat{H} - E|\mathbf{\Psi}\rangle & \langle \mathbf{\Psi}|\hat{H} - E|\mathbf{P} \otimes \boldsymbol{\nu} \pm \boldsymbol{\nu} \otimes \mathbf{P}\rangle \\ \langle \mathbf{P} \otimes \boldsymbol{\nu} \pm \boldsymbol{\nu} \otimes \mathbf{P}|\hat{H} - E|\mathbf{\Psi}\rangle & \langle \mathbf{P} \otimes \boldsymbol{\nu} \pm \boldsymbol{\nu} \otimes \mathbf{P}|\hat{H} - E|\mathbf{P} \otimes \boldsymbol{\nu} \pm \boldsymbol{\nu} \otimes \mathbf{P}\rangle \end{bmatrix} \begin{bmatrix} a_0 \\ \mathbf{Q} \end{bmatrix} = 0 \tag{5.15}$$

for the $\mathbf{Q}$ optimization. In addition to the $\mathbf{P}$ and $\mathbf{Q}$ vectors themselves, the coefficient a_0 couples the two eigenvalue equations. The full wavefunction for the system can then be defined as

$$|\mathbf{\Psi}\rangle = \sum_i c_i \left(\mathbf{P}_i \otimes \mathbf{Q}_i \pm \mathbf{Q}_i \otimes \mathbf{P}_i\right) \tag{5.16}$$

where the c_j coefficients scale each $\mathbf{P}, \mathbf{Q}$ pair that comprise the wavefunction.

5.3.2 Projected $\boldsymbol{\sigma}$ Formation

Since our factorization scheme relies on separation of the α and β components of the CI vector, a reasonable starting point for $\boldsymbol{\sigma}$ formation is Olsen's CI equations[128], where $\boldsymbol{\sigma}$ is factorized into three terms

$$\boldsymbol{\sigma} = \boldsymbol{\sigma_1} + \boldsymbol{\sigma_2} + \boldsymbol{\sigma_3} \tag{5.17}$$

with

$$\boldsymbol{\sigma_1}[\alpha,\beta] = \sum_{\alpha'}\sum_{kl}^{m} \gamma_{kl}^{\alpha} h'_{kl} \mathbf{C}[\alpha',\beta] + \frac{1}{2}\sum_{\alpha'}\sum_{ijkl}^{m} \gamma_{ij}^{\alpha}\gamma_{kl}^{\alpha}(ij|kl)\mathbf{C}[\alpha',\beta] \tag{5.18}$$

$$\boldsymbol{\sigma_2}[\alpha,\beta] = \sum_{\beta'}\sum_{kl}^{m} \gamma_{kl}^{\beta} h'_{kl} \mathbf{C}[\alpha,\beta'] + \frac{1}{2}\sum_{\beta'}\sum_{ijkl}^{m} \gamma_{ij}^{\beta}\gamma_{kl}^{\beta}(ij|kl)\mathbf{C}[\alpha,\beta'] \tag{5.19}$$

$$\boldsymbol{\sigma_3}[\alpha,\beta] = \sum_{\alpha'\beta'}\sum_{ijkl}^{m} \gamma_{ij}^{\beta}\gamma_{kl}^{\alpha}(ij|kl)\mathbf{C}[\alpha',\beta'] \tag{5.20}$$

where $\alpha, \alpha', \beta, \beta'$ index occupation strings and γ are the 1-particle coupling coefficients. Note that we have combined the one-electron part of the two-electron integrals $(ij|kl)$ with the one-electron integrals h_{kl} for computational efficiency

$$h'_{kl} = h_{kl} - \sum_j (kj|jl) \tag{5.21}$$

Linear transformation of $\boldsymbol{\sigma}$ holding either $\mathbf{P}$ or $\mathbf{Q}$ constant produces terms according to

$$\boldsymbol{\sigma}^{\mathbf{P}}_{\alpha'} = \sum_{\alpha\beta\beta'} \mathbf{Q}^T_{\beta'} \left(\hat{H} | \mathbf{Q}_\beta \mathbf{P}^T_\alpha \rangle \right) \tag{5.22}$$

$$\boldsymbol{\sigma}^{\mathbf{Q}}_{\beta'} = \sum_{\alpha\alpha'\beta} \mathbf{P}^T_{\alpha'} \left(\hat{H} | \mathbf{P}_\alpha \mathbf{Q}^T_\beta \rangle \right) \tag{5.23}$$

Note that we only present the projection for the asymmetric wavefunction. Obtaining the (anti)symmetric projected $\boldsymbol{\sigma}$ is straightforward by interchanging $\mathbf{P}$ and $\mathbf{Q}$. Factorizing the projected $\boldsymbol{\sigma}$ results in 3 terms

$$\begin{aligned} \boldsymbol{\sigma}^{\mathbf{P}}_{\mathbf{1}} &= \sum_{\alpha\beta\beta'} \mathbf{Q}^T_{\beta'} \left(\sum_{kl} \gamma^\alpha_{kl} h'_{kl} + \frac{1}{2} \sum_{ijkl} \gamma^\alpha_{ij} \gamma^\alpha_{kl} (ij|kl) \right) \mathbf{Q}_\beta \mathbf{P}^T_\alpha \\ &= \left(\sum_\beta \mathbf{Q}^T_\beta \mathbf{Q}_\beta \right) \sum_\alpha \sum_{kl} \gamma^\alpha_{kl} h'_{kl} \mathbf{P}^T_\alpha + \frac{1}{2} \sum_{ijkl} \gamma^\alpha_{ij} \gamma^\alpha_{kl} (ij|kl) \mathbf{P}^T_\alpha \end{aligned} \tag{5.24}$$

$$\begin{aligned} \boldsymbol{\sigma}^{\mathbf{P}}_{\mathbf{2}} &= \sum_{\alpha\beta\beta'} \mathbf{Q}^T_{\beta'} \left(\sum_{kl} \gamma^\beta_{kl} h'_{kl} + \frac{1}{2} \sum_{ijkl} \gamma^\beta_{ij} \gamma^\beta_{kl} (ij|kl) \right) \mathbf{Q}_\beta \mathbf{P}^T_\alpha \\ &= \sum_{\beta\beta'} \mathbf{Q}^T_{\beta'} \left(\sum_{kl} \gamma^\beta_{kl} h'_{kl} + \frac{1}{2} \sum_{ijkl} \gamma^\beta_{ij} \gamma^\beta_{kl} (ij|kl) \right) \mathbf{Q}_\beta \sum_\alpha \mathbf{P}^T_\alpha \end{aligned} \tag{5.25}$$

$$\begin{aligned} \boldsymbol{\sigma}^{\mathbf{P}}_{\mathbf{3}} &= \sum_{\alpha\beta\beta'} \mathbf{Q}^T_{\beta'} \sum_{ijkl} \gamma^\beta_{ij} \gamma^\alpha_{kl} (ij|kl) \mathbf{Q}_\beta \mathbf{P}^T_\alpha \\ &= \sum_{ijkl} \left(\sum_{\beta\beta'} \mathbf{Q}^T_{\beta'} \gamma^\beta_{ij} \mathbf{Q}_\beta \right) \left(\sum_\alpha \gamma^\alpha_{kl} \mathbf{P}^T_\alpha \right) (ij|kl) \end{aligned} \tag{5.26}$$

for the $\mathbf{P}$ factorization and

$$\begin{aligned}
\boldsymbol{\sigma}_{\mathbf{1}}^{\mathbf{Q}} &= \sum_{\alpha\alpha'\beta} \mathbf{P}_{\alpha'}^{T} \sum_{kl} \gamma_{kl}^{\alpha} h'_{kl} + \frac{1}{2} \sum_{ijkl} \gamma_{ij}^{\alpha} \gamma_{kl}^{\alpha} (ij|kl) \mathbf{P}_{\alpha} \mathbf{Q}_{\beta}^{T} \\
&= \left(\sum_{\alpha\alpha'} \mathbf{P}_{\alpha}^{T} \sum_{kl} \gamma_{kl}^{\alpha} h'_{kl} + \frac{1}{2} \sum_{ijkl} \gamma_{ij}^{\alpha} \gamma_{kl}^{\alpha} (ij|kl) \mathbf{P}_{\alpha} \right) \sum_{\beta} \mathbf{Q}_{\beta}^{T}
\end{aligned} \tag{5.27}$$

$$\begin{aligned}
\boldsymbol{\sigma}_{\mathbf{2}}^{\mathbf{Q}} &= \sum_{\alpha\alpha'\beta} \mathbf{P}_{\alpha'}^{T} \sum_{kl} \gamma_{kl}^{\beta} h'_{kl} + \frac{1}{2} \sum_{ijkl} \gamma_{ij}^{\beta} \gamma_{kl}^{\beta} (ij|kl) \mathbf{P}_{\alpha} \mathbf{Q}_{\beta}^{T} \\
&= \left(\sum_{\alpha} \mathbf{P}_{\alpha}^{T} \mathbf{P}_{\alpha} \right) \sum_{\beta} \sum_{kl} \gamma_{kl}^{\beta} h'_{kl} + \frac{1}{2} \sum_{ijkl} \gamma_{ij}^{\beta} \gamma_{kl}^{\beta} (ij|kl) \mathbf{Q}_{\beta}^{T}
\end{aligned} \tag{5.28}$$

$$\begin{aligned}
\boldsymbol{\sigma}_{\mathbf{3}}^{\mathbf{Q}} &= \sum_{\alpha\alpha'\beta} \mathbf{P}_{\alpha'}^{T} \sum_{ijkl} \gamma_{ij}^{\beta} \gamma_{kl}^{\alpha} (ij|kl) \mathbf{P}_{\alpha} \mathbf{Q}_{\beta}^{T} \\
&= \sum_{ijkl} \left(\sum_{\alpha\alpha'} \mathbf{P}_{\alpha'}^{T} \gamma_{ij}^{\alpha} \mathbf{P}_{\alpha} \right) \left(\sum_{\beta} \gamma_{kl}^{\alpha} \mathbf{Q}_{\beta}^{T} \right) (ij|kl)
\end{aligned} \tag{5.29}$$

for the $\mathbf{Q}$ terms. We forgo the distinction between α and β labels for the coupling coefficients γ_{ij} for the remainder of this work. For each projected $\boldsymbol{\sigma}$ formation three state vectors are required in addition to the current trial vector. For the $\mathbf{P}$ optimization, for example, the current $\mathbf{Q}$ vector is needed in addition to each of the previously converged $\mathbf{P}$ and $\mathbf{Q}$ vectors.

5.3.3 Computational Methods

5.3.3.1 Coupling Coefficient Formation

Data structures required for evaluation of $\boldsymbol{\sigma}$ include the one-particle coupling coefficients $\boldsymbol{\gamma}$, the orbital labels $\mathbf{l}$, and the configuration labels $\mathbf{IJ}$. Our program constructs these lists at the beginning of the calculation rather than computing elements on the fly as needed. While forming these lists for configuration spaces typical of direct CI calculations does not require exceptional computational effort, we discovered that extremely large rrFCI spaces require exceptionally large lists resulting in unreasonable computation times. We have developed an efficient GPU accelerated algorithm for construction of the 1-particle coupling coefficients and related data structures in a dense format.

Algorithm 7 GPU accelerated coupling coefficient $\boldsymbol{\gamma}$, orbital label $\mathbf{l}$, and configuration label **IJ** formation. **bin_rep** is a precomputed input array of integers defining the orbital occupations. *stradr*() is a function that determines the string address using lexical ordering as defined by Knowles and Handy[73] (Equations 11,12), and *get_parity*() is a function that determines the number of particle swaps required to bring the configurations I and J into maximum coincidence. **occ** and **vac** are compressed lists of occupied and vacant orbitals for a given string.

```
GPU vectorize over strings I
    for m < N_o do
        Decompose bin_rep[I] into arrays occ[] and vac[]
    end for
    counter = 0
    for m < N_e do
        work = bin_rep[I]
        work -= 2^occ[m]
        for n < N_o − N_e do
            work += 2^vac[n]                          ▷ Add a particle to orbital n
            IJ[I][counter] ← stradr()                 ▷ Get string address
            work -= 2^vac[n]                          ▷ Remove a particle from orbital n
            l[I][counter] = occ[m] * N_o + vac[n]
            γ[I][counter] ← get_parity()              ▷ Get the parity of the excitation
            counter += 1
        end for
    end for
end GPU vectorize
```

The outer GPU vectorized loop of Algorithm 7 can be tiled over to allow for large configuration spaces and hierarchical vectorization over multiple GPUs and/or compute nodes. The *stradr*() function is an implementation of Equations 11 and 12 from Knowles and Handy [73], where the binomial coefficients are retrieved from a precomputed list (i.e. Pascal's triangle) rather than calculated on the fly for efficiency purposes. The *get_parity* function is an abstraction that can be implemented trivially in several ways (we compute the value inline). This algorithm produces dense data structures in the minimum number of operations, $(N_s * N_e(N_o - N_e))$, where N_e and N_o are the number of α electrons and orbitals in the configuration space, respectively. Even so, larger configuration spaces require storage on the order of hundreds of GB. For the reader's convenience, Table 5.1 provides configuration space and array sizes for some of the CAS spaces used in this study.

Table 5.1: Configuration space sizes and memory requirements for supporting data structures (in GB). Note that storage for $\mathbf{P}, \mathbf{Q}$, $\boldsymbol{\sigma}$, and Davidson trial vectors have been omitted.

CAS Size	Strings	Memory Usage (GB)
(18,18)	48,620	< 0.1
(20,20)	184,756	0.2
(22,22)	705,432	1.1
(24,24)	2,704,156	4.8
(26,26)	10,400,600	21.6
(28,28)	40,116,600	94.7
(30,30)	155,117,520	428.1

5.3.3.2 Eigenvalue Problem

Our program solves a nonlinear eigenvalue problem using a two-step Davidson solver, where iterations occur between $\mathbf{P}$ and $\mathbf{Q}$ vector optimization until they are self-consistently converged. The inner (micro) iterations were performed using a generalized Davidson algorithm to solve the non-orthonormal subspace eigenvalue problem. The outer (macro) iterations are considered converged when the micro-iterations for each of the $\mathbf{P}$ and $\mathbf{Q}$ vectors converge in a single step. We "balance" the eigenvalue problem by scaling the projected metric matrix as recommended by Parrish et al.[132] and by Furche et al.[47]. A residual norm convergence criteria of $||r|| = 1.0 \times 10^{-6}$ was used for the micro-iterations.

To begin the iterative procedure for the $\mathbf{P}$ and $\mathbf{Q}$ vector optimizations we must supply a guess vector for each. In the first macro-iteration, the guess for each of $\mathbf{P}_0$ and $\mathbf{Q}_0$ is the vector corresponding to the Hartree-Fock determinant, i.e. $\langle 0, 1, 0, 0, 0, ... \rangle$, where the first vector element corresponds to the coefficient a_0 that couples the $\mathbf{P}$ and $\mathbf{Q}$ eigenvalue problems. In the second and subsequent macro-iterations the guess for the $\mathbf{P}$ vector becomes $\langle 1, 0, 0, 0, 0, ... \rangle$, which states that the previously converged $\mathbf{P}, \mathbf{Q}$ pairs are used as the $\mathbf{P}$ guess, and the $\mathbf{Q}$ guess is a unit vector in the direction of the configuration corresponding to the current macro-iteration, i.e. if we are adding the third $\mathbf{P}, \mathbf{Q}$ pair, the $\mathbf{Q}$ guess is $\langle 0, 0, 0, 1, 0, ... \rangle$. This permits efficient sampling of the configuration space while avoiding linear dependence of the added trial vectors. Previous work[77] suggested using a uniform

$\mathbf{Q}$ guess, i.e. $\langle 0, 1/\sqrt{N_s}, 1/\sqrt{N_s}, 1/\sqrt{N_s}, 1/\sqrt{N_s}, ...\rangle$. Additionally, $\mathbf{Q}$ vector guesses derived from projected residual vector formation were investigated in the present work. Interestingly, the unit guess sampling the configuration space results in the best convergence, with the other approaches producing linearly dependent trial vectors after mH accuracy was obtained.

Preconditioning of the Davidson routine was performed using the reference determinant energy modified by orbital energy differences as described originally by Evangelisti[18]. An advantage of orbital energy difference preconditioning is that no spin contamination is introduced into the trial vectors, and convergence of the micro-iterations generally requires $10-30$ iterations. We have reconstructed the final wavefunction $\mathbf{\Psi}$ for a variety of manageable configuration spaces (i.e. (16,16) and smaller) by taking the linear combination of $\mathbf{P}, \mathbf{Q}$ outer products according to Equation 5.16 to evaluate the extent of which spin contamination occurs in both our ansatz as well as in the original form given by Koch and Dalgaard[77]. The spin purity was determined for both $\mathbf{\Psi}$ and for each contributing $\mathbf{P}, \mathbf{Q}$ pair through direct calculation of $\langle \hat{S}^2 \rangle$. In this way we were able to quantify the spin contamination present in the ansatz described in Equation 5.5 while simultaneously verifying the integrity of the (anti)symmetric form.

5.3.3.3 Metric Formation

Since we are solving a generalized eigenvalue problem it is necessary to build the subspace metric matrix. We form the projected metric according to

$$\mathbf{B} = \begin{pmatrix} \langle \mathbf{\Psi} | \mathbf{\Psi} \rangle & \langle \mathbf{\Psi} | \boldsymbol{\mu} \otimes \mathbf{Q} \pm \mathbf{Q} \otimes \boldsymbol{\mu} \rangle \\ \langle \boldsymbol{\mu} \otimes \mathbf{Q} \pm \mathbf{Q} \otimes \boldsymbol{\mu} | \mathbf{\Psi} \rangle & \langle \boldsymbol{\mu} \otimes \mathbf{Q} \pm \mathbf{Q} \otimes \boldsymbol{\mu} | \boldsymbol{\mu} \otimes \mathbf{Q} \pm \mathbf{Q} \otimes \boldsymbol{\mu} \rangle \end{pmatrix} \begin{pmatrix} a_0 \\ \mathbf{P} \end{pmatrix} \tag{5.30}$$

for the $\mathbf{P}$ vector optimization and

$$\mathbf{B} = \begin{pmatrix} \langle \mathbf{\Psi} | \mathbf{\Psi} \rangle & \langle \mathbf{\Psi} | \mathbf{P} \otimes \boldsymbol{\nu} \pm \boldsymbol{\nu} \otimes \mathbf{P} \rangle \\ \langle \mathbf{P} \otimes \boldsymbol{\nu} \pm \boldsymbol{\nu} \otimes \mathbf{P} | \mathbf{\Psi} \rangle & \langle \mathbf{P} \otimes \boldsymbol{\nu} \pm \boldsymbol{\nu} \otimes \mathbf{P} | \mathbf{P} \otimes \boldsymbol{\nu} \pm \boldsymbol{\nu} \otimes \mathbf{P} \rangle \end{pmatrix} \begin{pmatrix} a_0 \\ \mathbf{Q} \end{pmatrix} \tag{5.31}$$

for the **Q** vector optimization. The algorithm for computing the metric is low enough scaling that we have implemented it on the host (CPU). Details are given in Algorithm 8.

Algorithm 8 Metric formation for **P** vector optimization. N_t is the number of converged product terms. **B** is the projected metric vector (output) of dimension $N_s + 1$, **c** are the coefficients of the previously converged **P** and **Q** (input), and **C** is the trial vector for the augmented eigenvalue problem (input).

for $i, j < N_t$ **do**
  $\mathbf{B}[0] \mathrel{+}= 2.0 * \mathbf{c}_i * \mathbf{c}_j * \left(\mathbf{P}_i \cdot \mathbf{P}_j\right)\left(\mathbf{Q}_i \cdot \mathbf{Q}_j\right)$ ▷ Upper left block of metric
  $\mathbf{B}[0] \pm= 2.0 * \mathbf{c}_i * \mathbf{c}_j * \left(\mathbf{P}_i \cdot \mathbf{Q}_j\right)\left(\mathbf{Q}_i \cdot \mathbf{P}_j\right)$
end for
Allocate work$[N_S]$
for $i < N_t$ **do**
  for $j < N_S$ **do**
    $\mathbf{work}[j] \mathrel{+}= 2.0 * \mathbf{c}_i * \left(\mathbf{Q}_t \cdot \mathbf{Q}_i\right) * \mathbf{P}_i[j]$
    $\mathbf{work}[j] \pm= 2.0 * \mathbf{c}_i * \left(\mathbf{Q}_t \cdot \mathbf{P}_i\right) * \mathbf{Q}_i[j]$
  end for
end for
$\mathbf{B}[0] \mathrel{+}= (\mathbf{work} \cdot \mathbf{C})$ ▷ Upper right block of metric
for $i < N_S$ **do**
  $\mathbf{B}[i+1] \mathrel{+}= a_0 * \mathbf{work}[i]$ ▷ Bottom left block of metric
end for
for $i < N_S$ **do**
  $\mathbf{B}[i+1] \mathrel{+}= 2.0 * (\mathbf{Q}_t \cdot \mathbf{Q}_t) * \mathbf{C}[i]$ ▷ Bottom right block of metric
  $\mathbf{B}[i+1] \pm= 2.0 * (\mathbf{Q}_t \cdot \mathbf{C}) * \mathbf{Q}_t[i]$
end for

Projected metric formation requires double precision dot product (DDOT) and $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{y}$ (DAXPY) operations, both of which are performed using highly optimized library routines[64].

5.3.3.4 Projected Sigma Formation

We have developed projected $\boldsymbol{\sigma}$ formation algorithms based on a hybrid of the Olsen factorization[128] and the Knowles and Handy vector machine FCI algorithm[73]. Algorithms $9 - 11$ describe **P** vector optimization projected $\boldsymbol{\sigma}$ formation using an orbital label driven method, directly analogous to the original FCI approach described by Olsen.

Algorithm 9 Factorized $\boldsymbol{\sigma_1}$ formation algorithm.

$Q = \pm(\mathbf{Q_l} \cdot \mathbf{Q_r})$
for $k, l < N_O$ **do**
  $\mathbf{F} \leftarrow 0.0$ ▷ Work array of size N_S
  for $I < N_S$ **do**
    $\alpha \leftarrow \mathbf{ij\_to\_I}[k, l, I]$
    $\alpha' \leftarrow \mathbf{ij\_to\_J}[k, l, I]$
    $\mathbf{F}[\alpha] \leftarrow \boldsymbol{\gamma}[k][l] * \mathbf{P_r}[\alpha']$
  end for
  $\boldsymbol{\sigma}$ += $Q * \mathbf{h'_{kl}}[k][l] * \mathbf{F}$ ▷ (DAXPY)
  for $i, j < N_O$ **do**
    for $I < N_S$ **do**
      $\alpha \leftarrow \mathbf{ij\_to\_I}[k, l, I]$
      $\alpha' \leftarrow \mathbf{ij\_to\_J}[k, l, I]$
      $\boldsymbol{\sigma}[\alpha']$ += $0.5 * Q * \boldsymbol{\gamma}[i][j] * (\mathbf{ij}|\mathbf{kl}) * \mathbf{F}[\alpha]$
    end for
  end for
end for

Algorithm 10 Factorized $\boldsymbol{\sigma_2}$ formation algorithm.

for $k, l < N_O$ **do**
  $\mathbf{F} \leftarrow 0.0$ ▷ Work array of size N_S
  for $I < N_S$ **do**
    $\alpha \leftarrow \mathbf{ij\_to\_I}[k, l, I]$
    $\alpha' \leftarrow \mathbf{ij\_to\_J}[k, l, I]$
    $\mathbf{F}[\alpha] \leftarrow \boldsymbol{\gamma}[k][l] * \mathbf{Q_r}[\alpha']$
  end for
  $\mathbf{s}$ += $\mathbf{h'_{kl}}[k][l] * \mathbf{F}$ ▷ (DAXPY)
  for $i, j < N_O$ **do**
    for $I < N_S$ **do**
      $\alpha \leftarrow \mathbf{ij\_to\_I}[k, l, I]$
      $\alpha' \leftarrow \mathbf{ij\_to\_J}[k, l, I]$
      $\mathbf{s}[\alpha']$ += $0.5 * \boldsymbol{\gamma}[i][j] * (\mathbf{ij}|\mathbf{kl}) * \mathbf{F}[\alpha]$
    end for
  end for
end for
$PQ = \pm(\mathbf{P_r} \cdot \mathbf{s})$
$\boldsymbol{\sigma}$ += $PQ * \mathbf{Q_l}$ ▷ (DAXPY)

Algorithm 11 Factorized $\boldsymbol{\sigma_3}$ formation algorithm.

for $k, l < N_O$ **do**
  $\mathbf{F} \leftarrow 0.0$ ▷ Work array of size N_S
  for $I < N_S$ **do**
    $\alpha \leftarrow$ **ij_to_I**$[k, l, I]$
    $\alpha' \leftarrow$ **ij_to_J**$[k, l, I]$
    $\mathbf{F}[\alpha] \leftarrow \boldsymbol{\gamma}[k][l] * \mathbf{Q_r}[\alpha']$
  end for
  $PQ = \pm(\mathbf{P_r} \cdot \mathbf{F})$
  for $i, j < N_O$ **do**
    for $I < N_S$ **do**
      $\alpha \leftarrow$ **ij_to_I**$[k, l, I]$
      $\alpha' \leftarrow$ **ij_to_J**$[k, l, I]$
      $\boldsymbol{\sigma}[\alpha]$ += $0.5 * PQ * \boldsymbol{\gamma}[i][j] * (\mathbf{ij}|\mathbf{kl}) * \mathbf{Q_l}[\alpha']$
    end for
  end for
end for

The $\mathbf{Q_l}$, $\mathbf{Q_r}$, and $\mathbf{P_r}$ correspond to the left $\mathbf{Q}$ vector, the right $\mathbf{Q}$ vector, and the $\mathbf{P}$ vector in Equation 5.22. Each of the $\mathbf{P} \otimes \mathbf{Q}$ and the $\mathbf{Q} \otimes \mathbf{P}$ term projected $\boldsymbol{\sigma}$ can be formed (for both $\mathbf{P}$ and $\mathbf{Q}$ vector optimizations) using these algorithms and interchanging the ordering and identities of the input vectors. In the above algorithms the $\boldsymbol{\gamma}$ are indexed by the orbital labels i, j, and the **ij_to_I** and **ij_to_J** matrices contain as elements the configuration string labels α and α'.

Vectorized algorithms require contiguous access of data structures to avoid taking costly performance hits. For example, in Algorithm 9, elements of the $\boldsymbol{\gamma}$ matrix are accessed contiguously, and even though both the $\mathbf{P_r}$ and $\boldsymbol{\sigma}$ vectors are accessed randomly, gathering of the $\mathbf{P_r}$ elements into the $\mathbf{F}$ matrix allows the higher scaling contraction with the two-electron integrals $(\mathbf{ij}|\mathbf{kl})$ to be performed using coalesced memory access patterns. Gather-scatter techniques are also used in Algorithm 10 to reduce memory pressure. The highest scaling operation in $\boldsymbol{\sigma_3}$ formation does not benefit from contiguous memory access, however, where the $(\mathbf{ij}|\mathbf{kl})$ are contracted directly with the $\mathbf{Q_l}$ vector (Algorithm 11). Efforts to improve access patterns here do not translate well to GPU hardware, where repeated manipulation of arrays must be performed to avoid strided memory access. Instead, we present a configuration label driven scheme for formation of the projected $\boldsymbol{\sigma}$ vector for the $\mathbf{P}$ vector optimization

in in Algorithms 12 − 14.

Algorithm 12 GPU vectorized projected $\boldsymbol{\sigma_1}$ algorithm

$Q = \pm(\mathbf{Q_l} \cdot \mathbf{Q_r})$
GPU vectorize over strings α, α' differing from α by zero or one occupations
  $ij \leftarrow \mathbf{l}[\alpha][\alpha']$
  $\mathbf{D}[ij, \alpha]$ += $\boldsymbol{\gamma}[\alpha][\alpha']\mathbf{Pr}[\alpha']$
end GPU vectorize
GPU vectorize over strings α
  $ij \leftarrow \mathbf{l}[\alpha][\alpha]$
  $\mathbf{D}[ij, \alpha]$ += $\mathbf{P_r}[\alpha]$
end GPU vectorize
$\boldsymbol{\sigma}$ += $Q * \mathbf{D} * \mathbf{h'_{kl}}$ ▷ (DGEMV)
$\mathbf{E} \leftarrow 0.5 * Q * (\mathbf{ij}|\mathbf{kl}) * \mathbf{D}$ ▷ (DGEMM)
GPU vectorize over strings α, α' differing from α by zero or one occupations
  $ij \leftarrow \mathbf{l}[\alpha][\alpha']$
  $\boldsymbol{\sigma}[\alpha']$ += $\boldsymbol{\gamma}[\alpha][\alpha']\mathbf{E}[ij, \alpha]$
end GPU vectorize
GPU vectorize over strings α
  $ij \leftarrow \mathbf{l}[\alpha][\alpha]$
  $\boldsymbol{\sigma}[\alpha]$ += $\mathbf{E}[ij, \alpha]$
end GPU vectorize

Algorithm 13 GPU vectorized projected $\boldsymbol{\sigma_2}$ algorithm

GPU vectorize over strings α, α' differing from α by zero or one occupations
  $ij \leftarrow \mathbf{l}[\alpha][\alpha']$
  $\mathbf{D}[ij, \alpha]$ += $\boldsymbol{\gamma}[\alpha][\alpha']\mathbf{Q_r}[\alpha']$
end GPU vectorize
GPU vectorize over strings α
  $ij \leftarrow \mathbf{l}[\alpha][\alpha]$
  $\mathbf{D}[ij, \alpha]$ += $\mathbf{Q}_r[\alpha]$
end GPU vectorize
$\mathbf{s}$ += $\mathbf{D} * \mathbf{h'_{kl}}$ ▷ (DGEMV)
$\mathbf{E} \leftarrow 0.5 * (\mathbf{ij}|\mathbf{kl}) * \mathbf{D}$ ▷ (DGEMM)
GPU vectorize over strings α, α' differing from α by zero or one occupations
  $ij \leftarrow \mathbf{l}[\alpha][\alpha']$
  $\mathbf{s}[\alpha']$ += $\boldsymbol{\gamma}[\alpha][\alpha']\mathbf{E}[ij, \alpha]$
end GPU vectorize
GPU vectorize over strings α
  $ij \leftarrow \mathbf{l}[\alpha][\alpha]$
  $\mathbf{s}[\alpha']$ += $\mathbf{E}[ij, \alpha]$
end GPU vectorize
$PQ = \pm(\mathbf{P_r} \cdot \mathbf{s})$
$\boldsymbol{\sigma}$ += $PQ * \mathbf{Q_l}$ ▷ (DAXPY)

Algorithm 14 GPU vectorized projected $\boldsymbol{\sigma_3}$ algorithm.

GPU vectorize over strings α, α' differing from α by zero or one occupations
  $ij \leftarrow \mathbf{l}[\alpha][\alpha']$
  $\mathbf{D}[ij, \alpha] \mathrel{+}= \boldsymbol{\gamma}[\alpha][\alpha']\mathbf{Q_r}[\alpha']$
end GPU vectorize
GPU vectorize over strings α
  $ij \leftarrow \mathbf{l}[\alpha][\alpha]$
  $\mathbf{D}[ij, \alpha] \mathrel{+}= \mathbf{Q_r}[\alpha]$
end GPU vectorize
$\mathbf{f} \mathrel{+}= \mathbf{D} * \mathbf{P_r}$ ▷ (DGEMV)
$\mathbf{D} \leftarrow 0.0$
GPU vectorize over strings α, α' differing from α by zero or one occupations
  $ij \leftarrow \mathbf{l}[\alpha][\alpha']$
  $\mathbf{D}[ij, \alpha] \mathrel{+}= \boldsymbol{\gamma}[\alpha][\alpha']\mathbf{Q_l}[\alpha']$
end GPU vectorize
GPU vectorize over strings α
  $ij \leftarrow \mathbf{l}[\alpha][\alpha]$
  $\mathbf{D}[ij, \alpha] \mathrel{+}= \mathbf{Q_l}[\alpha]$
end GPU vectorize
$\mathbf{E} \leftarrow (\mathbf{ij}|\mathbf{kl}) * \mathbf{D}$ ▷ (DGEMM)
$\boldsymbol{\sigma} \mathrel{+}= \mathbf{E} * \mathbf{f}$ ▷ (DGEMV)

The one-particle coupling coefficient matrix $\boldsymbol{\gamma}$ and orbital labels $\mathbf{l}$ are indexed by configurations in Algorithms 12 − 14. In contrast to the orbital driven formulation, rate-limiting two-electron integral contractions are performed using matrix-matrix multiplication to maximize data locality. Improved memory access comes at the cost of an increase in scaling, but this is more than offset by the overall savings achieved by taking advantage of high-performance vectorized matrix multiplication.

The GPU kernels in the above algorithms bear striking resemblance to those described in our previous work on direct CI[40]. By forming the projected $\boldsymbol{\sigma}$ vector directly we benefit from a significant reduction in scaling: in rrFCI we are able to eliminate the loop over β strings entirely. Further, direct CI kernels that contribute to the $\boldsymbol{\sigma}$ array must use atomic operations to avoid data collisions, the effect of which is harmful to computational performance. The rrFCI projected $\boldsymbol{\sigma}$ algorithm described above is able to restrict atomic operations to the $\boldsymbol{\sigma_1}$ term, using DAXPY and DGEMV operations for the $\boldsymbol{\sigma_2}$ and $\boldsymbol{\sigma_3}$ terms. We neglect description of our tiling scheme, which allows configuration spaces of arbitrary size (subject to memory constraints) for clarity. Due to the high cost associated with CPU host to

GPU device memory transfers we select the largest tile size possible (limited by GPU device memory) to minimize incurring memory transfer penalties.

5.4 Results and Discussion

5.4.1 Computational Details

The rrFCI method was implemented in the TeraChem GPU accelerated electronic structure package using the Compute Unified Device Architecture (CUDA) API[125] and the NVidia CUDA Basic Linear Algebra Subroutines library (cuBLAS)[124]. Benchmark calculations were performed using a single core of an Intel Xeon E5-2699 @2.2GHz and a single NVidia K40c GPU. Configuration spaces are defined according to the notation (X, Y), where X corresponds to the number of electrons and Y corresponds to the number of orbitals in the configuration space. Geometries for all molecules used in this work are reported in Cartesian coordinates in the Supporting Information.

5.4.2 Algorithm Performance

Computational performance of $\boldsymbol{\sigma}$ vector formation depends solely on the number of active electrons and orbitals. Benchmark calculations were performed on an ethylene dimer system at HF-CAS-rrCI/cc-pVDZ using symmetric configuration spaces ranging from 18 through 30 orbitals. Results are presented in Figure 5.1 and in Table 5.2.

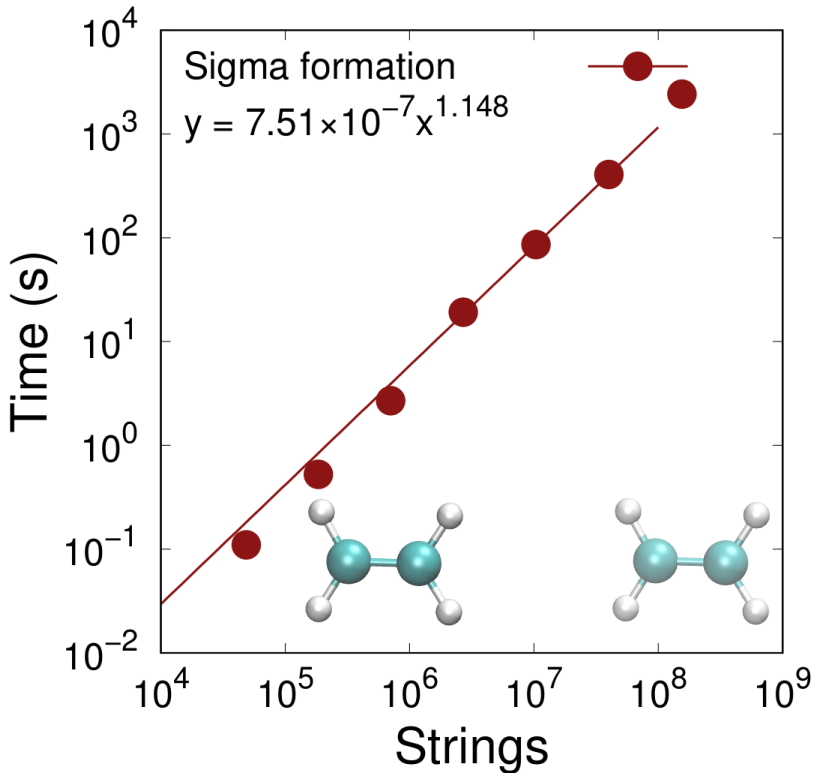

Figure 5.1: Sigma vector formation times in seconds for ethylene dimer benchmark system using HF-CAS-rrCI/cc-pVDZ and active spaces ranging from (18,18) through (30,30). Linear regression was performed to determine the scaling exponent and is shown as a solid line. The (30,30) active space data point was not included in the fit as described in the text.

Configuration spaces having fewer than 24 orbitals correspond to $N_s < S_{tile}$, where S_{tile} is the tile size, and no CPU—GPU memory transfer operations were necessary. Configuration spaces having between 24 and 28 orbitals exhibit similar scaling with a larger prefactor as evidenced by the rigidly shifted points relative to the fit line in Figure 5.1. The largest configuration space, (30,30) [$N_{string} = 155,117,520$], required a special, smaller tile size to accommodate the extremely large (> 1 GB) $\mathbf{P}, \mathbf{Q}$, and $\boldsymbol{\sigma}$ vector size. As a result, we did not include this data point in the fit as it is not representative of the algorithmic performance, instead it demonstrates the relatively high cost of CPU—GPU memory transfer. The linear fit to configuration spaces ranging from (18,18) [$N_{string} = 48,620$] through (28,28) [$N_{string} = 40,116,600$] gives a scaling factor of 1.148, similar to scalings reported for direct CI $\boldsymbol{\sigma}$ vector[40], two-particle reduced density matrix[41] and SA-CASSCF direct Hessian

matrix[166] formation scaling factors previously reported.

Table 5.2: Algorithm timings in seconds for configuration spaces ranging from (18,18) through (30,30). $\boldsymbol{\sigma}$ timings correspond to a combined total of 6 function calls, 2 each of $\boldsymbol{\sigma_1}$, $\boldsymbol{\sigma_2}$, $\boldsymbol{\sigma_3}$. Coupling coefficient and string label lists require formation only once per calculation, while metric and $\boldsymbol{\sigma}$ formation are required once per micro-iteration. Only functions that take non-negligible time are reported in the $\boldsymbol{\sigma}$ breakdown. Differences between the full $\boldsymbol{\sigma}$ time and the sum of the individual function calls correspond to CPU memory allocation and other low-scaling operations.

Function	**Active Space**						
	(18,18)	**(20,20)**	**(22,22)**	**(24,24)**	**(26,26)**	**(28,28)**	**(30,30)**
lists	0.026	0.111	0.501	2.350	10.338	46.038	2086.945
metric	<0.001	<0.001	0.004	0.025	0.119	0.454	1.759
$\boldsymbol{\sigma}$	0.110	0.525	2.694	19.180	86.087	406.627	2418.517
			$\boldsymbol{\sigma}$ components				
GPU memcpy()	<0.001	0.002	0.006	5.011	19.690	87.048	710.873
GPU memset()	<0.001	0.021	0.095	0.503	1.970	8.793	42.400
$\mathbf{D_{ij}}$	0.027	0.113	0.566	2.292	11.248	45.153	169.265
$\mathbf{D_{ii}}$	<0.001	0.003	0.013	0.052	0.231	0.900	12.370
$\sigma_{\mathbf{ij}}$	0.011	0.047	0.237	0.972	4.753	19.239	82.188
$\sigma_{\mathbf{ii}}$	<0.001	0.001	0.005	0.020	0.090	0.358	5.509
DGEMM	0.058	0.296	1.596	9.185	43.410	225.407	1207.285
DGEMV	0.007	0.036	0.154	0.802	3.341	14.183	80.479
DDOT	<0.001	<0.001	<0.001	0.001	0.004	0.014	0.057
DAXPY	<0.001	<0.001	<0.001	<0.001	0.003	0.010	0.039
ij lookup	<0.001	0.003	0.014	0.308	1.207	4.983	23.934

Table 5.2 provides timings for several rrFCI calculation components with configuration spaces ranging from (18,18) through (30,30). Coupling coefficient and other list formation requires less computational effort than $\boldsymbol{\sigma}$ formation in every case. To reduce the storage requirements for the largest lists we considered forming elements of these lists as needed during $\boldsymbol{\sigma}$ formation, but even neglecting CPU—GPU memory transfer overhead it is advantageous to precalculate and store these lists when possible. If larger configuration spaces than system memory allows for are desired these lists may be formed directly during the $\boldsymbol{\sigma}$ formation at a computational cost proportional to a standard $\boldsymbol{\sigma}$ formation.

Projected metric formation incurs only marginal computational expense for even the

largest configuration spaces. Instead, as expected, the rate-limiting step is $\boldsymbol{\sigma}$ formation. Timings for each function comprising $\boldsymbol{\sigma}$ formation show that for every configuration space the matrix-matrix multiply is the most expensive step, demonstrating that our algorithm is implemented efficiently. Memory transfer between the host (CPU) and device (GPU) becomes increasingly important with larger configuration spaces. Active spaces smaller than (24,24) show negligible contribution due to CPU—GPU memory transfer, as their data structures reside entirely in GPU memory and have dimension smaller than the tile size, but at configuration spaces of (24,24) and larger the high cost of CPU—GPU memory transfer becomes relevant, comprising nearly 30% of the overall cost of the projected $\boldsymbol{\sigma}$ formation for the largest active space. The GPU kernels corresponding to off-diagonal $\mathbf{D}$ and $\boldsymbol{\sigma}$ formation are the next most costly operations, followed by the remaining linear algebra operations (DGEMV, DDOT, DAXPY). Finally, as noted above, the largest configuration space required an extremely small tile size (1024), and this is reflected in both the list and the $\boldsymbol{\sigma}$ formation times accordingly.

5.4.3 Acene Absolute Energy Convergence

Linear polycyclic aromatic hydrocarbons, or acenes, possess interesting electronic properties making them suitable for incorporation into organic semiconductor materials[]. Correlation of the π valence electrons provides a reasonably accurate description of the electronic structure, making complete active space methods such as CASSCF or CASCI well-suited for these systems. We have calculated the ground singlet and triplet states for napthalene and anthracene using both HF-CASCI/cc-pVDZ and HF-CAS-rrCI/cc-pVDZ with full-π valence active spaces ((10,10) and (14,14) for naphthalene and anthracene, respectively) and report the energy differences between the rank-reduced and exact CI approaches in Figures 5.2 and 5.3. In addition to their interesting electronic characteristics, the calculations described in this section are examples of configuration spaces commonly referred to as "stout" CI, where the number of orbitals is similar to the number of electrons. UB3LYP/6-31G(d) singlet and

triplet optimized geometries from [52] were used in the present work and are available in the Supporting Information.

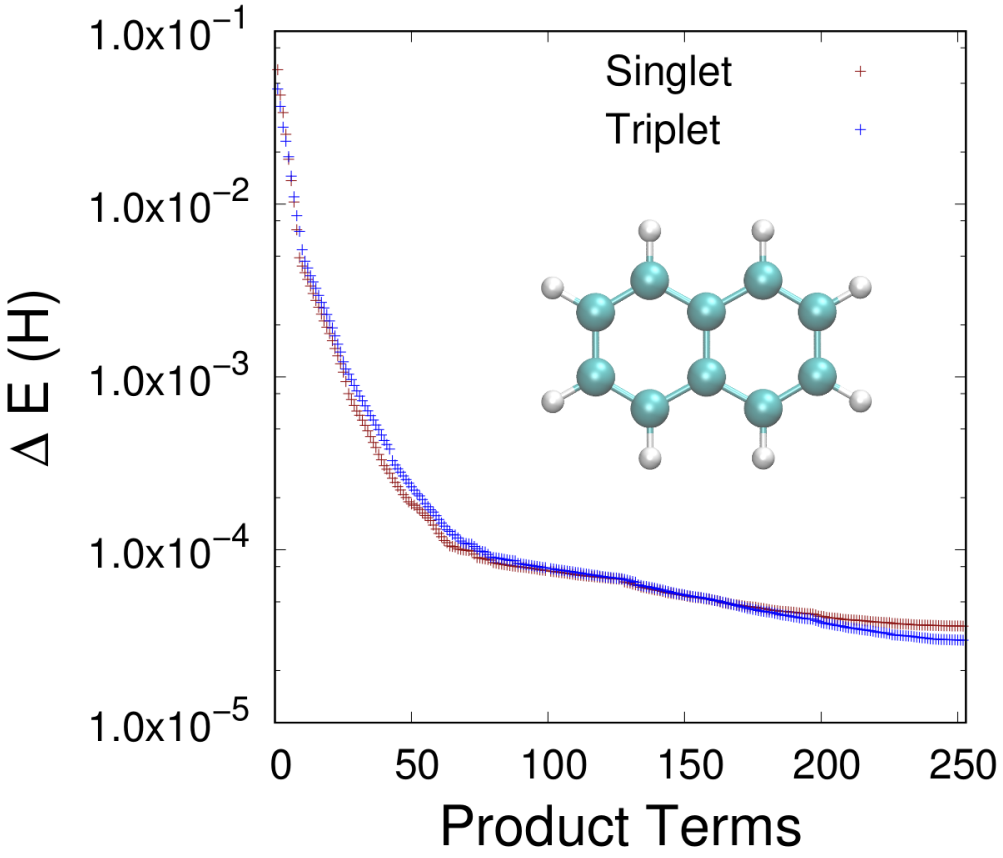

Figure 5.2: Naphthalene $E_{rrFCI}-E_{FCI}$ for singlet and triplet states at HF-CAS-(10,10)-CI/cc-pVDZ. The singlet optimized structure is depicted with carbon and hydrogen atoms shown in teal and white, respectively.

The rrFCI energy rapidly converges to within mH accuracy of the FCI energy after 30 product terms, representing inclusion of rank structure of the CI vector having the largest influence on the final energy. Once an accuracy of 1.0×10^{-4} H is obtained, convergence slows considerably and displays asymptotic behavior. Inclusion of 252 product states corresponds to the full rank of the CI vector, and we obtain a final accuracy approaching 1.0×10^{-5} H. Since we solve a nonorthogonal eigenvalue problem, product terms comprised of $\mathbf{P}$ and $\mathbf{Q}$ vectors may contain redundant information, resulting in a failure to converge to the exact FCI solution when the number of product terms added equals the FCI vector rank.

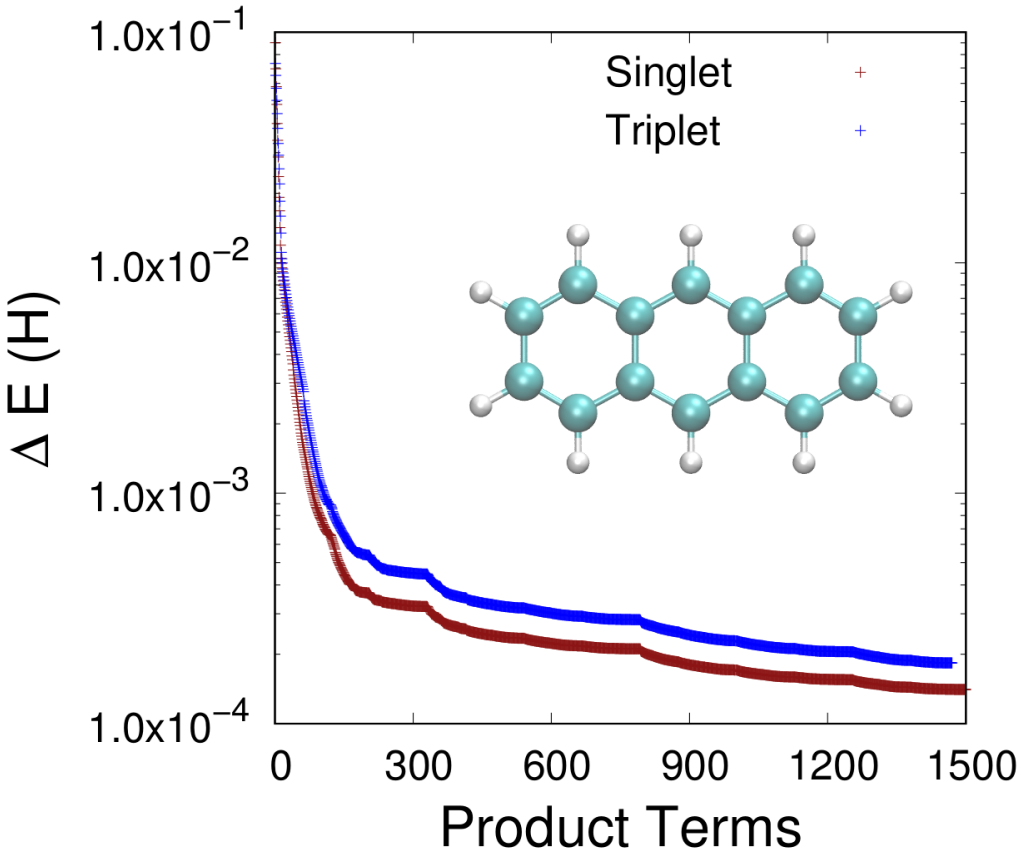

Figure 5.3: Anthracene $E_{rrFCI}-E_{FCI}$ for singlet and triplet states at HF-CAS-(14,14)-CI/cc-pVDZ. The singlet optimized structure is depicted with carbon and hydrogen atoms shown in teal and white, respectively.

The (14,14) CAS space of anthracene approaches the upper size limit of routinely performed CI calculations, providing an opportunity to showcase the convergence behavior of the rrFCI method for a non-trivial problem. Similar to the naphthalene calculation, mH accuracy is achieved relatively quickly, requiring fewer than 100 product terms (representing $< 3\%$ of the full rank of the CI vector, 3432 in this case). Following addition of ~ 150 product terms, convergence behavior decays before asymptotically approaching an accuracy of 1.0×10^{-4} H. Steps in the convergence at ~ 350 and ~ 800 product terms occur as artifacts of our guess vector procedure.

5.4.4 Relative Property Convergence: Acene Singlet-Triplet Gaps

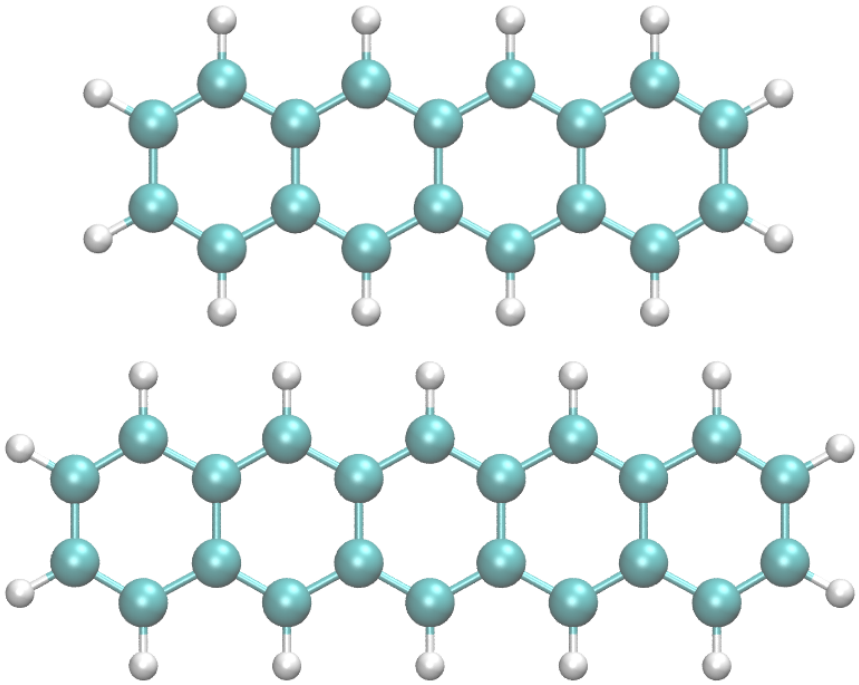

Figure 5.4: Singlet optimized tetracene (above) and pentacene (below) geometries. Carbon and hydrogen atoms depicted in teal and white, respectively.

In addition to absolute energy convergence of the shortest acenes compared with FCI, we have also investigated the singlet—triplet energy gap for the acene series having $2-5$ rings. Structures for napthalene and anthracene are depicted in Figures 5.2 and 5.3, respectively, while tetracene and pentacene are shown in Figure 5.4. Singlet—triplet energy gaps for each system are depicted in Figure 5.5.

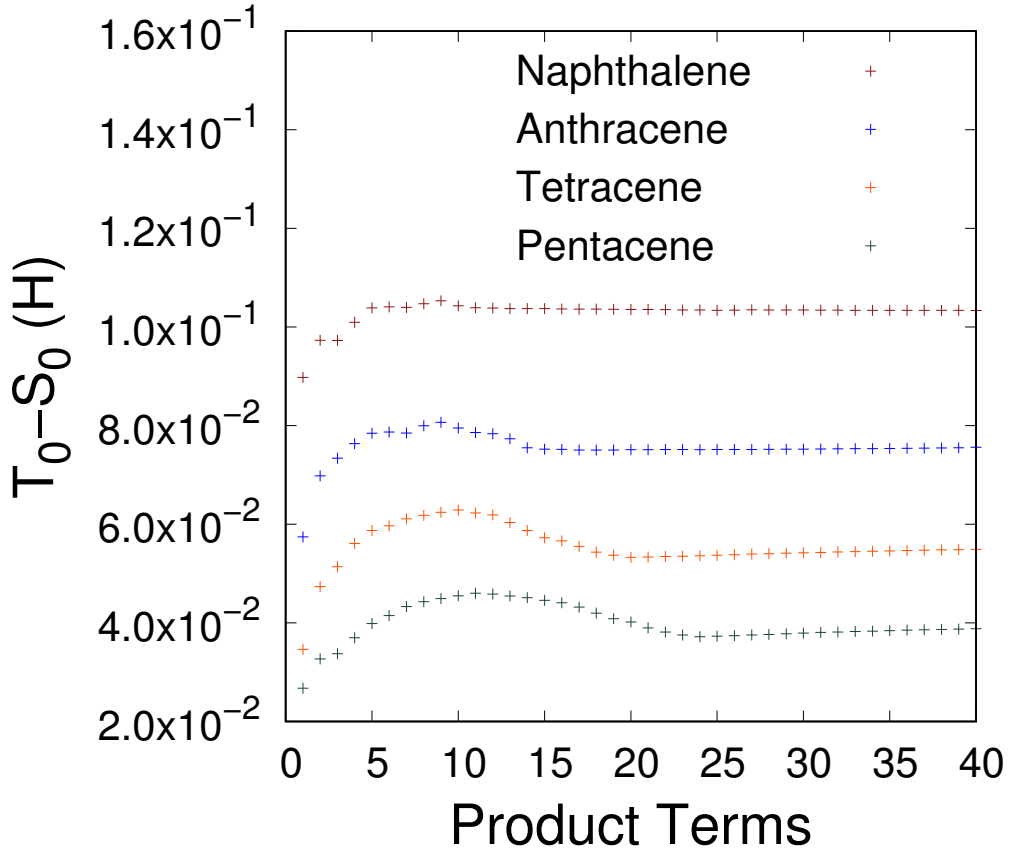

Figure 5.5: Singlet-triplet energy gap (in units of Hartree) for the acene series naphthalene through pentacene at HF-CAS-rrCI/cc-pVDZ where the CAS for each system is comprised of the full π valence.

While absolute energy convergence to within mH accuracy requires $\sim$ 60 product terms for naphthalene and $\sim$ 150 product terms for anthracene, convergence of a relative property, in this case the singlet—triplet gap, requires fewer product terms to achieve similar convergence. The S_0-T_0 gap for naphthalene is converged after only 12 product terms. Only slightly worse convergence is observed for the longer acenes, with anthracene, tetracene, and pentacene requiring 15, 20, and 24 product terms to stabilize, respectively.

Comparison of our HF-CAS-rrCI energies with the local DMRG results of Hachmann is given in Table 5.3, along with estimates to the experimental S_0-T_0 gaps for each system.

Given the disparity in CAS spaces between the rrCI and DMRG results reported above, the agreement between the two methods is excellent. In each case, rrCI lies within $\sim$ 4

Table 5.3: S_0-T_0 gap energies given in kcal/mol calculated at HF-CAS-rrCI using full π valence CASs compared with local density matrix renormalization group (DMRG) results as described by [52]. DMRG results were obtained using a "double" π valence space (i.e. naphthalene, anthracene, tetracene, and pentacene used (10,20), (14,28), (18,36), and (22,44) configuration spaces) and a cc-pVDZ basis set. Experimental energy gaps for naphthalene[21], anthracene[150], tetracene[149], and pentacene[26] are estimates. Number of product terms for rrCI results is reported in parentheses.

	CAS-rrCI	**DMRG**	**Expt.**
Naphthalene	65.1 (12)	61.0	61.0
Anthracene	47.2 (15)	44.0	43.1
Tetracene	33.4 (20)	31.9	29.3
Pentacene	23.3 (24)	23.4	19.8

kcal/mol of both the DMRG and the estimated experimental energies. Even more impressive is how few product terms as a function of full CI space rank are required to obtain this level of agreement: naphthalene requires 4.7% of the full rank, anthracene 0.4%, tetracene 0.04% and pentacene only 0.003%. While we must take care not to make broad generalizations based on a single series of calculations, the results presented here are encouraging and motivate us to pursue refinement and continued development of the rrFCI approach.

5.4.5 Nitrogen Bond Dissociation

FCI is often used as a benchmarking tool for evaluating lower-cost approximations. One commonly assessed task is calculation of the molecular energy as a bond is stretched towards the dissociation limit, a particularly difficult case as both static and dynamic electronic correlation regimes are encountered. We have applied the rrFCI method to the molecular nitrogen system, varying the bond length between 0.8 and 1.8 Å , and compare our rrFCI/cc-pVDZ results with the FCI/cc-pVDZ ground state energies in Figure 5.6. All 10 valence electrons are correlated in each series of calculations, corresponding to a (10,28) configuration space.

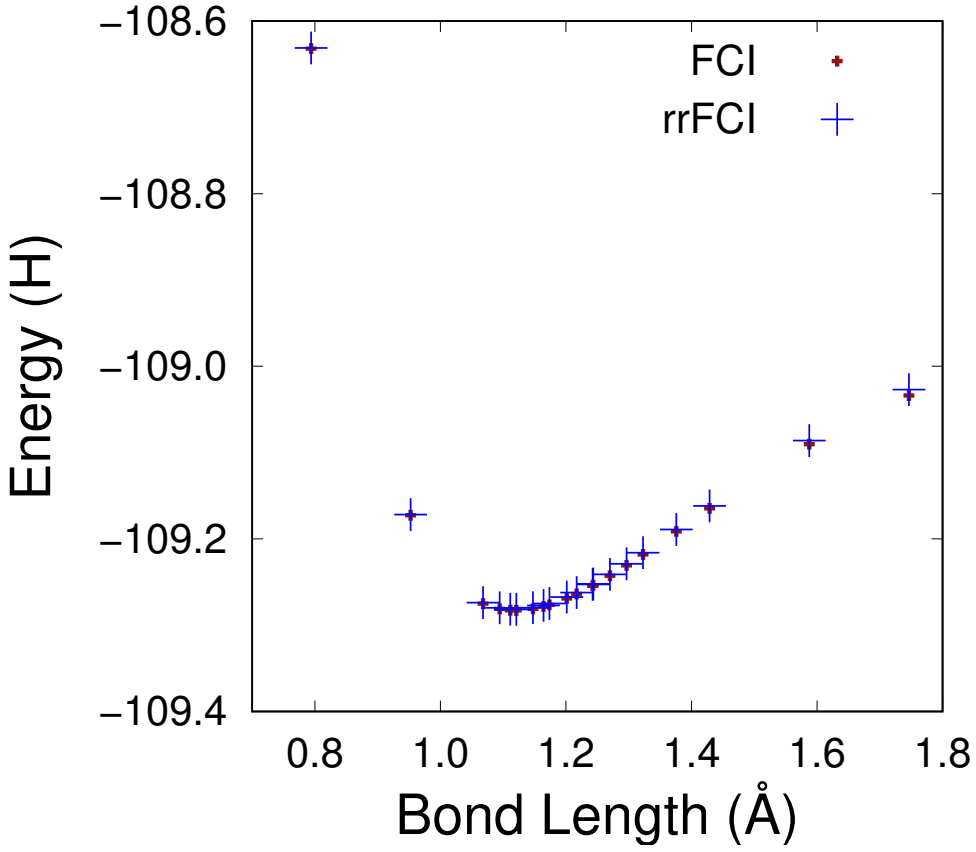

Figure 5.6: Molecular N_2 dissociation curve calculated at rrFCI/cc-pVDZ and FCI/cc-pVDZ[82]. rrFCI calculations include 100 product terms each.

While the configuration space corresponds to 9.6 billion determinants (without symmetry), excellent convergence is achieved by rrFCI compared with the exact FCI method in only 100 product terms. Figure 5.7 depicts the energy difference ($\Delta E = E_{rrFCI} - E_{FCI}$) between the two methods in units of Hartree. It is interesting to note that rrFCI calculations at or near the equilibrium geometry converge more rapidly than the stretched geometries, as can be seen by the higher energies relative to exact FCI when the nitrogen—nitrogen distance is greater than 1.2 Å. Increasing the number of product terms for the stretched geometry calculations continues to reduce the errors observed relative to the FCI energy. The low cost-to-accuracy ratio of rrFCI makes it an ideal tool for use in performing benchmark type calculations for evaluation of lower-cost alternatives.

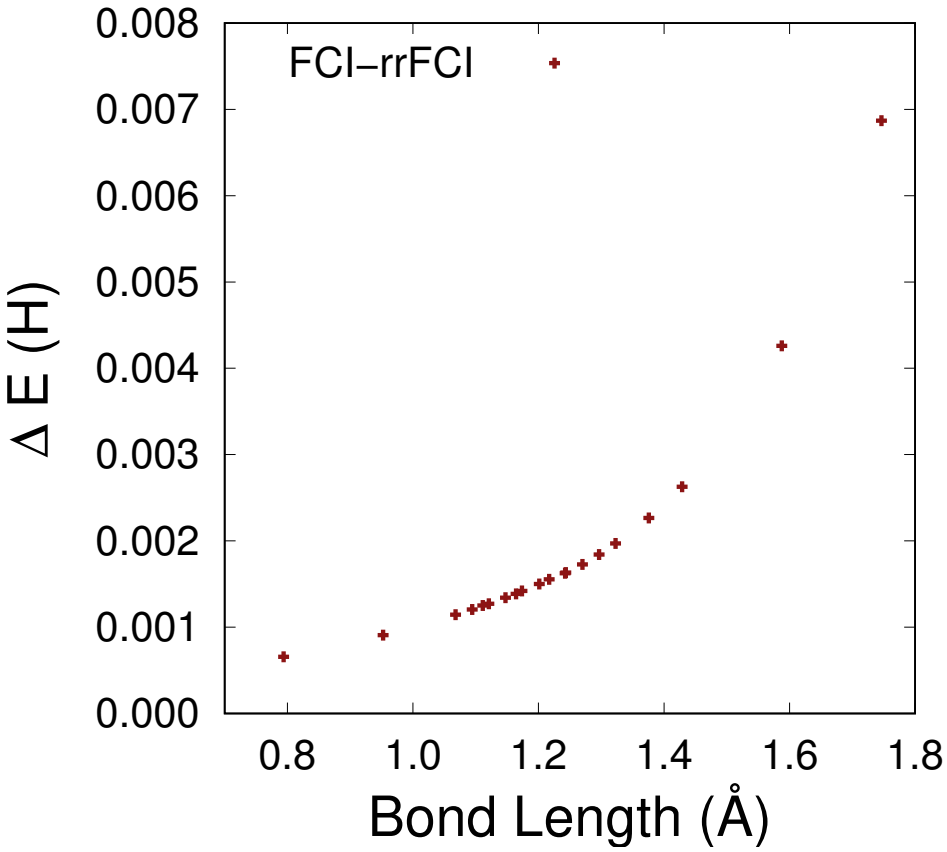

Figure 5.7: Molecular N_2 dissociation curve (rrFCI-FCI) energy differences (given in units of Hartree) calculated using cc-pVDZ. rrFCI calculations include 100 product terms each.

5.5 Conclusions

We have presented a low-cost FCI alternative for both singlet and triplet ground states that scales according to $\sqrt{N_{det}}$, allowing routine calculation of unprecedented configuration space sizes for general systems. GPU acceleration of rate-limiting components of the algorithm, including projected $\boldsymbol{\sigma}$ and 1-particle coupling coefficient formation, expand the size of accessible configuration spaces to $\mathcal{O}(10^{16})$ determinants while achieving sub-mH accuracy. We have applied our methods to the full π valence space CI calculations of acenes having $2-5$ polycyclic aromatic rings, demonstrating excellent agreement both with absolute energy convergence relative to FCI and to relative property convergence of S_0-T_0 energy gaps using both experimental estimates as well as with previous DMRG studies. Finally, we in-

vestigated dissociation of molecular nitrogen, and found that our results compare favorably against benchmark FCI studies. Given the success of rrFCI in describing the ground state singlet and triplet wavefunctions, we are currently working to extend our approach to allow for excited state calculations and the evaluation of molecular properties.

5.6 Acknowledgement

B.S.F. and B.G.L. are grateful for support by National Science Foundation grant CHE-1565634 and for computational resources from the Extreme Science and Engineering Discovery Environment through grant TG-CHE140101. S.S. was supported by an NSF Graduate Fellowship. T.J.M is a cofounder of PetaChem, LLC.

CHAPTER 6

CONCLUSIONS AND FUTURE DIRECTIONS

In the preceding chapters we have described the development of tools suitable for use in the characterization of molecular systems poorly described by a single Slater determinant such as those involved in photochemical processes. When possible, we have taken advantage of the parallel processing power offered by graphical processing units (GPUs) to accelerate electronic structure calculations and molecular dynamics simulations. Recognizing that configuration interaction based methods such as SA-CASSCF and those based on CASCI perform especially well on statically correlated systems, in Chapter 2 we describe our implementation of a GPU accelerated direct CI algorithm based on the vectorized algorithm of Knowles and Handy[73]. We have demonstrated the near-linear scaling of $\boldsymbol{\sigma}$ vector formation with respect to number of configurations, which, combined with the fast electron repulsion integral (ERI) transformation code that leverages sparsity in the atomic orbital basis to achieve quasi-quadratic scaling with respect to basis set size, permits CI calculations on systems having more than 10^3 basis functions with configuration spaces on the order of 10^8 determinants on time-scales of minutes to tens of minutes. The determinantal direct CI code described in Chapter 2 has been coupled with a variety of CASCI methods, including Hartree-Fock (HF-), improved virtual orbital (IVO-), configuration interaction singles natural orbital (CISNO-), and unrestricted natural orbital (UNO-) CASCI, as well as serving as a standalone FCI implementation and as the CI component of large active space CASSCF and SA-CASSCF.

In addition to calculation of ground and excited state energies, it is generally desirable to calculate the analytical nuclear energy gradient and nonadiabatic coupling vectors for use in the context of stationary point and minimum energy conical intersection (MECI) optimizations as well as for use in molecular dynamics simulations. Further, molecular properties such as the dipole moment and transition dipole moment are often useful for comparison

against spectroscopic data. In each case, the generalized 1- and 2-particle reduced density matrix (RDM) is needed to be formed efficiently. Methods requiring rotations of the molecular orbitals, such as CASSCF, also require formation of the 1-RDM for calculation of the orbital gradient. In Chapter 3 we described our GPU accelerated algorithms for formation of the generalized 1- and 2-RDMs, demonstrating that the performance of RDM formation is comparable to that achieved with our direct CI implementation.

Generalization of our direct CI code to allow the calculation of open-shell systems was straightforward, requiring only the generation of guess vectors having both the desired spin symmetry and a large overlap with the target eigenvectors. In Chapter 4 we briefly described our guess vector procedure. Preconditioned iterative diagonalization methods for determinant basis eigenvalue problems are prone to numerical instability, however, and we have presented several approaches in Chapter 4 for ensuring the spin purity and the correct target $\langle \hat{S}^2 \rangle$ of the converged CI vectors. These approaches allow the calculation of energies and properties, including a measure of the single excitation character, for several electronic states of both open- and closed-shell systems. To demonstrate this capability we described the lowest several excited states of an open-shell silver cluster at the SA-CASSCF level of theory. Our work suggests that multireference electronic structure methods are necessary for providing a qualitatively accurate description of systems exhibiting localized surface plasmon resonance character such as noble metal clusters.

Configuration interaction methods scale very poorly with respect to the number of active electrons and orbitals. To extend CI to generalized systems having larger configuration spaces we have developed the rank-reduced CI (rrCI) approach. Chapter 5 describes our formulation and GPU implementation of rrCI, an approximation to FCI suitable for systems having configuration spaces on the order of 10^{16} determinants. rrFCI makes no assumptions about the dimensionality or topology of the system, in contrast to methods based on tensor networks such as the density matrix renormalization group (DMRG), and is therefore capable of being applied to arbitrary systems. We have verified the accuracy of rrCI by comparing

against systems where the FCI energies are obtainable, including short (2 and 3 ring) linear polyaromatic hydrocarbons (acenes) and the nitrogen dimer dissociation, and by comparing against DMRG results for acenes having 4 and 5 rings.

The availability of GPU accelerated direct CI has enabled the routine calculation of multireference wavefunctions for systems approaching the nanoscale. Among other things, this allows for the location and characterization of minimum energy conical intersections (MECIs), using fully ab initio approaches, of small nanoparticles (< 2 nm). One example of recent work that achieves this is the characterization of low-lying oxide-defect-induced MECIs in silicon nanoparticles[159]. In this work, MECIs for each of 9 oxide defect systems were located using the CIOpt MECI optimization program[89] in conjunction with the CISNOr-CASCI method as implemented in the TeraChem electronic structure package. Three types of defects were investigated, an epoxide having three adjacent silicon atoms and one hydrogen atom bonded to the epoxide silicon atoms (type 1), an epoxide having two adjacent silicon atoms and two adjacent hydrogen atoms bonded to the epoxide silicon atoms (type 2), and a silicon—oxygen double bond defect. For each of the three types of defect, each of a small, medium, and large cluster was examined. For the type 1 epoxide and silicon—oxygen double bond system the clusters contained 14, 29, and 44 silicon atoms, while the type 2 epoxide clusters contained 15, 25, and 50 silicon atoms. Figure 6.1 depicts the MECI geometry of the largest cluster for each defect type (including the total number of basis functions) and Figure 6.2 provides more detailed information for each of the MECI geometries including definition of the connectivity for each defect type and images of the highest occupied molecular orbital (HOMO) and lowest unoccupied molecular orbital (LUMO).

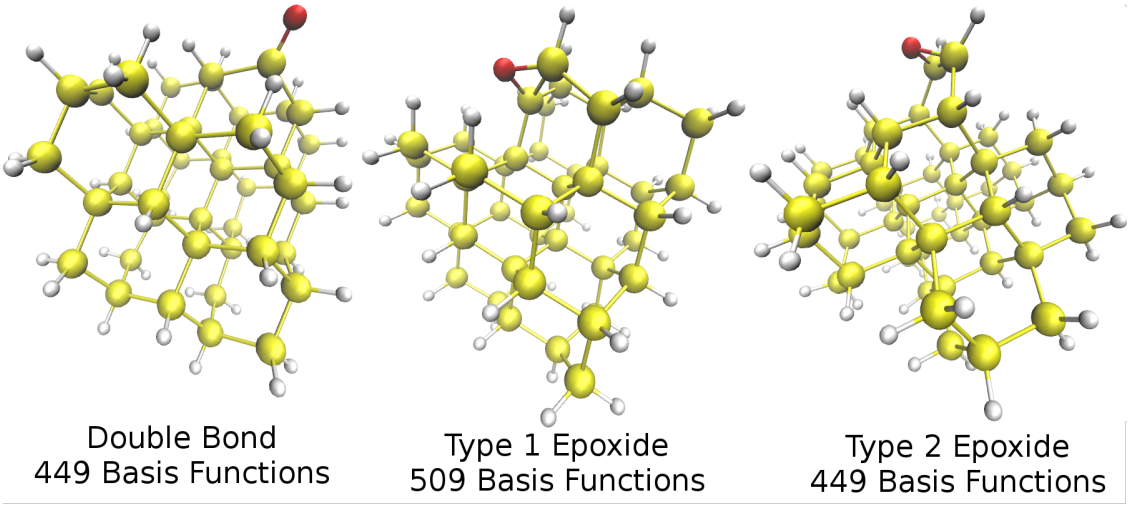

Figure 6.1: Silicon clusters with oxygen defects at their MECI geometries. From left to right, molecular formulae are $Si_{44}H_{44}O$ (with Si=O bond), $Si_{50}H_{50}O$ (with Si—O—Si epoxide), and $Si_{44}H_{44}O$ (with Si—O—Si epoxide)

Calculations were performed using CISNOr-CASCI/LANL2DZ for each molecular cluster. Active spaces for the epoxide double bond systems were chosen to be (2,4) and (4,3), respectively. The 44 silicon atom cluster contained 449 basis functions and the 50 silicon atom cluster 509 basis functions. Optimizations were performed using numerical energy gradients as the analytical energy gradient formulation for the relaxed density matrix variant for CISNO-CASCI is not yet available. Our high-performance direct CI implementation allowed the optimization of these large clusters, where each single point electronic energy calculation was completed within minutes.

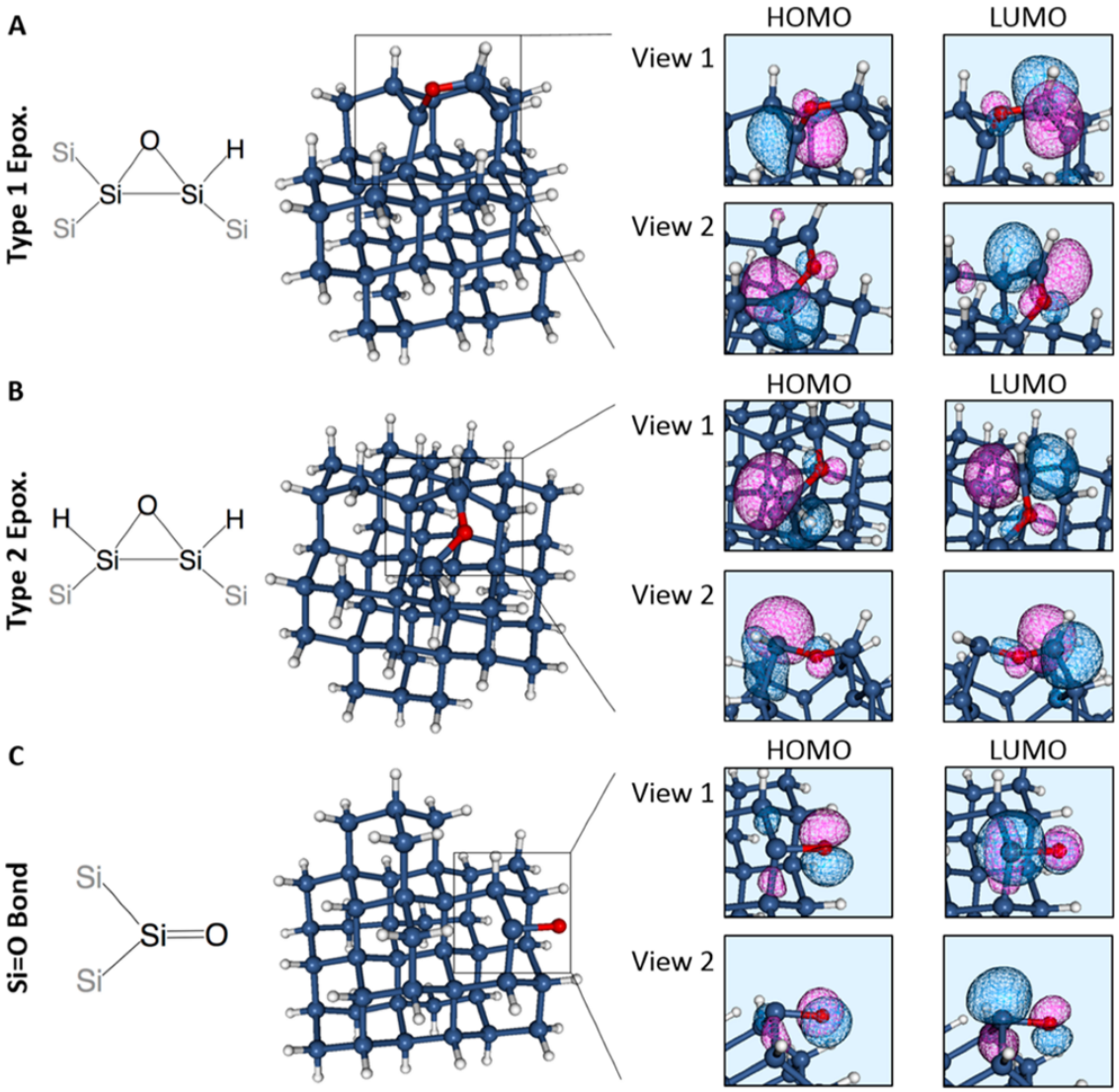

Figure 6.2: Geometric and orbital information for each of the largest type 1 epoxide, Si=O double bond, and type 2 epoxide defect silicon clusters. The first column depicts the defect connectivity, the second column shows the optimized MECI geometry, and the third column provides different views of the HOMO and LUMO for each system. Image courtesy of "Defect-Induced Conical Intersections Promote Nonradiative Recombination", Y. Shu, B. S. Fales, and B. G. Levine, *Nano Lett.*, 15, 6247-6253, 2015. Copyright 2015 American Chemical Society.

The availability of fast CASCI energies enabled by our GPU accelerated direct CI implementation combined with our robust spin purification procedures allows investigation of the dynamics and electronic structure of classes of materials that were previously difficult or impossible to study. One such molecular system is the boron-doped silicon nanoparticle

$Si_{37}BH_{42}$ measuring $1.3 \times 0.96 \times 0.82$ nm, where an interface silicon atom is replaced by a boron atom. Using a combination of ab initio molecular dynamics (AIMD) and the CIOpt program we have located a low-lying MECI between the ground and first excited doublet states using the floating occupation molecular orbital (FOMO-) CAS-(5,3)-CI/LANL2DZ level of theory. The MECI geometry energy relative to the D_0 minimum energy is 0.68 eV. The Franck-Condon and MECI geometries are depicted in Figures 6.3 and 6.4.

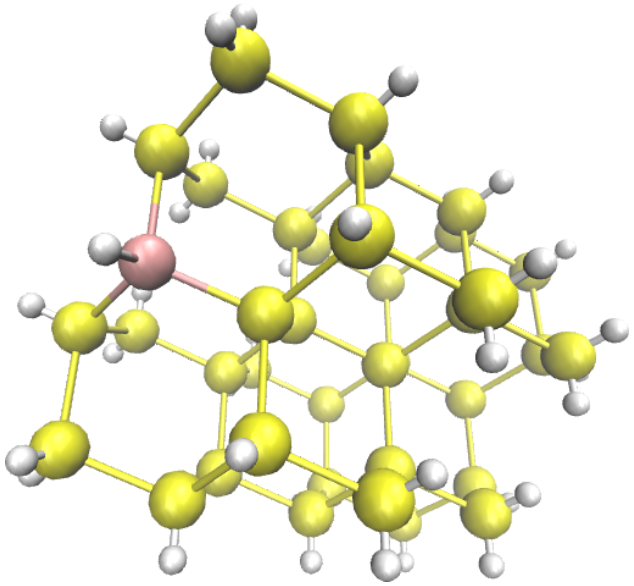

Figure 6.3: Boron defect silicon cluster $Si_{37}BH_{42}$ D_0 minimum geometry calculated at FOMO-CAS-(5,3)-CI/LANL2DZ. Silicon atoms are depicted in yellow, hydrogen atoms in white, and boron atoms in pink.

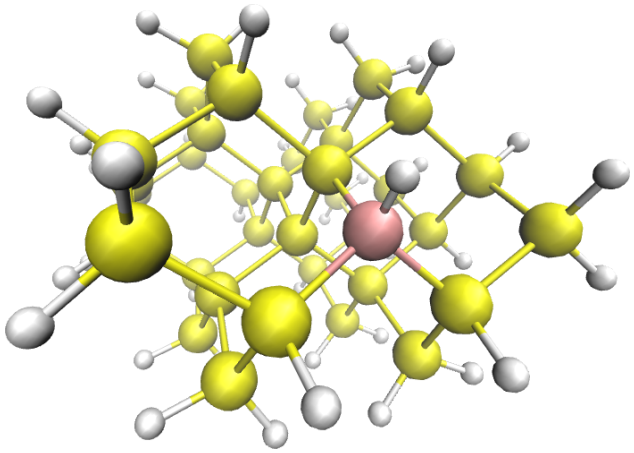

Figure 6.4: Boron defect silicon cluster $Si_{37}BH_{42}$ D_0-D_1 MECI geometry calculated at FOMO-CAS-(5,3)-CI/LANL2DZ. Silicon atoms are depicted in yellow, hydrogen atoms in white, and boron atoms in pink.

The MECI geometry differs from the D_0 minimum geometry by a flattening of the boron atom against the surface of the remainder of the cluster. Detailed geometric parameters are given in Table 6.1.

Table 6.1: $Si_{37}BH_{42}$ D_0 minimum and MECI geometry parameters. Angles given in units of degrees and bond lengths given in Å.

	D_0 Minimum	**D_0-D_1 MECI**
Si—B—Si angle	107.537	104.156
	107.537	111.959
	111.334	112.025
B—Si distance (Å)	2.27416	2.13878
	2.27417	2.13361
	2.38139	2.21514

Figure 6.5 depicts the molecular orbitals corresponding to the singly occupied molecular orbital (SOMO) for the D_0 minimum and the SOMO and LUMO for the D_0-D_1 MECI geometries. The SOMO and LUMO are strongly localized at the position of the boron atom in each case. This localization of electron density at the defect site was also observed in the oxide defect-induced MECIs described above. While the results for the boron-defect silicon clusters are preliminary, we have demonstrated here the existence of low-lying defect-induced MECIs through incorporation of the electron-deficient boron atom into a silicon nanocluster.

Further work to include molecular dynamics and a study based on the size dependence of the MECI energy of boron defect-induced silicon cluster MECIs will be forthcoming in future work.

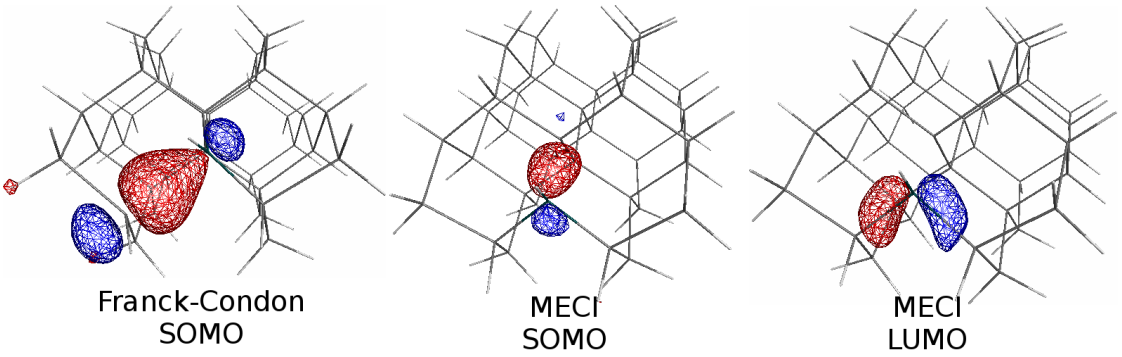

Figure 6.5: Singly occupied molecular orbital (SOMO) and lowest unoccupied molecular orbital (LUMO) for the D_0 minimum and MECI geometries. Silicon atoms depicted in grey, hydrogen atoms in white, and boron atoms in blue/green.

While the computational scaling of CI based methods is daunting, recent developments in vector hardware technology have provided motivation to develop new algorithms to take advantage of their massively parallel processing power. This is especially good news for the computational chemistry community interested in studying the photodynamics of molecular systems, as CASSCF and CASCI perform particularly well in describing these systems. The methods described in this work will allow the investigation of larger molecular systems at higher levels of accuracy (both due to methodological developments and to expanding the size of the correlated region, or the configuration space) in the coming years. The work presented here is a good first step, but much remains to be done still. Specifically, two areas that require immediate attention are 1) the CI scaling problem and 2) the dynamic/static electron correlation problem. While rrCI provides a strong foundation for future development of both reduced rank and FCI approximation approaches to solve the first problem, rrCI suffers from sub-optimal convergence to the FCI energy. A trial vector generation scheme based on projected preconditioned residual vectors will likely improve this situation, though our early attempts have proven unsuccessful. Inclusion of ideas from tensor networks (i.e.

DMRG) and from stochastics (i.e. "Heat-Bath" selected CI[62] or QMC-FCI[24, 174]) may allow us to further develop our approach to both improve accuracy and reduce computational scaling. The second point, that related to improving our description of electronic correlation, is currently in the development stage in the form of multireference CI truncated to single excitations (MR-CIS). Completion of this work will immediately allow multireference description of high-lying single excitations for molecular systems approaching the nanoscale. Our work with RDMs can naturally be extended to allow direct vectorized calculation of the higher-order 3- and 4-RDMs required for internally contracted MR-CISD.

APPENDICES

APPENDIX A

SUPPORTING INFORMATION FOR: NANOSCALE MULTIREFERENCE QUANTUM CHEMISTRY: FULL CONFIGURATION INTERACTION ON GRAPHICAL PROCESSING UNITS

A.1 Supporting Tables

Table A.1: CASCI/LANL2DZ times-to-solution using (6,6), (12,12), and (16,16) active spaces. Times-to-solution for the HF step, which is required for orbital determination but is not included in the CASCI times reported here, are shown in the final column for comparison.

System	**Time-to-Solution (s)**			
	(6,6)	**(12,12)**	**(16,16)**	**HF**
$Si_{15}H_{25}$	0.66	3.76	1093.50	1.98
$Si_{25}H_{30}$	1.87	8.29	949.80	4.71
$Si_{44}H_{44}$	6.45	23.98	1096.28	17.57
$Si_{50}H_{50}$	8.57	30.34	1007.76	20.96
$Si_{72}H_{64}$	23.59	77.00	827.96	57.50

Table A.2: CASCI/6-31G** times-to-solution using (6,6), (12,12), and (16,16) active spaces. Times-to-solution for the HF step, which is required for orbital determination but is not included in the CASCI times reported here, are shown in the final column for comparison.

System	**Time-to-Solution (s)**			
	(6,6)	**(12,12)**	**(16,16)**	**HF**
$Si_{15}H_{25}$	11.05	35.02	534.41	25.94
$Si_{25}H_{30}$	35.64	105.62	596.27	93.06
$Si_{44}H_{44}$	130.08	360.56	1030.23	442.82
$Si_{50}H_{50}$	167.67	469.92	1217.45	557.67
$Si_{72}H_{64}$	468.98	1190.91	2370.34	1610.08

Table A.3: CASCI/6-31G** times-to-solution using (6,6), (12,12), and (16,16) active spaces. Times-to-solution for the HF step, which is required for orbital determination but is not included in the CASCI times reported here, are shown in the final column for comparison.

System	**Time-to-Solution (s)**			
	(6,6)	**(12,12)**	**(16,16)**	**HF**
Pyrazine	0.69	3.69	1865.11	1.67
Dimelamine	2.84	11.11	693.82	8.55
Dimelem	7.26	22.58	1056.78	31.61
Trimelem	18.31	54.18	638.72	73.27
Hexamelem	70.33	191.49	789.57	267.18

Table A.4: Comparison of the times required to perform ERI transformations with different size active spaces and basis set. All calculations used the LANL2DZ basis and were performed on a single NVidia K40 GPU.

System	**No. of Basis Functions**	**ERI Transformation Time (s)**		
		(6,6)	**(12,12)**	**(16,16)**
$Si_{15}H_{25}$	160	0.63	2.40	4.74
$Si_{25}H_{30}$	260	1.84	6.89	13.55
$Si_{44}H_{44}$	440	6.32	22.48	43.23
$Si_{50}H_{50}$	500	8.37	28.93	55.46
$Si_{72}H_{64}$	704	23.10	75.43	139.49

Table A.5: Comparison of the times required to perform ERI transformations with different size active spaces and single-electron basis. All calculations used the 6-31G** basis and were performed on a single NVidia K40 GPU.

System	**No. of Basis Functions**	**ERI Transformation Time (s)**		
		(6,6)	**(12,12)**	**(16,16)**
$Si_{15}H_{25}$	385	10.93	33.53	59.70
$Si_{25}H_{30}$	625	35.22	103.78	182.85
$Si_{44}H_{44}$	1056	128.38	357.19	616.45
$Si_{50}H_{50}$	1200	165.27	465.97	798.16
$Si_{72}H_{64}$	1688	462.46	1182.64	2064.69

Table A.6: Comparison of the times required to perform ERI transformations with different size active spaces and single-electron basis. All calculations used the 6-31G** basis and were performed on a single NVidia K40 GPU.

System	**No. of Basis Functions**	**ERI Transformation Time (s)**		
		(6,6)	**(12,12)**	**(16,16)**
Pyrazine	110	0.48	1.72	3.26
Dimelamine	300	2.79	9.28	17.80
Dimelem	510	7.06	20.56	42.29
Trimelem	720	17.83	52.05	98.31
Hexamelem	1380	67.31	186.80	376.04

A.2 Supporting Geometries

We provide Cartesian coordinates for all structures given in Figure 1 and in Tables A.1 to A.6 in units of Angstrom (Å).

Pyrazine ($C_4N_2H_4$)

	X	Y	Z
C	0.043212	-0.111863	0.030960
C	1.178081	-0.158657	1.998825
C	2.321112	-0.600233	1.329035
C	1.186051	-0.553557	-0.638847
N	0.021661	0.092331	1.359971
N	2.342440	-0.804539	0.000080
H	1.187513	-0.001787	3.071560
H	-0.877047	0.083135	-0.508064
H	1.176552	-0.710913	-1.711343
H	3.241428	-0.795320	1.867838

Dimelamine ($C_6N_{11}H_9$)

	X	Y	Z
N	2.2076378511	0.3366740014	-0.2491142011
N	1.4616722720	-0.3240918931	-2.4081095971
N	3.7860252231	0.0800842421	-2.0162794288
N	0.0037423528	-0.0286033035	-0.5866094819
N	-2.2139297041	-0.1725213723	-0.1787275342
N	-3.7314268265	-0.9223597888	-1.8565940014
N	-1.3908222715	-0.7773072106	-2.3259824586
N	-2.8839705982	-1.4653100833	-3.9162227545
N	-4.4596967444	-0.3545005494	0.2426816480
N	3.0125811747	-0.5863427228	-4.0687906108
N	4.4390186762	0.7464633228	0.0785422526
C	1.2675651705	-0.0168310297	-1.1356836912
C	2.7486447034	-0.2597166795	-2.7848564862
C	3.4438384137	0.3700107000	-0.7558860162
C	-1.2415678086	-0.3464460210	-1.0837040264
C	-3.4336666542	-0.4825783421	-0.6287101083
C	-2.6656873940	-1.0443493699	-2.6513251768
H	-0.0134136335	0.2559822597	0.3823963955
H	-4.2853555857	0.0963855978	1.1239490687
H	4.2322927053	0.8406383088	1.0576757283
H	-5.3975398235	-0.4477801314	-0.1066915186
H	5.3886793411	0.6486319306	-0.2359130750
H	-3.7982575594	-1.8040714731	-4.1602344709
H	3.9370321363	-0.4214874449	-4.4270853047
H	-2.0850867819	-1.6840331924	-4.4859063128
H	2.2350673656	-0.7204407554	-4.6917548375

Dimelem ($C_{12}N_{19}H_9$)

	X	Y	Z
N	-0.5849539468	-0.0409195008	11.3471237143
C	-1.6105221883	-0.0532634031	10.5127966265
N	-1.3695566937	-0.0476416601	9.1271349637
C	-0.0524361611	-0.0320515460	8.6418075902
N	0.9522440023	-0.0207343559	9.4931088259
C	0.6369179026	-0.0265955625	10.7989923930
C	-2.4478845576	-0.0610798782	8.2289667416
N	-2.1999244207	-0.0582770517	6.9288189691
C	-0.9122105798	-0.0419526545	6.5575712936
N	0.1613358259	-0.0291724594	7.3294986307
N	-2.8557211443	-0.0759050914	10.9537870723
C	-3.8240747929	-0.0811017475	10.0265577559
N	-3.6838159592	-0.0714778805	8.6923447553
N	-0.7837959925	-0.0380440876	5.1881709343
C	0.2970595209	-0.0219893576	4.3378263568
N	-0.0890650351	-0.0247302549	3.0542729311
C	0.8688554981	-0.0101919131	2.1412791996
N	2.2079983906	0.0066920201	2.5604091685
C	2.5143669915	0.0081280233	3.9302246045
N	1.5275934262	-0.0064503129	4.8212793682
C	3.2415596365	0.0218783659	1.6065896778
N	2.9367881395	0.0202836204	0.3209757649
C	1.6327981527	0.0039435184	0.0100103966
N	0.5880423006	-0.0113548042	0.8513250829
N	3.7743937666	0.0239498760	4.3129185332
C	4.7005863620	0.0378335092	3.3399359398
N	4.4985057494	0.0375911349	2.0158800850
N	1.3366925432	0.0026601949	-1.2970943034
N	5.9769193111	0.0537775565	3.7466137054
N	-5.0829085413	-0.0907467044	10.4864930786
N	1.6706623948	-0.0170910843	11.6511070138
H	2.6065507082	-0.0057053812	11.2829191994
H	1.4908743411	-0.0183469600	12.6405484032
H	6.7108015184	0.0645465365	3.0591443591
H	6.1752108986	0.0546056401	4.7326776418
H	0.3718161481	-0.0089518418	-1.5802581481
H	2.0848585310	0.0137574194	-1.9689308551
H	-1.6721857195	-0.0491248881	4.7040980350
H	-5.8439022769	-0.1176444012	9.8296001571
H	-5.2406600503	-0.1121316329	11.4794653377

Buckyball (C_{60})

	X	Y	Z
C	-1.1241617717	-5.2742619223	0.5527495948
C	-1.4857298279	-4.2262077848	1.5235253960
C	-2.9515997271	-4.2312085234	1.6740267898
C	-2.5014368213	-6.3284282971	-1.2209644950
C	-1.4017970718	-6.1011107388	-2.1752113628
C	-0.2295193322	-5.4852415479	-1.7511815223
C	-0.0866038169	-5.0598337376	-0.3475811111
C	-0.7892682398	-3.0228632184	1.5391949124
C	0.3104695308	-2.7955704465	0.5849592400
C	0.6515590414	-3.7845011662	-0.3310282651
C	-3.6383604742	-3.0326069958	1.8316836466
C	-2.9003235762	-1.7572625818	1.8482468091
C	-3.7737179086	-6.1093536544	-1.9315282743
C	-5.4243318047	-3.8052402103	0.2930188874
C	-4.9106053294	-2.8134855710	1.1210989726
C	-1.9944955885	-5.7415558695	-3.4755453234
C	-3.4604302222	-5.7466400929	-3.3250337570
C	0.9648405221	-3.4217144160	-1.7245166468
C	0.4202879964	-4.4728447558	-2.6021701676
C	-6.0171481919	-3.4456397480	-1.0072873947
C	-5.6556646019	-4.4937418445	-1.9780465377
C	-3.7164610060	-0.7498890991	1.1478412994
C	-4.9588742849	-1.4026243189	0.6984630780
C	-0.1389946272	-4.1335009193	-3.8290801506
C	-1.3814038391	-4.7862353281	-4.2784820952
C	-5.3601028166	-4.1514833912	-3.2928957434
C	-4.2305518980	-4.7961341912	-3.9859511352
C	-5.5181700709	-1.0632899264	-0.5284684512
C	-6.0627012597	-2.1144273161	-1.4060969210
C	-1.6374522232	0.2104876728	0.1943910010
C	-3.1033751613	0.2054069617	0.3449033518
C	0.5578067520	-1.0423764314	-1.1525850415
C	0.9192961791	-2.0904877938	-2.1233442291
C	-2.1975676294	-3.7788799180	-4.9788791861
C	-3.5807066278	-3.7836358863	-4.8369101783
C	-0.2584469991	-0.0350332637	-1.8530045403
C	-1.3241574952	0.5732118936	-1.1991083344
C	-3.6960562172	0.5649807376	-0.9554377832
C	-4.8683449123	-0.0508694511	-1.3794373414
C	-5.7494195704	-1.7516195268	-2.7995886472
C	-5.4083083378	-2.7405683768	-3.7156057618
C	-5.0112824330	-0.4763071785	-2.7830596997
C	-2.5964467407	0.7922859453	-1.9096950594
C	-4.8394114983	-5.5011002111	-1.2776190834
C	-4.6964815146	-5.0755686026	0.1259109952
C	-3.4960357827	-5.2823051627	0.7963578392
C	-2.3666111336	-5.9269395271	0.1033805367
C	-1.5171478602	-1.7525066545	1.7062734550
C	-0.8673100176	-0.7399940142	0.8552948619
C	0.2622308883	-1.3846521603	0.1622912243
C	-4.3085992948	-2.5132713619	-4.6698204917
C	-3.9736822675	-0.2618760364	-3.6834186856
C	-3.6121618596	-1.3099182087	-4.6541506917
C	-1.4595000257	-2.5035243215	-4.9623075909
C	-0.1872752763	-2.7226581858	-4.2517234068
C	-2.1462519647	-1.3049217280	-4.8046595477
C	0.3264623031	-1.7308812785	-3.4236431228

C	-0.4013811715	-0.4605738889	-3.2565517532
C	-2.7312510739	0.3908260260	-3.2339928786
C	-1.6018340157	-0.2538324547	-3.9269994820

Silicon Nanoparticle ($Si_{72}H_{64}$)

	X	Y	Z
Si	0.0094082690	-0.0217856785	-0.0041189783
Si	0.0287568817	-0.0369390760	2.2907826028
Si	2.1828977668	0.0005978371	-0.7665567945
Si	2.1507663857	0.0212629496	-3.0697874537
Si	-1.0958575986	-1.8993161290	-0.7520833573
Si	-1.0828216766	1.8573975826	-0.7710306531
Si	-1.1131674195	-1.8695131003	-3.0553195693
Si	-1.0995050593	1.8692869896	-3.0757804295
Si	0.0126518547	-3.7741937631	-0.0262698815
Si	3.2631139344	-1.8890587256	-0.0379951239
Si	0.1870276216	-3.8321467245	2.2368921855
Si	3.2431181747	-2.0471232602	2.2271422961
Si	1.1183405612	-1.9133059624	3.0067345789
Si	2.1349236483	-3.6935387643	-0.8405631427
Si	-2.2086391565	-0.0003455136	-3.8373992795
Si	1.0692944837	1.8984844326	-3.8523185205
Si	1.0565563165	-1.8581475439	-3.8331032957
Si	-2.1855986323	3.7396933953	-3.8425784865
Si	-2.1779287793	3.7116921623	-6.1269822388
Si	-3.1612061219	1.8569349622	-6.9833554402
Si	-0.0864892538	3.6340571784	-6.9989576248
Si	-2.1940259643	-0.0134287585	-6.1329213045
Si	1.0577232211	1.8672759855	-6.1479131262
Si	-0.0264597641	-0.0154280946	-6.8212365321
Si	2.1484914297	-3.7360475197	-3.1134503251
Si	1.0610191916	-1.8851736526	-6.1217088919
Si	4.2872185561	-3.7009594537	-3.9081231764
Si	1.0458190756	-5.5742921794	-3.8944903734
Si	-0.0187016739	-3.7608206237	-6.8435852534
Si	3.2213993234	-1.8880004554	-6.8554084207
Si	1.0919392775	-5.6140721759	-6.1599294162
Si	4.2820493794	-3.7709036002	-6.1730988554
Si	-2.1303750385	0.0052494383	3.0329543603
Si	1.0703935816	1.8603901871	3.0200264528
Si	-3.2305751075	1.8830486061	0.0203009401
Si	-0.0319211369	3.7363233936	0.0058233210
Si	-4.3147837052	3.7306173986	-3.0167515778
Si	-1.1144070621	5.5848599669	-3.0283734080
Si	-1.1455161136	5.5749307867	-0.7573328079
Si	-0.0444318533	3.6989650227	2.2872261750
Si	-3.1797265140	1.8825902756	2.3009515756
Si	-4.2792480980	3.7592247904	-0.7456307775
Si	-2.1737228941	3.7658880553	3.0612349979
Si	-3.2574905178	5.6123459019	0.0636787554
Si	-3.2463744781	5.6189222791	2.3270003390
Si	-3.1308784467	-1.8914433321	-6.9953369129
Si	-4.3581026854	0.0126721429	-3.0327837951
Si	-2.2078789639	-3.7386605103	-3.8123456332
Si	-2.1633319391	-3.7307837243	-6.0953857771
Si	-0.8742211250	-5.7008831298	-0.8329919680
Si	-1.0769805996	-5.5842294145	-3.0856384286
Si	-3.2557353099	-1.8710293982	0.0257196589
Si	-4.2592702956	0.0125083102	-0.7605294398
Si	-5.5043442320	-1.8552939911	-3.6243452654
Si	-4.4424024314	-3.6420188694	-0.7571275158
Si	-4.3509658303	-3.7111707658	-3.0213820649
Si	3.2228227657	-5.6250477753	-6.9229403066

Si	3.2352355342	1.8919599867	-0.0026604041
Si	2.1336678602	3.7738736556	-3.0641372866
Si	2.0974573105	3.6962840471	-0.7905690346
Si	5.3546231784	-1.8608056360	-3.1123075177
Si	4.2685417290	-0.0095293579	-6.1231339691
Si	4.3123000422	0.0363176939	-3.8398384936
Si	5.3552533313	2.0514357597	-0.7986448021
Si	4.3130810780	3.8193347404	-3.7001445165
Si	5.3586379192	1.9121733965	-3.0610301479
Si	3.1926157724	2.0724466896	2.2585741588
Si	-3.3618749813	-1.7436801891	2.2884868090
Si	-5.5024968376	1.8901245978	-3.5903215893
Si	1.0683831792	5.6977124362	-3.6221991321
Si	5.3705201082	-2.0732064723	-0.8574710175
Si	3.1379568270	1.7238884028	-7.0433902178
H	3.2448539483	-5.6741717608	-8.3704539532
H	3.9101168929	-6.8158891002	-6.4672541648
H	0.3915044508	-6.7945164948	-6.6298969072
H	5.6495756255	-3.7494096467	-6.6569973611
H	4.9736893797	-4.8708655269	-3.3916786088
H	3.2012103674	-1.8456299534	-8.3061422544
H	1.7217190303	-6.7519585882	-3.3818172801
H	-0.0534241655	-3.7258578282	-8.2941857024
H	5.6339428907	-0.0080563645	-6.6153337237
H	6.7227318497	-1.8565790626	-3.5973161248
H	-1.7584159953	-6.7711041562	-3.5689892514
H	-2.8500091691	-4.9152832746	-6.5773317197
H	-2.1180089968	-6.0474821747	-0.1811400955
H	0.0397769131	-6.7796411942	-0.5174357696
H	2.8250936236	-4.8790575296	-0.3648349741
H	5.8672708362	-3.3935335210	-0.5276631326
H	6.2883937157	-1.1459976561	-0.2334033060
H	6.7279669394	1.9275356853	-3.5410534032
H	3.8647409577	2.9708893237	-6.9459348409
H	2.9957846470	1.4787568248	-8.4644320048
H	-0.0319578801	-0.0136263766	-8.2733196808
H	-4.5665680402	-1.9014824061	-6.8224810303
H	-2.9155509700	-1.9011461079	-8.4281353452
H	-5.0503097886	-4.8899927990	-3.4977032935
H	-4.5997512040	5.6589628234	2.8416623128
H	-2.5884907749	6.8107734152	2.8216619848
H	-3.9404476488	6.7916216022	-0.4339157188
H	-2.1545524042	3.7442558333	4.5116704300
H	0.6718242568	4.8716825346	2.7548130347
H	-0.4344134504	6.7537394450	-0.2972956210
H	-5.6535275538	3.7225543137	-0.2799442311
H	-4.5496540940	1.8418230583	2.7788703467
H	-5.0059707492	4.9143249323	-3.4940799035
H	-1.8010257619	6.7716926799	-3.5045644476
H	1.0794495145	1.8599211965	4.4713799916
H	-2.1263496779	0.0022573788	4.4844031297
H	-5.6277457941	0.0112404928	-0.2751913807
H	1.1337834617	-1.9339190855	4.4575256823
H	2.7827028096	4.8819179921	-0.3079673479
H	-2.8626320550	4.8925912884	-6.6191939765
H	-6.7392560036	1.8980229565	-2.8355313030
H	-5.9009761274	1.9003412589	-4.9808038215
H	-2.9652490372	-2.9891705662	2.9063731329
H	-4.7421907937	-1.5425695897	2.6812703383
H	3.6825141373	3.3880181232	2.6165510700

H	4.0988484923	1.1358797480	2.8850135281
H	1.2362329669	6.0435306398	-5.0169197180
H	1.6857971672	6.7820303130	-2.8853878421
H	-6.7563880267	-1.8574679002	-2.8957950163
H	-5.8719413221	-1.8458520565	-5.0230103338
H	-5.8262851408	-3.4843601297	-0.3582284170
H	-4.0091890841	-4.8820453202	-0.1528229992
H	-1.0794444776	-4.1352044513	2.8657879289
H	1.0609450907	-4.9299193367	2.5975097240
H	4.1216875044	-1.0695115617	2.8295171304
H	3.7839250738	-3.3374271147	2.6025077466
H	5.8870110711	3.3451093057	-0.4216190537
H	6.2317284984	1.0794082569	-0.1839094804
H	4.4459501943	4.0538958290	-5.1203835319
H	4.9673577754	4.9414173749	-3.0584173283
H	-0.1999838496	3.4635847756	-8.4331271709
H	-4.5934099006	1.8476849874	-6.7839076612
H	0.6387453033	4.8715038054	-6.8145560557
H	-2.9744127212	1.8613643255	-8.4200346775

Silicon Particle ($Si_{15}H_{20}$)

	X	Y	Z
Si	-1.8326498103	0.1884666153	-4.4209603450
Si	-4.0411889911	0.7292359818	-4.8982925586
Si	-5.4565015957	-0.7764675086	-3.7871797578
Si	-5.6899583912	-0.3005518780	-1.4970147295
Si	-3.7847741353	-0.5142240433	-0.1456414114
Si	-2.1454216737	1.1190537610	-0.5374012855
Si	-2.9812790090	3.2833124807	-0.8852804720
Si	-4.9559471685	3.3622632141	-2.1492578427
Si	-6.5359613909	1.8771719256	-1.2497972162
Si	-0.9126832442	0.3626028443	-2.3610034063
Si	-4.7049362039	2.9340276342	-4.4453590603
H	-4.9529024954	-2.1596380504	-3.9766292359
H	-6.7075679053	-1.2481451872	-0.9634173791
H	-4.2261570766	0.5227810044	-6.3618848130
H	-4.2528203932	-0.3440465796	1.2559276577
H	-1.2291535083	1.1763939269	0.6333888196
H	-3.1744331654	-1.8569753464	-0.2778444459
H	-6.6791291927	2.1691414862	0.2026899767
H	-5.4823762362	4.7510891011	-2.0354092739
H	-3.7198570168	3.8813687245	-5.0324871107
H	-1.9250573117	4.1060756025	-1.5296553576
H	-3.2789434708	3.8815537043	0.4417709435
H	-0.9848201941	-0.1365209706	-5.5923549928
H	0.5413198319	0.1554043233	-2.1637954011
Si	-8.6067285830	2.1187091760	-2.3221717467
Si	-8.3278316855	1.6580581992	-4.6025803492
Si	-6.7938639147	3.1951766898	-5.4939101382
Si	-7.5528542251	-0.5468011857	-4.8257416492
H	-9.5907868107	1.1770678096	-1.7294064431
H	-9.1110491534	3.5032387187	-2.1338838748
H	-9.6299712262	1.8026905163	-5.3063500737
H	-7.2984829629	4.5764371099	-5.2824475209
H	-6.6361162075	2.9765997682	-6.9548810832
H	-8.5397421434	-1.4714538850	-4.2105310204
H	-7.4227791241	-0.8944478593	-6.2644208895

Silicon Particle ($Si_{25}H_{30}$)

	X	Y	Z
Si	-1.8288732924	0.0332496713	-4.6133953704
Si	-4.0400382665	0.6645529746	-4.9002941633
Si	-5.4465227349	-0.7798035968	-3.6983845833
Si	-5.6875762357	-0.1908221631	-1.4576980408
Si	-3.7456126096	-0.4394762897	-0.1649545662
Si	-6.5286391333	2.0246375771	-1.2520538666
Si	-0.8588579239	0.2238828082	-2.5917347150
Si	-2.0605645239	1.0684838252	-0.7961589916
Si	-2.9436793934	3.2741611520	-0.9516746629
Si	-4.9164930919	3.5109220631	-2.1783975777
Si	-4.6186309337	2.8982360498	-4.4674991990
H	-4.9354481111	-2.1700265012	-3.7918524723
H	-4.3484192043	0.4675443847	-6.3447674497
H	-3.2338472941	-1.8278286401	-0.2806990800
H	-3.6034201128	3.8031042216	-5.0721224065
H	-1.9083973135	4.1819228263	-1.5170850431
H	-1.0692714565	-0.3215439513	-5.8341887505
H	0.5924141379	0.0135910135	-2.3852388509
H	-1.0960192519	1.1502529858	0.3373272537
H	-6.7160269483	-1.0982639535	-0.8750923566
Si	-4.2552752761	0.0583497997	2.0701721326
Si	-4.8970175221	2.3091337984	2.2145357712
Si	-6.8838077842	2.5709097487	0.9886875503
Si	-8.5330892112	2.2487268623	-2.4547555688
Si	-8.2268684083	1.6247412842	-4.6862015175
Si	-7.5558101824	-0.6214342223	-4.7140376846
Si	-3.2144459309	3.7023370514	1.3512492251
Si	-6.6113384019	2.9803994909	-5.7135910492
Si	-9.3488284875	4.4057823155	-2.2578882010
Si	-7.7889725193	5.9862622156	-3.0283378069
Si	-5.7525890102	5.7171221028	-1.9870697606
Si	-7.7377847928	4.7444333682	1.1584650305
Si	-9.7744627683	4.8607333245	0.0062622488
Si	-6.2767427378	6.3449494920	0.2302224723
H	-9.5529240831	1.3383667358	-1.8615137216
H	-9.5216618699	1.7777980589	-5.4030576951
H	-7.1382965606	4.3497886929	-5.9013566495
H	-6.2792270938	2.4319675960	-7.0543836418
H	-8.5492693926	-1.4371443193	-3.9698024492
H	-7.4847082654	-1.1217105631	-6.1108334835
H	-5.3698353046	-0.7984624132	2.5492673293
H	-3.0638586559	-0.1939777799	2.9204876148
H	-5.1458918500	2.6789347815	3.6340066812
H	-1.9222589948	3.3535823126	1.9968442261
H	-3.4992817596	5.1238301437	1.6499971395
H	-7.8967273372	1.6276363643	1.5382320294
H	-10.6067053903	4.5247470263	-3.0422501165
H	-10.3416434718	6.2257908895	0.1552412402
H	-10.7476105222	3.8817358301	0.5545868041
H	-7.9383401258	5.0710726663	2.5959585041
H	-4.7918525067	6.7061231838	-2.5483670712
H	-5.0690415362	6.5424373335	1.0597897118
H	-6.9958021535	7.6442246818	0.1524949984
H	-8.2931131612	7.3316037646	-2.6460631435
H	-7.6841349190	5.9492388852	-4.5048863785

Silicon Particle ($Si_{44}H_{44}$)

	X	Y	Z
Si	0.1079080836	0.5664593538	1.7288689966
Si	0.1989840814	-0.0598489464	3.9405322433
Si	3.2894533756	2.0736105752	0.8187907578
Si	2.0146996999	0.1241074122	0.7234754931
Si	3.1868132444	-1.8803310540	1.0035233118
H	-1.0491195392	-0.0912725276	4.7571837979
H	2.4557702417	3.2614638839	0.4905439554
H	-0.5690398652	1.8735912585	1.5570407385
Si	1.5271309677	-1.9333097645	4.4499516562
Si	1.9414309599	-1.8761730582	6.7691606510
Si	3.1608467239	-3.8015019101	7.3570750061
Si	5.2521323318	-3.7741073760	6.2637829560
Si	6.3527929616	-1.8053350904	6.9328377392
Si	5.1755345167	0.0881508941	6.1850626536
Si	4.8142028351	0.0726173309	3.8499939885
Si	6.8573378379	0.0671855176	2.6634030027
Si	8.1888259797	-1.8347536026	3.0234491376
Si	6.9104353354	-3.7775438867	2.6787841064
Si	6.4013273356	-3.7818775560	0.3720388732
Si	5.2046354258	-1.8397243211	-0.2240815201
Si	6.4516997756	0.0816906482	0.3438321973
Si	1.4764761609	1.9007694048	4.3977019373
Si	3.5326415212	1.9584403914	3.1978846962
Si	1.9073576827	-3.7734182327	0.4073503274
Si	-0.1171130421	-3.8667064231	1.5919998148
Si	0.3318773978	-3.8819154215	3.8918552738
Si	1.6086276087	-5.7578683426	4.4910536872
Si	3.6550838679	-5.7193554010	3.3186660823
Si	3.1974479016	-5.6665198662	0.9989344037
Si	3.6159391870	-1.8928837525	3.3209185894
Si	4.8902884338	-3.7939168291	3.9223367022
Si	3.1127964552	0.0942301543	7.3182168748
H	0.6710478014	3.0918157925	4.0015376495
H	4.2832631602	3.1584112976	3.6750160817
H	7.7639772268	0.0008257547	-0.3589089527
H	7.6203893421	1.2950179538	3.0298007321
H	9.2538882997	-1.8293764062	1.9836655811
H	7.6991486018	-3.7785883314	-0.3637147762
H	0.8803785264	-6.9997485147	4.1026567311
H	-0.9431819922	-3.8825049640	4.6657576140
H	3.3780223689	0.0973008018	8.7849593375
H	0.6342844980	-1.8825576430	7.4876154878
H	3.4197583841	-3.7784595205	8.8254777591
H	6.3916435736	-1.7752614038	8.4204859561
H	5.9344740104	1.3179072599	6.5516701554
H	2.3983336648	-6.8840042246	0.6733110927
H	7.7551735030	-1.8060793425	6.4471263101
H	8.8580768039	-1.8323364887	4.3484136247
Si	1.4919544678	-3.7772736742	-1.9064956802
Si	3.5086643386	-3.7849324941	-3.1051432522
Si	4.7493401836	-5.6934205350	-2.5350238572
Si	4.7348070423	-1.8689329803	-2.5293513989
Si	5.2050014125	-5.7223369892	-0.2287765017
H	0.6919298199	-4.9847101988	-2.2485224629
H	0.6906934148	-2.5770458109	-2.2585499182
H	3.2326039096	-3.7818577504	-4.5691296061
H	6.0158767288	-1.8309445455	-3.2856847465

H	3.9638598039	-0.6500255914	-2.8969038266
H	3.9963584642	-6.9215452077	-2.9064715664
H	6.0320384079	-5.7049227855	-3.2895622362
H	-0.8423400886	-5.1069003971	1.2046774814
H	-0.9740349672	-2.7047002851	1.2417060360
Si	5.3553376968	2.0580922062	-0.2851294922
H	5.1540233451	2.1145087654	-1.7575795357
H	6.1858541909	3.2277202546	0.1090950055
Si	1.8934528524	2.0021285927	6.7068987016
H	2.6699471795	3.2234534163	7.0507324399
H	0.6031550471	2.0552737072	7.4455625100
Si	8.2263934553	-5.7014011129	2.9988708574
Si	6.9360043442	-7.6317813629	2.6472561083
Si	6.4095404491	-7.6560590711	0.3581160705
Si	4.9479871559	-7.6040305358	3.9052191697
H	9.2948968370	-5.6607597444	1.9635700545
H	8.8901938674	-5.7080505020	4.3270163029
H	7.6660534469	-7.7172813709	-0.4365669787
H	5.5992347176	-8.8571423582	0.0198479848
H	7.7304916858	-8.8433886732	2.9965857900
Si	1.9980459679	-5.7730313629	6.8073859794
Si	3.2412007903	-7.6886746555	7.3583576581
Si	6.4147464022	-5.6917267804	6.9745793539
Si	5.3040103755	-7.6261680339	6.2349601169
H	4.1742784561	-8.8314774547	3.5587188049
H	6.1069935801	-8.8295373032	6.5930502045
H	2.4742298588	-8.9028377950	6.9694420639
H	3.4740055044	-7.7469286362	8.8264301888
H	6.4446945863	-5.6932519864	8.4623859524
H	7.8193042154	-5.6578538397	6.4950497310
H	0.6914802718	-5.7896708050	7.5246131427

Silicon Particle ($Si_{50}H_{50}$)

	X	Y	Z
Si	1.2699619180	1.3357222701	4.4287132058
Si	0.5706104690	-0.2039438766	5.9105993710
Si	1.7286418225	-2.1926423255	6.2170343509
Si	3.9266119805	-1.9949227168	7.0246125372
Si	5.1905004358	0.7273432768	4.4381451618
Si	5.5220376807	-1.4291647303	5.3645125905
Si	5.4692926648	-3.0414011861	3.6259503031
Si	2.9907580674	-0.8193925467	1.6989458110
Si	3.2066111175	0.9729592501	3.1975489904
Si	3.3150626130	-2.9556690118	2.6522591283
Si	1.7069637721	-3.5279424296	4.2890527996
H	1.6500622876	-0.7814023688	1.0592788572
H	3.4327969532	2.1931701687	2.3727360394
H	0.9974647199	-2.9495390885	7.2720792640
H	0.3558721051	-3.5038008688	3.6710657729
H	-0.5126992756	0.0803134282	6.8798041380
H	0.6839930228	2.6940566495	4.3648828893
Si	4.4102886742	-4.1521482305	7.8448033385
Si	4.2801459612	-5.7442934041	6.1077607128
Si	5.9091035528	-5.2110100196	4.4794805926
Si	8.0402818739	-5.2576962143	5.5014927009
Si	9.7736626004	-4.6574705750	4.0402207956
Si	9.2762034443	-2.6106750742	3.0084403598
Si	9.3099879484	-1.0003466278	4.7268063867
Si	9.0011454218	1.1534305436	3.8443722809
Si	6.8752891981	1.1446373533	2.8396272743
Si	5.3798759078	2.4386886144	6.0397293486
Si	7.4219559538	2.2577805214	7.1752860384
Si	9.1083088859	2.7042527469	5.6061780940
Si	7.6700746758	-1.4909770547	6.3555080763
Si	8.0755420749	-3.6503119257	7.2250705019
Si	6.5150034900	-4.1907945576	8.8945057360
Si	6.6618540245	-2.5803124310	10.6002304522
Si	6.1696574606	-0.4522225404	9.7440473242
Si	3.9963466908	-0.5452528574	8.8751844446
Si	7.1364880024	-2.5775966553	2.0014202096
Si	6.7651795669	-0.4290028254	1.0862554661
Si	4.6365338630	-0.3890423374	0.0728585468
Si	3.2299532390	-4.5587775880	0.9164645878
Si	4.8423840294	-4.0653843160	-0.7240117221
Si	4.4864785264	-1.9533405772	-1.6605985293
Si	2.1448198349	-5.6930070761	5.1065313474
Si	1.9921841758	-7.2660383913	3.3814748829
Si	3.6032317906	-6.7151439927	1.7792457037
Si	5.7555133837	-6.7812916549	2.7205951538
Si	7.7322474292	0.1123669523	8.0826961366
Si	10.9160777612	-2.1014638976	1.4014894274
Si	10.5441415435	0.0271460096	0.5000392403
Si	10.6365296166	1.6146939576	2.2211718201
Si	8.4283789094	0.0632473613	-0.5028500019
Si	7.2906977961	-6.3665151868	0.9993278810
Si	6.9745044750	-4.1658961151	0.2603028976
Si	4.7093972446	-7.8722192290	7.0121047124
Si	6.8265129636	-7.9185295361	8.0139787265
Si	6.9323758575	-6.3418867857	9.7424733770
Si	8.4434039011	-7.3913054349	6.4044774667
H	5.9979607910	-8.1382942942	3.2844082554

H	4.7748390983	-5.1158382259	-1.7771581000
H	3.5612291756	-7.6858754666	0.6514545713
H	1.8743688662	-4.5106156882	0.3024164932
H	4.4019699548	0.9841535072	-0.4522175521
H	6.5930181345	2.4972803817	2.2815782974
H	9.4438447241	-3.6459490686	7.8161584651
H	1.1348933459	-5.9944591577	6.1582118899
H	3.3612742788	-4.4783732308	8.8516770062
H	10.6409731161	-1.0575015756	5.3952960568
H	12.2549100777	-2.1595396185	2.0434070763
H	10.8706159653	-3.1316583022	0.3314152684
H	11.5875645975	0.3322925207	-0.5160413038
H	8.1542099144	1.3985601615	-1.0931120689
H	8.3663966405	-0.9513642465	-1.5859356324
H	11.9802199887	1.6007514349	2.8556180616
H	10.3999477091	2.9761708208	1.6743732148
H	11.0015157401	-4.4596723608	4.8557062254
H	10.0512049758	-5.7203785868	3.0443251683
H	5.4996959367	-1.6862206401	-2.7132842261
H	3.1367317503	-1.8788441888	-2.2756160807
H	0.6329721502	-7.2186081256	2.7844442982
H	2.2302352827	-8.6392618873	3.8952548058
H	5.3581919155	3.7291267187	5.3004336068
H	4.2243670999	2.4335717642	6.9709037772
H	8.9420336786	4.0842848947	5.0797008934
H	10.4473389864	2.6145778968	6.2441660631
H	7.4724920254	3.2676067995	8.2675329228
H	9.0887761348	0.0731266117	8.6969195612
H	6.2284459729	0.5593706734	10.8340647545
H	8.0405608499	-2.6038456791	11.1534404419
H	5.7158263023	-2.9088833190	11.6981813538
H	3.0826128022	-1.1009011371	9.9089118121
H	3.4892812146	0.8003200022	8.5085289133
H	8.6837154779	-6.6005288639	1.4449117553
H	6.9982450032	-7.2847298814	-0.1325413732
H	7.9877228169	-3.8212602894	-0.7749305503
H	3.6583129045	-8.1919143259	8.0127144156
H	4.6472044364	-8.8943223126	5.9351574928
H	5.9254135613	-6.6545946128	10.7895339431
H	8.2777432285	-6.3706855158	10.3727812413
H	7.0898785969	-9.2780677306	8.5589184002
H	8.4188926116	-8.4032801972	5.3163387979
H	9.7972547320	-7.4047069158	7.0170470991

Trimelem ($C_{18}N_{27}H_9$)

	X	Y	Z
N	-3.3533847378	0.1907807841	10.5234496757
C	-2.2097116774	0.0465354605	9.8810737520
N	-2.2207653613	-0.1110901351	8.4863377096
C	-3.4278347017	-0.0556651488	7.7718675467
N	-4.5619264882	0.0883617685	8.4309393187
C	-4.4671038171	0.1881854323	9.7690142496
C	-1.0309661389	-0.4021322606	7.8133550614
N	-1.0516137610	-0.5838266797	6.5050408153
C	-2.1996343120	-0.3221751816	5.8905255061
N	-3.3978398279	-0.1209818921	6.4413979464
N	-1.0414621004	0.0761024550	10.5212618158
C	0.0435008596	-0.1326066001	9.7736221800
N	0.0984548030	-0.4843824250	8.4940790115
N	-2.1651053483	-0.2914889126	4.5079928543
C	-1.0314530781	-0.0423627589	3.7567592826
N	-1.0352982362	-0.5219236822	2.5122807662
C	0.1076034673	-0.3991094260	1.8362600870
N	1.2340000266	0.1181003289	2.4941823521
C	1.0967177273	0.6789556214	3.7643473904
N	-0.0829908501	0.6683575056	4.3553809956
C	2.4904355324	0.1246116932	1.8675434231
N	2.5823181101	-0.2693728242	0.6102938561
C	1.4402180585	-0.6738607236	0.0264402892
N	0.2174409896	-0.7845750208	0.5795530357
N	2.1551864810	1.1781473884	4.3790525984
C	3.3257550341	0.9426257594	3.8098190969
N	3.5586757216	0.4963751354	2.5683094113
N	1.5318542425	-1.0249914147	-1.2627945874
N	4.4715318308	1.1371858338	4.5663360855
C	4.5611198052	1.0521024055	5.9474594022
N	5.7697759876	0.6889519644	6.3966514220
C	5.8552250812	0.4145819661	7.6958630982
N	4.6834529692	0.4135594569	8.4694378550
C	3.4939328994	0.8876504526	7.9153502075
N	3.4733624928	1.2976185594	6.6589193419
C	4.7081306720	-0.0098134679	9.8072827916
N	3.5552108500	-0.1351097313	10.4657642897
C	2.4619869303	0.2537568678	9.8078780783
N	2.3911893081	0.8821609135	8.6403367817
N	5.8651270748	-0.3089335953	10.3658005532
C	6.9520523398	-0.2072457124	9.5773506901
N	7.0041923159	0.1099285545	8.2713877753
N	1.2544070110	-0.0033527263	10.4300500543
N	8.1266061243	-0.4705088553	10.1647461632
N	-5.6246542543	0.3083239654	10.4317293319
H	2.4316401888	-0.9855320072	-1.7197193098
H	0.7073140749	-1.3620368209	-1.7386842801
H	-3.0306655137	-0.4752359731	4.0136332439
H	5.3309932163	0.8952156152	4.0820117755
H	8.1385455573	-0.7408353330	11.1377363129
H	8.9710021400	-0.4325563079	9.6120724401
H	-6.4884270014	0.3358429726	9.9091477694
H	-5.6061706655	0.4131825663	11.4359956199
H	1.2551186739	-0.1177258244	11.4371456416

Hexamelem ($C_{36}N_{52}H_{12}$)

	X	Y	Z
N	-3.2672721284	-1.1468226535	10.7538463018
C	-2.1664427271	-0.9309695605	10.0639891857
N	-2.2623355942	-0.6873506921	8.6950861021
C	-3.4935255521	-0.8378856263	8.0429033791
N	-4.5927310457	-0.9015892141	8.7795379445
C	-4.4097437317	-0.9515093168	10.1094551541
C	-1.1289026310	-0.3096485158	7.9782621655
N	-0.0216666681	0.0006217636	8.6501730280
C	0.0037019581	-0.3673628729	9.9158447711
N	-0.9669053795	-0.9132919825	10.6320872667
N	-3.4923397212	-0.9237500051	6.7153311901
C	-2.3088479443	-0.7559030846	6.1219032993
N	-1.1865616684	-0.2862833221	6.6649245160
N	1.2568715030	-0.2237854357	10.5960949062
C	2.3464071452	-0.9927818398	10.0760204119
N	2.6418601851	-0.7853540333	8.8070269623
C	3.4985540467	-1.6423028749	8.2595734337
N	4.2288639760	-2.4908814920	9.0897956565
C	3.9359163483	-2.5415685402	10.4544872968
N	2.8787939893	-1.8761491615	10.9059509561
N	3.6613625067	-1.7283329841	6.9567599827
C	4.4185437059	-2.7331219431	6.5254297384
N	5.2794414523	-3.4677418868	7.2344274131
C	5.2268961615	-3.3154121797	8.5539552342
N	4.7100179218	-3.2556940436	11.2439581830
C	5.8039375938	-3.7740872887	10.6909848525
N	6.0651916054	-3.9188632052	9.3866734884
N	4.3248953376	-3.0308358704	5.1829576030
C	3.2012119455	-2.8032553869	4.4071722687
N	2.0511713557	-2.6766019300	5.0565436428
C	1.0079657410	-2.2917330662	4.3457567476
N	1.1280190107	-2.1727385444	2.9589354560
C	2.3431558106	-2.5008945971	2.3349712644
N	3.4030120137	-2.7822480935	3.0917528095
N	2.4115972990	-2.4963159222	1.0196415402
C	1.2905571005	-2.1604541614	0.3581838071
N	0.1035655457	-1.8035311622	0.8787819732
C	0.0106338239	-1.8018936535	2.1925113891
N	-0.1477270930	-2.0187698712	4.9219925703
C	-1.0994301715	-1.5186919762	4.1439471484
N	-1.1076480285	-1.4328134120	2.8159761426
N	1.3665747134	-2.1787051533	-0.9763691812
N	-2.2449844815	-1.0767530028	4.7828220478
N	6.7584308847	-4.2636312647	11.5529105102
N	-5.5530365646	-0.7518487822	10.8541177408
H	7.3999973546	-4.9587830211	11.1975759786
H	5.1007459821	-3.5115796328	4.7492078463
H	-6.3877826322	-0.5838367483	10.3063881664
H	2.2393061079	-2.4207828637	-1.4146451327
H	0.5594985010	-1.9176459134	-1.5175059325
H	-3.1055085179	-1.0807486391	4.2530827905
C	1.3045879825	0.2961840309	11.8807428641
N	0.1815076795	0.8162635383	12.3689364597
N	2.4891386406	0.2674129865	12.4850942199
C	2.4820200751	0.4350968547	13.7920603023
C	0.1464759565	1.0147210335	13.6745669146
N	1.2801781687	0.7289098191	14.4431325054

N	-0.9472676713	1.4349675911	14.2876553506
N	3.5789290118	0.2647266225	14.5131991812
C	1.2417293694	0.8330387663	15.8413827905
C	3.4176614477	0.1974230738	15.8209149265
C	-0.9397160099	1.3913556497	15.6128792668
N	0.0934265165	1.1594862187	16.4272679785
N	2.3294328003	0.5272928544	16.5348839002
N	-2.1511078492	1.6351580866	16.2384588586
N	4.4565154604	-0.3137984454	16.5826727782
C	-3.3823431061	1.3704964527	15.6745708673
H	-2.1297242170	1.9891717267	17.1843836095
N	-4.4200620830	2.0412373635	16.1725652312
N	-3.3969060430	0.4507272310	14.7164483527
C	-4.4990898124	0.3614732078	14.0034990026
N	-4.5437402950	-0.3861876417	12.9137436586
C	-5.6093630583	-0.2477036283	12.1486356043
N	-6.7578891255	0.3671226759	12.4470369947
C	-6.7963937537	1.0128336503	13.6104372189
N	-5.6231067735	1.0979014824	14.3785023106
C	-5.5847807616	1.8759228566	15.5468737973
N	-6.6931888820	2.4513901402	15.9679208970
N	-7.8906549498	1.6023755452	14.0488562300
C	-7.7909605924	2.2672008444	15.2134185479
N	-8.9179168492	2.8240772958	15.6703958777
H	-8.8926824788	3.3486289413	16.5285172843
H	-9.7639540836	2.7299730063	15.1342114175
C	5.3998361064	-1.2353514381	16.1609808194
N	5.9159184074	-1.9813022751	17.1446102690
N	5.6634378759	-1.3173257968	14.8671008076
C	6.3038394431	-2.3983437275	14.4647619471
N	6.4325245360	-2.6722571649	13.1796820192
C	6.8902479856	-3.8785645884	12.8787673106
N	6.8155539200	-3.2913565339	15.4072017704
C	6.6824300878	-3.0022882994	16.7758329307
N	7.4834451411	-4.7543887034	13.6854107632
C	7.5015650823	-4.4395708359	14.9816829091
N	8.1064022399	-5.1934364838	15.8754514134
N	7.2954502868	-3.7733871334	17.6535347922
C	7.9947579776	-4.8086878779	17.1607769455
N	8.6485168639	-5.5582017504	18.0550449656
H	8.5838210930	-5.3247682515	19.0315276451
H	9.1708405427	-6.3570884441	17.7371559556
H	4.2608453511	-0.3525608660	17.5753065198

APPENDIX B

SUPPORTING INFORMATION FOR: ROBUST AND EFFICIENT SPIN PURIFICATION FOR DETERMINANTAL CONFIGURATION INTERACTION

B.1 Demonstration that Spin Contamination Arises from Numerical Error

To better understand the behavior of wave function convergence using a typical iterative eigenvalue solver, we have calculated the 20 lowest doublet states of the ethylene anion at the CASCI/6-31G** level of theory using Hartree-Fock canonical orbitals (HF-CASCI) with geometry as depicted in Figure B.1 and Cartesian coordinates reported below. A (7,8) active space is employed. We have used a standard Davidson-Liu algorithm with guess vectors as described in the main document and with a Hamiltonian subspace dimension of 400. Eigenvectors were converged using a residual norm threshold of $||r|| = 1.0 \times 10^{-6}$, a value generally considered suitable for calculation of analytic energy gradients or orbital gradients (i.e. for the complete active space self-consistent field, CASSCF, method). The Evangelisti preconditioner is used. The $\langle \hat{S}^2 \rangle$ value was determined for each root at each iteration and is given in Figure B.1.

As is evident in Figure B.1, the $\langle \hat{S}^2 \rangle$ values for many roots diverge from the desired $\langle \hat{S}^2 \rangle$ value beginning at iteration 7. Of the twenty roots calculated, only thirteen converge to doublet states ($\langle \hat{S}^2 \rangle = 0.75$), while the remaining seven converge to quartet states ($\langle \hat{S}^2 \rangle = 3.75$). Of course, the average value of $\hat{S}^2$ does not guarantee that the state is a pure spin eigenfunction. Instead we must consider a sufficient condition for verifying the purity of the states, the variance of $\hat{S}^2$, or $\sigma_v = \langle \hat{S}^4 \rangle - \langle \hat{S}^2 \rangle^2$. Each converged state was indeed found to be a pure spin eigenfunction, having a variance less than 1×10^{-10}.

To obtain a clearer picture of the nature of this contamination, we focus on a representative root (root 7). The $\langle \hat{S}^2 \rangle$ value was calculated at each of four points of the Davidson-Liu

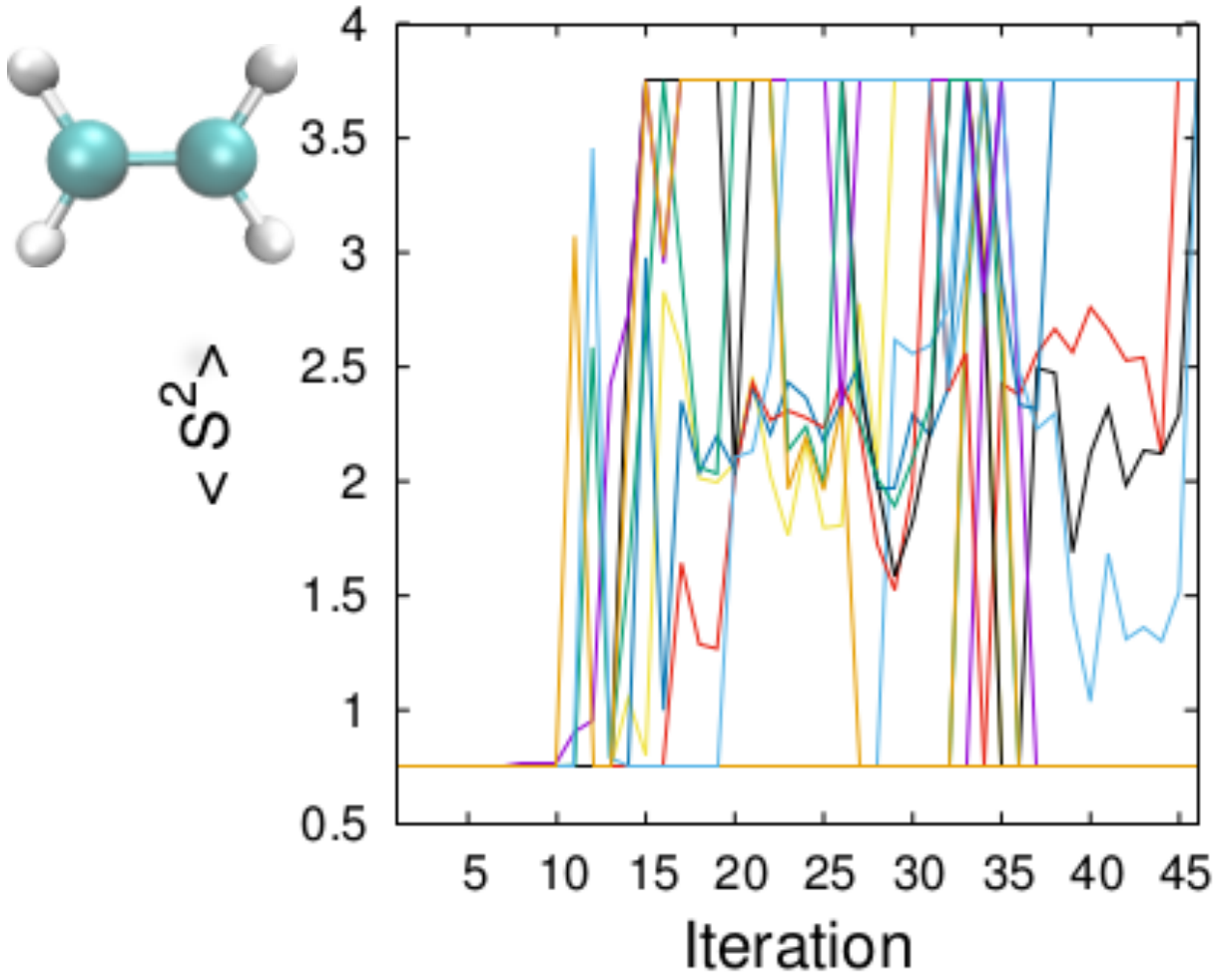

Figure B.1: $\langle \hat{S}^2 \rangle$ for 20 roots of $C_2H_4^-$ using HF-CAS-(7,8)-CI/6-31G**. A standard Davidson-Liu algorithm was used with orbital energy difference preconditioning.

algorithm, after residual vector formation, preconditioning, trial vector orthogonalization, and $\boldsymbol{\sigma}$ formation. These values are depicted in Figure B.2. The inset shows the region surrounding iterations $5-10$ in greater detail. In iteration 8, a small amount of spin contamination ($\langle \hat{S}^2 \rangle = 0.75005224727341$) in the residual vector is exacerbated by each of the preconditioning, orthogonalization, and $\boldsymbol{\sigma}$ vector formation steps. Careful examination of $\langle \hat{S}^2 \rangle$ at each of the four test locations in earlier iterations for each of the contaminated roots reveals that a small amount of spin contamination ($\sim 1 \times 10^{-12}$) occurs with approximately equal probability at each of the test locations, suggesting that the contamination is random and numerical in nature.

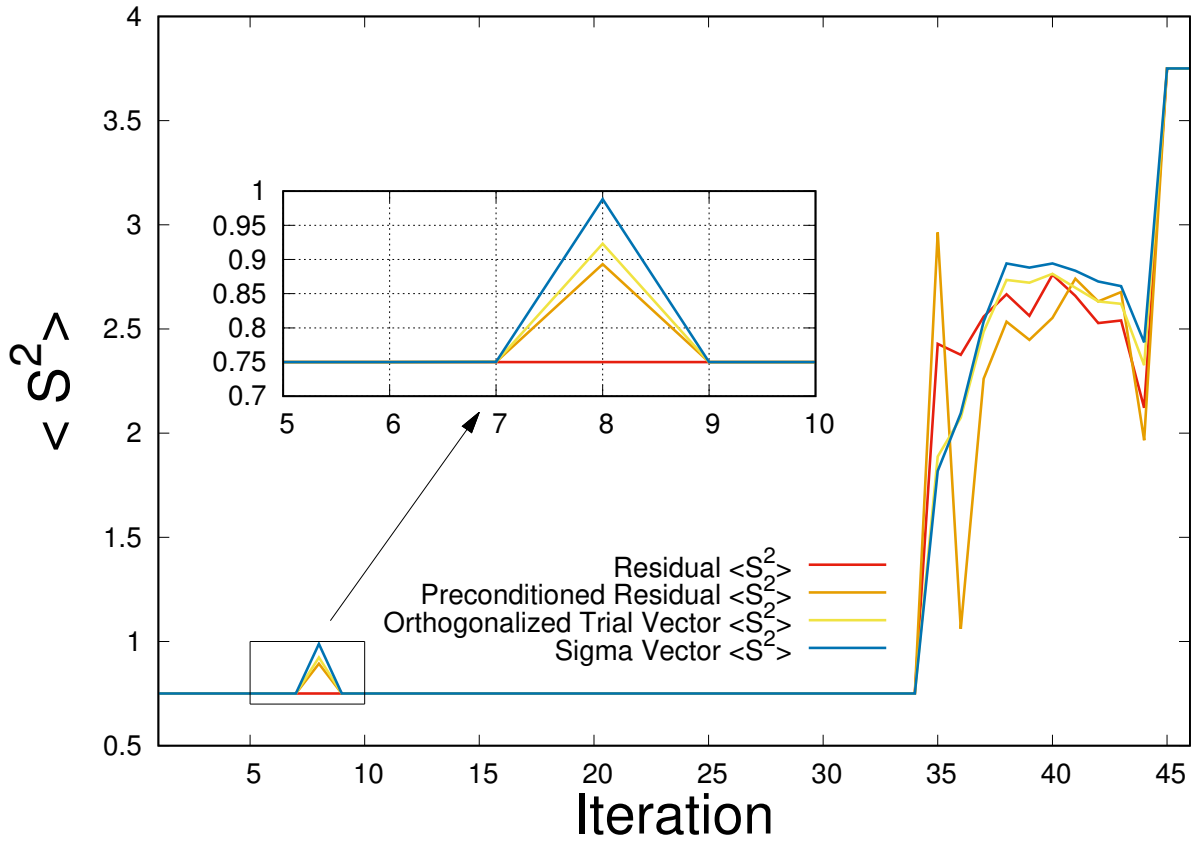

Figure B.2: $\langle \hat{S}^2 \rangle$ for root 7 of $C_2H_4^-$ using HF-CAS-(7,8)-CI/6-31G**. A standard Davidson-Liu algorithm was used with orbital energy difference preconditioning.

This phenomenon is reproducible across a wide range of active spaces where differing numbers of roots are requested, with the trend being that contamination occurs with increasing frequency as the number of requested roots and the size of the configuration space is increased. Additionally, spin contamination can be observed in moderately sized configuration spaces where fewer roots are requested by arbitrarily tightening the residual convergence threshold. Each of these observations supports the hypothesis that numerical instability is the dominant cause of the spin contamination in this case, motivating us to seek a technical solution to the problem.

B.2 Interior Spin States, Second-Order Projection, and Spin Purity Thresholds

In the paper we only considered states with maximal m_s (singlets with $m_s = 0$, doublets with $m_s = 1/2$, etc.), but in principle our spin purification schemes can target states with lower m_s. Though computing such interior spin states is not a common practice, we have investigated the effectiveness of our methods on such states. We calculate the lowest 12 $m_s = 0.5$ quartet states of the ethylene anion, using spin penalty, first-order projection, and Löwdin projection purification, and present the results in Table B.1.

Table B.1: Number of iterations required and times-to-solution for convergence of 12 quartet states of anionic ethylene at the HF-CAS-(13,12)-CI/6-31G** level. Results are shown for the spin penalty method as a function of the spin penalty parameter, α, and for the first-order and Löwdin spin projection methods, with $||r|| = 1.0 \times 10^{-6}$ in both cases. The number of states converging to the incorrect spin symmetry are given in parentheses.

α	Iterations	σ Formations	CI Time-to-Solution (s)
		Spin Penalty	
0.00	52 (9)	361	550.65
0.01	62 (8)	356	586.29
0.02	88 (5)	343	624.77
0.05	64 (1)	336	581.99
0.10	79	431	723.83
0.15	96	538	1073.61
0.20	117	619	1145.88
		First-Order Projection	
—	30	234	2125.58
		Löwdin Projection	
—	—	—	—

In contrast to results given for the singlet and doublet states, where projection purification excels at facilitating convergence to the correct eigenvectors, penalty purification provides a more efficient strategy than either of the projection methods (first-order projection has difficulty purifying the trial vectors, leading to a large time-to-solution, and Löwdin projection calculation fails to converge) when $m_s = 1/2$ quartet states are sought. The complexity of the problem is reflected in the large α values required to allow penalty method purified CI

convergence. A penalty method time-to-solution of 723.83 seconds, with a value of $\alpha = 0.10$, is nearly 3× faster than the solution provided by the first-order projection method (2125.58 s).

It is conceivable that a higher-order projector could promote better performance of the projection method. To test this idea we implemented a second-order projector according to

$$\boldsymbol{\chi} = \left(\mathbf{S}^2 - \mathbf{I}\langle\hat{S}^2\rangle_{target}\right)^2 \mathbf{c} \tag{B.1}$$

A second test system, a small Ag_{11} icosahedral cluster with geometry given in Cartesian coordinates below, was used to provide an additional example of behavior of each of the spin penalty, first-order projection, and Löwdin projection methods while evaluating the relative performance of the second-order projector. Calculation of the lowest 20 doublet states of Ag_{11} at the HF-CAS-(11,11)-CI/LANL2DZ level of theory with various forms of spin purification is described in Table B.2, and the analogous quartet calculation results are presented in Table B.3.

Table B.2: Number of iterations required and times-to-solution for convergence of 20 doublet states of Ag_{11} at the HF-CAS-(11,11)-CI/LANL2DZ level. Results are shown for the spin penalty method as a function of the spin penalty parameter, α, and for each of the first-order, second-order, and Löwdin spin projection methods, with $||r|| = 1.0 \times 10^{-6}$ in each case. The number of states converging to the incorrect spin symmetry are given in parentheses.

α	Iterations	σ Formations	CI Time-to-Solution (s)
		Spin Penalty	
0.00	30 (6)	343	128.51
0.01	33	515	217.45
0.02	50	736	317.81
0.05	90	1214	536.20
0.10	124	1821	831.92
0.15	174	2387	1134.63
0.20	212	2837	1315.25
		First-Order Projection	
—	17	292	335.98
		Second-Order Projection	
—	17	293	4832.94
		Löwdin Projection	
—	17	291	375.26

For the doublet states, the first-order projection method is slightly more efficient than the Löwdin projector (335 vs 375 seconds), while the second-order projector performs very poorly. Interestingly the penalty method gives the fastest solution (at a value $\alpha = 0.01$) of only 217.45 seconds. The mediocre performance of the second-order projector is surprising, but can be understood by considering the effect of multiple applications of the projector to a trial vector. A single application acts to remove the contaminating component having the largest magnitude, and subsequent applications tend to remove contaminants having decreasing magnitude. Unfortunately, each additional projection may reintroduce contamination through numerical instability, resulting in overall worse purification behavior than both first-order and Löwdin projection.

Table B.3: Number of iterations required and times-to-solution for convergence of 20 quartet states of Ag_{11} at the HF-CAS-(11,11)-CI/LANL2DZ level. Results are shown for the spin penalty method as a function of the spin penalty parameter, α, and for the first-order and Löwdin spin projection methods, with $||r|| = 1.0 \times 10^{-6}$ in each case. The number of states converging to the incorrect spin symmetry are given in parentheses.

$\boldsymbol{\alpha}$	**Iterations**	$\boldsymbol{\sigma}$ **Formations**	**CI Time-to-Solution (s)**
		Spin Penalty	
0.00	26 (14)	366	131.04
0.01	36 (1)	534	236.33
0.02	58	767	346.62
0.05	99	1258	574.43
0.10	138	1844	874.89
0.15	173	2375	1126.56
0.20	223	2853	1321.49
		First-Order Projection	
—	—	—	—
		Löwdin Projection	
—	19	289	1631.02

The $m_s = 1/2$ quartet state calculation of the Ag_{11} cluster behaves similarly as the doublet state calculation, where penalty purification achieves superior results when compared to the projection methods, requiring only 346 seconds ($\alpha = 0.02$) vs 1631 seconds for Löwdin projection. For this system, first-order projection fails to purify the trial vectors sufficiently, resulting in convergence failure of the CI calculation. It is clear that the effectiveness of the purification method depends strongly on the nature of the configuration space, with the overall trend being that first-order projection purification provides the best performance in cases where projection works, but in extreme cases penalty purification exhibits the most robust behavior of all methods, albeit at the cost of determining the appropriate parameter α.

Projection purification depends on the threshold for spin purity of the trial vector. Table B.4 shows the effect of varying the value between $1.0 \times 10^{-9} - 1.0 \times 10^{-15}$ using first-order purification applied to the ethylene anion at HF-CAS-(7,8)-CI/6-31G** with $||r|| = 1.0 \times 10^{-6}$. The results shown are applicable to all projection type methods, since the

purification threshold is independent of how the trial vector is purified.

Table B.4: CI convergence of 20 doublet states of ethylene anion at HF-CAS-(7,8)-CI/6-31G**, $||r|| = 1.0 \times 10^{-6}$, as a function of trial vector spin purity threshold using first-order projection.

Threshold	**Iterations**	**Number of Projections**
1.0×10^{-9}	—	—
1.0×10^{-10}	40	2053
1.0×10^{-11}	40	2260
1.0×10^{-12}	40	2482
1.0×10^{-13}	40	2700
1.0×10^{-14}	40	2915
1.0×10^{-15}	40	3156

A minimum threshold of 1.0×10^{-10} is required to ensure spin pure convergence of the CI problem (the case with a threshold of 1.0×10^{-9} fails to converge), but further tightening the threshold does not improve CI convergence, and only serves to incur additional computational cost due to the increased number of projection operations. The threshold is expected to change as the CI convergence criteria is changed (i.e. if we require $||r|| = 1.0 \times 10^{-8}$, we might expect a purity threshold of 1.0×10^{-14} to be more suitable), but since the residual tolerance is fixed at $||r|| = 1.0 \times 10^{-6}$ for CI calculations where orbital or analytic energy gradients are desired, a threshold of 1.0×10^{-10} seems to be generally applicable to all configuration spaces.

B.3 Preconditioning for Penalty Purification

Here we consider that, in the context of penalty purification, the preconditioner may need to be altered to be consistent with the penalized Hamiltonian. Both the Davidson and Evangelisti preconditioners were modified to account for the constant diagonal element shift of $\alpha \left(\mathrm{S}_{ii}^2 - \langle \hat{S}^2 \rangle_{target} \right)^2$, and we have tested the modified preconditioners by calculating the 20 lowest doublet states of the ethylene anion in Figure B.1 using HF-CAS-(7,8)-CI/6-31G**, $||r|| = 1.0 \times 10^{-6}$,and $\alpha = 0.10$ as shown in Table B.5. Surprisingly, the unshifted precon-

Table B.5: Number of iterations required for convergence of 20 states of anionic ethylene using HF-CAS-(7,8)-CI/6-31G** as a function of preconditioner choice used with the spin penalty purification method, $\alpha = 0.10$ and $||r|| = 1.0 \times 10^{-6}$.

Preconditioner	**Iterations**
Davidson Unshifted	51
Davidson Shifted	57
Evangelisti Unshifted	40
Evangelisti Shifted	43

ditioner outperforms the shifted preconditioner using both Davidson and Evangelisti style preconditioning. A more careful look at the nature of the Hamiltonian modification provides some insight into this behavior. The contribution from the $\mathbf{S^2}$ matrix is non-diagonally dominant. For example consider single excitations from a closed shell reference. Off-diagonal elements between coupled determinants have values of ± 1, while diagonal elements also have value of one, for both of the occupation-equivalent α and β singly-excited determinants. An alternative way of viewing the problem is that the open-shell singlet and triplet configurations are indistinguishable in the preconditioner whether or not the spin penalty contribution is included. By neglecting the penalty in the preconditioner neither state is penalized, and including it penalizes both states, leading to diminished convergence.

B.4 Ag_{19} SA-CASSCF CI Vector Coefficients and Orbital Occupancies

Table B.6: Orbital occupancies and CI vector coefficients for the 12 lowest doublet states of Ag_{19} calculated using SA-12-CAS-(11,11)-SCF/LANL2DZ. Table entries "X" denote doubly occupied orbitals, "/" α electron occupied orbitals, and "\" β electron occupied orbitals.

State	Coefficient	Orbital										
		176	177	178	179	180	181	182	183	184	185	186
D_0	0.931	X	X	X	X	X	/					
D_1	0.931	X	X	X	X	/	X					
D_2	0.907	X	X	X	X	X		/				
D_2	-0.134	X	X	X	X	X						/
D_2	-0.121	X	X	X	X		X	/				
D_3	0.925	X	X	X	X	X			/			
D_3	-0.117	X	X	X	X		X		/			
D_4	-0.907	X	X	X	X	X				/		
D_4	0.116	X	X	X	X		X			/		
D_4	-0.116	X	X	X	X	X						/
D_5	0.888	X	X	X	X	X					/	
D_5	0.201	X	X	X	X	X						/
D_5	0.131	X	X	X	X	/	\			/		
D_6	-0.805	X	X	X	X	X						/
D_6	0.270	X	X	X	X	/	\		/			
D_6	0.230	X	X	X	X	/	/		\			
D_6	0.181	X	X	X	X	X					/	
D_6	-0.141	X	X	X	X	X		/				
D_6	0.136	X	X	X	X	X				/		
D_7	-0.868	X	X	X	/	X	X					
D_7	0.290	X	X	/	X	X	X					
D_8	0.730	X	X	X	X	/	/	\				
D_8	-0.452	X	X	X	X	\	/	/				
D_8	0.277	X	X	X	X	/	\	/				
D_8	0.183	X	X	X	X	/	/			\		
D_8	0.132	X	X	X	X	/	\			/		
D_9	-0.631	X	X	X	X	/	/			\		
D_9	-0.444	X	X	X	X	/	\			/		
D_9	-0.368	X	X	X	X	\	/	/				
D_9	-0.256	X	X	X	X	/	\	/				
D_9	0.187	X	X	X	X	\	/			/		
D_9	0.119	X	X	X	X	X					/	
D_9	0.113	X	X	X	X	/	/	\				
D_{10}	0.630	X	X	X	X	/	\	/				
D_{10}	0.466	X	X	X	X	\	/	/				
D_{10}	-0.329	X	X	X	X	/	/			\		
D_{10}	0.199	X	X	X	X	/	/					\

Continued on next page

Table B.6 *(cont'd)*

State	Coefficient	Orbital										
		176	177	178	179	180	181	182	183	184	185	186
D_{10}	-0.178	X	X	X	X	/	\			/		
D_{10}	0.164	X	X	X	X	/	/	\				
D_{10}	0.152	X	X	X	X	\	/			/		
D_{10}	-0.101	X	X	X	X	\	/					/
D_{11}	-0.692	X	X	X	X	/	\		/			
D_{11}	-0.420	X	X	X	X	/	/		\			
D_{11}	-0.338	X	X	X	X	X						/
D_{11}	-0.272	X	X	X	X	\	/		/			

B.5 Supporting Geometries

We provide Cartesian coordinates for the ethylene and silver structures used in this work in units of Angstrom (Å).

Ethylene (C_2H_4)

	X	Y	Z
C	0.35673483	-0.05087227	-0.47786734
C	1.61445821	-0.06684947	-0.02916681
H	-0.14997206	0.87780529	-0.62680155
H	-0.16786485	-0.95561368	-0.69426370
H	2.15270896	0.84221076	0.19314809
H	2.16553127	-0.97886933	0.15232587

Small Silver Cluster (Ag_{11})

	X	Y	Z
Ag	0.958652	-0.279967	1.212703
Ag	0.005323	2.137448	0.018571
Ag	1.143187	-2.160336	-0.874374
Ag	3.518286	-0.894511	0.133855
Ag	1.683504	0.501181	-1.540101
Ag	-0.956338	-0.279447	-1.208969
Ag	-1.697899	0.495065	1.544716
Ag	-2.785287	1.764762	-0.708986
Ag	-3.520960	-0.880522	-0.147912
Ag	2.807778	1.753710	0.692652
Ag	-1.156247	-2.157383	0.877846

Large Silver Cluster (Ag_{19})

	X	Y	Z
Ag	-0.223357	-1.023232	2.404318
Ag	-2.602282	0.562832	-2.463318
Ag	-2.573188	-2.352427	-1.548365
Ag	2.203440	0.549933	-2.390372
Ag	3.773015	-0.174255	-0.004604
Ag	-0.226951	-1.024053	-2.405452
Ag	2.128632	2.312016	0.001861
Ag	-2.600437	0.560283	2.465417
Ag	2.212257	0.548246	2.386126
Ag	-0.224276	1.789238	1.606984
Ag	2.214150	-2.236559	1.483518
Ag	-1.727027	-0.242038	-0.000106
Ag	-0.226709	-2.731956	0.002640
Ag	-2.470578	2.409640	0.003766
Ag	1.051719	-0.221501	0.000342
Ag	-0.118414	4.076477	-0.001229
Ag	-2.572912	-2.352702	1.549742
Ag	-0.225624	1.789461	-1.605864
Ag	2.208543	-2.239402	-1.485402

APPENDIX C

SUPPORTING INFORMATION FOR: LARGE SCALE ELECTRONIC CORRELATION CALCULATIONS: RANK-REDUCED FULL CONFIGURATION INTERACTION

C.1 Supporting Geometries

We provide Cartesian coordinates for the ethylene dimer and acene structures used in this work in units of Angstrom (Å).

Ethylene Dimer $(C_2H_4)_2$

	X	Y	Z
C	0.35673483	-0.05087227	-0.47786734
C	1.61445821	-0.06684947	-0.02916681
H	-0.14997206	0.87780529	-0.62680155
H	-0.16786485	-0.95561368	-0.69426370
H	2.15270896	0.84221076	0.19314809
H	2.16553127	-0.97886933	0.15232587
C	6.35673483	-0.05087227	-0.47786734
C	7.61445821	-0.06684947	-0.02916681
H	5.85000000	0.87780529	-0.62680155
H	5.84000000	-0.95561368	-0.69426370
H	8.15270896	0.84221076	0.19314809
H	8.16553127	-0.97886933	0.15232587

Singlet Naphthalene ($C_{10}H_8$)

	X	Y	Z
C	-2.433661	0.708302	0.000000
C	-2.433661	-0.708302	0.000000
H	-3.378045	-1.245972	0.000000
H	-3.378045	1.245972	0.000000
C	-1.244629	1.402481	0.000000
C	-1.244629	-1.402481	0.000000
C	-0.000077	0.717168	0.000000
C	-0.000077	-0.717168	0.000000
H	-1.242734	2.490258	0.000000
H	-1.242734	-2.490258	0.000000
C	1.244779	1.402533	0.000000
C	1.244779	-1.402533	0.000000
C	2.433606	0.708405	0.000000
C	2.433606	-0.708405	0.000000
H	1.242448	2.490302	0.000000
H	1.242448	-2.490302	0.000000
H	3.378224	1.245662	0.000000
H	3.378224	-1.245662	0.000000

Triplet Naphthalene ($C_{10}H_8$)

	X	Y	Z
C	2.488282	0.681831	0.000000
C	2.488282	-0.681831	0.000000
H	3.419575	-1.241120	0.000000
H	3.419575	1.241120	0.000000
C	1.238411	1.400670	0.000000
C	1.238411	-1.400670	0.000000
C	-0.000001	0.725296	0.000000
C	-0.000001	-0.725296	0.000000
H	1.247313	2.487396	0.000000
H	1.247313	-2.487396	0.000000
C	-1.238411	1.400665	0.000000
C	-1.238411	-1.400665	0.000000
C	-2.488281	0.681833	0.000000
C	-2.488281	-0.681833	0.000000
H	-1.247325	2.487390	0.000000
H	-1.247325	-2.487390	0.000000
H	-3.419563	1.241141	0.000000
H	-3.419563	-1.241141	0.000000

Singlet Anthracene ($C_{14}H_{10}$)

	X	Y	Z
C	-3.660857	0.713070	0.000000
C	-3.660857	-0.713070	0.000000
H	-4.607652	-1.246404	0.000000
H	-4.607652	1.246404	0.000000
C	-2.479632	1.407088	0.000000
C	-2.479632	-1.407088	0.000000
C	-1.224085	0.722636	0.000000
C	-1.224085	-0.722636	0.000000
H	-2.477321	2.494657	0.000000
H	-2.477321	-2.494657	0.000000
C	0.000029	1.403284	0.000000
C	0.000029	-1.403284	0.000000
C	1.224035	0.722646	0.000000
C	1.224035	-0.722646	0.000000
H	-0.000033	2.491609	0.000000
H	-0.000033	-2.491609	0.000000
C	2.479679	1.407102	0.000000
C	2.479679	-1.407102	0.000000
C	3.660841	0.713104	0.000000
C	3.660841	-0.713104	0.000000
H	2.477242	2.494663	0.000000
H	2.477242	-2.494663	0.000000
H	4.607703	1.246313	0.000000
H	4.607703	-1.246313	0.000000

Triplet Anthracene ($C_{14}H_{10}$)

	X	Y	Z
C	3.706046	0.691777	0.000000
C	3.706046	-0.691777	0.000000
H	4.641764	-1.243830	0.000000
H	4.641764	1.243830	0.000000
C	2.480846	1.395697	0.000000
C	2.480846	-1.395697	0.000000
C	1.255174	0.720576	0.000000
C	1.255174	-0.720576	0.000000
H	2.484966	2.483134	0.000000
H	2.484966	-2.483134	0.000000
C	0.000000	1.406192	0.000000
C	0.000000	-1.406192	0.000000
C	-1.255173	0.720575	0.000000
C	-1.255173	-0.720575	0.000000
H	-0.000001	2.493700	0.000000
H	-0.000001	-2.493700	0.000000
C	-2.480846	1.395696	0.000000
C	-2.480846	-1.395696	0.000000
C	-3.706045	0.691777	0.000000
C	-3.706045	-0.691777	0.000000
H	-2.484968	2.483133	0.000000
H	-2.484968	-2.483133	0.000000
H	-4.641762	1.243835	0.000000
H	-4.641762	-1.243835	0.000000

Singlet Tetracene ($C_{18}H_{12}$)

	X	Y	Z
C	4.889184	0.715267	0.000000
C	4.889184	-0.715267	0.000000
H	5.837094	-1.246479	0.000000
H	5.837094	1.246479	0.000000
C	3.711380	1.409387	0.000000
C	3.711380	-1.409387	0.000000
C	2.450840	0.725941	0.000000
C	2.450840	-0.725941	0.000000
H	3.709630	2.496931	0.000000
H	3.709630	-2.496931	0.000000
C	1.235427	1.406152	0.000000
C	1.235427	-1.406152	0.000000
C	0.000016	0.726124	0.000000
C	0.000016	-0.726124	0.000000
H	1.235305	2.494321	0.000000
H	1.235305	-2.494321	0.000000
C	-1.235526	1.406147	0.000000
C	-1.235526	-1.406147	0.000000
C	-2.450799	0.725948	0.000000
C	-2.450799	-0.725948	0.000000
H	-1.235335	2.494309	0.000000
H	-1.235335	-2.494309	0.000000
C	-3.711407	1.409399	0.000000
C	-3.711407	-1.409399	0.000000
C	-4.889136	0.715307	0.000000
C	-4.889136	-0.715307	0.000000
H	-3.709510	2.496929	0.000000
H	-3.709510	-2.496929	0.000000
H	-5.837057	1.246476	0.000000
H	-5.837057	-1.246476	0.000000

Triplet Tetracene ($C_{18}H_{12}$)

	X	Y	Z
C	-4.926099	0.698684	0.000000
C	-4.926099	-0.698684	0.000000
H	-5.865246	-1.244975	0.000000
H	-5.865246	1.244975	0.000000
C	-3.716632	1.397315	0.000000
C	-3.716632	-1.397315	0.000000
C	-2.486849	0.717519	0.000000
C	-2.486849	-0.717519	0.000000
H	-3.717388	2.484893	0.000000
H	-3.717388	-2.484893	0.000000
C	-1.231329	1.404734	0.000000
C	-1.231329	-1.404734	0.000000
C	0.000000	0.730940	0.000000
C	0.000000	-0.730940	0.000000
H	-1.236114	2.492628	0.000000
H	-1.236114	-2.492628	0.000000
C	1.231329	1.404734	0.000000
C	1.231329	-1.404734	0.000000
C	2.486848	0.717519	0.000000
C	2.486848	-0.717519	0.000000
H	1.236114	2.492628	0.000000
H	1.236114	-2.492628	0.000000
C	3.716633	1.397314	0.000000
C	3.716633	-1.397314	0.000000
C	4.926099	0.698684	0.000000
C	4.926099	-0.698684	0.000000
H	3.717391	2.484892	0.000000
H	3.717391	-2.484892	0.000000
H	5.865245	1.244978	0.000000
H	5.865245	-1.244978	0.000000

Singlet Pentacene ($C_{22}H_{14}$)

	X	Y	Z
C	6.117721	0.716456	0.000000
C	6.117721	-0.716456	0.000000
H	7.065952	-1.247064	0.000000
H	7.065952	1.247064	0.000000
C	4.941657	1.410527	0.000000
C	4.941657	-1.410527	0.000000
C	3.678746	0.727647	0.000000
C	3.678746	-0.727647	0.000000
H	4.939919	2.498020	0.000000
H	4.939919	-2.498020	0.000000
C	2.467582	1.407836	0.000000
C	2.467582	-1.407836	0.000000
C	1.226500	0.728534	0.000000
C	1.226500	-0.728534	0.000000
H	2.467624	2.495998	0.000000
H	2.467624	-2.495998	0.000000
C	-0.000028	1.408524	0.000000
C	-0.000028	-1.408524	0.000000
C	-1.226469	0.728539	0.000000
C	-1.226469	-0.728539	0.000000
H	-0.000017	2.496573	0.000000
H	-0.000017	-2.496573	0.000000
C	-2.467625	1.407846	0.000000
C	-2.467625	-1.407846	0.000000
C	-3.678713	0.727655	0.000000
C	-3.678713	-0.727655	0.000000
H	-2.467643	2.496006	0.000000
H	-2.467643	-2.496006	0.000000
C	-4.941671	1.410535	0.000000
C	-4.941671	-1.410535	0.000000
C	-6.117708	0.716482	0.000000
C	-6.117708	-0.716482	0.000000
H	-4.939849	2.498024	0.000000
H	-4.939849	-2.498024	0.000000
H	-7.065930	1.247100	0.000000
H	-7.065930	-1.247100	0.000000

Triplet Pentacene ($C_{22}H_{14}$)

	X	Y	Z
C	-6.148662	0.703577	0.000000
C	-6.148662	-0.703577	0.000000
H	-7.090330	-1.245586	0.000000
H	-7.090330	1.245586	0.000000
C	-4.949456	1.399985	0.000000
C	-4.949456	-1.399985	0.000000
C	-3.712886	0.717632	0.000000
C	-3.712886	-0.717632	0.000000
H	-4.948444	2.487564	0.000000
H	-4.948444	-2.487564	0.000000
C	-2.464142	1.403071	0.000000
C	-2.464142	-1.403071	0.000000
C	-1.244230	0.729785	0.000000
C	-1.244230	-0.729785	0.000000
H	-2.468456	2.491214	0.000000
H	-2.468456	-2.491214	0.000000
C	0.000000	1.408308	0.000000
C	0.000000	-1.408308	0.000000
C	1.244230	0.729785	0.000000
C	1.244230	-0.729785	0.000000
H	0.000000	2.496125	0.000000
H	0.000000	-2.496125	0.000000
C	2.464142	1.403071	0.000000
C	2.464142	-1.403071	0.000000
C	3.712886	0.717632	0.000000
C	3.712886	-0.717632	0.000000
H	2.468456	2.491214	0.000000
H	2.468456	-2.491214	0.000000
C	4.949456	1.399985	0.000000
C	4.949456	-1.399985	0.000000
C	6.148662	0.703577	0.000000
C	6.148662	-0.703577	0.000000
H	4.948445	2.487564	0.000000
H	4.948445	-2.487564	0.000000
H	7.090329	1.245586	0.000000
H	7.090329	-1.245586	0.000000

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] M. L. Abrams and C. D. Sherrill. Natural Orbitals as Substitutes for Optimized Orbitals in Complete Active Space Wavefunctions. *Chem. Phys. Lett.*, 395:227–232, 2004.

[2] M. L. Abrams and C. D. Sherrill. Natural Orbitals as Substitutes for Optimized Orbitals in Complete Active Space Wavefunctions. *Chem. Phys. Lett.*, 395(4-6):227–232, 2004.

[3] C. M. Aikens, S. Li, and G. C. Schatz. From Discrete Electronic States to Plasmons: TDDFT Optical Absorption Properties of Ag-n (n = 10, 20, 35, 56, 84, 120) Tetrahedral Clusters. *J. Phys. Chem. C*, 112(30):11272–11279, JUL 31 2008.

[4] B. J. Alder and T. E. Wainwright. Phase Transition for a Hard Sphere System. *J. Chem. Phys.*, 27(5):1208–1209, 1957.

[5] X. Andrade and A. Aspuru-Guzik. Real-Space Density Functional Theory on Graphical Processing Units: Computational Approach and Comparison to Gaussian Basis Set Methods. *J. Chem. Theory Comput.*, 9(10):4360–4373, 2013.

[6] F. Aquilante, L. Gagliardi, T. B. Pedersen, and R. Lindh. Atomic Cholesky Decompositions: A Route to Unbiased Auxiliary Basis Sets for Density Fitting Approximation with Tunable Accuracy and Efficiency. *J. Chem. Phys.*, 130:154107, 2009.

[7] F. Aquilante, T. B. Pedersen, and R. Lindh. Low-Cost Evaluation of the Exchange Fock Matrix from Cholesky and Density Fitting Representations of the Electron Repulsion Integrals. *J. Chem. Phys.*, 126:194106, 2007.

[8] F. Aquilante, T. B. Pedersen, R. Lindh, B. O. Roos, A. S. De Meras, and H. Koch. Accurate Ab Initio Density Fitting for Multiconfigurational Self-Consistent Field Methods. *J. Chem. Phys.*, 129(2):024113, 2008.

[9] A. Asadchev, V. Allada, J. Felder, B. M. Bode, M. S. Gordon, and T. L. Windus. Uncontracted Rys Quadrature Implementation of Up to G Functions on Graphical Processing Units. *J. Chem. Theory Comput.*, 6(3):696–704, 2010.

[10] A. Asadchev and M. S. Gordon. New Multithreaded Hybrid CPU/GPU Approach to Hartree-Fock. *J. Chem. Theory Comput.*, 8(11):4166–4176, 2012.

[11] A. Asadchev and M. S. Gordon. Fast and Flexible Coupled Cluster Implementation. *J. Chem. Theory Comput.*, 9(8):3385–3392, 2013.

[12] E. Badaeva, J. W. May, J. Ma, D. R. Gamelin, and X. S. Li. Characterization of Excited-State Magnetic Exchange in Mn2+-Doped ZnO Quantum Dots Using Time-Dependent Density Functional Theory. *J. Phys. Chem. C*, 115(43):20986–20991, 2011.

[13] G.-T. Bae and C. M. Aikens. Time-Dependent Density Functional Theory Studies of Optical Properties of Ag Nanoparticles: Octahedra, Truncated Octahedra, and Icosahedra. *J. Phys. Chem. C*, 116:10356–10367, 2012.

[14] M. J. Bearpark, M. A. Robb, and H. B. Schlegel. A Direct Method for the Location of the Lowest Energy Point on a Potential Surface Crossing. *Chem. Phys. Lett.*, 223:269–274, 1994.

[15] N. H. F. Beebe and J. Linderberg. Simplifications in the Generation and Transformation of Two-Electron Integrals in Molecular Calculations. *Int. J. Quantum Chem.*, XII:683–705, 1977.

[16] M. Ben-Nun, J. Quenneville, and T. J. Martínez. Ab Initio Multiple Spawning: Photochemistry from First Principles Quantum Molecular Dynamics. *J. Phys. Chem. A*, 104(22):5161–5175, 2000.

[17] M. Bénard and W. G. Laidlaw. Hartree Fock Instabilities in the $S_4N_4^{2+}$ Dication. *Theor. Chim. Acta*, 70:17–24, 1986.

[18] G. L. Bendazzoli and S. Evangelisti. A Vector and Parallel Full Configuration-Interaction Algorithm. *J. Chem. Phys.*, 98(4):3141–3150, 1993.

[19] F. Berencz. The Relationship Between Löwdin's Spin Projection Operator and Pratt's Spin Operator. *Acta Phys. Hung.*, XII(1):47–53, 1960.

[20] K. Bhaskaran-Nair, W. J. Ma, S. Krishnamoorthy, O. Villa, H. J. J. van Dam, E. Apra, and K. Kowalski. Noniterative Multireference Coupled Cluster Methods on Heterogeneous CPU-GPU Systems. *J. Chem. Theory Comput.*, 9(4):1949–1957, 2013.

[21] J. B. Birks. *Photophysics of Aromatic Molecules*. Wiley, London, 1970.

[22] J. M. Bofill and P. Pulay. The Unrestricted Natural Orbital-Complete Active Space (UNO-CAS) Method - an Inexpensive Alternative to the Complete Active Space-Self-Consistent-Field (CAS-SCF) Method. *J. Chem. Phys.*, 90(7):3637–3646, 1989.

[23] V. Bonačić-Koutecký, V. Veyret, and R. Mitrić. Ab Initio Study of the Absorption Spectra of Ag_n $(n = 5 - 8)$ Clusters. *J. Chem. Phys.*, 115(22):10450–10460, 2001.

[24] G. H. Booth, A. J. W. Thom, and A. Alavi. Fermion Monte Carlo Without Fixed Nodes: A Game of Life, Death, and Annihilation in Slater Determinant Space. *J. Chem. Phys.*, 131:054106, 2009.

[25] R. J. Buenker and S. D. Peyerimhoff. Individualized Configuration Selection in CI Calculations with Subsequent Energy Extrapolation. *Theoret. Chim. Acta*, 35:33–58, 1974.

[26] J. Burgos, M. Pope, C. E. Swenberg, and R. R. Alfano. Heterofission in Pentacene-Doped Tetracene Single Crystals. *Phys. Status Solidi B*, 83:249, 1977.

[27] G. Chambaud, B. Levy, and P. Millie. Ab Initio Hartree-Fock Instabilities in Closed-Shell Molecular Systems. *Theor. Chim. Acta*, 48:103–118, 1978.

[28] G. K. L. Chan and M. Head-Gordon. Highly Correlated Calculations with a Polynomial Cost Algorithm: A Study of the Density Matrix Renormalization Group. *J. Chem. Phys.*, 116(11):4462–4476, 2002.

[29] G. K.-L. Chan and S. Sharma. The Density Matrix Renormalization Group in Quantum Chemistry. *Annu. Rev. Phys. Chem.*, 62:465–481, 2011.

[30] S. Chattopadhyay, R. K. Chaudhuri, and K. F. Freed. Geometry Optimization of Radicaloid Systems Using Improved Virtual Orbital-Complete Active Space Configuration Interaction (IVO-CASCI) Analytical Gradient Method. *J. Phys. Chem. A*, 115(16):3665–3678, 2011.

[31] T. S. Chwee and E. A. Carter. Density Fitting of Two-Electron Integrals in Local Multireference Single and Double Excitation Configuration Interaction Calculations. *Mol. Phys.*, 108(19-20):2519–2526, 2010.

[32] B. F. Curchod, C. Rauer, P. Marquetand, L. González, and T. J. Martínez. Communication: GAIMS-Generalized Ab Initio Multiple Spawning for Both Internal Conversion and Intersystem Crossing Processes. *J. Chem. Phys.*, 144:101102, 2016.

[33] E. R. Davidson. The Iterative Calculation of a Few of the Lowest Eigenvalues and Corresponding Eigenvectors of Large Real-Symmetric Matrices. *J. Comput. Phys.*, 17:87–94, 1975.

[34] M. G. Delcey, L. Freitag, T. B. Pedersen, F. Aquilante, R. Lindh, and L. Gonzalez. Analytical Gradients of Complete Active Space Self-Consistent Field Energies Using Cholesky Decomposition: Geometry Optimization and Spin-State Energetics of a Ruthenium Nitrosyl Complex. *J. Chem. Phys.*, 140(17):174103, 2014.

[35] A. E. DePrince and J. R. Hammond. Coupled Cluster Theory on Graphics Processing Units I. The Coupled Cluster Doubles Method. *J. Chem. Theory Comput.*, 7(5):1287–1295, 2011.

[36] A. E. DePrince, M. R. Kennedy, B. G. Sumpter, and C. D. Sherrill. Density-Fitted Singles and Doubles Coupled Cluster on Graphics Processing Units. *Mol. Phys.*, 112(5-6):844–852, 2014.

[37] R. N. Diffenderfer and D. R. Yarkony. Use of the State-Averaged MCSCF Procedure: Application to Radiative Transitions in MgO. *J. Phys. Chem.*, 86:5098–5105, 1982.

[38] W. Duch. GRMS or Graphical Representation of Model Spaces. In G. Berthier, M. J. S. Dewar, H. Fischer, K. Fukui, G. G. Hall, J. Hinze, H. H. Jaffe, J. Jortner, W. Kutzelnigg, K. Ruedenberg, and J. Tomasi, editors, *Lecture Notes in Chemistry*, volume 1. Springer Verlag, New York, 1986.

[39] S. Evangelisti, G. L. Bendazzoli, R. Ansaloni, F. Duri, and E. Rossi. A One Billion Determinant Full CI Benchmark on the Cray T3D. *Chem. Phys. Lett.*, 252:437–446, 1996.

[40] B. S. Fales and B. G. Levine. Nanoscale Multireference Quantum Chemistry: Full Configuration Interaction on Graphical Processing Units. *J. Chem. Theory Comput.*, 11:4708–4716, 2015.

[41] B. S. Fales, Y. Shu, B. G. Levine, and E. G. Hohenstein. Complete Active Space Configuration Interaction From State-Averaged Configuration Interaction Singles Natural Orbitals: Analytic First Derivatives and Derivative Coupling Vectors. *J. Chem. Phys., submitted*, 2017.

[42] D. A. Fedorov, S. R. Pruitt, K. Keipert, M. S. Gordon, and S. A. Varganov. Ab Initio Multiple Spawning Method for Intersystem Crossing Dynamics: Spin-Forbidden Transitions between 3B(1) and (1)A(1) States of GeH2. *J. Phys. Chem. A*, 120(18):2911–2919, MAY 12 2016.

[43] R. Fournier and H. Dhillon. Geometric Structure of Silver Clusters With and Without Adsorbed Cl and Hg. *Comp. Theor. Chem*, 1021:26–34, 2013.

[44] R. A. Friesner, R. B. Murphy, M. D. Beachy, M. N. Ringnalda, W. T. Pollard, B. D. Dunietz, and Y. Cao. Correlated Ab Initio Electronic Structure Calculations for Large Molecules. *J. Phys. Chem. A*, 103(13):1913–1928, 1999.

[45] M. Frisch, I. N. Ragazos, M. A. Robb, and H. B. Schlegel. An Evaluation of 3 Direct MC-SCF Procedures. *Chem. Phys. Lett.*, 189(6):524–528, 1992.

[46] H. Fukutome. Spin Density Wave and Charge Transfer Wave in Long Conjugated Molecules. *Progr. Theoret. Phys.*, 40(5):998–1012, 1968.

[47] F. Furche, B. T. Krull, B. D. Nguyen, and J. Kwon. Accelerating Molecular Property Calculations with Nonorthonormal Krylov Space Methods. *J. Chem. Phys.*, 144:174105, 2016.

[48] Z. Gan and R. J. Harrison. Calibrating Quantum Chemistry: A Multi-Teraflop, Parallel Vector, Full-Configuration Interaction Program for the Cray-X1. *Proceedings of the 2005 ACM/IEEE SC—05 Conference (SC05)*, 2005.

[49] M. Garavelli, F. Bernardi, M. Olivucci, T. Vreven, S. Klein, P. Celani, and M. A. Robb. Potential-Energy Surfaces for Ultrafast Photochemistry. *Faraday Discuss.*, 110:51–70, 1998.

[50] G. Granucci, M. Persico, and A. Toniolo. Direct Semiclassical Simulation of Photochemical Processes with Semiempirical Wave Functions. *J. Chem. Phys.*, 114(24):10608–10615, 2001.

[51] E. B. Guidez and C. M. Aikens. Theoretical Analysis of the Optical Excitation Spectra of Silver and Gold Nanowires. *Nanoscale*, 4(14):4190–4198, 2012.

[52] J. Hachmann, J. J. Dorando, M. Avilés, and G. K.-L. Chan. The Radical Character of the Acenes: A Density Matrix Renormalization Group Study. *J. Chem. Phys.*, 127:134309, 2007.

[53] N. C. Handy. Multi-Root Configuration Interaction Calculations. *Chem. Phys. Lett.*, 74(2):280–283, 1980.

[54] E. Hao and G. Schatz. Electromagnetic Fields Around Silver Nanoparticles and Dimers. *J. Chem. Phys.*, 120(1):357–366, JAN 1 2004.

[55] M. Head-Gordon, J. A. Pople, and M. J. Frisch. MP2 Energy Evaluation by Direct Methods. *Chem. Phys. Lett.*, 153(6):503–506, 1988.

[56] T. Helgaker, P. Jørgensen, and J. Olsen. *Molecular Electronic-Structure Theory*. Wiley, Chichester, 2000.

[57] E. G. Hohenstein. Analytic Formulation of Derivative Coupling Vectors for Complete Active Space Configuration Interaction Wavefunctions with Floating Occupation Molecular Orbitals. *J. Chem. Phys.*, 145:174110, 2016.

[58] E. G. Hohenstein, M. E. F. Bouduban, C. Song, N. Luehr, I. S. Ufimtsev, and T. J. Martínez. Analytic First Derivatives of Floating Occupation Molecular Orbital-Complete Active Space Configuration Interaction on Graphical Processing Units. *J. Chem. Phys.*, 143:014111, 2015.

[59] E. G. Hohenstein, S. I. L. Kokkila, R. M. Parrish, and T. J. Martínez. Tensor Hypercontraction Equation-of-Motion Second-Order Approximate Coupled Cluster: Electronic Excitation Energies in $\mathcal{O}(N^4)$ Time. *J. Phys. Chem. B*, 117:12972–12978, 2013.

[60] E. G. Hohenstein, N. Luehr, I. S. Ufimtsev, and T. J. Martínez. An Atomic Orbital Based Formulation of the Complete Active Space Self-Consistent Field Method on Graphical Processing Units. *J. Chem. Phys.*, 142:224103, 2015.

[61] E. G. Hohenstein, R. M. Parrish, and T. J. Martínez. Tensor Hypercontraction Density Fitting. I. Quartic Scaling Second- and Third-Order Møller-Plesset Perturbation Theory. *J. Chem. Phys.*, 137:044103, 2012.

[62] A. A. Holmes, N. M. Tubman, and C. J. Umrigar. Heat-Bath Configuration Interaction: An Efficient Selected Configuration Interaction Algorithm Inspired by Heat-Bath Sampling. *J. Chem. Theory Comput.*, 12:3674–3680, 2016.

[63] B. Huron, J. P. Malrieu, and P. Rancurel. Iterative Perturbation Calculations of Ground and Excited State Energies From Multiconfigurational Zeroth-Order Wavefunctions. *J. Chem. Phys.*, 58(12):5745–5759, 1973.

[64] Intel. *Intel Math Kernel Library Developer Reference 2017 — C*, 2017.

[65] C. M. Isborn, N. Luehr, I. S. Ufimtsev, and T. J. Martínez. Excited-State Electronic Structure with Configuration Interaction Singles and Tamm-Dancoff Time-Dependent Density Functional Theory on Graphical Processing Units. *J. Chem. Theory Comput.*, 7(6):1814–1823, 2011.

[66] J. Ivanic. Direct Configuration Interaction and Multiconfigurational Self-Consistent-Field Method for Multiple Active Spaces with Variable Occupations. I. Method. *J. Chem. Phys.*, 119(18):9364–9376, 2003.

[67] P. K. Jain, X. Huang, I. H. El-Sayed, and M. A. El-Sayed. Noble Metals on the Nanoscale: Optical and Photothermal Properties and Some Applications in Imaging, Sensing, Biology, and Medicine. *Acc. Chem. Res.*, 41(12, SI):1578–1586, DEC 2008.

[68] P. Karadakov and O. Castaño. Aromaticity of Annulenes and Annulene Ions with $4\nu+2$ Electrons from the Viewpoint of the Theory of Hartree-Fock Instabilities. *Theor. Chim. Acta*, 70:25–34, 1986.

[69] S. Keller, K. Boguslawski, T. Janowski, M. Reiher, and P. Pulay. Selection of Active Spaces for Multiconfigurational Wavefunctions. *J. Chem. Phys.*, 142(24):244104, 2015.

[70] M. Kertész, J. Koller, and A. Ažman. Nuclear Distortion of the Equidistant Arrangement in Trans-Polyacetylene, $(CH)_x$. *Int. J. Quantum Chem.*, XVIII:645–650, 1980.

[71] G. Knizia, W. Li, S. Simon, and H.-J. Werner. Determining the Numerical Stability of Quantum Chemistry Algorithms. *J. Chem. Theory Comput.*, 7:2387–2398, 2011.

[72] P. J. Knowles. Very Large Full Configuration-Interaction Calculations. *Chem. Phys. Lett.*, 155(6):513–517, 1989.

[73] P. J. Knowles and N. C. Handy. A New Determinant Based Full Configuration Interaction Method. *Chem. Phys. Lett.*, 111(4,5):315–321, 1984.

[74] P. J. Knowles and N. C. Handy. A Determinant Based Full Configuration-Interaction Program. *Comput. Phys. Commun.*, 54(1):75–83, 1989.

[75] P. J. Knowles and N. C. Handy. Unlimited Full Configuration Interaction Calculations. *J. Chem. Phys.*, 91(4):2396–2398, 1989.

[76] P. J. Knowles and H. J. Werner. An Efficient 2nd-Order MC SCF Method for Long Configuration Expansions. *Chem. Phys. Lett.*, 115(3):259–267, 1985.

[77] H. Koch and E. Dalgaard. A Variational Matrix Decomposition Applied to Full Configuration-Interaction Calculations. *Chem. Phys. Lett.*, 198(1,2):51–58, 1992.

[78] H. Koch, A. S. de Meras, and T. B. Pedersen. Reduced Scaling in Electronic Structure Calculations Using Cholesky Decompositions. *J. Phys. Chem.*, 118(21):9481–9484, 2003.

[79] J. Koutecký. Unrestricted Hartree-Fock Solutions for Closed-Shell Molecules. *J. Chem. Phys.*, 46:2443–2445, 1967.

[80] D. B. Krisiloff, C. M. Krauter, F. J. Ricci, and E. A. Carter. Density Fitting and Cholesky Decomposition of the Two-Electron Integrals in Local Multireference Configuration Interaction Theory. *J. Chem. Theory Comput.*, 11:5242–5251, 2015.

[81] J. Kussmann and C. Ochsenfeld. Preselective Screening for Linear-Scaling Exact Exchange-Gradient Calculations for Graphics Processing Units and General Strong-Scaling Massively Parallel Calculations. *J. Chem. Theory Comput.*, 11(3):918–922, 2015.

[82] H. Larsen, J. Olsen, P. Jørgensen, and O. Christiansen. Full Configuration Interaction Benchmarking of Coupled-Cluster Models for the Lowest Energy Singlet Surfaces of N_2. *J. Chem. Phys.*, 113:6677, 2000.

[83] S. S. Leang, A. P. Rendell, and M. S. Gordon. Quantum Chemical Calculations Using Accelerators: Migrating Matrix Operations to the NVIDIA Kepler GPU and the Intel Xeon Phi. *J. Chem. Theory Comput.*, 10(3):908–912, 2014.

[84] R. Lefebvre and R. Prat. On the Projection of Slater Determinants. *Chem. Phys. Lett.*, 1:388–390, 1967.

[85] R. Lefebvre and R. F. Prat. Etudes en Methode de Hartree-Fock avec projection. I. Fonctions Propres de S^2. Evaluation de l'energie. *Int. J. Quantum Chem.*, III:93–105, 1969.

[86] O. Legeza, J. Roder, and B. A. Hess. Controlling the Accuracy of the Density-Matrix Renormalization-Group Method: The Dynamical Block State Selection Approach. *Phys. Rev. B*, 67(12):125114, 2003.

[87] M. L. Leininger, C. D. Sherrill, W. D. Allen, and H. F. Schaefer III. Systematic Study of Selected Diagonalization Methods for Configuration Interaction Matrices. *J. Comput. Chem.*, 22:1574–1589, 2001.

[88] B. G. Levine, J. D. Coe, and T. J. Martínez. Optimizing Conical Intersections Without Derivative Coupling Vectors: Application to Multistate Multireference Second-Order Perturbation Theory (MS-CASPT2). *J. Phys. Chem. B*, 112:405–413, 2008.

[89] B. G. Levine, J. D. Coe, A. M. Virshup, and T. J. Martínez. Implementation of Ab Initio Multiple Spawning in the MOLPRO Quantum Chemistry Package. *Chem. Phys.*, 347(1-3):3–16, MAY 23 2008.

[90] B. G. Levine, C. Ko, J. Quenneville, and T. J. Martínez. Conical Intersections and Double Excitations in Time-Dependent Density Functional Theory. *Mol. Phys.*, 104(5-7):1039–1051, 2006.

[91] B. G. Levine, J. E. Stone, and A. Kohlmeyer. Fast Analysis of Molecular Dynamics Trajectories with Graphics Processing Units—Radial Distribution Function Histogramming. *J. Comput. Phys.*, 230(9):3556–3569, 2011.

[92] R. Lindh, J. Olsen, and B. Roos. Low-Rank Configuration Interaction With Orbital Optimization - the LR SCF Approach. *Chem. Phys. Lett.*, 148(4):276–280, 1988.

[93] B. Liu. The Simultaneous Expansion Method for the Iterative Solution of Several of the Lowest-Lying Eigenvalues and Corresponding Eigenvectors of Large Real-Symmetric Matrices. *Numerical Algorithms in Chemistry: Algebraic Methods*, pages 49–53, 1978.

[94] P.-O. Löwdin. Quantum Theory of Many-Particle Systems. III. Extension of the Hartree-Fock Scheme to Include Degenerate Systems and Correlation Effects. *Phys. Rev.*, 97(6):1509–1520, 1955.

[95] Z. Lu and S. Matsika. High-Multiplicity Natural Orbitals in Multireference Configuration Interaction for Excited States. *J. Chem. Theory Comput.*, 8(2):509–517, 2012.

[96] Z. Lu and S. Matsika. High-Multiplicity Natural Orbitals in Multireference Configuration Interaction for Excited State Potential Energy Surfaces. *J. Phys. Chem. A*, 117:7421–7430, 2013.

[97] Z. Lu and S. Matsika. High-Multiplicity Natural Orbitals in Multireference Configuration Interaction for Excited State Potential Energy Surfaces. *J. Phys. Chem. A*, 117(32):7421–7430, 2013.

[98] N. Luehr, I. S. Ufimtsev, and T. J. Martínez. Dynamic Precision for Electron Repulsion Integral Evaluation on Graphical Processing Units (GPUs). *J. Chem. Theory Comput.*, 7(4):949–954, 2011.

[99] D. I. Lyakh. An Efficient Tensor Transpose Algorithm for Multicore CPU, Intel Xeon Phi, and NVidia Tesla GPU. *Comput. Phys. Commun.*, 189:84–91, 2015.

[100] P. Lykos and G. W. Pratt. Discussion on the Hartree-Fock Approximation. *Rev. Mod. Phys.*, 35:496–501, 1963.

[101] D. X. Ma, G. L. Manni, and L. Gagliardi. The Generalized Active Space Concept in Multiconfigurational Self-Consistent Field Methods. *J. Chem. Phys.*, 135(4):044128, 2011.

[102] W. J. Ma, S. Krishnamoorthy, O. Villa, and K. Kowalski. GPU-Based Implementations of the Noniterative Regularized-CCSD(T) Corrections: Applications to Strongly Correlated Systems. *J. Chem. Theory Comput.*, 7(5):1316–1327, 2011.

[103] W. J. Ma, S. Krishnamoorthy, O. Villa, K. Kowalski, and G. Agrawal. Optimizing Tensor Contraction Expressions for Hybrid CPU-GPU Execution. *Cluster Computing-the Journal of Networks Software Tools and Applications*, 16(1):131–155, 2013.

[104] J. K. L. MacDonald. On the Modifed Ritz Variation Method. *Phys. Rev.*, 46:828, 1934.

[105] J. D. C. Maia, G. A. U. Carvalho, C. P. Mangueira, S. R. Santana, L. A. F. Cabral, and G. B. Rocha. GPU Linear Algebra Libraries and GPGPU Programming for Accelerating MOPAC Semiempirical Quantum Chemistry Calculations. *J. Chem. Theory Comput.*, 8(9):3072–3081, 2012.

[106] G. L. Manni, D. X. Ma, F. Aquiliante, J. Olsen, and L. Gagliardi. SplitGAS Method for Strong Correlation and the Challenging Case of Cr_2. *J. Chem. Theory Comput.*, 9(8):3375–3384, 2013.

[107] R. L. Martin. Natural Transition Orbitals. *J. Chem. Phys.*, 118(11):4775–4777, 2003.

[108] T. J. Martínez, A. Mehta, and E. A. Carter. Pseudospectral Full Configuration-Interaction. *J. Chem. Phys.*, 97(3):1876–1880, 1992.

[109] S. A. Maurer, J. Kussmann, and C. Ochsenfeld. Communication: A Reduced Scaling J-Engine Based Reformulation of SOS-MP2 Using Graphics Processing Units. *J. Chem. Phys.*, 141(5):051106, 2014.

[110] G. A. Meek, A. D. Baczewski, D. J. Little, and B. G. Levine. Polaronic Relaxation by Three-Electron Bond Formation in Graphitic Carbon Nitrides. *J. Phys. Chem. C*, 118(8):4023–4032, 2014.

[111] G. A. Meek and B. G. Levine. The Best of Both Reps-Diabatized Gaussians on Adiabatic Surfaces. *J. Chem. Phys.*, 145(18), NOV 14 2016.

[112] M. M. Mestechkin. Restricted Hartree-Fock Method Instability. *Int. J. Quantum Chem.*, XIII:469–481, 1978.

[113] Y. P. Miao and K. M. Merz. Acceleration of Electron Repulsion Integral Evaluation on Graphics Processing Units via Use of Recurrence Relations. *J. Chem. Theory Comput.*, 9(2):965–976, 2013.

[114] Y. P. Miao and K. M. Merz. Acceleration of High Angular Momentum Electron Repulsion Integrals and Integral Derivatives on Graphics Processing Units. *J. Chem. Theory Comput.*, 11(4):1449–1462, 2015.

[115] A. O. Mitrushenkov. Passing the Several Billions Limit in FCI Calculations on a Mini-Computer. *Chem. Phys. Lett.*, 217(5-6):559–565, 1994.

[116] A. O. Mitrushenkov and Y. Y. Dmitriev. Passing the Several Billion Limit in FCI Calculations on a Mini-Computer. A Norm-Consistent Zero CI Threshold Estimate Within the Dynamic CI Approach. *Chem. Phys. Lett.*, 235:410–413, 1995.

[117] A. O. Mitrushenkov, G. Fano, F. Ortolani, R. Linguerri, and P. Palmieri. Quantum Chemistry Using the Density Matrix Renormalization Group. *J. Chem. Phys.*, 115(15):6815–6821, 2001.

[118] G. Moritz and M. Reiher. Decomposition of Density Matrix Renormalization Group States into a Slater Determinant Basis. *J. Chem. Phys.*, 126(24):244109, 2007.

[119] H. Nakano and K. Hirao. A Quasi-Complete Active Space Self-Consistent Field Method. *Chem. Phys. Lett.*, 317(1-2):90–96, 2000.

[120] N. Nakatani and G. K.-L. Chan. Efficient Tree Tensor Network States (TTNS) for Quantum Chemistry: Generalizations of the Density Matrix Renormalization Group Algorithm. *J. Chem. Phys.*, 138:134113, 2013.

[121] I. H. Nayyar, E. R. Batista, S. Tretiak, A. Saxena, D. L. Smith, and R. L. Martin. Localization of Electronic Excitations in Conjugated Polymers Studied by DFT. *J. Phys. Chem. Lett.*, 2(6):566–571, 2011.

[122] M. A. Nitsche, M. Ferreria, E. E. Mocskos, and M. C. G. Lebrero. GPU Accelerated Implementation of Density Functional Theory for Hybrid QM/MM Simulations. *J. Chem. Theory Comput.*, 10(3):959–967, 2014.

[123] H. Nobutoki. Broken-Symmetry Orbitals for the Quasidiabatic Electronic States of Polymers. *Int. J. Quantum Chem.*, 74:745–752, 1999.

[124] NVidia. *cuBLAS Library*, 2017.

[125] NVidia. *CUDA C Programming Guide*, 2017.

[126] R. Olivares-Amaya, M. A. Watson, R. G. Edgar, L. Vogt, Y. Shao, and A. Aspuru-Guzik. Accelerating Correlated Quantum Chemistry Calculations Using Graphical Processing Units and a Mixed Precision Matrix Multiplication Library. *J. Chem. Theory Comput.*, 6(1):135–144, 2010.

[127] J. Olsen, P.-A. Malmqvist, B. Roos, R. Lindh, and P.-O. Widmark. A Non-Linear Approach to Configuration Interaction. The Low-Rank CI Method (LR CI). *Chem. Phys. Lett.*, 133(2):91–101, 1987.

[128] J. Olsen, B. Roos, P. Jørgensen, and H. J. A. Jensen. Determinant Based Configuration Interaction Algorithms for Complete and Restricted Configuration Interaction Spaces. *J. Chem. Phys.*, 89:2185, 1988.

[129] N. S. Ostlund. Complex and Unrestricted Hartree-Fock Wavefunctions. *J. Chem. Phys.*, 57(7):2994–2997, 1972.

[130] J. Paldus. Many-Electron Correlation Problem. A Group Theoretical Approach. In H. Eyring and D. Henderson, editors, *Theoretical Chemistry: Advances and Perspectives*. Academic Press, New York, NY, 1976.

[131] S. M. Parker, T. Seideman, M. A. Ratner, and T. Shiozaki. Communication: Active-Space Decomposition for Molecular Dimers. *J. Chem. Phys.*, 139(2):021108, 2013.

[132] R. M. Parrish, E. G. Hohenstein, and T. J. Martínez. "Balancing" the Block Davidson-Liu Algorithm. *J. Chem. Theory Comput.*, 12:3002–3007, 2016.

[133] R. M. Parrish, E. G. Hohenstein, T. J. Martínez, and C. D. Sherrill. Tensor Hypercontraction. II. Least-Squares Renormalization. *J. Chem. Phys.*, 137:224106, 2012.

[134] R. M. Parrish, E. G. Hohenstein, N. F. Schunk, C. D. Sherrill, and T. J. Martínez. Exact Tensor Hypercontraction: A Universal Technique for the Resolution of Matrix Elements of Local Finite-Range N-Body Potentials in Many-Body Quantum Problems. *Phys. Rev. Lett.*, 111:132505, 2013.

[135] J. K. Percus and A. Rotenberg. Exact Eigenfunctions of Angular Momentum by Rotational Projection. *J. Math Phys.*, 3(5):928–932, 1962.

[136] D. M. Potts, C. M. Taylor, R. K. Chaudhuri, and K. F. Freed. The Improved Virtual Orbital-Complete Active Space Configuration Interaction Method, a "Packageable" Efficient Ab Initio Many-Body Method for Describing Electronically Excited States. *J. Chem. Phys.*, 114(6):2592–2600, 2001.

[137] G. W. Pratt, Jr. Eigenfunctions of S^2 by a Spin Operator Method. *Phys. Rev.*, 92(2):278–288, 1953.

[138] A. Rahman. Correlations in the Motion of Atoms in Liquid Argon. *Phys. Rev.*, 136(2A):A405–A411, 1964.

[139] I. Roeggen and E. Wisloff-Nilssen. On the Beebe-Linderberg Two-Electron Integral Approximation. *Chem. Phys. Lett.*, 132(2):154–160, 1986.

[140] Z. Rolik, Ágnes Szabados, and P. R. Surján. A Sparse Matrix Based Full-Configuration Interaction Algorithm. *J. Chem. Phys.*, 128(14):144101, 2008.

[141] B. Roos. A New Method For Large-Scale CI Calculations. *Chem. Phys. Lett.*, 15:153–159, 1972.

[142] B. Roos and P. Siegbahn. The Direct Configuration Interaction Method from Molecular Integrals. In H. F. Schaefer III, editor, *Methods in Electronic Structure Theory*, page 277. Plenum Press, New York, NY, 1977.

[143] B. Roos, P. R. Taylor, and P. E. M. Siegbahn. A Complete Active Space SCF Method (CASSCF) Using a Density Matrix Formulated Super-CI Approach. *Chem. Phys.*, 48:157–173, 1980.

[144] B. O. Roos. The Complete Active Space SCF Method in a Fock-Matrix-Based Super-CI Formulation. *Int. J. Quantum Chem.*, 14:175–189, 1980.

[145] B. O. Roos and P. E. M. Siegbahn. A Direct CI Method with a Multiconfigurational Reference State. *Int. J. Quantum Chem.*, 17:485–500, 1980.

[146] B. O. Roos and P. M. Siegbahn. Methylene Singlet—Triplet Separation. An Ab Initio Configuration Interaction Study. *J. Am. Chem. Soc.*, 99(23):7716–7718, 1977.

[147] B. O. Roos and P. R. Taylor. A Complete Active Space SCF Method (CASSCF) Using A Density-Matrix Formulated Super-CI Approach. *Chem. Phys.*, 48(2):157–173, 1980.

[148] K. Ruedenberg, M. W. Schmidt, M. M. Gilbert, and S. T. Elbert. Are Atoms Intrinsic to Molecular Electronic Wavefunctions? I. The FORS Model. *Chem. Phys.*, 71:41–49, 1982.

[149] N. Sabbatini, M. T. Indelli, M. T. Gandolfi, and V. Balzani. Quenching of Singlet and Triplet Excited States of Aromatic Molecules by Europium Ions. *J. Phys. Chem.*, 86:3585, 1982.

[150] J. Schiedt and R. Weinkauf. Photodetachment Photoelectron Spectroscopy of Mass Selected Anions: Anthracene and the Anthracene-H_2O Cluster. *Chem. Phys. Lett.*, 266:201, 1997.

[151] H. B. Schlegel. Optimization of Equilibrium Geomtries and Transition States. *J. Comput. Chem.*, 3:214–218, 1982.

[152] M. W. Schmidt and M. S. Gordon. The Construction and Interpretation of MCSCF Wavefunctions. *Annu. Rev. Phys. Chem.*, 49:233–266, 1998.

[153] J. B. Schriber and F. A. Evangelista. Communication: An Adaptive Configuration Interaction Approach for Strongly Correlated Electrons with Tunable Accuracy. *J. Chem. Phys.*, 144:161106, 2016.

[154] S. I. L. K. Schumacher, E. G. Hohenstein, R. M. Parrish, L.-P. Wang, and T. J. Martínez. Tensor Hypercontraction Second-Order Møller-Plesset Perturbation Theory: Grid Optimization and Reaction Energies. *J. Chem. Theory Comput.*, 11:3042–3052, 2015.

[155] M. Schutz, R. Lindh, and H. J. Werner. Integral-Direct Electron Correlation Methods. *Mol. Phys.*, 96(4):719–733, 1999.

[156] R. Seeger and J. A. Pople. Self-Consistent Molecular Orbital Methods. XVIII. Constraints and Stability in Hartree-Fock Theory. *J. Chem. Phys.*, 66(7):3045–3050, 1977.

[157] I. Shavitt. Matrix Element Evaluation in the Unitary Group Approach to the Electron Correlation Problem. *Int. J. Quantum Chem.*, 12:5–32, 1978.

[158] C. D. Sherrill and H. F. Schaefer III. The Configuration Interaction Method: Advances in Highly Correlated Approaches. In M. C. Z. Per-Olov Löwdin, John R. Sabin and E. Brandas, editors, *Advances in Quantum Chemistry*, volume 34, pages 143 – 269. Academic Press, 1999.

[159] Y. Shu, B. S. Fales, and B. G. Levine. Defect-Induced Conical Intersections Promote Nonradiative Recombination. *Nano Lett.*, 15:6247–6253, 2015.

[160] Y. Shu, E. G. Hohenstein, and B. G. Levine. Configuration Interaction Singles Natural Orbitals: An Orbital Basis for an Efficient and Size Intensive Multireference Description of Electronic Excited States. *J. Chem. Phys.*, 142(2):024102, 2015.

[161] Y. Shu and B. G. Levine. Reducing the Propensity for Unphysical Wavefunction Symmetry Breaking in Multireference Calculations of the Excited States of Semiconductor Clusters. *J. Chem. Phys.*, 139(7):074102, 2013.

[162] Y. N. Shu and B. G. Levine. Simulated Evolution of Fluorophores for Light Emitting Diodes. *J. Chem. Phys.*, 142(10):104104, 2015.

[163] P. Siegbahn, A. Heiberg, B. Roos, and B. Levy. A Comparison of the Super-CI and the Newton-Raphson Scheme in the Complete Active Space SCF Method. *Phys. Scr.*, 21:323–327, 1980.

[164] P. E. M. Siegbahn. A New Direct CI Method for Large CI Expansions in a Small Orbital Space. *Chem. Phys. Lett.*, 109(5):417–423, 1984.

[165] P. Slavicek and T. J. Martínez. Ab Initio Floating Occupation Molecular Orbital-Complete Active Space Configuration Interaction: An Efficient Approximation to CASSCF. *J. Chem. Phys.*, 132(23):234102, 2010.

[166] J. W. Snyder, Jr., B. S. Fales, E. G. Hohenstein, B. G. Levine, and T. J. Martínez. A Direct-Compatible Formulation of the Coupled Perturbed Complete Active Space Self-Consistent Field Equations on Graphical Processing Units. *J. Chem. Phys.*, 146:174113, 2017.

[167] J. W. Snyder, Jr., E. G. Hohenstein, N. Luehr, and T. J. Martínez. An Atomic Orbital Based Formulation of Analytical Gradients and Nonadiabatic Coupling Vector Elements for the Complete Active Space Self-Consistent Field Method on Graphical Processing Units. *J. Chem. Phys.*, 143:154107, 2015.

[168] C. Song and T. J. Martínez. Atomic Orbital-Based SOS-MP2 with Tensor Hypercontraction. I. GPU Based Tensor Construction and Exploiting Sparsity. *J. Chem. Phys.*, 144:174111, 2016.

[169] C. Song and T. J. Martínez. Atomic Orbital-Based SOS-MP2 with Tensor Hypercontraction. II. Local Tensor Hypercontraction. *J. Chem. Phys.*, 146:034104, 2016.

[170] P. L. Stiles, J. A. Dieringer, N. C. Shah, and R. R. Van Duyne. Surface-Enhanced Raman Spectroscopy. *Ann. Rev. Anal. Chem.*, 1:601–626, 2008.

[171] P. G. Szalay, T. Muller, G. Gidofalvi, H. Lischka, and R. Shepard. Multiconfiguration Self-Consistent Field and Multireference Configuration Interaction Methods and Applications. *Chem. Rev.*, 112(1):108–181, 2012.

[172] K. Tanaka, H. Kobayashi, S. Yamanaka, K. Yoshizawa, and T. Yamabe. A Stability Condition for the Hartree-Fock Solution of the Infinite One-Dimensional System. *J. Chem. Phys.*, 91(6):3724–3728, 1989.

[173] P. R. Taylor. Lossless Compression of Wavefunction Information Using Matrix Factorization: A "gzip" for Quantum Chemistry. *J. Chem. Phys.*, 139:074113, 2013.

[174] S. Ten-no. Stochastic Determination of Effective Hamiltonian for the Full Configuration Interaction Solution of Quasi-Degenerate Electronic States. *J. Chem. Phys.*, 138:164126, 2013.

[175] S. Ten-no and S. Iwata. Multiconfiguration Self-Consistent Field Procedure Employing Linear Combination of Atomic-Electron Distributions. *J. Chem. Phys.*, 105(9):3604–3611, 1996.

[176] D. J. Thouless. *The Quantum Mechanics in Many-Body Systems.* Academic Press, New York, 1961.

[177] A. V. Titov, I. S. Ufimtsev, N. Luehr, and T. J. Martínez. Generating Efficient Quantum Chemistry Codes for Novel Architectures. *J. Chem. Theory Comput.*, 9:213–221, 2013.

[178] N. M. Tubman, J. Lee, T. Y. Takeshita, M. Head-Gordon, and K. B. Whaley. A Deterministic Alternative to the Full Configuration Interaction Quantum Monte Carlo Method. *J. Chem. Phys.*, 145:044112, 2016.

[179] I. S. Ufimtsev and T. J. Martínez. Quantum Chemistry on Graphical Processing Units. 1. Strategies for Two-Electron Integral Evaluation. *J. Chem. Theory Comput.*, 4(2):222–231, 2008.

[180] I. S. Ufimtsev and T. J. Martínez. Quantum Chemistry on Graphical Processing Units. 2. Direct Self-Consistent Field Implementation. *J. Chem. Theory Comput.*, 5(4):1004–1015, 2009.

[181] I. S. Ufimtsev and T. J. Martínez. Quantum Chemistry on Graphical Processing Units. 3. Analytical Energy Gradients, Geometry Optimization, and First Principles Molecular Dynamics. *J. Chem. Theory Comput.*, 5(10):2619–2628, 2009.

[182] J. Čížek and J. Paldus. Stability Conditions for the Solutions of the Hartree-Fock Equations for Atomic and Molecular Systems. Application to the π-Electron Model of Cyclic Polyenes. *J. Chem. Phys.*, 47(10):3976–3985, 1967.

[183] L. Vogt, R. Olivares-Amaya, S. Kermes, Y. Shao, C. Amador-Bedolla, and A. Aspuru-Guzik. Accelerating Resolution-of-the-Identity Second-Order Moller-Plesset Quantum Chemistry Calculations with Graphical Processing Units. *J. Phys. Chem.*, 112A(10):2049–2057, 2008.

[184] W. R. Wadt and P. J. Hay. Ab Initio Effective Core Potentials for Molecular Calculations - Potentials for Main Group Elements Na to Bi. *J. Chem. Phys.*, 82(1):284–298, 1985.

[185] L. Wang, S. D. Fried, S. G. Boxer, and T. E. Markland. Quantum Delocalization of Protons in the Hydrogen-Bond Network of an Enzyme Active Site. *Proc. Natl. Acad. Sci. U.S.A.*, 111(52):18454–18459, 2014.

[186] L. P. Wang, A. Titov, R. McGibbon, F. Liu, V. S. Pande, and T. J. Martínez. Discovering Chemistry With an Ab Initio Nanoreactor. *Nat. Chem.*, 6(12):1044–1048, 2014.

[187] L.-W. Wang and A. Zunger. Electronic Structure Pseudopotential Calculations of Large ($\sim$ 1000 Atoms) Si Quantum Dots. *J. Phys. Chem.*, 98:2158–2165, 1994.

[188] L.-W. Wang and A. Zunger. Solving Schrödinger's Equation Around a Desired Energy: Application to Silicon Quantum Dots. *J. Chem. Phys.*, 100(3):2394–2397, 1994.

[189] H. J. Werner and P. J. Knowles. A Second Order Multiconfiguration SCF Procedure with Optimum Convergence. *J. Chem. Phys.*, 82(11):5053–5063, 1985.

[190] H.-J. Werner, P. J. Knowles, G. Knizia, F. R. Manby, M. Schütz, P. Celani, T. Korona, R. Lindh, A. Mitrushenkov, G. Rauhut, K. R. Shamasundar, T. B. Adler, R. D. Amos, A. Bernhardsson, A. Berning, D. L. Cooper, M. J. O. Deegan, A. J. Dobbyn, F. Eckert, E. Goll, C. Hampel, A. Hesselmann, G. Hetzer, T. Hrenar, G. Jansen, C. Köppl, Y. Liu, A. W. Lloyd, R. A. Mata, A. J. May, S. J. McNicholas, W. Meyer, M. E. Mura, A. Nicklass, D. P. O'Neill, P. Palmieri, D. Peng, K. Pflüger, R. Pitzer, M. Reiher, T. Shiozaki, H. Stoll, A. J. Stone, R. Tarroni, T. Thorsteinsson, and M. Wang. MOLPRO, Version 2010.1, a Package of Ab Initio Programs, 2010. see http://www.molpro.net.

[191] H.-J. Werner, F. R. Manby, and P. J. Knowles. Fast Linear Scaling Second-Order Møller-Plesset Perturbation Theory (MP2) Using Local and Density Fitting Approximations. *J. Chem. Phys.*, 118(18):8149–8160, 2003.

[192] S. R. White and R. L. Martin. Ab Initio Quantum Chemistry Using the Density Matrix Renormalization Group. *J. Chem. Phys.*, 110(9):4127–4130, 1999.

[193] K. A. Wilkinson, P. Sherwood, M. F. Guest, and K. J. Naidoo. Acceleration of the GAMESS-UK Electronic Structure Package on Graphical Processing Units. *J. Comput. Chem.*, 32(10):2313–2318, 2011.

[194] K. A. Willets and R. P. Van Duyne. Localized Surface Plasmon Resonance Spectroscopy and Sensing. *Ann. Rev. Phys. Chem.*, 58:267–297, 2007.

[195] X. Wu, A. Koslowski, and W. Thiel. Semiempirical Quantum Chemical Calculations Accelerated on a Hybrid Multicore CPU-GPU Computing Platform. *J. Chem. Theory Comput.*, 8(7):2272–2281, 2012.

[196] T. Yamada and S. Hirata. Singlet and Triplet Instability Theorems. *J. Chem. Phys.*, 143:114112, 2015.

[197] K. Yamaguchi. Electronic Structures of Antiaromatic Molecules. *Chem. Phys. Lett.*, 35(2):230–235, 1975.

[198] D. R. Yarkony. Conical Intersections: Diabolical and Often Misunderstood. *Acc. Chem. Res.*, 31(8):511–518, 1998.

[199] P. Yasaei, B. Kumar, R. Hantehzadeh, M. Kayyalha, A. Baskin, N. Repnin, C. H. Wang, R. F. Klie, Y. P. Chen, P. Kral, and A. Salehi-Khojin. Chemical Sensing with Switchable Transport Channels in Graphene Grain Boundaries. *Nat. Commun.*, 5:4911, 2014.

[200] K. Yasuda and H. Maruoka. Efficient Calculation of Two-Electron Integrals for High Angular Basis Functions. *Int. J. Quantum Chem.*, 114(9):543–552, 2014.

[201] T. Yoshikawa and H. Nakai. Linear-Scaling Self-Consistent Field Calculations Based on Divide-and-Conquer Method Using Resolution-of-Identity Approximation on Graphical Processing Units. *J. Comput. Chem.*, 36(3):164–170, 2015.

[202] K. Yoshizawa, A. Ito, K. Tanaka, and T. Yamabe. Unrestricted Hartree-Fock Method for Infinite Systems with Antiferromagnetic Array: Analysis of Antiferromagnetic State of Trans-Polyacetylene. *Int. J. Quantum Chem.*, 45:391–400, 1993.

[203] S. Zarrabian, C. R. Sarma, and J. Paldus. Vectorizable Approach to Molecular CI Problems Using Determinantal Basis. *Chem. Phys. Lett.*, 155(2):183–188, 1989.