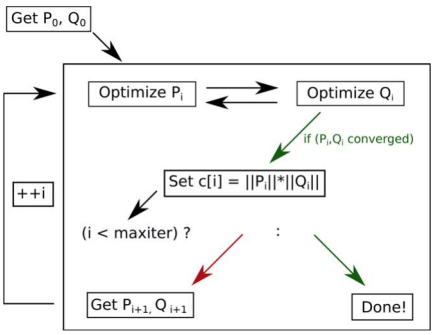
Large-Scale Electron Correlation Calculations: Rank-Reduced Full Configuration Interaction

B. Scott Fales,[†,‡] Stefan Seritan,[†,‡] Nick F. Settje,[†,‡] Benjamin G. Levine,[§] Henrik Koch,[∥] and Todd J. Martínez*[,†,‡]

[†]Department of Chemistry and the PULSE Institute, Stanford University, Stanford, California 94305, United States
[‡]SLAC National Accelerator Laboratory, Menlo Park, California 94305, United States
[§]Department of Chemistry, Michigan State University, East Lansing, Michigan 48824, United States
[∥]Department of Chemistry, Norwegian University of Science and Technology, 7491 Trondheim, Norway

ABSTRACT: We present the rank-reduced full configuration interaction (RR-FCI) method, a variational approach for the calculation of extremely large full configuration interaction (FCI) wave functions. In this report, we show that RR-FCI can provide ground state singlet and triplet energies within kcal/mol accuracy of full CI (FCI) with computational effort scaling as the square root of the number of determinants in the CI space (compared to conventional FCI methods which scale linearly with the number of determinants). Fast graphical processing unit (GPU) accelerated projected $\boldsymbol{\sigma} = \mathbf{Hc}$ matrix−vector product formation enables calculations with configuration spaces as large as 30 electrons in 30 orbitals, corresponding to an FCI calculation with over 2.4×10^{16} configurations. We apply this method in the context of complete active space configuration interaction calculations to acenes with 2−5 aromatic rings, comparing absolute energies against FCI when possible and singlet/triplet excitation energies against both density matrix renormalization group (DMRG) and experimental results. The dissociation of molecular nitrogen was also examined using both FCI and RR-FCI. In each case, we found that RR-FCI provides a low cost alternative to FCI, with particular advantages when relative energies are desired.



1. INTRODUCTION

The development of accurate and efficient methods that can describe electron correlation in complex molecules 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 a given basis set. Besides being formally exact, FCI is also conceptually simpler than other high accuracy methods. Unfortunately, because of its high computational expense and factorial scaling with respect to molecular size, FCI is rarely used for routine chemical investigations. Instead, it is usually used as a benchmarking tool for quantifying the relative accuracy of more tractable approximate methods.

Benchmark FCI calculations typically include all configurations arising from exciting any of the electrons within the entire orbital space. Another use of FCI is in the context of active space methods, where the configurations include all electronic excitations for a limited number of "active" electrons in a limited number of "active" orbitals. This defines the complete active space configuration interaction (CASCI) method. When the orbitals are also varied to minimize the resulting energy, the method is termed complete active space self-consistent field (CASSCF)[1−3] or fully optimized reaction space (FORS).[4] Both CASCI and CASSCF require solution of the FCI problem in a user-defined configuration space (or active space). These multiconfigurational approaches were developed to describe static electron correlation, where the electronic structure is poorly characterized by a single Slater determinant. This situation often arises in polyradicaloid systems that are encountered in many photochemical processes, in transition metal systems, and in molecules where covalent bonds have been stretched far from the equilibrium distance.

FCI may be considered a mature electronic structure method. The development of direct CI methods reduced memory requirements significantly, since these avoid explicit formation and diagonalization of the full Hamiltonian matrix.[5] The success of direct CI has motivated development of numerical approaches designed for the problem of iterative matrix diagonalization.[6] Additional progress by Siegbahn[7] resulted in approaches even better suited for modern computing by leveraging fast matrix−vector and matrix−matrix operations, and further refinement in the context of vector processing machines has been performed.[8−10] In FCI programs, Slater determinants are often used en lieu of spin-adapted configuration state functions due to the convenient factorization of the determinantal CI vector into α and β strings and because the one-particle coupling coefficients become simple integers.[11] Olsen's determinant-based algorithms for

solving the FCI and restricted active space CI (RASCI) problems enabled early calculations of unprecedented size.[12,13] Even when using cutting edge hardware and highly efficient algorithms, configuration spaces for high-throughput calculations are still limited to approximately 10^{10} determinants.[14] 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 computational time and memory constraints.

Efforts to improve scaling can be focused on either (or both) of the many-electron configuration space or one-electron orbital space dimensions. These are of course linked, since the size of the configuration space is a factorial function of the number of orbitals. However, the distinction arises because the formal scaling of direct CI is $O(N_{\text{det}}^2 N_o^4)$, where N_{det} is the number of configurations (determinants in our case) and N_o is the number of (active) molecular orbitals (MOs). In practice, the sparsity of the Hamiltonian matrix reduces this to $O(N_{\text{det}} N_o^4)$. Algorithms having smaller operation counts have been described, for example by Olsen,[12] but these are difficult to implement efficiently for vector machines. Approaches that attempt to improve the N_o^4 (orbital space) scaling include pseudospectral methods,[15,16] density fitting and Cholesky decomposition,[17−22] and tensor hypercontraction.[23−29] These approaches all decompose the fourth-order tensor representing the MO basis electron repulsion integrals as a product of lower-order tensors. This decomposition can decrease both the scaling of the integral transformation and the computational cost of the $\boldsymbol{\sigma} = \mathbf{Hc}$ vector formation during the direct CI iterations. Unfortunately, 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. Thus, schemes aimed at improving the orbital space scaling have been of limited utility so far. This motivates the development of methods that directly reduce the scaling with respect to the size of the configuration space.

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[30−33] and stochastics.[34] 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-FCI[35,36] (QMC-FCI) have been demonstrated in some cases to provide accurate descriptions of correlated Fermionic systems. Unfortunately, uncontrolled errors resulting from the constraint imposed by the antisymmetry principle (i.e., the Fermion sign problem) remain open challenges in QMC-FCI.[37] Alternatively, variational methods based on tensor networks, such as density matrix renormalization group[38,39] (DMRG), provide high-accuracy deterministic solutions for both one-dimensional and branched/tree-like systems using matrix product states and tree tensor network states,[40] respectively. Systems possessing higher orders of quantum entanglement, however, are still challenging problems for DMRG.

Another productive approach for reducing the configuration space scaling relies on taking advantage of sparsity in the CI vector. Examples include methods by Knowles,[41,42] Mitrushenkov,[43,44] and Rolik.[45] 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 a spectral resolution of the CI coefficient matrix[46] to compress the wave function. They used a second-order Newton−Raphson scheme to solve the CI with single and double replacements (CISD) problem. LR-CISD 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 wave functions were employed. Lindh et al. followed this work by implementing a low-rank multiconfiguration self-consistent field (LR-SCF) method[47] and demonstrated that polarizabilities were better described using LR-SCF than with LR-CI, though wave function convergence was hindered by strong coupling between orbital rotations and single excitations. Attacking the CI problem from another angle, Koch and Dalgaard formulated a variational matrix decomposition using nonlinear optimization methods to solve the FCI problem.[48] A pilot version of a program performing the decomposition was applied to the ground state of beryllium atom. Rapid convergence to the FCI energies within mH accuracy was observed in this case after 25 iterations, and 95% of the correlation energy was obtained after only 10 iterations.

Both the LR-CI and variational matrix decomposition methods rely on the assumption that 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 reported a study where the singular 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.[49] We refer to this as the low-rank SVD (LR-SVD) method. Several representative FCI cases were examined, including examples of 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) CI matrices. These naming conventions were derived from the shape of the graphical representations of the corresponding CI spaces,[50] where the slim graph is tall and narrow and the stout graph is more symmetrical. 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). Instead of demonstrating the low rank of the converged CI vectors, however, it was observed that a relatively large rank of the CI vector was required to achieve the desired accuracy. The high computational cost of the SVD operation ($N_{\text{det}}^{3/2}$ versus $\approx N_{\text{det}}$ for a $\boldsymbol{\sigma}$ build in standard direct CI) and the absence of a reduction in trial vector length in later LR-SVD iterations imply that the LR-SVD approach has few advantages in large-scale production calculations.

As described above, writing the CI vector ($\mathbf{c}$) in matrix form ($\mathbf{C}$), 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,[48] where the product space vectors are determined variationally through solution of a nonlinear multiscale eigenvalue problem (a type of super-CI approach, similar to that which is often used in two-step CASSCF implementations[51,52]), in developing a rank-reduced FCI (RR-FCI) formulation. The structure of the paper is as follows: in Section 2, we introduce the augmented eigenvalue problem for the product space vectors (Section 2.1),

describe projected $\boldsymbol{\sigma}$ formation (Section 2.2), and provide implementation details including one-particle coupling coefficient formation (Section 2.3.1), eigenvalue problem solution (Section 2.3.2), metric formation (Section 2.3.3), and projected $\boldsymbol{\sigma}$ formation (Section 2.3.4). Next, we demonstrate computational performance of the RR-FCI method (Section 3.2) before applying RR-FCI to acenes having 2−5 polycyclic aromatic rings (Section 3.3) and to molecular nitrogen dissociation (Section 3.5). We conclude with a brief review of the RR-FCI method before discussing future directions of work.

2. METHODS

The second-quantized electronic Hamiltonian is

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

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 spin-averaged single-particle excitation operator.

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 focus on the calculation of the ground state (lowest for a given spin) eigenvalue/eigenvector pair. Future work will explore the determination of higher-lying (excited) states. The rate-limiting step in the direct CI procedure is the formation of the matrix−vector product $\boldsymbol{\sigma}$ defined as

$$\sigma_I = \sum_J H_{IJ}c_J \tag{2}$$

Formally, $\boldsymbol{\sigma}$ formation requires $O(N_{\text{det}}^2)$ operations, but modern algorithms exploit Hamiltonian sparsity to achieve an effective scaling approaching $O(N_{\text{det}})$. In this work, we describe an approach that reduces the formal scaling to $O(N_{\text{det}})$ with an observed effective scaling of $O(\sqrt{N_{\text{det}}})$ for the rate-limiting $\boldsymbol{\sigma}$ formation.

2.1. Eigenvalue Problem. We begin by describing the variational decomposition approach taken by Koch and Dalgaard. We largely restrict our attention to singlet spin states in the following. In a determinantal basis, the full CI wave function can be written as

$$|\psi\rangle = \sum_{I=1}^{N_{\text{det}}} d_I|\Phi_I\rangle = \sum_{J=1}^{N_\alpha}\sum_{K=1}^{N_\beta} C_{JK}|\phi_J^\alpha, \phi_K^\beta\rangle \tag{3}$$

where N_α and N_β are the number of α and β strings, and $N_{\text{det}} = N_\alpha N_\beta$ is the number of determinants, labeled as $|\Phi_I\rangle$ with coefficients d_I or as a product of α/β strings $|\phi_J^\alpha,\phi_K^\beta\rangle$ with coefficients C_{JK}. The wave function coefficients, $\mathbf{C}$, expressed as a matrix according to the second equality in eq 3, can be decomposed as

$$\mathbf{C} = \sum_i^{N_r} c_i(\mathbf{P}^i(\mathbf{Q}^i)^\dagger) \equiv \sum_i^{N_r} c_i(\mathbf{P}^i \otimes \mathbf{Q}^i) = \sum_i^{N_r} c_i\Psi_i \tag{4}$$

where N_r (the allowed rank, or equivalently, the number of product terms) is less than or equal to the full rank of $\mathbf{C}$, c_i are product term coefficients, and $\mathbf{P}$ and $\mathbf{Q}$ are product state vectors of length N_α and N_β, respectively. In the following, we adopt the convention of normalizing the $\mathbf{P}^i$ and $\mathbf{Q}^i$ vectors as follows:

$$\Psi_i = \mathbf{P}^i \otimes \mathbf{Q}^i \tag{5}$$

$$\Psi_i^*\Psi_i = Tr((\mathbf{Q}^i \otimes \mathbf{P}^i)(\mathbf{P}^i \otimes \mathbf{Q}^i)) = |\mathbf{P}^i|^2|\mathbf{Q}^i|^2 = 1 \tag{6}$$

$$|\mathbf{P}^i|^2 = |\mathbf{Q}^i|^2 = 1 \tag{7}$$

The vectors $\mathbf{P}$ and $\mathbf{Q}$ correspond to coefficients of product space basis functions $\boldsymbol{\phi}^\alpha$ and $\boldsymbol{\phi}^\beta$. This provides us with an expression for the initial wave function, $|\psi_0\rangle$

$$\begin{aligned}|\psi_0\rangle &= \sum_J^{N_\alpha}\sum_K^{N_\beta}|\phi_J^\alpha, \phi_K^\beta\rangle P_J^0 Q_K^0 = \sum_J^{N_\alpha}\sum_K^{N_\beta}|P_J^0, Q_K^0\rangle \\ &= (\boldsymbol{\phi}^\alpha)^T(\mathbf{P}^0 \otimes \mathbf{Q}^0)\boldsymbol{\phi}^\beta\end{aligned} \tag{8}$$

where P_J^0 and Q_K^0 are elements of $\mathbf{P}^0$ and $\mathbf{Q}^0$. Equivalently, we can also expand the initial wave function in terms of the product space basis sets

$$|\psi_0\rangle = \sum_J^{N_\alpha}\sum_K^{N_\beta}|\phi_J^\alpha, Q_K^0\rangle P_J^0 \tag{9}$$

$$|\psi_0\rangle = \sum_J^{N_\alpha}\sum_K^{N_\beta}|P_J^0, \phi_K^\beta\rangle Q_K^0 \tag{10}$$

The full wave function can be expressed as

$$|\psi\rangle = \sum_i^{N_r} c_i|\psi_i\rangle = \sum_i^{N_r} c_i\sum_J^{N_\alpha}\sum_K^{N_\beta}|\phi_J^\alpha, \phi_K^\beta\rangle P_J^i Q_K^i \tag{11}$$

Applying the variational principle to the full wave function leads to coupled equations for the first set of product terms

$$\sum_J^{N_\alpha}\sum_K^{N_\beta}\langle\phi_{J'}^\alpha, Q_K|\hat{H} - E|\phi_J^\alpha, Q_K\rangle P_J = 0 \quad \forall\ J' \tag{12}$$

$$\sum_J^{N_\alpha}\sum_K^{N_\beta}\langle P_J, \phi_{K'}^\beta|\hat{H} - E|P_J, \phi_K^\beta\rangle Q_K = 0 \quad \forall\ K' \tag{13}$$

Addition of the second and subsequent product terms result in coupled augmented eigenvalue problems

$$\begin{bmatrix}\langle\psi|\hat{H} - E|\psi\rangle & \sum_K^{N_\beta}\langle\psi|\hat{H} - E|\phi_J^\alpha, Q_K\rangle \\ \sum_K^{N_\beta}\langle\phi_{J'}^\alpha, Q_K|\hat{H} - E|\psi\rangle & \sum_K^{N_\beta}\langle\phi_{J'}^\alpha, Q_K|\hat{H} - E|\phi_J^\alpha, Q_K\rangle\end{bmatrix}\begin{bmatrix}a_0 \\ \mathbf{P}\end{bmatrix} = \begin{bmatrix}0 \\ \mathbf{0}\end{bmatrix} \tag{14}$$

for the $\mathbf{P}$ optimization and

$$\begin{bmatrix}\langle\psi|\hat{H} - E|\psi\rangle & \sum_J^{N_\alpha}\langle\psi|\hat{H} - E|P_J, \phi_K^\beta\rangle \\ \sum_J^{N_\alpha}\langle P_J, \phi_K^\beta|\hat{H} - E|\psi\rangle & \sum_J^{N_\alpha}\langle P_J, \phi_{K'}^\beta|\hat{H} - E|P_J, \phi_K^\beta\rangle\end{bmatrix}\begin{bmatrix}a_0 \\ \mathbf{Q}\end{bmatrix} = \begin{bmatrix}0 \\ \mathbf{0}\end{bmatrix} \tag{15}$$

for the $\mathbf{Q}$ optimization, where a_0 is a scalar which couples the eigenvalue problems. As previously described, this approach provides fast convergence to within mH accuracy of the FCI energy. Unfortunately, this formulation is prone to spin contamination, and only singlet states may be described.[48] We can

remedy both of these issues by modifying this ansatz to be a linear combination of products of **P** and **Q** vectors.

$$\Psi_i = \mathbf{P}^i \otimes \mathbf{Q}^i \pm \mathbf{Q}^i \otimes \mathbf{P}^i \tag{16}$$

where the singlet corresponds to the symmetric case (+) and the triplet to the antisymmetric case (−). This linear combination requires **P** and **Q** to be of length N_{str}, where $N_{str} = N_\alpha = N_\beta$ is the number of α or β strings for the restricted case described here. From this point on, we assume that $N_\alpha = N_\beta$ and leave it to future work to generalize the formalism further. While wave function symmetry/antisymmetry alone does not guarantee that the final eigenvector is a spin eigenfunction (quintet states can, in principle, contaminate the singlet states, for example), we find that in practice this is not a serious issue. We have performed a detailed analysis of the expectation value of S^2 for the RR-FCI wave function in several test cases and refer the interested reader to the Supporting Information. The coupled equations for the initial set of product space vectors then become

$$\sum_{J,K}^{N_{str}} |\langle(\phi_{J'}^{\alpha}, Q_K) \pm (Q_K, \phi_{J'}^{\beta})|\hat{H} - E|(\phi_J^{\alpha}, Q_K) \pm (Q_K, \phi_J^{\beta})\rangle|P_J = 0 \quad \forall\ J' \tag{17}$$

$$\sum_{J,K}^{N_{str}} |\langle(P_J, \phi_{K'}^{\beta}) \pm (\phi_{K'}^{\alpha}, P_J)|\hat{H} - E|(P_J, \phi_K^{\beta}) \pm (\phi_K^{\alpha}, P_J)\rangle|Q_K = 0 \quad \forall\ K' \tag{18}$$

giving the eigenvalue problem for the subsequent product terms as

$$\begin{bmatrix} \langle\psi|\hat{H} - E|\psi\rangle & \sum_K^{N_{str}} \langle\psi|\hat{H} - E|(\phi_J^{\alpha}, Q_K) \pm (Q_K, \phi_J^{\beta})\rangle \\ \sum_K^{N_{str}} \langle(\phi_{J'}^{\alpha}, Q_K) \pm (Q_K, \phi_{J'}^{\beta})|\hat{H} - E|\psi\rangle & \sum_K^{N_{str}} \langle(\phi_{J'}^{\alpha}, Q_K) \pm (Q_K, \phi_{J'}^{\beta})|\hat{H} - E|(\phi_J^{\alpha}, Q_K) \pm (Q_K, \phi_J^{\beta})\rangle \end{bmatrix} \begin{bmatrix} a_0 \\ \mathbf{P} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{0} \end{bmatrix} \tag{19}$$

for the **P** optimization and

$$\begin{bmatrix} \langle\psi|\hat{H} - E|\psi\rangle & \sum_J^{N_{str}} \langle\psi|\hat{H} - E|(P_J, \phi_K^{\beta}) \pm (\phi_K^{\alpha}, P_J)\rangle \\ \sum_J^{N_{str}} \langle(P_J, \phi_{K'}^{\beta}) \pm (\phi_{K'}^{\alpha}, P_J)|\hat{H} - E|\psi\rangle & \sum_J^{N_{str}} \langle(P_J, \phi_{K'}^{\beta}) \pm (\phi_{K'}^{\alpha}, P_J)|\hat{H} - E|(P_J, \phi_K^{\beta}) \pm (\phi_K^{\alpha}, P_J)\rangle \end{bmatrix} \begin{bmatrix} a_0 \\ \mathbf{Q} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{0} \end{bmatrix} \tag{20}$$

for the **Q** optimization. The full wave function can be expressed as

$$|\psi\rangle = \sum_i^{N_r} c_i \sum_{J,K}^{N_{str}} |(\phi_J^{\alpha}, \phi_K^{\beta}) \pm (\phi_K^{\alpha}, \phi_J^{\beta})\rangle P_J^i Q_K^i \tag{21}$$

2.2. 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,[12] where $\boldsymbol{\sigma}$ is separated into three terms

$$\sigma = \sigma_1 + \sigma_2 + \sigma_3 \tag{22}$$

with

$$(\sigma_1)_{J'K} = \sum_J^{N_\alpha} \sum_{kl}^{N_o} \gamma_{kl}^{JJ'} h'_{kl} C_{JK} + \frac{1}{2} \sum_{J,M}^{N_\alpha} \sum_{ijkl}^{N_o} \gamma_{ij}^{JM} \gamma_{kl}^{MJ'} (ij|kl) C_{JK} \tag{23}$$

$$(\sigma_2)_{JK'} = \sum_K^{N_\beta} \sum_{kl}^{N_o} \overline{\gamma}_{kl}^{KK'} h'_{kl} C_{JK} + \frac{1}{2} \sum_{K,M}^{N_\beta} \sum_{ijkl}^{N_o} \overline{\gamma}_{ij}^{KM} \overline{\gamma}_{kl}^{MK'} (ij|kl) C_{JK} \tag{24}$$

$$(\sigma_3)_{J'K'} = \sum_J^{N_\alpha} \sum_K^{N_\beta} \sum_{ijkl}^{N_o} \overline{\gamma}_{ij}^{KK'} \gamma_{kl}^{JJ'} (ij|kl) C_{JK} \tag{25}$$

where J,J' index α occupation strings, K,K' index β occupation strings, M indexes both α and β strings, and γ,$\overline{\gamma}$ are the one-particle coupling coefficients defined as

$$\gamma_{ij}^{JJ'} = \langle\phi_J^{\alpha}|\hat{E}_{ij}|\phi_{J'}^{\alpha}\rangle \tag{26}$$

$$\overline{\gamma}_{ij}^{KK'} = \langle\phi_K^{\beta}|\hat{E}_{ij}|\phi_{K'}^{\beta}\rangle \tag{27}$$

with $\hat{\boldsymbol{E}}_{ij}$ a single particle generator either between α occupation strings J and J' or β occupation strings K and K' corresponding to an excitation between orbitals i and j. In the case we consider here, where $N_\alpha = N_\beta$, the α/β coupling coefficients are equal

$$\overline{\gamma}_{ij}^{KK'} = \langle\phi_K^{\beta}|\hat{\boldsymbol{E}}_{ij}|\phi_{K'}^{\beta}\rangle = \gamma_{ij}^{KK'} = \langle\phi_K^{\alpha}|\hat{\boldsymbol{E}}_{ij}|\phi_{K'}^{\alpha}\rangle \tag{28}$$

Note that in eqs 23−25, we have combined the one-electron part of the two-electron integrals $(ij|kl)$ with the one-electron integrals h_{kl} for computational efficiency

$$h'_{kl} = h_{kl} - \sum_j^{N_o} (kj|jl) \tag{29}$$

Two different types of $\boldsymbol{\sigma}$ formations correspond to eqs 12 and 13. Projection onto the **P** vector space results in

$$\sigma_{J'}^{P} = \sum_J^{N_\alpha} \sum_K^{N_\beta} Q_K \langle\phi_{J'}^{\alpha}, \phi_K^{\beta}|\hat{H}|\phi_J^{\alpha}, \phi_K^{\beta}\rangle P_J Q_K \tag{30}$$

while projection onto the **Q** vector space gives

$$\sigma_{K'}^{Q} = \sum_J^{N_\alpha} \sum_K^{N_\beta} P_J \langle\phi_J^{\alpha}, \phi_{K'}^{\beta}|\hat{H}|\phi_J^{\alpha}, \phi_K^{\beta}\rangle P_J Q_K \tag{31}$$

We can think about projected $\boldsymbol{\sigma}$ formation in terms of operations that span the full determinantal space. Figure 1 depicts how a standard CI program can be used to form the projected $\boldsymbol{\sigma}$ vector corresponding to eq 31.

Frame 1 from Figure 1 defines formation of the rank 1 full CI vector **C** from the **P** and **Q** vectors. The outer product of the **P** and **Q** vectors forms a matrix $\mathbf{C}_{JK}$. Expanding this matrix

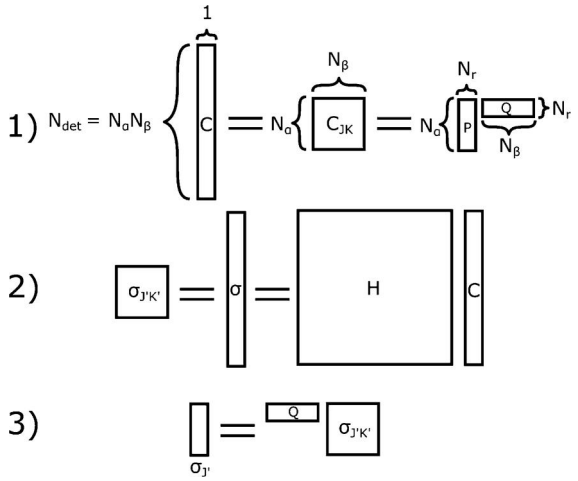


Figure 1. Pictorial representation of projected $\boldsymbol{\sigma}$ formation using a full CI program. Subscripts correspond to basis indices of determinant space vectors represented as matrices in product space. Array dimensions are provided in the uppermost panel where N_α and N_β are the numbers of α and β strings and N_r is the rank of the decomposition (number of product terms in RR-FCI).

produces the standard full CI vector, **C**. The second frame corresponds to formation of the $\boldsymbol{\sigma}$ vector, which is then folded into matrix form, $\boldsymbol{\sigma}_{J'K'}$. Finally, $\boldsymbol{\sigma}_{J'K'}$ is then projected by the **Q** vector in frame 3 to give the projected $\boldsymbol{\sigma}$ vector, $\boldsymbol{\sigma}^{\mathbf{P}}$. While this approach assists in understanding the nature of the data structures involved in the algorithm, direct formation of the factorized $\boldsymbol{\sigma}$ vector is needed to avoid unnecessary computation and storage.

Factorizing the projected $\boldsymbol{\sigma}$ results in three terms

$$(\sigma_1)^P_{J'} = \sum_J^{N_{str}} \sum_K^{N_{str}} Q_K \left(\sum_{kl}^{N_o} \gamma_{kl}^{JJ'} h'_{kl} + \frac{1}{2} \sum_M^{N_{str}} \sum_{ijkl}^{N_o} \gamma_{ij}^{JM} \gamma_{kl}^{MJ'} (ij|kl) \right) P_J Q_K$$
$$= \left(\sum_K^{N_{str}} Q_K Q_K \right) \sum_J^{N_{str}} \left(\sum_{kl}^{N_o} \gamma_{kl}^{JJ'} h'_{kl} + \frac{1}{2} \sum_M^{N_{str}} \sum_{ijkl}^{N_o} \gamma_{ij}^{JM} \gamma_{kl}^{MJ'} (ij|kl) \right) P_J \quad (32)$$

$$(\sigma_2)^P_{J'} = \sum_J^{N_{str}} \sum_K^{N_{str}} Q_J \left(\sum_{kl}^{N_o} \overline{\gamma}_{kl}^{KJ'} h'_{kl} + \frac{1}{2} \sum_M^{N_{str}} \sum_{ijkl}^{N_o} \overline{\gamma}_{ij}^{KM} \overline{\gamma}_{kl}^{MJ'} (ij|kl) \right) P_J Q_K$$
$$= \left(\sum_J^{N_{str}} P_J Q_J \right) \sum_K^{N_{str}} \left(\sum_{kl}^{N_o} \overline{\gamma}_{kl}^{KJ'} h'_{kl} + \frac{1}{2} \sum_M^{N_{str}} \sum_{ijkl}^{N_o} \overline{\gamma}_{ij}^{KM} \overline{\gamma}_{kl}^{MJ'} (ij|kl) \right) Q_K \quad (33)$$

$$(\sigma_3)^P_{J'} = \sum_J^{N_{str}} \sum_{K,K'}^{N_{str}} Q_{K'} \sum_{ijkl}^{N_o} \overline{\gamma}_{ij}^{KK'} \gamma_{kl}^{JJ'} (ij|kl) P_J Q_K$$
$$= \sum_{ijkl}^{N_o} \left(\sum_{K,K'}^{N_{str}} Q_{K'} \overline{\gamma}_{ij}^{KK'} Q_K \right) \left(\sum_J^{N_{str}} \gamma_{kl}^{JJ'} P_J \right) (ij|kl) \quad (34)$$

for the **P** factorization and

$$(\sigma_1)^Q_{K'} = \sum_J^{N_{str}} \sum_K^{N_{str}} P_K \left(\sum_{kl}^{N_o} \gamma_{kl}^{JK'} h'_{kl} + \frac{1}{2} \sum_M^{N_\alpha} \sum_{ijkl}^{N_o} \gamma_{ij}^{JM} \gamma_{kl}^{MK'} (ij|kl) \right) P_J Q_K$$
$$= \left(\sum_K^{N_{str}} P_K Q_K \right) \sum_J^{N_{str}} \left(\sum_{kl}^{N_o} \gamma_{kl}^{JK'} h'_{kl} + \frac{1}{2} \sum_M^{N_{str}} \sum_{ijkl}^{N_o} \gamma_{ij}^{JM} \gamma_{kl}^{MK'} (ij|kl) \right) P_J \quad (35)$$

$$(\sigma_2)^Q_{K'} = \sum_J^{N_{str}} \sum_K^{N_{str}} P_J \left(\sum_{kl}^{N_o} \overline{\gamma}_{kl}^{KK'} h'_{kl} + \frac{1}{2} \sum_M^{N_{str}} \sum_{ijkl}^{N_o} \overline{\gamma}_{ij}^{KM} \overline{\gamma}_{kl}^{MK'} (ij|kl) \right) P_J Q_K$$
$$= \left(\sum_J^{N_{str}} P_J P_J \right) \sum_K^{N_{str}} \left(\sum_{kl}^{N_o} \overline{\gamma}_{kl}^{KK'} h'_{kl} + \frac{1}{2} \sum_M^{N_{str}} \sum_{ijkl}^{N_o} \overline{\gamma}_{ij}^{KM} \overline{\gamma}_{kl}^{MK'} (ij|kl) \right) Q_K \quad (36)$$

$$(\sigma_3)^Q_{K'} = \sum_{J,J'}^{N_{str}} \sum_K^{N_{str}} P_{J'} \sum_{ijkl}^{N_e} \overline{\gamma}_{ij}^{KK'} \gamma_{kl}^{JJ'} (ij|kl) P_J Q_K$$
$$= \sum_{ijkl}^{N_o} \left(\sum_K^{N_{str}} \overline{\gamma}_{ij}^{KK'} Q_K \right) \left(\sum_{J,J'}^{N_{str}} P_{J'} \gamma_{kl}^{JJ'} P_J \right) (ij|kl) \quad (37)$$

for the **Q** terms. Since we focus on the restricted case where $N_\alpha = N_\beta$, we do not distinguish between $\boldsymbol{\gamma}$ and $\overline{\boldsymbol{\gamma}}$ for the remainder of this work. Note that this restriction also leads to a loss of the correspondence between α/β and J/K indices in eqs 32−37. For each projected $\boldsymbol{\sigma}_2$ formation, three state vectors are required in addition to the current trial vector. For the **P** optimization, for example, the current **Q** vector is needed in addition to each of the previously converged **P** and **Q** vectors, and the **P** being optimized is taken to be the current trial vector.

2.3. Computational Methods. *2.3.1. Coupling Coefficient Formation.* Data structures required for evaluation of $\boldsymbol{\sigma}$ include the one-particle coupling coefficients $\boldsymbol{\gamma}$, the orbital labels **l**, and the configuration labels **IJ**. When possible, our program constructs and stores these lists in memory at the beginning of the calculation rather than computing elements on-the-fly. Forming these lists for configuration spaces typical of direct CI calculations is a routine task and does not require exceptional computational effort. In contrast, the large configuration spaces accessible to RR-FCI require extremely large lists. To handle these cases, we have developed an efficient GPU accelerated algorithm listed in Algorithm 1 for construction of the one-particle coupling coefficients and related data structures in a lexical (and dense) format. The approach described here is also well suited for forming these lists on-the-fly if desired.

Algorithm 1 Vectorized formation of coupling coefficients $\boldsymbol{\gamma}$, orbital labels **l**, and configuration labels **IJ**. N_e is the number of α electrons and **bin_rep** is a precomputed integer array defining the orbital occupations using binary representation. *stradr*() is a function that determines the string address using lexical ordering as defined by Knowles and Handy[8] (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 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 loop of Algorithm 1 can be tiled to enable vectorization over multiple GPUs and/or compute nodes. The *stradr*() function is an implementation of eqs 11 and 12 from Knowles and Handy,[8] where the binomial coefficients are retrieved from a precomputed list (i.e., Pascal's triangle) rather than calculated on-the-fly. 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, $(N_{str} * N_e(N_o - N_e))$, 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 1 provides configuration space and array sizes for some of the CAS spaces used in this study.

Table 1. Active space sizes and memory requirements for supporting data structures including orbital labels and coupling coefficients (in GB)[a]

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

[a]Note that storage for P, Q, σ, and Davidson trial vectors are not included.

2.3.2. Eigenvalue Problem. Our program solves a nonlinear eigenvalue problem using a two-step Davidson approach, where we iterate over **P** and **Q** vector optimization until they are self-consistently converged. The procedure we use is given graphically in Figure 2. The inner (micro)iterations are performed using a generalized Davidson algorithm to solve the nonorthonormal subspace eigenvalue problem. The outer (macro)iterations are considered converged when the microiterations for each of the **P** and **Q** vectors converge in a single step. We "balance" the eigenvalue problem by scaling the projected metric matrix as recommended previously.[53,54] A residual norm convergence criterion of $||r|| = 1.0 \times 10^{-6}$ is used for the microiterations.

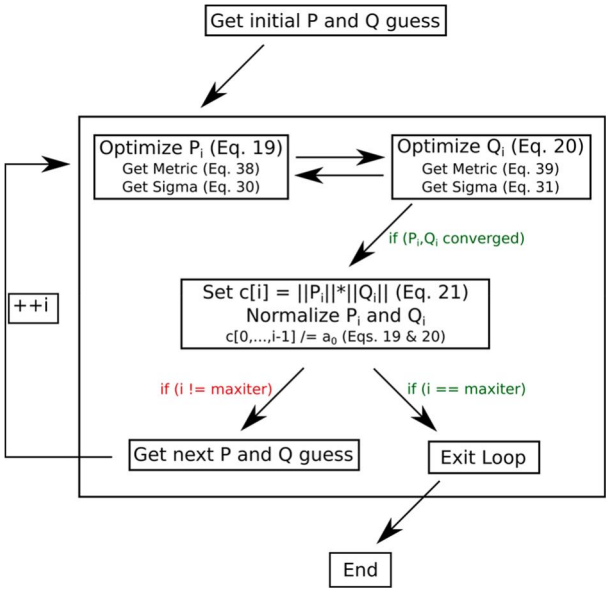


Figure 2. RR-FCI program flow. Note that in the first iteration, eqs 19 and 20 are simplified because $|\Psi\rangle = 0$.

To begin the iterative procedure for the **P** and **Q** vector optimizations, we must provide a guess vector for each. In the first macroiteration, the guess for each of $\mathbf{P}^0$ and $\mathbf{Q}^0$ is the vector corresponding to the Hartree−Fock determinant, i.e., $\langle 0,1,0,0,0,\cdots\rangle$, where the first vector element corresponds to the coefficient a_0 that couples the **P** and **Q** eigenvalue problems. In the second and subsequent macroiterations, the guess for the **P** vector becomes $\langle 1,0,0,0,0,\cdots\rangle$, which corresponds to the previously converged **P**,**Q** pairs being used as the $\mathbf{P}^i$ guess. The $\mathbf{Q}^i$ guess corresponds to a unit vector in the direction of the ith macroiteration; i.e., if we are adding the third **P**,**Q** pair, the **Q** guess is $\langle 0,0,0,1,0,\cdots\rangle$. This permits efficient sampling of the configuration space while avoiding linear dependence of the added trial vectors. Other guesses for **Q** can be conceived; the original implementation used a uniform **Q** guess, i.e., $\langle 0, 1/\sqrt{N_{str}}, 1/\sqrt{N_{str}}, 1/\sqrt{N_{str}}, \ldots\rangle$, and **Q** vector guesses derived from projected residual vector formation (similar to typical trial vector formation in standard Davidson algorithms) appear promising. In our tests, we investigated all three of these options. Interestingly, the unit guess approach (uniformly sampling the configuration space) results in the best convergence, with the other approaches producing linearly dependent trial vectors shortly after mH accuracy is obtained.

Preconditioning of the Davidson routine is performed using the reference determinant energy modified by orbital energy differences as described by Evangelisti.[55] Advantages of orbital energy difference preconditioning are that spin contamination is not introduced into the trial vectors, and convergence is generally similar to that observed when using exact diagonal energies. Convergence of the RR-FCI microiterations generally requires 10−30 iterations. To test the propensity for spin contamination of each of the ansatzes described in eqs 11 and 21, we have reconstructed the final wave function $|\Psi\rangle$ for a variety of manageable configuration spaces (i.e., 16 electrons in 16 orbitals and smaller). The spin purity of the full wave function $|\Psi\rangle$ and for each contributing **P**,**Q** pair was assessed by directly calculating the expectation value of S^2. In doing so we confirmed the presence of spin contamination in the ansatz described in eq 11 and the absence of spin contamination in the wave function defined in eq 21.

2.3.3. Metric Formation. Since we are solving a generalized (nonorthonormal) eigenvalue problem, it is necessary to build the subspace metric matrix/trial vector product corresponding to eqs 19 and 20. We form the projected metric according to

$$\mathbf{B}^P = \begin{bmatrix} \langle\psi|\psi\rangle & \sum_K^{N_{str}} \langle\psi|(\phi_J^\alpha, Q_K) \pm (Q_K, \phi_J^\beta)\rangle \\ \sum_K^{N_{str}} \langle(\phi_{J'}^\alpha, Q_K) \pm (Q_K, \phi_{J'}^\beta)|\psi\rangle & \sum_K^{N_{str}} \langle(\phi_{J'}^\alpha, Q_K) \pm (Q_K, \phi_{J'}^\beta)||(\phi_J^\alpha, Q_K) \pm (Q_K, \phi_J^\beta)\rangle \end{bmatrix} \begin{bmatrix} a_0 \\ \mathbf{P} \end{bmatrix} \tag{38}$$

for the **P** vector optimization and

$$\mathbf{B}^Q = \begin{bmatrix} \langle\psi|\psi\rangle & \sum_K^{N_{str}} \langle\psi|(P_J, \phi_K^\beta) \pm (\phi_K^\alpha, P_J)\rangle \\ \sum_K^{N_{str}} \langle(P_{J'}, \phi_{K'}^\beta) \pm (\phi_{K'}^\alpha, P_{J'})|\psi\rangle & \sum_K^{N_{str}} \langle(P_{J'}, \phi_{K'}^\beta) \pm (\phi_{K'}^\alpha, P_{J'})||(P_J, \phi_K^\beta) \pm (\phi_K^\alpha, P_J)\rangle \end{bmatrix} \begin{bmatrix} a_0 \\ \mathbf{Q} \end{bmatrix} \tag{39}$$

for the **Q** vector optimization. The algorithm for computing the metric is low enough scaling that we have implemented it in serial on the host (CPU). Details are given in Algorithm 2. Projected metric formation requires double precision dot product (DDOT) and **y** = **Ax** + **y** (DAXPY) operations, both of which are performed using optimized linear algebra library routines.[56]

Algorithm 2 Metric formation for **P** vector optimization. N_{conv} is the number of converged product terms. **B** is the projected metric vector of dimension $N_{str}+1$, **c** are the coefficients of the previously converged **P** and **Q**, **C** is the trial vector for the augmented eigenvalue problem (in this case the current **P** vector), and $\mathbf{Q_{curr}}$ is the current **Q** vector corresponding to the trial vector **C**.

```
for i, j < N_conv do
    B[0] += 2.0 * a_0 * c_i * c_j * (P_i · P_j) (Q_i · Q_j)        ▷ Upper left block of metric
    B[0] ±= 2.0 * a_0 * c_i * c_j * (P_i · Q_j) (Q_i · P_j)
end for
Allocate work[N_str]
for i < N_conv do
    for j < N_str do
        work[j] += 2.0 * c_i * (Q_curr · Q_i) * P_i[j]
        work[j] ±= 2.0 * c_i * (Q_curr · P_i) * Q_i[j]
    end for
end for
B[0] += (work · C)                                                 ▷ Upper right block of metric
for i < N_str do
    B[i + 1] += a_0 * work[i]                                      ▷ Bottom left block of metric
end for
for i < N_str do
    B[i + 1] += 2.0 * (Q_curr · Q_curr) * C[i]                     ▷ Bottom right block of metric
    B[i + 1] ±= 2.0 * (Q_curr · C) * Q_curr[i]
end for
```

2.3.4. Projected $\boldsymbol{\sigma}$ Formation. We have developed projected $\boldsymbol{\sigma}$ formation algorithms based on a hybrid of the Olsen separation[12] and the Knowles and Handy FCI algorithm.[8] A configuration label-driven scheme for formation of $\boldsymbol{\sigma}^{\mathbf{P}}$ is described in Algorithms 3−5 . Orbital label-driven algorithms which more closely resemble the Olsen CI separation are presented in the Supporting Information.

Algorithm 3 GPU vectorized projected $\boldsymbol{\sigma_1}$ algorithm. $\mathbf{Q_l}$, $\mathbf{Q_r}$, and $\mathbf{P_r}$ correspond to the left **Q** vector, the right **Q** vector, and the **P** vector in Equation 32. Q is a floating point scalar, **l** are orbital indices given a string pair I, I', and **D** and **E** are floating point arrays.

```
Q = ±(Q_l · Q_r)
GPU vectorize over strings I, I′ differing from I by zero or one occupations
    ij ← l[I][I′]
    D[ij, I] += γ[I][I′]P_r[I′]                                    ▷ D_ij formation
end GPU vectorize
GPU vectorize over strings I
    ii ← l[I][I]
    D[ii, I] += P_r[I]                                             ▷ D_ii formation
end GPU vectorize
σ += Q * D * h′_kl                                                 ▷ DGEMV
E ← 0.5 * Q * (ij|kl) * D                                          ▷ DGEMM
GPU vectorize over strings I, I′ differing from I by zero or one occupations
    ij ← l[I][I′]
    σ[I′] += γ[I][I′]E[ij, I]                                      ▷ σ_ij formation - atomic operation
end GPU vectorize
GPU vectorize over strings I
    ii ← l[I][I]
    σ[I] += E[ii, I]                                               ▷ σ_ii formation - atomic operation
end GPU vectorize
```

Algorithm 4 GPU vectorized projected $\boldsymbol{\sigma_2}$ algorithm. $\mathbf{Q_l}$, $\mathbf{Q_r}$, and $\mathbf{P_r}$ correspond to the left **Q** vector, the right **Q** vector, and the **P** vector in Equation 33. PQ is a floating point scalar, **l** are orbital indices given a string pair I, I', **s** is an array that accumulates the temporary $\boldsymbol{\sigma}$ vector, and **D** and **E** are floating point arrays.

```
GPU vectorize over strings I, I′ differing from I by zero or one occupations
    ij ← l[I][I′]
    D[ij, I] += γ[I][I′]Q_r[I′]                                    ▷ D_ij formation
end GPU vectorize
GPU vectorize over strings I
    ii ← l[I][I]
    D[ii, I] += Q_r[I]                                             ▷ D_ii formation
end GPU vectorize
s += D * h′_kl                                                     ▷ DGEMV
E ← 0.5 * (ij|kl) * D                                              ▷ DGEMM
GPU vectorize over strings I, I′ differing from I by zero or one occupations
    ij ← l[I][I′]
    s[I′] += γ[I][I′]E[ij, I]                                      ▷ σ_ij formation - atomic operation
end GPU vectorize
GPU vectorize over strings I
    ii ← l[I][I]
    s[I′] += E[ii, I]                                              ▷ σ_ii formation - atomic operation
end GPU vectorize
PQ = ±(P_r · s)
σ += PQ * Q_l                                                      ▷ DAXPY
```

Algorithm 5 GPU vectorized projected $\boldsymbol{\sigma_3}$ algorithm. $\mathbf{Q_l}$, $\mathbf{Q_r}$, and $\mathbf{P_r}$ correspond to the left **Q** vector, the right **Q** vector, and the **P** vector in Equation 34. **l** are orbital indices given a string pair α, α', **s** is an array that accumulates the temporary $\boldsymbol{\sigma}$ vector, and **D** and **E** are floating point arrays.

```
GPU vectorize over strings I, I′ differing from I by zero or one occupations
    ij ← l[I][I′]
    D[ij, I] += γ[I][I′]Q_r[I′]                                    ▷ D_ij formation
end GPU vectorize
GPU vectorize over strings I
    ii ← l[I][I]
    D[ii, I] += Q_r[I]                                             ▷ D_ii formation
end GPU vectorize
s += D * P_r                                                       ▷ DGEMV
D ← 0.0
GPU vectorize over strings I, I′ differing from I by zero or one occupations
    ij ← l[I][I′]
    D[ij, I] += γ[I][I′]Q_l[I′]                                    ▷ D_ij formation
end GPU vectorize
GPU vectorize over strings I
    ii ← l[I][I]
    D[ii, I] += Q_l[I]                                             ▷ D_ii formation
end GPU vectorize
E ← (ij|kl) * D                                                    ▷ DGEMM
σ += E * s                                                         ▷ σ formation - DGEMV
```

The GPU kernels in the above algorithms strongly resemble those described in our previous direct CI work.[10] By forming the projected $\boldsymbol{\sigma}$ vector directly, we benefit from a significant reduction in scaling: in RR-FCI, 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. While hardware support for atomics ameliorates some of the computational overhead associated with their use, performance still suffers relative to algorithms that can avoid atomic operations altogether. The RR-FCI projected $\boldsymbol{\sigma}$ algorithm described above is able to restrict atomic operations to the $\boldsymbol{\sigma_1}$ and $\boldsymbol{\sigma_2}$ terms, using DAXPY and DGEMV operations to form the $\boldsymbol{\sigma_3}$ term. Each of the $\mathbf{PQ}^T$ and the $\mathbf{QP}^T$ projected $\boldsymbol{\sigma}$ vector combinations can be formed (for both **P** and **Q** vector optimizations) using these algorithms and interchanging the ordering and identities of the input vectors. In practice, we tile the loop indices to allow for configuration spaces of arbitrary size (subject to memory constraints imposed by the length of the product space vectors). We avoid description of our tiling scheme in Algorithms 3−5 for clarity and instead present this information in more detail in the Supporting Information. Because of the high cost associated with a 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.

3. RESULTS AND DISCUSSION

3.1. Computational Details. The RR-FCI method was implemented in the TeraChem GPU-accelerated electronic structure package using the Compute Unified Device Architecture (CUDA) API[57] and the NVidia CUDA Basic Linear Algebra Subroutines library (cuBLAS).[58] Benchmark calculations were performed using a single core of an Intel Xeon E5-2699 @2.2 GHz 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* to the number of orbitals. Geometries for all molecules used in this work are reported in Cartesian coordinates in the Supporting Information. The cc-pVDZ basis set (in Cartesian form including contaminants) is used in all calculations unless stated otherwise.

3.2. Algorithm Performance. The computational performance of $\boldsymbol{\sigma}$ vector formation depends solely on the number of active electrons and orbitals. Benchmark CASCI (RR-FCI and conventional FCI) calculations were performed on an ethylene dimer using molecular orbitals from a preceding Hartree−Fock calculation. The active spaces range from (10,10) through (30,30)

for RR-FCI and up to (16,16) for conventional FCI. Results are presented in Figure 3 and in Table 2. For comparison, we also

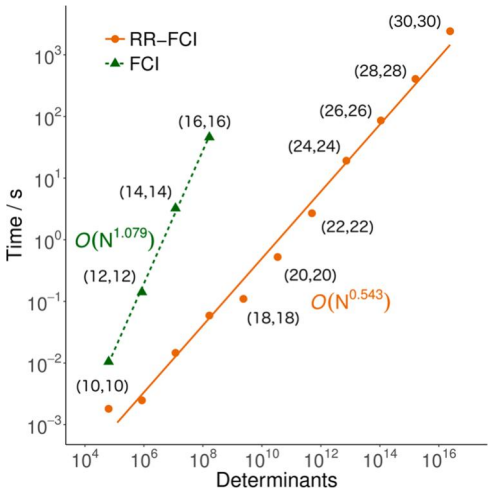


Figure 3. CASCI $\boldsymbol{\sigma}$ vector formation times in seconds for ethylene dimer RR-FCI (orange circles) and conventional FCI (green triangles). Basis set is cc-pVDZ in both cases. RR-FCI active spaces range from (10,10) through (30,30) and FCI active spaces from (10,10) through (16,16). Linear regression was performed to determine the scaling exponent and is shown as a solid (RR-FCI) or a dashed (FCI) line.

present GPU accelerated $\boldsymbol{\sigma}$ formation timings from our conventional GPU-accelerated FCI program.[10]

A linear regression fit of observed computational times shows that the conventional FCI $\boldsymbol{\sigma}$ formation scales linearly with the number of determinants, with a scaling exponent of 1.079. This is in agreement with previously reported results for computational scaling of direct CI $\boldsymbol{\sigma}$ vector,[10] two-particle reduced density matrix,[59] and SA-CASSCF direct Hessian matrix[60] formation. In contrast, RR-FCI projected $\boldsymbol{\sigma}$ vector formation scales with the square root of the number of determinants, with an observed scaling exponent of 0.543. The RR-FCI approach is more computationally efficient than FCI by a factor approaching $\sqrt{N_{\text{det}}}$.

Table 2 provides detailed timings for the most time-consuming portions of RR-FCI calculations with active spaces ranging from (18,18) through (30,30). The 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 considering CPU−GPU memory transfer overhead it is advantageous to precalculate and store these lists when possible. If one desires to use larger configuration spaces than system memory allows, 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. This step is performed using dense matrix operations, and analysis of sparsity in the intermediate matrices and two-electron integrals may offer additional performance advantages. Each of the $\boldsymbol{\sigma}_1$ and $\boldsymbol{\sigma}_3$ formations requires multiple (2) matrix multiplication operations (DGEMMs) while $\boldsymbol{\sigma}_2$ formation requires only a single DGEMM. This, combined with the increased number of GPU kernels ($\mathbf{D}$ and $\boldsymbol{\sigma}$ matrix formations) required, results in $\boldsymbol{\sigma}_2$ being less computationally expensive than $\boldsymbol{\sigma}_1$ and $\boldsymbol{\sigma}_3$. Additional details related to this may be found in the Supporting Information.

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 from CPU−GPU memory transfer, as their data structures reside entirely in GPU memory and have dimensions smaller than the tile size, but at configuration spaces of (24,24) and larger, the high cost of CPU−GPU memory transfer becomes apparent, comprising nearly 30% of the overall cost of the projected $\boldsymbol{\sigma}$ formation for the (30,30) active space. The GPU kernels corresponding to $\mathbf{D}_{ij}$ and $\boldsymbol{\sigma}_{ij}$ formation are the next most costly operations, followed by the remaining linear algebra operations (DGEMV, DDOT, DAXPY). Finally, the largest

Table 2. Algorithm Timings (s) for RR-CASCI with Active Spaces Ranging from (18,18) through (30,30).[a]

	Active Space						
Function	(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
Detailed timing breakdown for σ formation (see Algorithms 3−5)							
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
$\boldsymbol{\sigma}_{ij}$	0.011	0.047	0.237	0.972	4.753	19.239	82.188
$\boldsymbol{\sigma}_{ii}$	<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

[a] $\boldsymbol{\sigma}$ timings correspond to a combined total of 6 function calls, 2 each of $\boldsymbol{\sigma}_1$, $\boldsymbol{\sigma}_2$, and $\boldsymbol{\sigma}_3$. Coupling coefficient and string label lists are formed only once per calculation, while metric and $\boldsymbol{\sigma}$ formation are required once per microiteration. 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.

configuration space required an extremely small tile size (1024), and this is reflected in both the list and the σ formation times accordingly.

3.3. 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 CASCI and RR-CASCI with full π valence active spaces, (10,10) and (14,14) for naphthalene and anthracene, respectively. Energy differences between the rank-reduced and exact (FCI) approaches are shown in Figures 4 and 5.

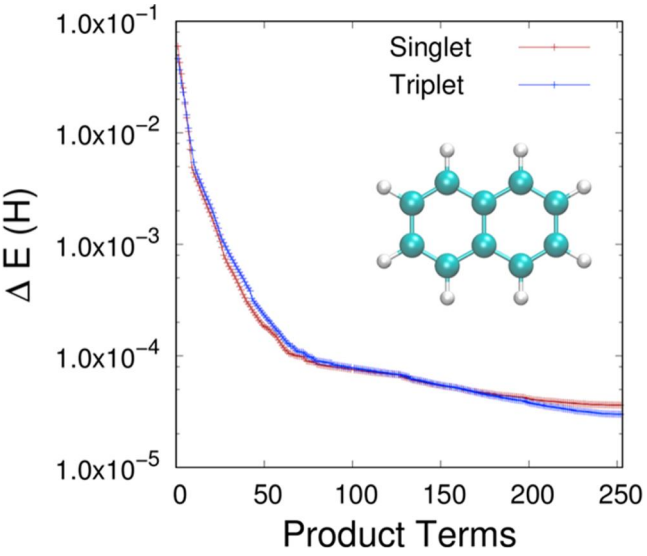


Figure 4. Energy error (relative to conventional CASCI, in Hartree) for RR-FCI with increasing number of product terms in napthalene. Lowest singlet state (brown) and lowest triplet state (blue) are shown using CAS(10,10)-CI with cc-pVDZ basis set. The singlet optimized structure is depicted with carbon and hydrogen atoms shown in teal and white, respectively.

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. Geometries for the singlet and triplet systems described in the following study were optimized using UB3LYP/6-31G(d) and are the same as those reported by Hachmann et al.[61] For the reader's convenience, they are available in the Supporting Information in Cartesian coordinates.

The RR-FCI energy rapidly converges to within mH accuracy of the FCI energy after 30 product terms, representing inclusion of the 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. Including 252 product states corresponds to the largest possible 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 **P** and **Q** vectors may contain redundant information. This occurs because we do not reoptimize **P** and **Q** vectors added in previous iterations. This impedes convergence to the exact FCI

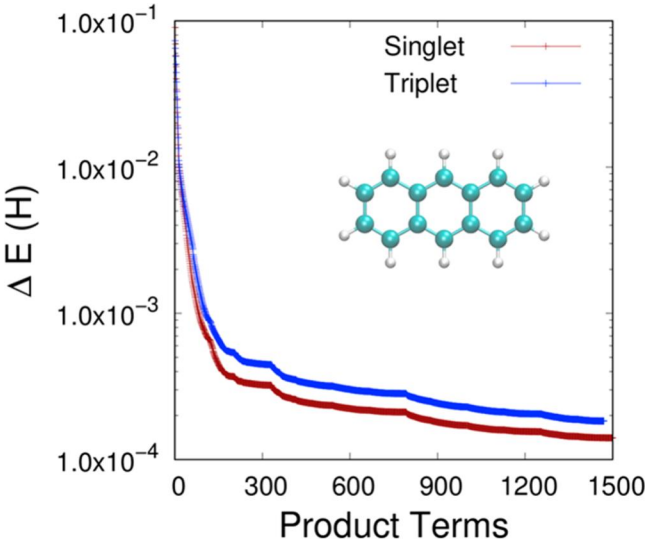


Figure 5. Energy error (relative to conventional CASCI, in Hartree) for RR-FCI with increasing number of product terms in anthracene. Lowest singlet state (brown) and lowest triplet state (blue) are shown using CAS(14,14)-CI with cc-pVDZ basis set. The singlet optimized structure is depicted with carbon and hydrogen atoms shown in teal and white, respectively.

solution when the number of product terms included equals the FCI vector rank.

The (14,14) CAS space of anthracene approaches the upper size limit of routinely performed CI calculations. 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. Irregularities in the convergence at ~350 and ~800 product terms are likely artifacts of our guess vector procedure.

3.4. Relative Energy Convergence: Acene Singlet–Triplet Gaps. 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. Singlet–triplet energy gaps for each system are depicted in Figure 6. While absolute energy convergence to within mH accuracy requires ~30 product terms for naphthalene and ~100 product terms for anthracene, convergence of a relative

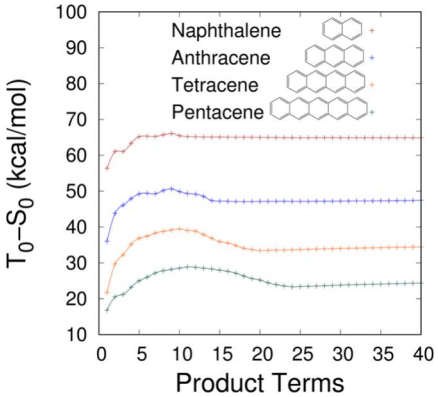


Figure 6. Singlet–triplet energy gap (in units of kcal/mol) for the acene series naphthalene through pentacene using RR-CASCI with the cc-pVDZ basis set. In each case, the active space comprises all π electrons in all valence π orbitals.

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 converge, respectively. A comparison of our RR-CASCI energies with the local DMRG results of Hachmann is given in Table 3, along with estimates of the experimental singlet−triplet gaps for each system.[62−65]

Table 3. Singlet−triplet Energy Splittings (kcal/mol) Calculated with RR-CASCI Using Full π Valence Active Spaces Compared with Local Density Matrix Renormalization Group (DMRG) Results As Described by Hachmann[61,a]

	Singlet−Triplet Splitting (kcal/mol)		
Acene	RR-CASCI	DMRG	Expt.
Naphthalene	65.1 (N_r = 12)	61.0	61.0
Anthracene	47.2 (N_r = 15)	44.0	43.1
Tetracene	33.4 (N_r = 20)	31.9	29.3
Pentacene	23.3 (N_r = 24)	23.4	19.8

[a]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, respectively] and a cc-pVDZ basis set. Experimental energy gaps for naphthalene, anthracene, tetracene, and pentacene are estimates. Number of product terms for RR-CASCI results is reported in parentheses.

Even with the disparity in CAS spaces between the RR-CI and DMRG results reported above, the agreement between the two methods is surprisingly good. In each case, RR-CI lies within ~4 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%. These results are encouraging and motivate us to pursue refinement and continued development of the RR-FCI approach.

3.5. Nitrogen Bond Dissociation. FCI is often used as a benchmarking tool for evaluating lower-cost approximations. A common benchmark is the computation of a potential energy curve for dissociation of a multiple bond, which can be particularly difficult since both static and dynamic electron correlation often contribute. We applied the RR-FCI and FCI methods to molecular nitrogen, varying the bond length from 0.8 to 1.8 Å. The RR-FCI and FCI results are compared in Figure 7. All 10 valence electrons are correlated, corresponding to a (10,28) configuration space. Conventional FCI calculations were performed using the Molpro electronic structure program.[66,67]

While the configuration space corresponds to 9.6 billion determinants (without symmetry), excellent agreement between RR-FCI and FCI is observed using only 100 product terms. Energy differences $[\Delta E = (E_{RR\text{-}FCI}) - (E_{FCI})]$ between the two methods are provided in the inset of Figure 7. It is interesting to note that RR-FCI 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. This demonstrates that our ansatz treats static and dynamic correlation in an unbalanced fashion. We are currently investigating this behavior with the intention of improving the treatment of static correlation. The low cost-to-accuracy ratio of RR-FCI makes it an ideal tool for use in performing benchmark type calculations for evaluation of lower-cost alternatives.

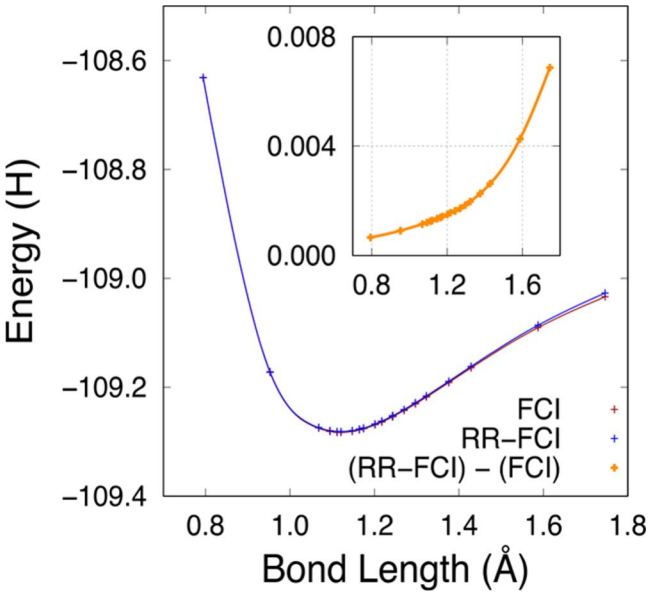


Figure 7. Molecular N_2 dissociation curve calculated with RR-FCI and conventional FCI using the cc-pVDZ basis set. RR-FCI calculations include 100 product terms each. Energy differences (in Hartree) between RR-FCI and FCI are given in the inset. Units for the inset are the same as the main plot. Time of a particular Davidson microiteration for N_2 (R_{NN} = 0.7938 Å) is 0.72 s, and optimization of the first and second product terms required 1127.47 and 605.28 s, respectively.

4. CONCLUSIONS

We have presented a low-cost FCI alternative for both singlet and triplet ground states with computational cost that scales as $\sqrt{N_{det}}$, allowing routine calculation of CASCI wave functions for large active spaces. GPU acceleration of rate-limiting components of the algorithm, including projected $\boldsymbol{\sigma}$ and one-particle coupling coefficient formation, expands the size of accessible configuration spaces to $O(10^{16})$ 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 for both absolute and relative energies compared to conventional FCI. Finally, we investigated dissociation of molecular nitrogen and found that our results compare favorably against benchmark FCI studies. Given the success of RR-FCI in describing the ground state singlet and triplet wave functions, we are currently working to extend our approach to allow for excited state calculations and the evaluation of molecular properties.

■ AUTHOR INFORMATION

Corresponding Author

*E-mail: toddjmartinez@gmail.com

ORCID

Stefan Seritan: 0000-0002-8808-6886
Benjamin G. Levine: 0000-0002-0356-0738
Henrik Koch: 0000-0002-8367-8727
Todd J. Martínez: 0000-0002-4798-8947

Notes

The authors declare the following competing financial interest(s): T.J.M. is a cofounder of PetaChem, LLC.

■ ACKNOWLEDGMENTS

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Scientific Discovery through Advanced Computing (SciDAC) program. B.G.L. acknowledges support from the National Science Foundation (CHE-1565634). Computational resources were provided through the Extreme Science and Engineering Discovery Environment[68] (Grant No. ACI-1548562, allocation TG-CHE140101) and the XStream computational resource supported by the NSF MRI program (Grant No. ACI-1429830). S.S. acknowledges support from an NSF Graduate Fellowship. H.K. acknowledges support from the Norwegian Research Council through FRINATEK project no. 263110/F20. T.J.M. is a cofounder of PetaChem, LLC.

■ REFERENCES

(1) Roos, B. O. The Complete Active Space SCF Method in a Fock-Matrix-Based Super-CI Formulation. *Int. J. Quantum Chem.* **1980**, *18*, 175−189.

(2) Siegbahn, P.; Heiberg, A.; Roos, B.; Levy, B. A Comparison of the Super-CI and the Newton-Raphson Scheme in the Complete Active Space SCF Method. *Phys. Scr.* **1980**, *21*, 323−327.

(3) Roos, B.; Taylor, P. R.; Siegbahn, P. E. M. A Complete Active Space SCF Method (CASSCF) Using a Density Matrix Formulated Super-CI Approach. *Chem. Phys.* **1980**, *48*, 157−173.

(4) Ruedenberg, K.; Schmidt, M. W.; Gilbert, M. M.; Elbert, S. T. Are Atoms Intrinsic to Molecular Electronic Wavefunctions? I. The FORS Model. *Chem. Phys.* **1982**, *71*, 41−49.

(5) Roos, B. A New Method For Large-Scale CI Calculations. *Chem. Phys. Lett.* **1972**, *15*, 153−159.

(6) Davidson, E. R. The Iterative Calculation of a Few of the Lowest Eigenvalues and Corresponding Eigenvectors of Large Real-Symmetric Matrices. *J. Comput. Phys.* **1975**, *17*, 87−94.

(7) Siegbahn, P. E. M. A New Direct CI Method for Large CI Expansions in a Small Orbital Space. *Chem. Phys. Lett.* **1984**, *109*, 417−423.

(8) Knowles, P. J.; Handy, N. C. A New Determinant Based Full Configuration Interaction Method. *Chem. Phys. Lett.* **1984**, *111*, 315−321.

(9) Harrison, R.; Zarrabian, S. An Efficient Implementation of the Full-CI Method Using an (n-2)-Electron Projection Space. *Chem. Phys. Lett.* **1989**, *158*, 393−398.

(10) Fales, B. S.; Levine, B. G. Nanoscale Multireference Quantum Chemistry: Full Configuration Interaction on Graphical Processing Units. *J. Chem. Theory Comput.* **2015**, *11*, 4708−4716.

(11) Handy, N. C. Multi-Root Configuration Interaction Calculations. *Chem. Phys. Lett.* **1980**, *74*, 280−283.

(12) Olsen, J.; Roos, B.; Jørgensen, P.; Jensen, H. J. A. Determinant Based Configuration Interaction Algorithms for Complete and Restricted Configuration Interaction Spaces. *J. Chem. Phys.* **1988**, *89*, 2185.

(13) Olsen, J.; Jørgensen, P.; Simons, J. Passing the One-Billion Limit in Full Configuration Interaction (FCI) Calculations. *Chem. Phys. Lett.* **1990**, *169*, 463−472.

(14) Gan, Z.; Harrison, R. J. In *Calibrating Quantum Chemistry: A Multi-Teraflop, Parallel-Vector, Full-Configuration Interaction Program for the Cray-X1*, Supercomputing, 2005. Proceedings of the ACM/IEEE SC 2005 Conference, 2005; pp 22−22.

(15) Friesner, R. A.; Murphy, R. B.; Beachy, M. D.; Ringnalda, M. N.; Pollard, W. T.; Dunietz, B. D.; Cao, Y. Correlated Ab Initio Electronic Structure Calculations for Large Molecules. *J. Phys. Chem. A* **1999**, *103*, 1913−1928.

(16) Martinez, T. J.; Mehta, A.; Carter, E. A. Pseudospectral Full Configuration-Interaction. *J. Chem. Phys.* **1992**, *97*, 1876−1880.

(17) Beebe, N. H. F.; Linderberg, J. Simplifications in the Generation and Transformation of Two-Electron Integrals in Molecular Calculations. *Int. J. Quantum Chem.* **1977**, *12*, 683−705.

(18) Røeggen, I.; Wisloff-Nilssen, E. On the Beebe-Linderberg Two-Electron Integral Approximation. *Chem. Phys. Lett.* **1986**, *132*, 154−160.

(19) Koch, H.; Sanchez de Meras, A. S.; Pedersen, T. B. Reduced Scaling in Electronic Structure Calculations Using Cholesky Decompositions. *J. Chem. Phys.* **2003**, *118*, 9481−9484.

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

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

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

(23) Hohenstein, E. G.; Parrish, R. M.; Martinez, T. J. Tensor Hypercontraction Density Fitting. I. Quartic Scaling Second- and Third-Order Møller-Plesset Perturbation Theory. *J. Chem. Phys.* **2012**, *137*, 044103.

(24) Parrish, R. M.; Hohenstein, E. G.; Martinez, T. J.; Sherrill, C. D. Tensor Hypercontraction. II. Least-Squares Renormalization. *J. Chem. Phys.* **2012**, *137*, 224106.

(25) Hohenstein, E. G.; Kokkila, S. I. L.; Parrish, R. M.; Martinez, T. J. Tensor Hypercontraction Equation-of-Motion Second-Order Approximate Coupled Cluster: Electronic Excitation Energies in $O(\hat{N}4)$ Time. *J. Phys. Chem. B* **2013**, *117*, 12972−12978.

(26) Parrish, R. M.; Hohenstein, E. G.; Schunck, N. F.; Sherrill, C. D.; Martinez, T. J. 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.* **2013**, *111*, 132505.

(27) Song, C.; Martinez, T. J. Atomic Orbital-Based SOS-MP2 with Tensor Hypercontraction. I. GPU Based Tensor Construction and Exploiting Sparsity. *J. Chem. Phys.* **2016**, *144*, 174111.

(28) Song, C.; Martinez, T. J. Atomic Orbital-Based SOS-MP2 with Tensor Hypercontraction. II. Local Tensor Hypercontraction. *J. Chem. Phys.* **2017**, *146*, 034104.

(29) Kokkila Schumacher, S. I. L.; Hohenstein, E. G.; Parrish, R. M.; Wang, L.-P.; Martinez, T. J. Tensor Hypercontraction Second-Order Møller-Plesset Perturbation Theory: Grid Optimization and Reaction Energies. *J. Chem. Theory Comput.* **2015**, *11*, 3042−3052.

(30) Huron, B.; Malrieu, J. P.; Rancurel, P. Iterative Perturbation Calculations of Ground and Excited State Energies From Multiconfigurational Zeroth-Order Wavefunctions. *J. Chem. Phys.* **1973**, *58*, 5745−5759.

(31) Buenker, R. J.; Peyerimhoff, S. D. Individualized Configuration Selection in CI Calculations with Subsequent Energy Extrapolation. *Theoret. Chim. Acta* **1974**, *35*, 33−58.

(32) Tubman, N. M.; Lee, J.; Takeshita, T. Y.; Head-Gordon, M.; Whaley, K. B. A Deterministic Alternative to the Full Configuration Interaction Quantum Monte Carlo Method. *J. Chem. Phys.* **2016**, *145*, 044112.

(33) Schriber, J. B.; Evangelista, F. A. Communication: An Adaptive Configuration Interaction Approach for Strongly Correlated Electrons with Tunable Accuracy. *J. Chem. Phys.* **2016**, *144*, 161106.

(34) Holmes, A. A.; Tubman, N. M.; Umrigar, C. J. Heat-Bath Configuration Interaction: An Efficient Selected Configuration

Interaction Algorithm Inspired by Heat-Bath Sampling. *J. Chem. Theory Comput.* **2016**, *12*, 3674−3680.
(35) Booth, G. H.; Thom, A. J. W.; Alavi, A. Fermion Monte Carlo Without Fixed Nodes: A Game of Life, Death, and Annihilation in Slater Determinant Space. *J. Chem. Phys.* **2009**, *131*, 054106.
(36) Ten-no, S. Stochastic Determination of Effective Hamiltonian for the Full Configuration Interaction Solution of Quasi-Degenerate Electronic States. *J. Chem. Phys.* **2013**, *138*, 164126.
(37) Shepherd, J. J.; Scuseria, G. E.; Spencer, J. S. Sign Problem in Full Configuration Interaction Quantum Monte Carlo: Linear and Sublinear Representation Regimes for the Exact Wave Function. *Phys. Rev. B: Condens. Matter Mater. Phys.* **2014**, *90*, 155130.
(38) White, S. R.; Martin, R. L. Ab Initio Quantum Chemistry Using the Density Matrix Renormalization Group. *J. Chem. Phys.* **1999**, *110*, 4127−4130.
(39) Chan, G. K.-L.; Sharma, S. The Density Matrix Renormalization Group in Quantum Chemistry. *Annu. Rev. Phys. Chem.* **2011**, *62*, 465−481.
(40) Nakatani, N.; Chan, G. K.-L. Efficient Tree Tensor Network States (TTNS) for Quantum Chemistry: Generalizations of the Density Matrix Renormalization Group Algorithm. *J. Chem. Phys.* **2013**, *138*, 134113.
(41) Knowles, P. J. Very Large Full Configuration-Interaction Calculations. *Chem. Phys. Lett.* **1989**, *155*, 513−517.
(42) Knowles, P. J.; Handy, N. C. Unlimited Full Configuration Interaction Calculations. *J. Chem. Phys.* **1989**, *91*, 2396−2398.
(43) Mitrushenkov, A. O. Passing the Several Billions Limit in FCI Calculations on a Mini-Computer. *Chem. Phys. Lett.* **1994**, *217*, 559−565.
(44) Mitrushenkov, A. O.; Dmitriev, Y. Y. 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.* **1995**, *235*, 410−413.
(45) Rolik, Z.; Szabados, A.; Surjan, P. A Sparse Matrix Based Full-Configuration Interaction Algorithm. *J. Chem. Phys.* **2008**, *128*, 144101.
(46) Olsen, J.; Malmqvist, P.-A.; Roos, B. o.; Lindh, R.; Widmark, P.-O. A Non-Linear Approach to Configuration Interaction. The Low-Rank CI Method (LR CI). *Chem. Phys. Lett.* **1987**, *133*, 91−101.
(47) Lindh, R.; Olsen, J.; Roos, B. o. Low-Rank Configuration Interaction With Orbital Optimization - the LR SCF Approach. *Chem. Phys. Lett.* **1988**, *148*, 276−280.
(48) Koch, H.; Dalgaard, E. A Variational Matrix Decomposition Applied to Full Configuration-Interaction Calculations. *Chem. Phys. Lett.* **1992**, *198*, 51−58.
(49) Taylor, P. R. Lossless Compression of Wavefunction Information Using Matrix Factorization: A "gzip" for Quantum Chemistry. *J. Chem. Phys.* **2013**, *139*, 074113.
(50) Duch, W. *GRMS or Graphical Representation of Model Spaces*; Springer Verlag, 1986; Vol. *1*.
(51) Grein, F.; Chang, T. C. Multiconfiguration Wavefunctions Obtained by Application of the Generalized Brillouin Theorem. *Chem. Phys. Lett.* **1971**, *12*, 44−48.
(52) Grein, F.; Banerjee, A. A Multiconfiguration Method for Excited States of Atoms and Molecules. *Int. J. Quantum Chem.* **1975**, *9*, 147−154.
(53) Parrish, R. M.; Hohenstein, E. G.; Martinez, T. J. 'Balancing" the Block Davidson-Liu Algorithm. *J. Chem. Theory Comput.* **2016**, *12*, 3003−3007.
(54) Furche, F.; Krull, B. T.; Nguyen, B. D.; Kwon, J. Accelerating Molecular Property Calculations with Nonorthonormal Krylov Space Methods. *J. Chem. Phys.* **2016**, *144*, 174105.
(55) Bendazzoli, G. L.; Evangelisti, S. A Vector and Parallel Full Configuration-Interaction Algorithm. *J. Chem. Phys.* **1993**, *98*, 3141−3150.
(56) *Developer Reference for Intel Math Kernel Library 2017 − C*, 2017. https://software.intel.com/en-us/download/developer-reference-for-intel-math-kernel-library-2017-c (accessed June 2018).
(57) *CUDA C Programming Guide*, 2017. https://developer.download.nvidia.com/compute/DevZone/docs/html/C/doc/CUDA_C_Programming_Guide.pdf (accessed June 2018).
(58) *cuBLAS Library*, 2017. https://developer.download.nvidia.com/compute/DevZone/docs/html/CUDALibraries/doc/CUBLAS_Library.pdf (accessed June 2018).
(59) Fales, B. S.; Shu, Y.; Levine, B. G.; Hohenstein, E. G. Complete Active Space Configuration Interaction From State-Averaged Configuration Interaction Singles Natural Orbitals: Analytic First Derivatives and Derivative Coupling Vectors. *J. Chem. Phys.* **2017**, *147*, 094104.
(60) Snyder, J. J. W.; Fales, B. S.; Hohenstein, E. G.; Levine, B. G.; Martinez, T. J. A Direct-Compatible Formulation of the Coupled Perturbed Complete Active Space Self-Consistent Field Equations on Graphical Processing Units. *J. Chem. Phys.* **2017**, *146*, 174113.
(61) Hachmann, J.; Dorando, J. J.; Avilés, M.; Chan, G. K.-L. The Radical Character of the Acenes: A Density Matrix Renormalization Group Study. *J. Chem. Phys.* **2007**, *127*, 134309.
(62) Birks, J. B. *Photophysics of Aromatic Molecules*; Wiley, London, 1970.
(63) Schiedt, J.; Weinkauf, R. Photodetachment Photoelectron Spectroscopy of Mass Selected Anions: Anthracene and the Anthracene-H$_2$O Cluster. *Chem. Phys. Lett.* **1997**, *266*, 201.
(64) Sabbatini, N.; Indelli, M. T.; Gandolfi, M. T.; Balzani, V. Quenching of Singlet and Triplet Excited States of Aromatic Molecules by Europium Ions. *J. Phys. Chem.* **1982**, *86*, 3585.
(65) Burgos, J.; Pope, M.; Swenberg, C. E.; Alfano, R. R. Heterofission in Pentacene-Doped Tetracene Single Crystals. Phys. *Phys. Status Solidi B* **1977**, *83*, 249.
(66) Werner, H. J.; Knowles, P. J.; Knizia, G.; Manby, F. R.; Schutz, M. *MOLPRO*, version 2015.1, a package of ab initio programs, 2015.
(67) Knowles, P. J.; Handy, N. C. A Determinant Based Full Configuration Interaction Program. *Comput. Phys. Commun.* **1989**, *54*, 75−83.
(68) Towns, J.; Cockerill, T.; Dahan, M.; Foster, I.; Gaither, K.; Grimshaw, A.; Hazlewood, V.; Lathrop, S.; Lifka, D.; Peterson, G. D.; Roskies, R.; Scott, J. R.; Wilkens-Diehr, N. XSEDE: Accelerating Scientific Discovery. *Comput. Sci. Eng.* **2014**, *16*, 62−74.