An efficient deterministic perturbation theory for selected configuration interaction methods

Norm M. Tubman, Daniel S. Levine, Diptarka Hait, Martin Head-Gordon, K. Birgitta Whaley

University of California, Berkeley, California 94720, USA and Chemical Sciences Division, Lawrence Berkeley National Laboratory Berkeley, California 94720, USA

(Dated: August 8, 2018)

The interplay between advances in stochastic and deterministic algorithms has recently led to development of interesting new selected configuration interaction (SCI) methods for solving the many-body Schrödinger equation. The performance of these SCI methods can be greatly improved with a second order perturbation theory (PT2) correction, for which stochastic and hybrid-stochastic methods have recently been proposed as new tools to perform such calculations. In this work, we present a highly efficient, fully deterministic PT2 algorithm for SCI methods and demonstrate that our approach is orders of magnitude faster than recent proposals for stochastic SCI+PT2. We also show that it is important to have a compact reference SCI wave function, in order to obtain optimal SCI+PT2 energies. This indicates that it advantageous to use accurate search algorithms such as ‘ASCI search’ rather than more approximate approaches. Our deterministic PT2 algorithm is based on sorting techniques that have been developed for modern computing architectures and is inherently straightforward to use on parallel computing architectures. Related architectures such as GPU implementations can be also used to further increase the efficiency. Overall, we demonstrate that the algorithms presented in this work allow for efficient evaluation of trillions of PT2 contributions with modest computing resources.

I. INTRODUCTION

There has been a recent renaissance in the field of selected configuration interaction (SCI) algorithms [1–9] for tackling strongly correlated systems where traditional single reference theories like coupled cluster methods are likely to break down. Recently, SCI has been applied to systems that are larger than what is possible with conventional FCI algorithms [10–14]. This include simulations of transition metal systems [1, 4], a solver for dynamical mean field theory [15], cluster decompositions [16], and excited states [1, 7, 18]. One approach to SCI is called adaptive sampling configuration interaction (ASCI) [1, 9], which has been developed with fast and efficient algorithms to take advantage of modern computing architectures.

SCI methods typically include a second order perturbation theory (PT2) correction to account for the relatively small component of the correlation energy missed by the variational SCI wave function [19–21]. Indeed, it was also observed that the accuracy of ASCI can be increased dramatically by incorporation of a PT2 correction [1]. In very recent years, stochastic perturbative corrections have been developed for various SCI methods [8, 22], as well as for density matrix renormalization group (DMRG) [23] and for FCIQMC [24]. The apparent lack of fast deterministic PT2 algorithms in the literature has perhaps encouraged development of these recent stochastic approaches. Two of the newer stochastic PT2 developments for SCI are the semi-stochastic perturbation theory [22] and hybrid deterministic-stochastic [8] algorithms.

It is an interesting and still open question as to whether or not stochastic PT2 algorithms have inherent advantages over their deterministic counterparts. To address this question, in this work we introduce a fast deterministic approach to generate exact PT2 energies. We also demonstrate via comparison of run times, that this new deterministic algorithm is significantly faster than the recently proposed stochastic algorithms for SCI.

II. OVERVIEW OF ASCI

The main idea behind SCI methods (see Figure 1 for a flowchart of the ASCI approach) is to diagonalize the Hamiltonian within a Hilbert space in which only the most important degrees of freedom are identified and retained. SCI algorithms therefore attempt to generate the top contributing determinants to the full CI (FCI) wave function, in contrast to traditional FCI algorithms, which include all possible determinants. They also stand in contrast to truncated CI approaches, like CISD, CISDT, etc, which typically consider restricted excitations from the reference up to some given level of excitation [13, 28]. Truncation based upon excitation level leads to large size-consistency errors [28], that make such models unsuitable for most chemical applications, while SCI approaches like ASCI simply aim for a sufficiently close approximation to FCI that this issue is irrelevant.

Full details of the ASCI method are presented in previous works [1, 9], and we therefore provide only a brief overview here. In the ASCI method, a wave function \( \psi_k \) is improved upon over the course of several iterations. In each iteration \( k \), the current wave function and associated Hamiltonian matrix elements are used together to search the Hilbert space for new determinants. The search part of the algorithm requires two rules: a selection criterion to determine what part of Hilbert space to search (pruning), and a ranking criterion over the determinants that make it through the pruning, to determine the best determinants to include in the updated wave function \( \psi_{k+1} \). Two different pruning techniques have been discussed in the original ASCI and heatbath CI (HCI) papers: these we refer to as coefficient driven search [1] and integral
This equation can be reinterpreted as an iterative recipe for $j$ and $k$, where $H$ are determinants with coefficients $C_i$. We consider an expansion of an eigenstate in a basis of Slater determinants that constitute eigenstates of the Schrödinger equation. From a consistency relationship among the coefficients of the relative compactness of the variational wave functions. The algorithm however, is quite different and more approximate than the one used in ASCI (as will be shown later, this reduces the computational components are the search, diagonalization, and post-processing steps. The growth steps are done in the first set of iterations of ASCI to bring the variational wave function from the input wave function, and the output coefficients $A_i$ are estimates of coefficients of an improved wave function that is closer to an eigenstate. The $A_i$ coefficients are also related to a first-order perturbation estimate for CI coefficients in many body Epstein-Nesbet perturbation theory [29].

Since the goal of a SCI method is to include the most important weight determinants in the expansion, ASCI search uses a slightly modified version of Eqn. 2 to define a ranking, where $|A_i|$ is the rank value of the $i$th determinant. The novel algorithm used in ASCI search to calculate the ranking equation is both fast and accurate. It is described in detail in reference [9]. After the ranking values are calculated, the top $N$ determinants are then chosen for Hamiltonian construction and subsequent diagonalization to obtain $\psi_{k+1}$ and $E_{k+1}$. In practice, this iterative approach is successful in generating all the top contributions to the wave function, which is essential for obtaining highly accurate energies. The effect of the neglected determinants can be approximated via perturbation theory [1]. Additional considerations for performing other aspects of selected CI, such as Hamiltonian building, have also been described in recent work [9].

The PT2 energy correction $E_{PT2}$ is generally calculated with Epstein-Nesbet perturbation theory [30, 31], in which it is given by

$$E_{PT2} = \sum_i \frac{|\langle \psi | H | D_i \rangle|^2}{E_{ASCI} - h_i}.$$

This combination of ASCI search with Epstein-Nesbet PT2 is able to treat strongly correlated systems, including the Cr$_2$ dimer in the SVP basis, to chemical accuracy [1]. To the best of our knowledge, ASCI (as well as other SCI methods) and DMRG approaches [32, 33] are the only methods that have been able to treat Cr$_2$ to chemical accuracy in the SVP basis. We note that it is possible that auxiliary field quantum Monte Carlo will soon also reach this benchmark, using the new and improved wave functions that are currently under investigation [34].

III. DETERMINISTIC PERTURBATION THEORY WITH ASCI

1. The Fundamental Algorithm

Understanding modern computer algorithms and architectures is important in designing new simulation tools, since op-
The constraint approach is quite general in terms of the possible constraints that can be used. The implementation we present here uses what we call the triplet constraints. These are indexed by a set of three numbers that specify the three highest occupied alpha spin-orbitals in the Slater determinant. Many different sorting algorithms have been benchmarked with bitstrings generated within an ASCI simulation in recent work [9].
specify a class of determinants (PT2 contributions), and thus the triplet constraints are non-overlapping and complete. For our purposes, we order the bitstrings such that the alpha bitstrings are the most significant bits (orbitals). Thus in any system in which there are at least three or more alpha electrons, the triplet values will refer to the occupation of the top three alpha electrons.

This approach is summarized in Algorithm 2. It works by first constructing a loop over the triplet constraints. For each triplet constraint, we then consider each determinant in the variational ASCI wave function and generate all possible PT2 contributions (i.e., single and double excitations) from this that satisfy the given triplet constraint. These contributions are stored in an array and then sorted after all contributions have been generated. This aggregates all the terms that make up all PT2 contributions with the given constraint. A single pass over the sorted list is then sufficient for calculating the PT2 energy contribution with Eqn. 3.

The motivation for the triplet constraints comes from specific consideration of the simulations we are targeting. For systems with around 50 electrons, 200 orbitals, and up to 10 million determinants in the SCI reference wave function, we find that in most cases the largest number of contributions per constraint is 1 million to 100 million, which can easily fit into memory on most modern machines. Larger systems are also likely to fit in memory. In the instances where this is not the case, it is easy to modify the constraints, e.g., by specifying the top four orbitals (quadruplet constraint) instead of the top three. This flexibility leads to easy generation of memory and speed efficient approaches.

4. Parallelization

The algorithm that we have outlined above is straightforward to parallelize. While the triplet constraint described above shows a large variation in the number of PT2 contributions for each constraint, the load balancing issue can nevertheless be resolved by creating a work-list with an accurate estimate of the number of contributions in each constraint and distributing the work accordingly. Due to the flexibility of our constraints, specifically, the fact that each constraint can be broken up into subconstraints, load balancing becomes a matter of implementation and presents no inherent algorithmic difficulty. Furthermore, for a given constraint, the computational work can be parallelized, as most of the computational time is focused on generation of the PT2 contributions and sorting of these contributions. Thus openmp, MPI, and openmp MPI hybrid approaches can be used straightforwardly. To be explicit, our ASCI implementation on 20 cores is perfectly parallel, and can be directly compared to the 20 core SHCI simulations included in the results section. We present single core timings in the results section to make it easier to compare to other works for which benchmarks are presented for different numbers of processors.

IV. RESULTS

In this section, we present results for implementation of the triplet constraint evaluation of PT2 corrections to ASCI. We use a cutoff which ignores PT2 contributions (given by the numerator part $|c_i H_{ij}|$, where $i$ is in the variational space and $j$ is the perturbing determinant) that are less than $10^{-8}$ [4]. This is not strictly necessary, since the deterministic approach can handle all the contributions efficiently with only a reasonable amount of extra computational effort, at least for the systems tested in this work. To know whether to include a term we currently calculate $|c_i H_{ij}|$ for all contributions, however for terms that are neglected we save time by not calculating the denominators or sorting such terms. When one is interested in developing a lower accuracy deterministic PT2 approach, integral driven versions of this algorithm can be created and used to more efficiently neglect terms without having to calculate the numerator part explicitly [4]. We find the errors on the total energy that result from these neglected contributions are generally less than $10^{-7}$ Hartrees for the systems considered in this work. Additionally, neglecting these terms allows us to make explicit comparisons to previously published HCI results.

In Tables I, II, and III we demonstrate the performance of our approach using canonical Hartree-Fock orbitals, making comparison to the stochastic PT2 method used with HCI [22] (which referred to as SHCI). In Table III we also make comparison to the deterministic-stochastic hybrid PT2 implemented with CIPSI [8]. To compare deterministic results against stochastic results, we calculate the amount of time it takes to achieve an error bar of 0.1 mHa accuracy with a 95% likelihood. This would typically be called a $2\sigma$ error bar in the Monte Carlo literature. We also consider what error bars would be needed to achieve the same accuracy in an energy difference (since subtracting two quantities with the same error results in an error $\sqrt{2}$ larger than the original error). Since error bars asymptotically decrease with the amount of sampling as $1/\sqrt{N_{\text{samples}}}$, energy differences having the same error bars as the absolute energies require another factor of two greater computation time. Since these algorithms are all inherently parallelizable, we calculate the single core times to compare against previously published results that use slightly different architectures. We expect these comparison to be informative as all calculations are done on similar Intel processors, the details of which are described in the caption of the tables.

A. Molecules with Hartree-Fock orbitals

We have analyzed C$_2$, N$_2$ and F$_2$, at internuclear separations of 1.24253 Å, 1.0977 Å, and 1.4119 Å respectively, for testing purposes. Table I present timings for the triplet constraint approach compared with timings for the stochastic approaches. Our results indicate that the deterministic con-
strained PT2 approach is in general two orders of magnitude faster than the stochastic PT2 approach employed in SHCI with the cc-pVDZ basis. For cc-pVTZ and cc-pVQZ bases, the improvement over stochastic SHCI PT2 is between one and two orders of magnitude, as can be seen from Tables II and III. In Table III we also make comparison to deterministic-stochastic hybrid CIPSI in the deterministic limit (i.e., in the limit that the stochastic part of the algorithm is turned off) [8].

Our results show that ASCI generates more compact variational wave functions than HCI, which is expected (see also [9]). By compact we mean to make a comparison of the variational energy that can be attained for a given number of determinants. Although ASCI and HCI search the same variational space, ASCI finds a significantly better set of determinants. The only system in Table III that shows an exception to this is the case of F\(_2\) in a cc-pVQZ basis, without natural orbital rotations. While the reason for the lower HCI energy in this case is not entirely clear, it is however evident that CIPSI is also not able to obtain the HCI variational energy, even though it also uses a search much improved over HCI. The variational energies for ASCI and CIPSI are comparable when the ASCI wave function has \(2 \times 10^6\) determinants and the CIPSI wave function has \(4 \times 10^6\).

As can be seen in Tables I, II, and III the ASCI wave functions are sometimes close to 40% more compact than HCI wave functions with the same accuracy of variational energy. HCI appears unable to generate compact wave functions (as can be seen from variational energies in Tables I, II and III), reflecting more significant residual systematic errors. With the introduction of the fast ASCI search algorithm [9], it is now feasible to do larger, more accurate searches to generate better reference wave functions, which not only produces better variational energies but also improved PT2 predictions.

B. Molecules with Natural Orbital Rotations

We now demonstrate the effect of adding orbital rotations for the F\(_2\) molecule in Table IV. We find that there is some disagreement between previously published results for F\(_2\)/cc-pVQZ and our results which is illustrated in Fig. 2. The CIPSI results [8] were not fully converged and were not generated with orbital rotations, The previous SHCI results have large stochastic error bars that prevent any high accuracy comparisons [22]. Thus we are left with the FCIQMC energy of \(-199.3598(2)\) Ha [41] as the best previous benchmark. The FCIQMC result uses the initiator approximation and consequently it will be subject to systematic bias [41]. The best ASCI result presented here has an energy of \(-199.36082\) Ha using natural orbitals which is 1 mHa below the FCIQMC results. However, we are able to validate our results with a linear extrapolation from energies generated with the Hartree-Fock orbitals. We find the extrapolated energy agrees with our best natural orbital result to better than 0.15 mHa (see Fig.2). This close agreement provides numerical validation for the accuracy of the ASCI results over the more approximate results generated from CIPSI, SHCI, and FCIQMC, all of which appear to differ from ASCI by at least 1 mHa.

We also note that simulating F\(_2\) in the cc-pVQZ basis gave rise to some unique challenges related to orbital rotations. In particular, compared to the other simulations presented here, it was important to use large wave functions in initial ASCI iterations, before performing the first orbital rotation for F\(_2\).

Table V presents orbital rotated results for C\(_2\) and N\(_2\). For these systems there are no comparable results for the stochastic selected CI approaches but for C\(_2\), there are converged DMRG results. We see that the results from the ASCI+PT2 approach agree with the DMRG results to better than 0.25 mHa across all basis sets. For N\(_2\), there are FCIQMC results (within the initiator approximation) [41] that can be compared, although the the initiator approximation renders comparison with ASCI less informative.

C. TZV basis calculation for Cr\(_2\)

The Cr\(_2\) dimer in the SVP basis set (at 1.5Å) has provided a standard benchmark for testing the efficiency of strongly correlated methods [1, 4, 32, 33, 42]. To date, only DMRG and selected CI techniques, such as ASCI, have been able to converge the result to below 1 mHa accuracy. Benchmarks have
Table I. Comparison of ASCI+PT2 to HCI+PT2 for ground state energies in the cc-pVDZ basis. HCI results are taken from ref. [22]. To reach these system sizes, HCI uses a stochastic algorithm to perform the PT2 (the resulting method being called SHCI). The colors indicate which ASCI results are most comparable to the stochastic SHCI results, as determined by matching the variational energies as closely as possible. For those comparisons we calculate the speedup that can be found by using the ASCI deterministic PT2 approach. These ASCI speedup factors demonstrate the speedup of ASCI over stochastic SHCI both i) for obtaining absolute energies with an expected error of 0.1 mHa with a 95% probability "E(0.1mHa)", and ii) for obtaining the equivalent accuracy to determine a physical energy difference such as required for calculating energy gaps and atomization energies "E diff(0.1mHa)". In simulations where stochastic and deterministic PT2 algorithms use the same variational wave function, the deterministic results will be always more accurate because they do not have any stochastic error. The SHCI results were performed on nodes with 2 Intel Xeon E5-2680 v2 processors of 2.8 GHz, with 20 computational cores per node. The timings reported for SHCI 'stochastic' are taken from ref.[22], for which we then calculate the single core cost (of a 2.8 GHz processor) it would take to produce 0.1 mHa accuracy with a 95% confidence. The energies for SHCI are directly taken from reference [22], and the reported error bars are 1σ error bars. The ASCI simulations were performed on a single core of a Intel Xeon E5-2620 v5 processor of 2.10 GHz. For all simulations we calculate the equivalent single core time, and we scale the ASCI timings to be representative of a single 2.10 GHz core. For the PT2 simulations, we neglect contributions less than 10⁻⁸, as in Ref. [22]. The colors are used to highlight results that are most comparable to each other in terms of energy of the variational wave function.

| Comparisons | Dets | Basis | Time PT2 (secs) | Energy (Ha) | ASCI speedup factor |
|-------------|------|-------|-----------------|------------|---------------------|
| C₂(8,26) HCI [22] | 28566 | cc-pVDZ | 640 | -75.7217 | -75.7286(2) | E(0.1mHa) | E diff(0.1mHa) |
| C₂(8,26) ASCI | 10000 | cc-pVDZ | 2 | -75.7168 | -75.7285 | | |
| C₂(8,26) ASCI | 20000 | cc-pVDZ | 3 | -75.7212 | -75.7287 | 213 | 426 |
| C₂(8,26) ASCI | 100000 | cc-pVDZ | 26 | -75.7258 | -75.7285 | | |
| C₂(8,26) ASCI | 182145 | cc-pVDZ | 43 | -75.7263 | -75.7285 | | |
| N₂(10,26) HCI [22] | 37993 | cc-pVDZ | 160 | -109.2692(1) | | | |
| N₂(10,26) ASCI | 10000 | cc-pVDZ | 2 | -109.26419 | -109.27691 | | |
| N₂(10,26) ASCI | 30000 | cc-pVDZ | 7 | -109.26938 | -109.27691 | 23 | 45 |
| N₂(10,26) ASCI | 100000 | cc-pVDZ | 27 | -109.27335 | -109.27698 | | |
| F₂(14,26) HCI [22] | 88994 | cc-pVDZ | 11760 | -199.09921(7) | | | |
| F₂(14,26) ASCI | 10000 | cc-pVDZ | 4 | -199.08368 | -199.09921 | | |
| F₂(14,26) ASCI | 100000 | cc-pVDZ | 37 | -199.09265 | -199.09929 | 310 | 620 |
| F₂(14,26) ASCI | 300000 | cc-pVDZ | 109 | -199.09406 | -199.09933 | | |

Table II. Comparison of ASCI+PT2 to SHCI for ground state energies in the cc-pVTZ basis. SHCI results are taken from ref. [22]. See caption of Table I for more details. The colors are used to highlight results that are most comparable to each other in terms of energy of the variational wave function. These ASCI speedup factors demonstrate the speedup of ASCI over stochastic SHCI both i) for obtaining absolute energies with an expected error of 0.1 mHa with a 95% probability "E(0.1mHa)", and ii) for obtaining the equivalent accuracy to determine a physical energy difference "E diff(0.1mHa)". The ASCI simulations were performed on a single core of a Intel Xeon E5-2620 v5 processor of 2.10 GHz.

| Comparisons | Dets | Basis | Time PT2 (secs) | Energy (Ha) | ASCI speedup factor |
|-------------|------|-------|-----------------|------------|---------------------|
| C₂(8,58) HCI [22] | 142467 | cc-pVTZ | 2880 | -75.77339 | -75.7846(3) | E(0.1mHa) | E diff(0.1mHa) |
| C₂(8,58) ASCI | 50000 | cc-pVTZ | 60 | -75.76893 | -75.784113 | | |
| C₂(8,58) ASCI | 100000 | cc-pVTZ | 1177 | -75.77396 | -75.78447 | 24 | 48 |
| C₂(8,58) ASCI | 142467 | cc-pVTZ | 166 | -75.775386 | -75.784589 | | |
| N₂(10,58) HCI [22] | 189060 | cc-pVTZ | 11520 | -109.34058 | -109.37414 | | |
| N₂(10,58) ASCI | 100000 | cc-pVTZ | 19 | -109.34058 | -109.37414 | | |
| N₂(10,58) ASCI | 100000 | cc-pVTZ | 784 | -109.34058 | -109.37414 | 62 | 124 |
| N₂(10,58) ASCI | 300000 | cc-pVTZ | 501 | -109.34058 | -109.37414 | | |
| F₂(14,58) HCI [22] | 395744 | cc-pVTZ | 38880 | -199.2782 | -199.2984(9) | | |
| F₂(14,58) ASCI | 20000 | cc-pVTZ | 60 | -199.2543 | -199.295491 | | |
| F₂(14,58) ASCI | 100000 | cc-pVTZ | 295 | -199.27131 | -199.296290 | | |
| F₂(14,58) ASCI | 300000 | cc-pVTZ | 931 | -199.27014 | -199.296686 | 43 | 86 |
| F₂(14,58) ASCI | 395744 | cc-pVTZ | 1163 | -199.279209 | -199.296767 | | |

recently been generated with semi-core simulations using at least 24 electrons with the Dunning basis sets (CIPSI) [8], and with DMRG using DKH corrections with a triple zeta basis [23, 43]. Here we present results for Ahlrich triple zeta valence (TZV) basis set [44] (24e, 76o), which is expected to be manageable to converge for most selected CI implementations and for DMRG. Computation of our benchmark energies presented here takes only a few hours, as shown in Table VI. The naive deterministic approach presented in Algorithm 2 will be quite fast even in the absence of substantial optimiza-
Table III. Comparison of ASCI+PT2 to SHCI for ground state energies in the cc-pVQZ basis. SHCI results are taken from ref. [22]. See caption of Table I for more details. In this table we also provide a comparison to CIPSI with the stochastic hybrid approach for the F₂ molecule [8]. For the CIPSI simulations, the calculation is done with a hybrid stochastic method but in the limit that all terms are calculated, and thus there is no error bar. They were performed on a Intel Xeon E5-2680 at 2.70 GHz. We have scaled the CPU time to 2.80 GHz which is the speed of the Intel processors for which the HCI results were calculated. The CIPSI result requires a substantially larger number of determinants to have similar variational energies as ASCI. We note that for cc-pVQZ F₂, the HCl wave function has a lower variational energy than ASCI for less determinants. This is highly unusual and is the only simulation we have ever observed this property. As a result we make direct comparisons with CIPSI, and compare the compactness of the wave function. It can be seen that the ASCI wave function with 2 million determinants has a variational energy that is comparable to the CIPSI wave function with 4 million determinants. The colors are used to highlight results that are most comparable to each other in terms of energy of the variational wave function.

| Comparisons | Dets | Basis | Time PT2 (secs) | Energy (Ha) | ASCI speedup factor |
|-------------|------|-------|-----------------|-------------|---------------------|
| C₂(8,108) HCl [22] | 403071 | cc-pVQZ | 12800 | -75.7894 | -75.8018(4) |
| C₂(8,108) ASCI | 10000 | cc-pVQZ | 80 | -75.75108 | -75.799103 |
| C₂(8,108) ASCI | 10000 | cc-pVQZ | 670 | -75.78297 | -75.80128 |
| C₂(8,108) ASCI | 300000 | cc-pVQZ | 2020 | -75.79030 | -75.80192 |
| N₂(10,108) HCl [22] | 399644 | cc-pVQZ | 64800 | -109.3884 | -109.4035(0) |
| N₂(10,108) ASCI | 10000 | cc-pVQZ | 115 | -109.34421 | -109.40349 |
| N₂(10,108) ASCI | 10000 | cc-pVQZ | 990 | -109.38073 | -109.40448 |
| N₂(10,108) ASCI | 300000 | cc-pVQZ | 2865 | -109.38782 | -109.40491 |
| F₂(14,108) HCI [22] | 1053491 | cc-pVQZ | 142500 | -199.3463 | -199.3590(9) |
| F₂(14,108) CIPSI [8] | 4000000 | cc-pVQZ | 1533503 | -199.3417 | -199.3594 |

Table IV. Comparison of ASCI+PT2 to stochastic SHCI for ground state energies of F₂ with orbital rotations turned on. HCI results are taken from ref. [22]. The deterministic ASCI+PT2 approach has significantly improved performance over stochastic SHCI, compared to the results without orbital rotations in Tables I, II and III. The timings reported for SHCI ‘stochastic’ are taken from reference [22], for which we then calculate the single core cost (of a 2.80 GHz processor) it would take to produce 0.1 mHa accuracy with a 95% confidence. The energies for SHCI are directly taken from reference [22]. The deterministic ASCI+PT2 approach has significantly improved performance over stochastic SHCI, compared to the results without orbital rotations in Tables I, II and III. The timings reported for SHCI ‘stochastic’ are taken from reference [22], for which we then calculate the single core cost (of a 2.80 GHz processor) it would take to produce 0.1 mHa accuracy with a 95% confidence. The energies for SHCI are directly taken from reference [22], and the reported error bars are 1σ error bars. The colors are used to highlight results that are most comparable to each other in terms of energy of the variational wave function. A number of further speedups to this naive approach are however possible. In this section we describe some of the techniques that we have developed to accelerate the constraint PT2 approach and present the substantially faster run times achieved from these additional refinements.

The techniques described here are:

- Fast generation of triplet contributions
- Removing duplicates with the core determinants
- Spin symmetry ($Z_2$) for $S_z = 0$ calculations
- Data reuse for unique α and β strings (triplet constraint specific)
- Matrix element cutoff and fast diagonal matrix elements
- Parallelization with work lists.
- Generalized compression with hash functions
Table V. ASCI+PT2 with orbital rotations for C2 and N2. For C2, comparisons are made to DMRG [33]. For N2, comparisons are made to FCIQMC [41]. The DMRG results represent a highly converged benchmark, whereas the FCIQMC results are approximate, with uncontrolled errors deriving from the initiator approximation. As a result, the best ASCI results agree with DMRG to better than 0.25 mHa, while agreement with the approximate FCIQMC results is not quite as good. These calculations show much quicker convergence (with respect to number of determinants) than the respective simulations without orbital rotations, as expected. The column “Contributions” lists how the number of PT2 contributions, as well as the size of the subset that have values greater than 10^-8. The column “Unique α” lists the number of unique α bitstrings in the variational wave function.

A GPU could also be employed to reduce the sorting time even further. We generally find that a single GPU is about 10 times faster than a single CPU for sorting [9]. There is still much ongoing development of GPU sorting [35]. Much of the current runtime is used to move data onto and off the GPU, so a fully GPU implementation could have even further speed improvements.

1. Fast generation of triplet contributions

One of the main costs for the PT2 algorithm presented in this work is the cost for generating the contributions. For every constraint, one has to loop over all determinants. Thus the constraint generation routine will be called $N_{tdets}$ times for each triplet constraint considered. The most expensive part of generating the constrained PT2 contributions is creating the contributions from the double alpha or double beta excitations. To create these excitations quickly, we consider three cases for creating a list of pairs of virtual orbitals. Once these lists are created, every pairwise combination generates all excitations from $D_t$ that satisfy the triplet constraint. The different cases are described in Algorithm 3.
Algorithm 3 Generate fast triplet contributions

1: For each determinant $D_i$ in the trial wave function, generate only the relevant PT2 contributions that match the current triplet string $T$. We show the more complicated case of double excitations here. Do this by creating a reduced list of occupied pairs ($\{O\}$) and virtual pairs ($\{V\}$) as follows:

2: Notation: $\wedge$ is (bit-wise) logical AND, $\oplus$ is (bit-wise) logical XOR

3: Definition $T_{small}$: The smallest occupied in the triplet $T$

4: Create a bitmask $B$, which is equal to 1 for all orbitals greater than $T_{small}$.

5: If countbits($D_i \wedge T$) = 0 or countbits(($D_i \wedge B$) $\oplus T$) > 2, return empty ($\{O\}$) and ($\{V\}$) (There are too many differences between $D_i$ and $T$ to be fixed with a double excitation)

6: Begin case for creating $\{V\}$:

   - Case(1): If countbits($D_i \wedge T$) == 1, put the pair of orbitals that correspond to the bits in (($D_i \oplus T$) $\wedge T$) into ($\{V\}$)
   - Case(2): If countbits($D_i \wedge T$) == 2, put the pairs ($x$, (($D_i \oplus T$) $\wedge T$)) into ($\{V\}$), where $x$ is any unoccupied orbital smaller than $T_{small}$
   - Case(3): If countbits($D_i \wedge T$) == 3, put the pairs ($x$, $y$) into ($\{V\}$), where $x$ and $y$ are distinct unoccupied orbitals smaller than $T_{small}$

7: Begin case for creating $\{O\}$:

   - Case(1): If countbits(($D_i \wedge B$) $\oplus T$) == 2, put the pair of orbitals that correspond to the bits in (($D_i \wedge B$) $\oplus T$) into ($\{O\}$)
   - Case(2): If countbits(($D_i \wedge B$) $\oplus T$) == 1, put the pairs ($x$, (($D_i \wedge B$) $\oplus T$)) into ($\{O\}$), where $x$ is any occupied orbital less than $T_{small}$
   - Case(3): If countbits(($D_i \wedge B$) $\oplus T$) == 0, put the pairs ($x$, $y$) into ($\{O\}$), where $x$ and $y$ are distinct occupied orbitals smaller than $T_{small}$

8: Double loop over the elements of ($\{O\}$) and ($\{V\}$) to create all possible double excitations from $D_i$ with the orbitals in $T$ as the highest occupied.

2. Removal of duplicate PT2 contributions from the variational wave function

Only determinants absent from the variational SCI wave function contribute to the PT2 energy. However, the list of all possible single and double excitations out of the variational SCI wave function also contains all the determinants present in the SCI wave function (since the search algorithms should guarantee that any determinant in the variational wave function is connected to at least one other determinant in the variational wave function). This situation persists even after imposition of a constraint. Therefore measures must be taken to avoid contamination of the PT2 energy from the variational wave function determinants.

A simple approach to this would be to explicitly check whether a given excitation is present in the variational SCI wave function or not. However, this is quite inefficient, with a cost of verification scaling at least as $\log N_{tdets}$ for each excited state determinant (assuming an efficient algorithm such as binary search is used), and there would be $N_SDN_{tdets}$ such excitations. The net cost of this simple approach would therefore show a rather prohibitive $N_SDN_{tdets}\log N_SD$ scaling.

A more efficient approach is to calculate the PT2-like contributions from determinants present in the variational wave function in advance, and subtract these from the PT2-like energy computed from all possible excitations out of the variational wave function. This would yield a PT2 energy without any contamination from the determinants of the variational wave function. The PT2-like contribution from determinants in the variational wave function is calculated efficiently with only a linear $N_{tdets}$ scaling as follows:

1. For true PT2 contributions $|D_i\rangle$ that are not included in the variational wave function $|\psi\rangle$ (i.e., contributions not in the variational space), we attempt to find $\langle D_i | H | \psi \rangle$ by computing contributions from all variational wave function determinants that are a single or double excitation away from $|D_i\rangle$. This is exact if $|D_i\rangle$ itself is not in the variational wave function.

2. If $|D_i\rangle$ was in the variational wave function, with coefficient $c_i$, we instead calculate $\langle D_i | H | \psi \rangle - c_i \langle D_i | H | D_i \rangle$, where the negative term arises because the algorithm finds single and double excitations only, which misses the zero excitation diagonal term $c_i \langle D_i | H | D_i \rangle$.

3. To proceed, recall that $\langle D_i | H | \psi \rangle = E_{ASC1} \langle D_i | \psi \rangle = c_i E_{ASC1}$, since $|D_i\rangle$ and $|\psi\rangle$ both belong to the subset of the Hilbert space spanned by determinants employed in the variational wave function, and $|\psi\rangle$ is furthermore an eigenstate of $H$ within that subspace.

4. Construct a contribution of $\langle D_i | H | \psi \rangle - c_i \langle D_i | H | D_i \rangle = c_i (E_{ASC1} - H_{ii})$ for each $|D_i\rangle$ in the variational wave function. This corresponds to a PT2-like contribution of $\frac{|\langle D_i | H | \psi \rangle - c_i \langle D_i | H | D_i \rangle|^2}{E_{ASC1} - H_{ii}} = c_i^2 (E_{ASC1} - H_{ii})$.

5. These extra PT2-like contributions $c_i^2 (E_{ASC1} - H_{ii})$ are summed over all $N_{tdets}$ $|D_i\rangle$'s, and are then subtracted from the PT2-like energy computed by the algorithm, yielding the true PT2 energy.

3. Spin symmetry ($Z_2$) for $S_z = 0$ calculations

For systems in which the quantum number for $S_z = 0$, one can loop over all determinants at the end of an ASCI run and ensure that, for all $\alpha/\beta$ pairs, the bitstring corresponding to the swap of the $\alpha/\beta$ strings is also present. If not, the corresponding bitstring can be added in and the wave function re-diagonalized (large ASCI wave functions generally are very close to having this symmetry automatically). When the
ASCII wave function has this α/β symmetry, then the set of determinants contributing to the PT2 correction will also have this symmetry, and roughly half of the contributions can be ignored. This can be implemented in practice by only generating PT2 contributions for which the α bitstrings are not smaller than the β bitstrings. These contributions are then multiplied by 2 to account for the excluded cases.

4. Data reuse for unique α and β strings (triplet constraint specific)

We can also sort the ASCII wave function by the numerical α bitstrings prior to calculating the PT2 contribution. This permits reuse of information and calculations associated with a given α bitstring. For example, the double α excitations will be the same for all bitstrings with the same α bitstring, and the off-diagonal Hamiltonian matrix elements will also be the same for these excitations. Thus it is possible to reuse the generation of PT2 contributions and matrix element calculations over all reference determinants that differ only in the beta bitstring. This is a significant speedup, since the number of unique α bitstrings is often substantially smaller than the size of the variational wave function, and by percentage, goes down as the latter gets bigger. For virtually all simulations calculated here and in previous ASCII work [1, 9], the number of unique α bitstrings is generally less than 10% of the total number of variational determinants for wave functions of size 100,000. This often becomes less than 5% when using up to 1 million determinants. In Table V and VI, we demonstrate this trend by presenting the number of unique α bitstrings for each calculation.

5. Matrix element cutoff and fast diagonal matrix elements

During the above process, when calculating matrix elements, a cutoff is used for determining whether a given matrix element should be stored and used. For the simulations considered here, we generally use a cutoff of $10^{-8}$ for inclusion of matrix elements in the PT2 calculations. As noted above, a cutoff is not required, but it can slightly speed up the algorithm without any significant loss of accuracy for the calculations under consideration here. We then calculate the diagonal matrix element in the denominator of Eq. 3 only after checking that the matrix element in the numerator is larger than the cutoff value. For evaluation of the diagonal matrix elements, we use the fast diagonal matrix element algorithm that was presented in ref. [9] and is also included here in the appendix.

6. Parallelization with work lists

The constraint PT2 algorithm presented in this work can be calculated in parallel over the constraints without any communication between parallel units, as described in previous sections. After the contributions have been generated, it is possible to do a parallel sort over the contributions or, alternatively, to offload the sorting to a GPU. We have previously presented benchmarks for the parallelization and sorting with the Thrust library [45] and the IPS^4O parallel sort [36]. The amount of work for each triplet constraint scales with the number of PT2 contributions that are consistent with the constraint. Before calculating any of the PT2 contributions, it is possible to do a first pass over the ASCII wave function and determine how many contributions will be consistent with each triplet. This work list can be saved and used to make sure that enough memory is available to calculate the contributions for each triplet constraint. It can also be used to provide an efficient load distribution for parallel execution.

7. Generalized compression with hash functions

Going to larger basis sets becomes more costly for the ASCII approach, for several reasons. One of the biggest cost increases is due to the manipulation of larger bitstrings associated with a larger basis set. Some of this cost can be mitigated by using more compact bitstring representations. While the standard bitstring representation is suitable for non-selected CI algorithms, there might be better forms for SCI approaches.

Here we consider how hashing can be used to compress bitstrings for use in calculating the PT2 energy. Hashing is an extremely fast way to compress a bitstring. The one negative feature of hashing, namely that the compression can induce collisions, i.e., two long bitstrings can be compressed to the same shorter value, can be managed by taking advantage of hash functions designed to avoid certain types of collisions. For example many hash functions have the property called avalanching, where small changes of any bit in the uncompressed bitstring can lead to a larger difference in the hashed string. Such a feature is important for our purposes and is also well tested and understood for all widely used hash functions [46, 47]. Compression of bitstrings for selected CI applications is highly desirable and will very generally provide speedups when used in a sorting/hashing approach.

It is important to understand how often collisions may occur when hashing. As an example, there are over $10^{36}$ unique numbers that can be represented with a 128 bit integer, which is the length of bitstrings used in the SVP Cr calculation. Yet, at most, only on the order of $10^{13}$ PT2 contributions will be generated in a very large scale PT2 calculation. If we consider compressing 128 bitstrings to bitstrings of length 64, we expect that a collision is highly unlikely for applications of selected CI simulations. That is to say, the number of determinants being considered will not be dense in comparison to the number of values that can be represented with 64 bits (which is $≈ 10^{18}$).

For testing we considered 64 bit hash functions. The term 64 bits indicates the size of the output integer that is created when a longer bitstring is hashed. For the electronic struc-
VI. CONCLUSIONS

In this work, we have presented an efficient deterministic alternative for the second order perturbation theory (PT2) refinement to a selected CI method. This algorithm leverages a combination of fast sorting algorithms on modern computers with a well thought out algorithmic design. The result is a highly efficient algorithm that is in many cases two orders of magnitude faster than the recently proposed stochastic approaches to evaluation of PT2 corrections. This approach allowed us to converge the ground state energies of several different molecules which previous selected CI approaches have had difficulties converging, including F$_2$. It might also be possible to use the approaches developed here to produce a more efficient stochastic PT2 algorithm, though the lack of stochastic error would ensure that any deterministic algorithm would remain arbitrarily more accurate (ignoring numerical precision errors that would be present in both approaches).

We also found that the SCI+PT2 energy converges faster when a more compact variational wave function is used as the reference. Such wave functions can be readily obtained with the ASCI algorithm, and compactness can further be enhanced through natural orbital rotations. All these improvements to the efficiency of the ASCI approach and its PT2 refinement, leads to the conclusion that ASCI is well-suited to pursue many different electronic structure problems in chemistry and physics that have been difficult to pursue with other techniques and approaches.

VII. ACKNOWLEDGEMENTS

This work was supported through the Scientific Discovery through Advanced Computing (SciDAC) program funded by the U.S. Department of Energy, Office of Science, Advanced Scientific Computing Research and Basic Energy Sciences. We used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by the National Science Foundation Grant No. OCI-1053575. DH was supported by a Berkeley Fellowship. DSL and MHG acknowledge support from the Director, Office of Science, Office of Basic Energy Sciences, of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.
could be evaluated \textit{ab initio}. Algorithm 4 describes the protocol. Briefly, the contributions to the energy of the excited electrons are calculated and added to the reference determinant energy, while the contribution of the removed electron is subtracted. Calculating the array of partial contributions described in Algorithm 4 has to be done only once per reference determinant.

Algorithm 4: Fast Diagonal Matrix elements

1: (Precalculation step) Input Determinant $D_k$, the diagonal matrix element $H_{ii}$, and the one-electron integrals $h_{ij}$
2: (Precalculation step) Calculate the partial contribution: $p(i) = \sum_{ij} \langle ij| h_{ij} | ij \rangle$
3: Input $D_k$ (connected to $D_i$) with set of orbitals excited into ($A$) and the set of orbitals excited out of ($R$)
4: $E_{rem} = \sum_{i\in R} h_{ii} + p(i)$, $E_{add} = \sum_{i\in A} h_{ii} + p(i)$
5: $H_{kk} = H_{ii} - E_{rem} + E_{add} - \sum_{i\in R,j\in A} \langle ij| h_{ij} | ij \rangle$
6: if $A, R$ have two elements from the same spin space then $H_{kk} = H_{kk} + \langle R_i R_k | R_2 R_2 \rangle - \langle A_i A_k | A_2 A_2 \rangle - \langle R_i R_1 | A_2 A_2 \rangle - \langle R_2 R_2 | A_1 A_1 \rangle$
7: else if $A, R$ have two elements from different spin spaces then $H_{kk} = H_{kk} + \langle R_i R_0 | R_2 R_0 \rangle + \langle A_0 A_0 | A_3 A_3 \rangle - \langle A_0 A_0 | R_2 R_0 \rangle - \langle R_0 R_0 | A_3 A_3 \rangle$
8: end if

---

1. N. M. Tubman, J. Lee, T. Y. Takeshita, M. Head-Gordon, and K. B. Whaley, J. Chem. Phys. 145, 044112 (2016), http://dx.doi.org/10.1063/1.4955109.
2. F. A. Evangelista, J. Chem. Phys. 140, 124114 (2014).
3. P. J. Knowles, Mol. Phys. 113, 1655 (2015).
4. A. A. Holmes, N. M. Tubman, and C. J. Umrigar, J. Chem. Theory Comp. 12, 3674 (2016), http://dx.doi.org/10.1021/acs.jctc.6b00407.
5. J. B. Schirber and F. A. Evangelista, J. Chem. Phys. 144, 161106 (2016).
6. W. Liu and M. R. Hoffmann, J. Chem. Theory Comput. 12, 1169 (2016).
7. J. B. Schirber and F. A. Evangelista, Journal of Chemical Theory and Computation 13, 5354 (2017), pMID: 28892621, https://doi.org/10.1021/acs.jctc.7b00725.
8. Y. Garniron, A. Scemama, P.-F. Loos, and M. Caffarel, The Journal of Chemical Physics 147, 034101 (2017), https://doi.org/10.1063/1.4992127.
9. N. M. Tubman, C. D. Freeman, D. S. Levine, D. Hait, M. Head-Gordon, and K. B. Whaley, arXiv preprint arXiv:1807.00821 (2018).
10. P. E. M. Siegbahn, Chem. Phys. Lett. 109, 417 (1984).
11. F. A. Knowles and N. C. Handy, Chem. Phys. Lett. 111, 315 (1984).
12. J. Olsen, B. O. Roos, P. Jorgensen, and H. J. A. Jensen, J. Chem. Phys. 89, 2185 (1988).
13. C. D. Sherrill and H. F. S. III. (Academic Press, 1999) pp. 143 – 269.
14. Z. Gan and R. Harrison, in Supercomputing, 2005. Proceedings of the ACM/IEEE SC 2005 Conference (2005) pp. 22–22.
[48] “https://code.google.com/archive/p/xxhash/,” (2015).
[49] “https://github.com/leo-yuriev/t1ha,” (2016).
[50] “https://en.wikipedia.org/wiki/murmurhash,” (2015).
[51] “http://burtleburtle.net/bob/hash/spooky.html,” (2011).
[52] “http://www.isthe.com/chongo/tech/comp/fnv/,” (2015).
[53] “https://aras-p.info/blog/2016/08/09/more-hash-function-tests/,” (2013).