Large scale parallelization in stochastic coupled cluster

J. S. Spencer, V. A. Neufeld, W. A. Vigor, R. S. T. Franklin, and A. J. W. Thom

---

Large scale parallelization in stochastic coupled cluster

J. S. Spencer,[1,2] V. A. Neufeld,[3] W. A. Vigor,[4] R. S. T. Franklin,[3] and A. J. W. Thom[3,a)]

[1]*Department of Materials, Imperial College London, Exhibition Road, London SW7 2AZ, United Kingdom*
[2]*Department of Physics, Imperial College London, Exhibition Road, London SW7 2AZ, United Kingdom*
[3]*University Chemical Laboratory, Lensfield Road, Cambridge CB2 1EW, United Kingdom*
[4]*Department of Chemistry, Imperial College London, Exhibition Road, London SW7 2AZ, United Kingdom*

Coupled cluster theory is a vital cornerstone of electronic structure theory and is being applied to ever-larger systems. Stochastic approaches to quantum chemistry have grown in importance and offer compelling advantages over traditional deterministic algorithms in terms of computational demands, theoretical flexibility, or lower scaling with system size. We present a highly parallelizable algorithm of the coupled cluster Monte Carlo method involving sampling of clusters of excitors over multiple time steps. The behavior of the algorithm is investigated on the uniform electron gas and the water dimer at coupled-cluster levels including up to quadruple excitations. We also describe two improvements to the original sampling algorithm, *full non-composite*, and *multi-spawn* sampling. A stochastic approach to coupled cluster results in an efficient and scalable implementation at arbitrary truncation levels in the coupled cluster expansion. *Published by AIP Publishing.* https://doi.org/10.1063/1.5047420

I. INTRODUCTION

Coupled cluster (CC) methods[1] are of crucial importance in electronic structure and have been used to explore a variety of systems, including atoms and molecular systems,[1–5] the uniform electron gas,[6–14] and solids/other periodic systems.[15–25] Coupled-cluster singles doubles and perturbative triples [CCSD(T)],[26] where single and double excitations are included in the wavefunction ansatz and supplemented with the perturbative treatment of triple excitations, is commonly regarded as the "gold standard" of quantum chemistry and can frequently achieve[27] a chemical accuracy of 1 kcal/mol.

Despite these successes, coupled cluster is not without its drawbacks. Coupled cluster is systematically improvable, at least in principle, by increasing the excitation level included in the CC wavefunction ansatz. Doing so makes the conventional CC equations vastly more complicated and hence computational demanding. As a result, treating higher truncation levels is possibly only in specialist codes.[28] Conventional implementations of coupled cluster also rely heavily upon dense linear algebra, which does not scale well with increasing numbers of processors on parallel or heterogeneous computer architectures although recent work in linear algebra and tensor libraries are making impressive progress.[29,30]

One avenue for improving the computational efficiency of coupled cluster is to exploit the nearsighted nature of electron correlation and use local approximations.[31–34] Another approach, of increasing use in quantum chemistry and the broader electronic structure community, is to use stochastic methods; these have proven to provide low-scaling algorithms for electronic structure methods and typically exhibit excellent scaling with increasing processor count.[35–39] Local and stochastic methods may also be easily combined via a

localisation transformation of the mean-field single-particle orbitals.[40]

The full configuration interaction quantum Monte Carlo (FCIQMC) method[41,42] has been a major development in quantum chemistry. By sampling the action of the Hamiltonian, FCIQMC has been able to calculate exact properties for quantum systems inaccessible to conventional diagonalisation techniques.[41,43–46] The computational advantage of FCIQMC is largely through a representation of the FCI wavefunction which is significantly more compact than the full wavefunction, though still scaling factorially with the size of the Hilbert space sampled.

One of us (AJWT) subsequently used a similar approach to formulate a Monte Carlo approach to coupled cluster theory (CCMC),[47] inheriting the benefits of more compact storage and now scaling with the polynomial size of the truncated CC space. The initiator approximation can substantially improve the stochastic sampling of the wavefunction in both FCIQMC[42] and CCMC[11] although the latter requires careful extrapolation. The stochastic sampling of the coupled cluster wavefunction can be further improved by sampling only linked diagrams[48] and non-uniform sampling of the coupled cluster expansion,[49] and improved sampling of the action of the Hamiltonian.[50] The utility of CCMC has been demonstrated to calculate coupled cluster energies at up to the CCSDTQ56 level for molecular systems,[47,49] for the uniform electron gas,[11,14] and also been used to automatically generate the P subspace in the CC(P; Q) method.[51] However, due to the non-linearity of the coupled cluster equations, parallelization of the CCMC algorithm is less straightforward than a parallel FCIQMC implementation.[52]

We present a brief overview of the CCMC algorithm in Sec. II, which provides context to the problem. In Sec. III, we show that the CCMC algorithm can be efficiently parallelized by introducing an additional level of Monte Carlo sampling

a)ajwt3@cam.ac.uk

by considering only a subset of terms in the coupled cluster expansion per iteration. The accuracy and performance of this algorithm is investigated using the uniform electron gas and the water dimer. Section IV provides simple improvements to the original CCMC algorithm to improve stability and convergence of the CC wavefunction. We conclude in Sec. V.

II. COUPLED CLUSTER MONTE CARLO

The algorithms used to sample the FCI and coupled cluster wavefunctions have been previously detailed[11,41,47,49,53] and as such we summarise the key features relevant to this work here.

The coupled cluster wavefunction ansatz can be expressed as $|\Psi\rangle = N e^{\hat{T}} |D_{HF}\rangle$, where $|D_{HF}\rangle$ is the Hartree–Fock determinant, N controls the (intermediate) normalisation, and the cluster operator $\hat{T}$ is

$$\hat{T} = \sum_{\substack{i,a}} t_i^a \hat{c}_i^a + \sum_{\substack{i<j \\ a<b}} t_{ij}^{ab} \hat{c}_{ij}^{ab} + \cdots, \tag{1}$$

where $\{t_{i\cdots}^{a\cdots}\}$ is the set of amplitudes and $\hat{c}_{i\cdots}^{a\cdots}$ is an *excitor* comprising of a string of creation and annihilation operators. For convenience, we use $\hat{c}_i$ and t_i, such that $\hat{c}_i|D_{HF}\rangle$ produces $|D_i\rangle$ (up to a sign, as discussed later) and t_i is the corresponding amplitude, and rescale the amplitudes with an additional factor, t_{HF}, such that

$$|\Psi\rangle = t_{HF} e^{\hat{T}/t_{HF}} |D_{HF}\rangle. \tag{2}$$

Within this wavefunction ansatz, the coefficient of a given determinant, $\tilde{t}_j = \langle D_j | \Psi \rangle$, contains contributions from all sets of excitors which can be combined to produce that determinant.

As with FCIQMC, CCMC applies an approximate linear propagator, $1 - \delta\tau(\hat{H} - S)$, where $\delta\tau$ is the time step and S is an adjustable parameter to control proportionality, and which has the same eigenspectrum as $e^{-\delta\tau\hat{H}}$ for sufficiently small $\delta\tau$.[53] Applying this to $|\Psi\rangle$ and cancelling quadratic and higher-order terms[11,47] result in a form reminiscent of the propagation equation for FCIQMC,[41]

$$t_i(\tau + \delta\tau) = t_i(\tau) - \delta\tau \sum_j (H_{ij} - S\delta_{ij})\tilde{t}_j(\tau). \tag{3}$$

The similarity-transformed Hamiltonian can also be used in the projection the coupled cluster wavefunction,[48] and the approaches presented here equally apply to that formulation. The amplitudes are stochastically sampled by representing them using either particles with integer[41,47] or real (as opposed to integer) weights,[44,54] which has been shown to reduce the stochastic error within FCIQMC and readily applies to CCMC.

In FCIQMC, the particles on each occupied determinant are explicitly evolved; in CCMC, this would require one to first evaluate all possible $\tilde{t}_j$, which is computationally painful. Instead, we exploit the fact that Monte Carlo is a powerful tool for sampling high dimensional spaces and, in addition to stochastically sampling the action of the Hamiltonian, also sample the wavefunction ansatz. The algorithm used to sample the cluster expansion has been shown to have a significant impact on computational and statistical efficiency;[49] here we consider only the simplest approach. The cluster size, $s = [0, l+2]$, is selected according to an exponential distribution, i.e., $p_{size}(s) = 2^{-(s+1)}$, where l is the highest order term

in the cluster expansion.[55] The cluster is then generated by selecting s excitors from the current distribution, each with probability $|t_i|/(N_{total} - t_{HF})$, where N_{total} is the total current population. A cluster containing the same excitor more than once is discarded. Alternative approaches for sampling the cluster expansion are discussed in Sec. IV.

The dynamics for evolving the particles on a cluster are essentially identical to those in FCIQMC.[41] A simulation starts with a number of particles of unit weight on the Hartree–Fock determinant. The particles are then evolved by sampling the action of the Hamiltonian on each particle, allowing new particles to be created ("spawned"), and the particle to die (due to the sign of the Hamiltonian operator). At the end of each iteration, particles on the same excitor with opposite signs are removed ("annihilated") from the simulation, which aids the sign problem[53] and is a statistically exact process. Note that for clusters of size 2 and higher, the death step amounts to creating a particle of opposite sign on the corresponding excitor. Events which create particles on excitors which are not within the desired truncation level of the cluster operator are simply discarded in our current CCMC implementation.

Anti-commutation relationships in strings of creation and annihilation operators must be handled with care. A given excitor is required to be unique and hence an arbitrary excitor $\{\hat{c}_{ij\cdots l}^{ab\cdots e}\}$ must satisfy $i < j < \cdots < l$ and $a < b < \cdots < e$. Defining $\hat{c}_i$ ($\hat{c}_i^\dagger$) to annihilate (create) an electron in the ith spin-orbital, an excitor and a determinant can be expressed as

$$\hat{c}_{ij\cdots l}^{ab\cdots e} = \hat{c}_a^\dagger \hat{c}_b^\dagger \cdots \hat{c}_e^\dagger \hat{c}_l \cdots \hat{c}_j \hat{c}_i, \tag{4}$$

$$|D_i\rangle = |i_1 i_2 i_3 \cdots i_N\rangle = \hat{c}_{i_1}^\dagger \hat{c}_{i_2}^\dagger \hat{c}_{i_3}^\dagger \cdots \hat{c}_{i_N}^\dagger |0\rangle, \tag{5}$$

where $|0\rangle$ is the vacuum state and $i_1 < i_2 < \cdots < i_N$. Therefore, when collapsing a cluster, $\hat{c}_i \hat{c}_j \cdots \hat{c}_k$, to a single excitor, $\hat{c}_l$, a negative sign must be included as required by anticommutativity in order for the operators in the cluster to match the order in the single excitor. Similarly when an excitor is applied to the Hartree–Fock determinant, the resultant set of creation operators must be permuted in order to achieve the required ordering,

$$s_D(i) = \langle D_i | \hat{c}_i | D_{HF} \rangle. \tag{6}$$

The sign from collapsing a cluster is conveniently absorbed into the amplitude of the cluster, and the sign from converting to/from a determinant in the spawning step, such that the sign of the spawned particle, is determined by $-\text{sgn}(H_{ij} s_D(i) s_D(j))$.

The energy shift, S, is not known *a priori*. In keeping with other QMC methods,[41,56] S is updated to keep the population stable. In a simulation, S is initially held constant (typically at the Hartree–Fock energy) to allow the population to grow and is only adjusted once the population has reached a desired value. It is important to take the non-linear wavefunction ansatz into account during the constant-shift phase in order to ensure correct normalisation.[48]

The energy at a given time can, as with FCIQMC, be evaluated with a projected estimator. Again, it is simpler to sample the wavefunction using the same set of clusters, $\{\hat{c}_j\}$, chosen above,

$$E_{\text{proj.}} = t_{\text{HF}}^{-1} \sum_{\{\hat{c}_j\}} \langle D_{\text{HF}}|\hat{H}|D_j\rangle \frac{t_j s_D(\boldsymbol{j})}{p_{\text{cluster}}(\boldsymbol{j})}. \qquad (7)$$

A. Computational methods

All CCMC calculations are performed using a development version of HANDE. Most one- and two-body molecular integrals were obtained from restricted Hartree–Fock calculations performed in Psi4,[57] except for the study of three water molecules at large distances from each other where the integrals were obtained with PySCF[58] and localised with a Boys method[59] by PySCF. Floating-point weights were used to improve stochastic efficiency. Input files and raw data are available under a Creative Commons license at https://doi.org/10.17863/CAM.30359. Estimates of the stochastic error in CCMC simulations were obtained via a reblocking analysis.[60] QMC energies when quoted as a value with errorbar were verified to be unaffected by a population control bias by comparison to those obtained using a reweighting analysis.[56,61]

All data were analyzed using numpy,[62] pandas,[63] and pyblock,[64] and plots were produced using matplotlib[65] and seaborn.[66] Some QMC calculations have been restarted from calculations run with different settings.

III. PARALLELISATION

Using distributed computer architectures is advantageous both in terms of reducing the runtime of a calculation and in being able to treat larger systems due to the corresponding increase in available memory. The memory usage on a given processor of a QMC calculation in Slater determinant space is proportional to the number of states stored on that processor, whilst the computational workload is a function of both the number of states and the total population on the processor.[52] Ideally both would be evenly balanced across all processors, whilst the annihilation step requires all particles on the same determinant to be placed on the same processor. The size of the Hilbert space precludes a lookup table, and simply dividing the Hilbert space into chunks and assigning chunk(s) to a processor yields poor load balancing as the distribution of "important" states tends to be highly irregular in many chemical systems. Booth *et al.*[52] proposed a deterministic mapping of a determinant to a processor, $p(|D\rangle)$, in a time- and space-efficient manner,

$$p(|D\rangle) = \text{hash}(|D\rangle) \bmod N_{\text{p}}, \qquad (8)$$

where N_{p} is the number of processors and hash is a function which maps an arbitrary amount of data (here a representation of a determinant) to an integer over a fixed range. Crucially a good hash function returns different values for similar inputs, and hence determinants which are close in excitation space are mapped to different processors.[67]

CCMC introduces the additional complication that the cluster expansion must be sampled. One option, which we exploit, is to use a shared-memory paradigm (implemented in HANDE using OpenMP), where the cluster selection and evolution are distributed over threads. Distributing the set of states over multiple nodes,[68] as performed in FCIQMC, is not helpful as either each spawning event would involve

communication between nodes in order to randomly generate clusters. Instead, we again exploit Monte Carlo sampling: a node only samples the subset of clusters that can be formed from the excitors residing on that node. Crucially, the subset of clusters changes such that all clusters have an equal chance of being selected within a few time steps.

Concretely, Eq. (8) is extended to periodically change the processor of a given excitor,

$$o(|D\rangle, i_\tau) = (\text{hash}(|D\rangle) + i_\tau) \gg \nu_{\text{move}}, \qquad (9)$$

$$p(|D\rangle, i_\tau) = \text{hash}(|D\rangle \oplus o(|D\rangle, i_\tau)) \bmod N_{\text{p}}, \qquad (10)$$

where i_τ is the iteration index at time τ (i.e., $\tau/\delta\tau$), $2^{\nu_{\text{move}}}$ is a constant termed the "move frequency" and is discussed below, $a \gg b$ represents the right-shift bit operation on a, where the bits in a are moved to the right and the b least significant bits are removed, and $\oplus$ is the exclusive or bit operation. The offset function, $o(|D\rangle, i_\tau)$, discards the lower ν_{move} bits and hence changes value every $2^{\nu_{\text{move}}}$ iterations, where the iteration at which it first changes is determined by the value of hash$(|D\rangle)$. Hence the processor index of an excitor can change every $2^{\nu_{\text{move}}}$ iterations. Given a good hash function, p returns an even distribution of values in the range $[0, N_{\text{p}})$ and all clusters can still be sampled over a number of iterations. The probability of selecting a cluster of size s is scaled by N_{p}^{-s} to account for the probability of each excitor in the cluster being on the same processor. Excitors are efficiently redistributed at the same time as newly created particles are communicated to the appropriate node.

A. Parallelization scaling

The scaling of the parallelization algorithm is demonstrated on the water dimer in Fig. 1. All calculations were run with a different number of MPI processes divided into 12 OpenMP threads giving a total number of cores used. They were all restarted from a calculation that was run on 384 cores (32 MPI processes × 12 OpenMP threads) and used the same parameters. The speed-up was then evaluated as the ratio of time taken per iteration when using 384 cores over the current number of cores. Up to about 500 cores, the "strong
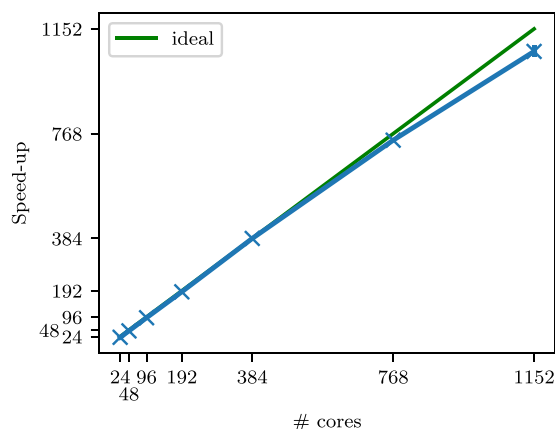

FIG. 1. Scaling of hybrid MPI + OpenMP CCSDT calculations on the water dimer using a jun-cc-pVDZ basis set performed using even selection.[49] 12 OpenMP threads were used per MPI process. Timings were taken from an equilibrated calculation on 384 cores restarted on different numbers of cores. Error bars are only visible for 1152 cores.

scaling" is approximately ideal. After 1000 cores, over 90% of ideal scaling is still achieved. The calculations used about 1.5×10^7 excips and had about 8×10^6 occupied states/excitors. The scaling depends upon the effect of load-balancing and the ratio of calculation to communication time, both of which reduce efficiency as the number of occupied excitors per core decreases. For calculations with over 10^4 excitors per core, we find no loss of computational efficiency upon parallelization. As system size increases, the number of excitors grows polymonially, so in this "weak scaling" regime, the algorithm displays perfect parallelization, over 1000 cores can be employed for sufficient calculation size.

B. Parallelization bias

The parallel CCMC algorithm can produce a biased estimate of the energy as not all excitors can form clusters with *all*

available excitors in the spawn step, and so a subspace is sampled each iteration. If the number of MPI processes is 1, then the complete CC space is sampled each iteration and there is no bias. Conversely if more than one MPI process is employed and $\nu_{\text{move}} = \infty$, then Eq. (10) reduces to Eq. (8) and clearly the CC wavefunction cannot be sampled as excitors are fixed on specific MPI processes and hence clusters involving excitors on different processes can never be sampled. Whilst the dominant factor controlling the accessible subspace of clusters per iteration is the number of MPI processes, the time step, $\delta\tau$, and (log of the) move frequency, ν_{move}, are also important, as decreasing either amounts to increasing the available subspace per unit of imaginary time.

To demonstrate these effects, we have evaluated the correlation between the locations of excitors on a trial system of 3813 determinants. The probability that a single excitor is on
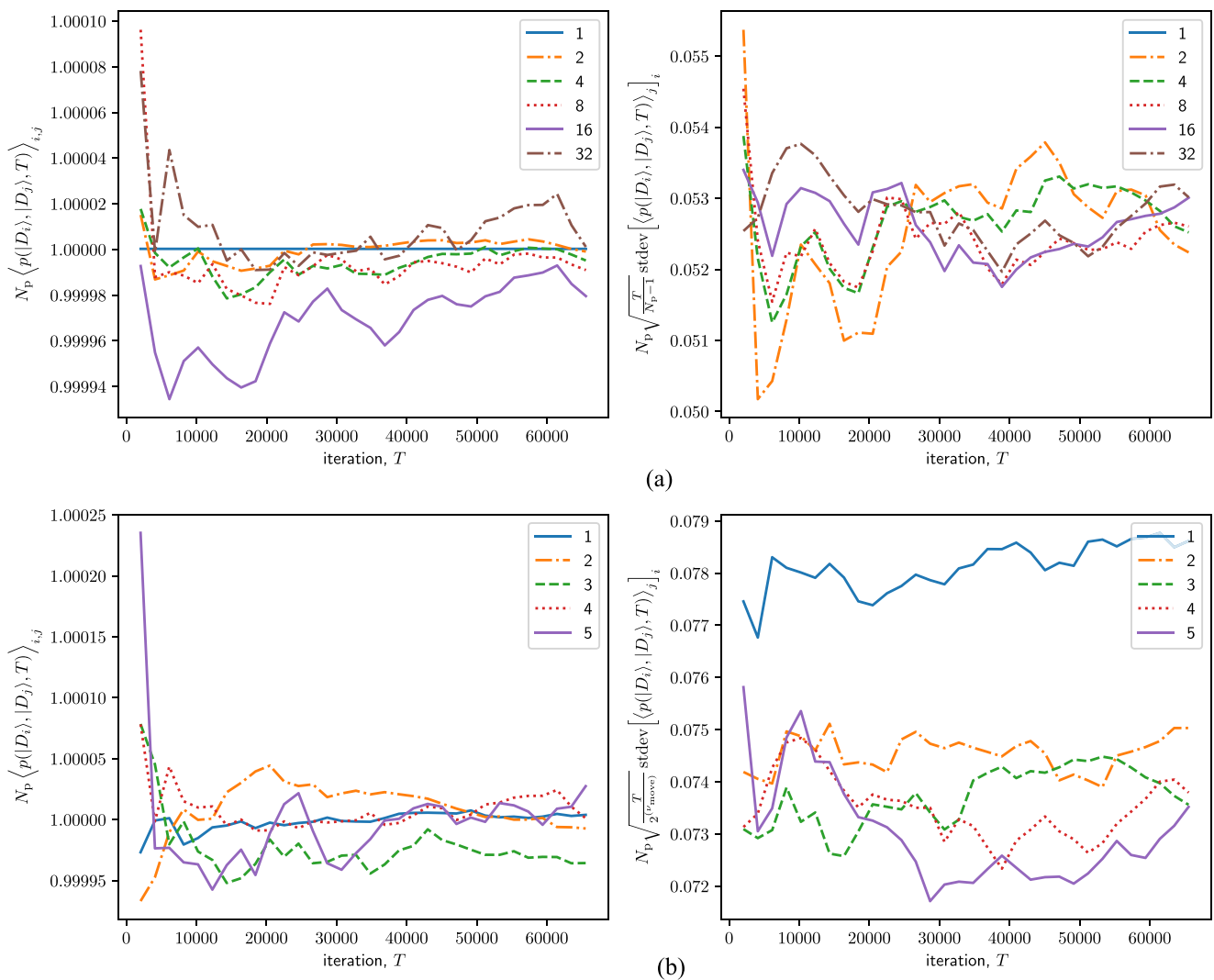

FIG. 2. Analysis of the co-distribution of excitors for different N_p and ν_{move} for a UEG system of 14 *e* in 186 spin-orbitals, consisting of 3813 excitors. $p(|D_i\rangle, |D_j\rangle, T)$ denotes the probability that excitors on $|D_i\rangle$ and $|D_j\rangle$ were on the same processor, averaging over all iterations up to T. (a) For $\nu_{\text{move}} = 4$, (*left*) the mean probability of two excitors sharing the same processor (rescaled by $1/N_p{}^{-1}$ such that the exactly correct value is unity); and (*right*) the standard deviation of the same probability (rescaled by $N_p\sqrt{\frac{T}{N_p-1}}$ to show the scaling behavior). The different lines correspond to different numbers of processors, N_p, given in the legend. (b) For $N_p = 32$, (*left*) the mean probability of two excitors sharing the same processor (rescaled by $1/N_p{}^{-1}$ such that the exactly correct value is unity); and (*right*) the standard deviation of the same probability (rescaled by $N_p\sqrt{\frac{T}{2^{\nu_{\text{move}}}}}$ to show the scaling behavior). The different lines correspond to different move frequencies, ν_{move}, given in the legend.

a specific processor may be regarded as a random event with probability $\frac{1}{N_p}$, and the long-time distribution of such events is expected therefore to follow a binomial distribution with this probability, giving an unbiased mean of $\frac{1}{N_p}$ and a variance which therefore scales with $\frac{N_p-1}{N_p^2}$. Similarly, a move frequency of ν_{move} moves an excitor's processor every $2^{\nu_{move}}$ iterations so decreases the variance by a factor of $2^{\nu_{move}}$. Both of these scaling effects translate directly to the correlated probability of two excitor locations. Figure 2 shows no notable bias in the time-averaged mean probability of two excitors coinciding and a standard deviation following the above scaling relationships. Any bias present due to the instantaneous probability distribution of excitors can therefore be reduced by decreasing N_p and ν_{move}.

As a test system to see a bias, we consider the three-dimensional uniform electron gas (3D UEG),[69–71] which conveniently allows for an easily adjustable Hilbert space and degree of correlation. Specifically, we calculate the CCSDT energy of the 14-e 3D UEG with 66 plane-wave spin-orbitals at $r_s = 0.5a_0$ and $r_s = 5a_0$, for which parallelization unbiased results using solely OpenMP parallelization are available.[14] The full non-composite cluster selection algorithm (Sec. IV B) was used to aid convergence. A discrepancy with a magnitude of 0.01 eV/electron is similar in magnitude to chemical accuracy[14,35,72] and represents an upper bound on any bias, which would preferably be negligible. Note that this is far from a production-level calculation: the CISDT Hilbert space for this system contains only 22 969 determinants.

Figure 3(a) shows the dependence of the bias in the CCSDT projected energy as a function of the number of MPI processes for the UEG. The bias increases with the number of MPI processes and is larger for $r_s = 5a_0$ than $r_s = 0.5a_0$. At 240 MPI processes, each MPI process has fewer than 100 excitors (assuming perfect load balancing) and so the subspace spanned by each MPI process at any given time step is very small. The degree of correlation is also important: 17% of excips are on the reference for $r_s = 0.5a_0$ compared to just 2% for $r_s = 5a_0$. As such, the relative importance of products of clusters increases with correlation. We have also performed a hybrid MPI-OpenMP calculation for $r_s = 5a_0$ using 20 MPI processes with 12 OpenMP threads per process for $r_s = 5a_0$ which agrees within error bars to the expected result and to the corresponding 20 MPI process calculation.

The bias can be reduced by decreasing the imaginary time a given subspace is sampled. Figures 3(b) and 3(c) show reducing the time step and move frequency, respectively, reduces the bias in the CCSDT energy using 240 MPI processes for the 3D 14-e UEG system. The bias remains smaller for the smaller values of r_s value with otherwise identical parameters.

We wish to emphasise this is a contrived setup to show that it is possible to obtain biased results in extreme parameter ranges.

Even ignoring computational and parallel efficiency, the small number of excitors per processor results in poor sampling of the cluster expansion and results in a biased sampling of the coupled cluster wavefunction. We typically set the number of MPI processes such that each process contains at least $\mathcal{O}(10^5)$
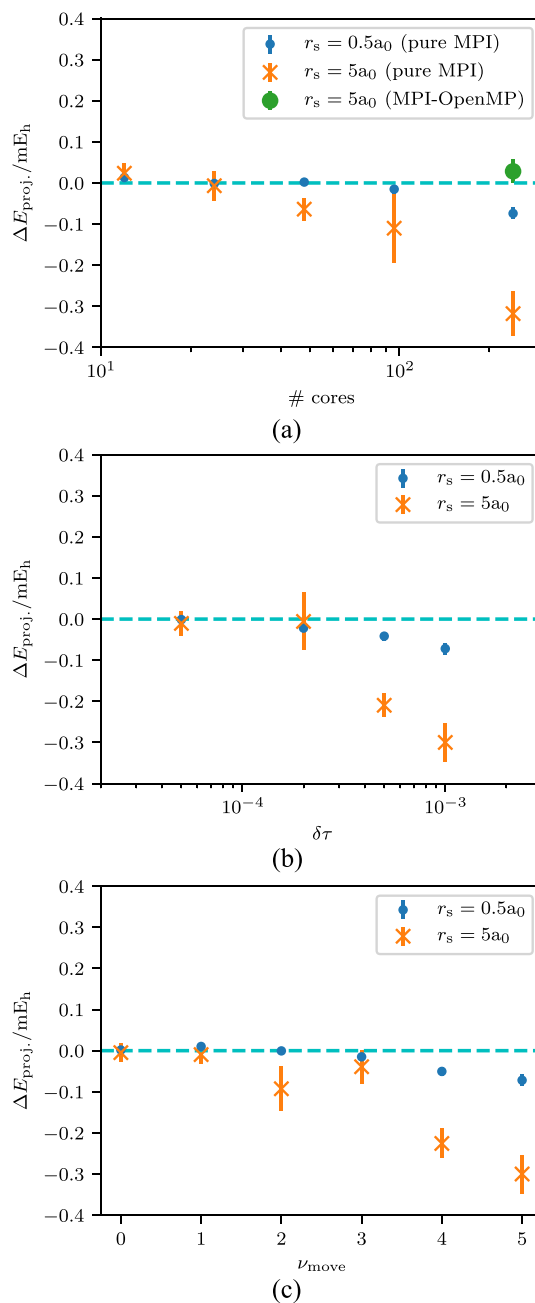

FIG. 3. The effects of the number of MPI processes, $\delta\tau$, ν_{move} on the deviation of the CCSDT projected energy from the unbiased value[14] for the 14-e 3D UEG with 66 spin-orbitals. An accuracy of ± 0.01 eV/electron corresponds to 5 mE$_h$ here as we have 14 electrons. This is outside of the range shown. A horizontal line at zero error is shown to guide the eye. (a) The bias in the CCSDT energy as a function of the number of cores using $\nu_{move} = 5$ and $\delta\tau = 0.001$. The MPI-OpenMP calculation used 12 OpenMP threads per MPI process. (b) The bias in the CCSDT energy as a function of $\delta\tau$ using 240 MPI processes and $\nu_{move} = 5$. (c) The bias in the CCSDT energy as a function of ν_{move} using 240 MPI processes and $\delta\tau = 0.001$.

excitors.[73] In addition to improved parallel efficiency, hybrid MPI-OpenMP parallelization greatly helps with this issue. As a demonstration of non-trivial calculations, we consider a larger UEG system and the water dimer.

The CISDTQ Hilbert space size of the 14-e UEG in a basis of 358 spin-orbitals is $2.6(1) \times 10^8$. Using 96 MPI processes (pure MPI parallelisation) gave an estimate of the CCSDTQ correlation energy at $r_s = 1a_0$ to be $-0.518\,75(7)$ E$_h$; using 8

MPI processes each with 12 OpenMP threads (i.e., the same total resources) gave an estimate of $-0.518\,66(7)\,E_h$. A previous study exploiting only OpenMP parallelization found the correlation energy to be $-0.518\,56(7)\,E_h$.[14] The hybrid calculation agrees with the previous result within 2 standard errors (individual standard errors added in quadrature), whereas the pure MPI calculation is close but does not agree within 2 standard errors. However, while the Hilbert space is of the order of 3×10^8, the number of occupied excitors relevant to the calculation is only about 7×10^6–8×10^6. This means that in the pure MPI case, significantly less than 10^5 excitors are on the same MPI process. We have used even selection[49] for this CCSDTQ calculation.

The CCSDT energy of the H_2O dimer at its CCSDTQ optimized geometry obtained by Lane[74] in the jun-cc-pVDZ basis set[75] is compared to deterministic results calculated using MRCC[28,76] in Table I. The CCMC calculation employed the heat bath excitation generator[77] with slight modifications[50] and was run on 32 MPI processes each using 12 OpenMP threads. The water dimer in the jun-cc-pVDZ basis has a Hilbert space of 1.16×10^7 at the CCSDT level. The stochastic wavefunction contained $\approx 7.9 \times 10^6$ excitors, resulting in $\approx 5.3 \times 10^4$ excitors per MPI process, assuming perfect load balancing. Both CCSDT results agree well with each other. The CCMC CCSDT result is resolvably different from the coupled-cluster single double (CCSD) and CCSDTQ energy within error bars, and no bias is visible.

To demonstrate the capabilities and indicate the future possibilities of parallelized CCMC, we have also studied a system of three water molecules separated at a large distance at the CCSDTQ level in a cc-pVDZ basis.[78] Using three molecules that are—for all practical purposes—infinitely separated has the advantage that the total energy can be calculated by other means, as three times the energy of a single water molecule at CCSDTQ. This calculation from individual molecules is necessary as the deterministic calculation has proven too computationally expensive to be performed in a reasonable time.[79]

The calculation was restarted from a CCSDT QMC calculation and run with 400 MPI processes using 12 OpenMP threads each for part of the calculation that we are analyzing. Even selection[49] and the heat bath uniform singles[77] excitation generator[80] have been used. In the CCMC calculation, the projected energy oscillates in the range -0.6507 to $-0.6524\,E_h$. The true correlation energy, as found by using MRCC[28] scaling up from one molecule, is $-0.6515\,E_h$ which is included

at around the middle of our range. The CCSDT correlation energy, found by the same method, is $-0.6501\,E_h$, which is outside of the quoted range for CCSDTQ with CCMC. To give more than merely a range or to give a smaller, more certain, range, the calculation would have to be run for longer. However, the intent of this study is not to find a known CCSDTQ value but to act as a demonstration that (parallelized) CCMC can give coupled cluster energies that are not feasible with deterministic coupled cluster codes. Current developments of CCMC will enable more precise large calculations in the future.

IV. IMPROVED STOCHASTIC SAMPLING

The efficiency of a stochastic coupled cluster calculation is highly dependent upon the algorithm used to sample the various steps within the algorithm. The original implementation used a simple and easy-to-implement algorithm for selecting clusters from which to spawn. This has proved to be increasingly inefficient as system size increases. Scott and Thom have shown that an "even-selection" algorithm which selects clusters with probabilities more closely corresponding to their amplitude dramatically increases the stability of calculations.[49] In this section, we describe some alternative approaches we have explored to improve the stochastic sampling[81] and first consider what metric we may use to measure this.

A. Shoulder heights

The stability of a stochastic coupled cluster calculation can be determined by whether its population has overcome a plateau or shoulder in its dynamics,[53] where the rate of total particle growth (with imaginary time) slows or stops for some time while the correct wavefunction is evolved. A plateau is commonly visible in FCIQMC calculations whereas CCMC calculations typically only contain a shoulder in the total population growth. After this point, a calculation emerges with a stable growth rate of both the total and reference particle populations. These factors are conveniently described on a shoulder plot,[11] which show a maximum in the ratio of total and reference populations at the shoulder. The population at the shoulder is indicative of the relative difficulty of a calculation and is affected by the parameters used to run it, and a best estimate is given with a low initial reference population and small time step; using larger time steps causes the shoulder population to increase. As the population on the reference also dictates the normalization of the calculation, too low an initial population leads to unstable calculations which do not experience a shoulder and merely "blow up." Examples of shoulder plots are given in Fig. 4.

In a stochastic coupled cluster calculation, the number of particles spawned at any step is directly proportional to the time step, and after a certain threshold, larger time steps will give a higher shoulder population. To therefore determine the effects of any algorithmic changes, we have used the parameters which give the lowest estimate of the shoulder, namely, a very small time step, and reducing the initial population, following a study in Ref. 49 which compared selection algorithms with appropriate time steps and initial populations.

TABLE I. Total energy $E_{\text{tot.}}$ and the correlation energy $E_{\text{corr.}}$ of the H_2O dimer in the jun-cc-pVDZ basis set using Hartree–Fock, deterministic coupled cluster from CCSD to CCSDTQ and CCSDT with CCMC using 32 MPI processes threaded into 12 OpenMP threads each. Even selection[49] has been used. The units are hartrees.

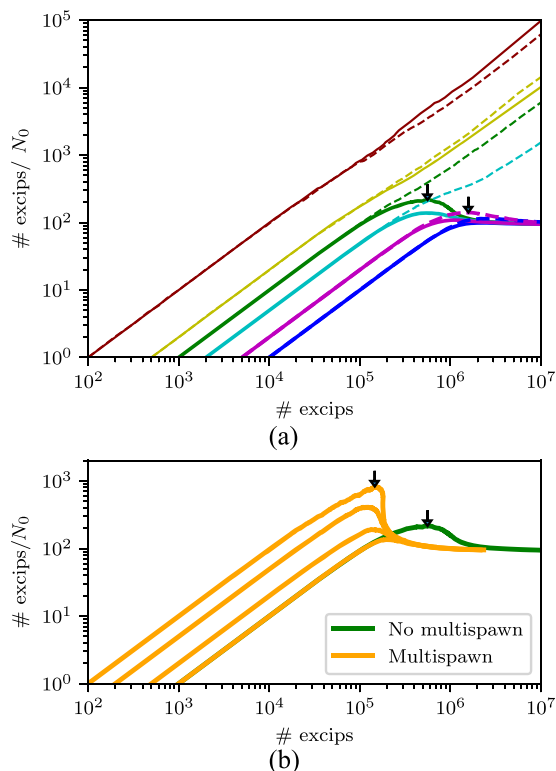| Method	$E_{\text{tot.}}/E_h$	$E_{\text{corr.}}/E_h$
Hartree–Fock	$-152.080\,419\,5$	0
CCSD deterministic	$-152.515\,827\,2$	$-0.435\,407\,682$
CCSDT deterministic	$-152.524\,119\,2$	$-0.443\,699\,666$
CCSDT QMC 12×32 cores		$-0.443\,69(7)$
CCSDTQ deterministic	$-152.525\,116\,4$	$-0.444\,696\,884$

FIG. 4. Shoulder plots for frozen-core CCSD on benzene in a 6-31G basis using the original (dashed) and full non-composite (solid) algorithm. The initial population can be read from the intercept with the abscissa. (*top*) The arrows indicate the best estimate of the shoulder, showing that a full non-composite calculation has lower shoulder (5.6×10^5) than the original algorithm (1.6×10^6). (*bottom*) Multispawn, using $A_{thresh} = 1$, lowering the shoulder to 1.5×10^5, showing stability for many fewer initial particles. Calculations used a time step of 5×10^{-4} and the renormalized excitation generators. $r_{CH} = 1.084$ Å and $r_{CC} = 1.397$ Å exploiting D_{2h} symmetry.

B. Non-composite clusters

We first look at reducing the noise due to stochastic sampling of the cluster expansion. The total number of cluster selections to be made has previously been chosen to equal the total amplitude of excips, in analogy with FCIQMC, where each discrete psip individually undergoes spawning and death events. In stochastic coupled cluster, clusters built up from single excips as well as multiple excips (known as composite clusters) are sampled, so the single-excip (non-composite) clusters have fewer samples taken than the total number of excips. The sampling of (rather than explicit iteration through) the non-composite clusters proves to be an additional source of stochastic noise. We therefore introduce a modification to the algorithm, *full non-composite* sampling, which explicitly iterates through the list of excips, performing spawning and death events on these individually. It is still necessary to sample the composite clusters, and these are sampled with the same number of samples as total excips. The effect of this sampling change is to reduce the number of particles at the shoulder, as shown in Fig. 4, and so calculations require fewer excips to be stable. Even though the computational effort increases since we are doing twice as many cluster selections, the number of minimum excips required is reduced by more than a factor of two when using the full non-composite cluster algorithm as shown by

the shoulder positions in the *top* part of Fig. 4. Figure 2 in Ref. 49 shows that at higher time steps, the memory cost as a function of number of attempts per unit imaginary time is lower when using the full non-composite algorithm compared to the original algorithm.

C. Multiple spawning events

The small time step regime reduces the plateau height because at sufficiently small time step all spawning attempts produce no more than one particle—there are no "blooms," in which a single (rare) spawning event creates a large number of new particles on the same excitor. Such rare events are undesirable both because of the inefficient exploration of the space (especially if the spawned particles do not have the same sign as the ground-state wavefunction) and the impact on population control. Unfortunately the computational efficiency of using such small time steps is low, as the majority of spawning attempts produce no particles at all. Once a cluster has been selected and collapsed to produce determinant j, the number of spawned particles depends on the amplitude of this cluster from which they are spawned. As this is selected stochastically, the amplitude is unbiased by dividing by the probability of cluster selection to give an effective amplitude, $A(j)$. The probability of spawning a particle from this cluster to determinant i is then given by

$$p_{spawn}(i \leftarrow j) = \delta\tau \frac{|A(j)H_{ij}|}{p_{gen}(i|j)}. \quad (11)$$

It is therefore possible to use the amplitude $A(j)$ to decide on the number of spawning attempts from that cluster. A larger number of attempts to spawn will proportionately increase $p_{gen}(i|j)$, the probability that spawning onto i is attempted from the excips on j, and consequently decrease the spawning probability, reducing any blooms. The choice of when to change the number of spawning attempts should depend on $A(j)$, as merely keeping it at a constant, say n, would have a similar effect as using a time step $\delta\tau/n$ and not account for the impact of rare events from clusters with large amplitudes. We have chosen to introduce a threshold A_{thresh}, such that the number of attempts is given by

$$n_{attempts} = \max\left(1, \left\lfloor \frac{A(j)}{A_{thresh}} \right\rfloor\right). \quad (12)$$

The effect of these multiple spawning changes is shown in the lower panel of Fig. 4, and in combination with the full non-composite sampling, shows a significant reduction in shoulder heights and a corresponding decrease in memory requirements for calculations. We finish with a note of caution when such sampling is used in combination with large-scale parallelization. If the cluster sampling is such that there are occasional amplitudes significantly larger in magnitude than A_{thresh}, a correspondingly large number of spawning attempts are made. In some calculations, $n_{attempts}$ can be occasionally of the order of 10^6. Should such events be unevenly distributed over processors, it can lead to a significant load-imbalance and reduction of parallel efficiency. In practice, we have only seen this effect when the number of excitors per process is under 10^5. In general, such effects are averted by the use of even selection[49] which ensures $|A(j)|$ takes an approximately a constant value

for all clusters. Even selection was used for the scaling plot in Fig. 1.

V. DISCUSSION

Overall we have shown that, despite the non-linearity of the coupled cluster ansatz, by introducing a stochastic algorithm, it is possible to perform massively parallel calculations at arbitrary orders of coupled cluster theory with great parallel efficiency and approximately ideal strong scaling up to 500 cores. Our parallelization scheme exploits the stochastic nature of the algorithm to sample combinations of excitors averaged over multiple cycles, and we have shown that the bias introduced can be made minimal (provided the number of MPI processes and other parameters are chosen sensibly), well below the intrinsic accuracy of the calculations themselves. Furthermore, the bias can be systematically reduced, and so confidence can be had in its magnitude and as systems studied become larger, the parallelization bias becomes smaller.

We contrast our parallelization scheme to that of the Cyclops tensor framework,[30] which requires explicit knowledge of the sparsity within the tensors of excitor amplitudes and performs deterministic CCSD and CCSDT calculations. While such a scheme produces numerically precise results, it cannot easily take advantage of the natural sparsity of amplitudes within excitation space so requires significantly more storage for amplitudes. While the polynomial scaling of the number of amplitudes with the system size allows a good weak scaling behavior, the requirement to communicate all of these results in relatively poor strong scaling behavior. Efforts are underway[82] to redesign deterministic algorithms using a dataflow paradigm; however, these require significant manual reorganisations of the code which appears infeasible for higher levels of coupled cluster theory.

We have also shown that the stochastic sampling can be improved using the *full non-composite* or the *multi-spawn* additions. The later *even selection* sampling[49] was inspired by and was built on top of the *full non-composite* algorithm and when using multiple MPI processes is more efficient than *multi-spawn* sampling which was a first step on the way to improving shoulder heights.

We close by noting that the approach described here is not restricted to just coupled cluster; rather the idea of sampling both the action of the Hamiltonian and the wavefunction ansatz is applicable to many other methods in quantum chemistry.

[1]R. J. Bartlett and M. Musiał, Rev. Mod. Phys. **79**, 291 (2007).

[2]K. Kowalski and P. Piecuch, J. Chem. Phys. **113**, 18 (2000).

[3]M. E. Harding, J. Vázquez, B. Ruscic, A. K. Wilson, J. Gauss, and J. F. Stanton, J. Chem. Phys. **128**, 114111 (2008).

[4]C. Hättig, W. Klopper, A. Köhn, and D. P. Tew, Chem. Rev. **112**, 4 (2012).

[5]A. Karton, Wiley Interdiscip. Rev.: Comput. Mol. Sci. **6**, 292 (2016).

[6]D. L. Freeman, Phys. Rev. B **15**, 5512 (1977).

[7]R. F. Bishop and K. H. Lührmann, Phys. Rev. B **17**, 3757 (1978).

[8]R. F. Bishop and K. H. Lührmann, Phys. Rev. B **26**, 5523 (1982).

[9]J. J. Shepherd and A. Grüneis, Phys. Rev. Lett. **110**, 226401 (2013); e-print arXiv:1310.6059.

[10]A. Roggero, A. Mukherjee, and F. Pederiva, Phys. Rev. B **88**, 115138 (2013); e-print arXiv:1304.1549.

[11]J. S. Spencer and A. J. W. Thom, J. Chem. Phys. **144**, 084108 (2016); e-print arXiv:1511.05752.

[12]J. McClain, J. Lischner, T. Watson, D. A. Matthews, E. Ronca, S. G. Louie, T. C. Berkelbach, and G. K.-L. Chan, Phys. Rev. B **93**, 235139 (2016); e-print arXiv:1512.04556.

[13]J. J. Shepherd, J. Chem. Phys. **145**, 031104 (2016); e-print arXiv:1605.05699.

[14]V. A. Neufeld and A. J. W. Thom, J. Chem. Phys. **147**, 194105 (2017); e-print arXiv:1706.09923.

[15]S. Hirata, I. Grabowski, M. Tobita, and R. J. Bartlett, Chem. Phys. Lett. **345**, 475 (2001).

[16]S. Hirata, R. Podeszwa, M. Tobita, and R. J. Bartlett, J. Chem. Phys. **120**, 2581 (2004).

[17]F. R. Manby, D. Alfè, and M. J. Gillan, Phys. Chem. Chem. Phys. **8**, 5178 (2006).

[18]S. J. Nolan, M. J. Gillan, D. Alfè, N. L. Allan, and F. R. Manby, Phys. Rev. B **80**, 165109 (2009).

[19]A. Grüneis, G. H. Booth, M. Marsman, J. Spencer, A. Alavi, and G. Kresse, J. Chem. Theory Comput. **7**, 2780 (2011).

[20]G. H. Booth, A. Grüneis, G. Kresse, and A. Alavi, Nature **493**, 365 (2013).

[21]A. Grüneis, J. Chem. Phys. **143**, 102817 (2015).

[22]K. Liao and A. Grüneis, J. Chem. Phys. **145**, 141102 (2016).

[23]P. Schwerdtfeger, R. Tonner, G. E. Moyano, and E. Pahl, Angew. Chem., Int. Ed. **55**, 12200 (2016).

[24]J. McClain, Q. Sun, G. K.-L. Chan, and T. C. Berkelbach, J. Chem. Theory Comput. **13**, 1209 (2017).

[25]T. Gruber, K. Liao, T. Tsatsoulis, F. Hummel, and A. Grüneis, Phys. Rev. X **8**, 021043 (2018).

[26]K. Raghavachari, G. W. Trucks, J. A. Pople, and M. Head-Gordon, Chem. Phys. Lett. **157**, 479 (1989).

[27]T. J. Lee and G. E. Scuseria, *Quantum Mechanical Electronic Structure Calculations with Chemical Accuracy* (Springer, The Netherlands, Dordrecht, 1995), pp. 47–108.

[28]MRCC, a quantum chemical program suite, written by M. Kllay, Z. Rolik, J. Csontos, P. Nagy, G. Samu, D. Mester, J. Cska, B. Szab, I. Ladjnszki, L. Szegedy, B. Ladczki, K. Petrov, M. Farkas, P. D. Mezei, and B. Hégel, see http://www.mrcc.hu.

[29]E. Agullo, J. Demmel, J. Dongarra, B. Hadri, J. Kurzak, J. Langou, H. Ltaief, P. Luszczek, and S. Tomov, J. Phys.: Conf. Ser. **180**, 012037 (2009).

[30]E. Solomonik, D. Matthews, J. R. Hammond, J. F. Stanton, and J. Demmel, J. Parallel Distrib. Comput. **74**, 3176 (2014).

[31]N. Flocke and R. J. Bartlett, J. Chem. Phys. **121**, 10935 (2004).

[32]M. Ziółkowski, B. Jansík, T. Kjærgaard, and P. Jørgensen, J. Chem. Phys. **133**, 014107 (2010).

[33]C. Riplinger and F. Neese, J. Chem. Phys. **138**, 034106 (2013).

[34]C. Riplinger, B. Sandhoefer, A. Hansen, and F. Neese, J. Chem. Phys. **139**, 134101 (2013).

[35]W. M. C. Foulkes, L. Mitas, R. J. Needs, and G. Rajagopal, Rev. Mod. Phys. **73**, 33 (2001).

[36]A. J. W. Thom and A. Alavi, Phys. Rev. Lett. **99**, 143001 (2007).

[37]S. Y. Willow, K. S. Kim, and S. Hirata, J. Chem. Phys. **137**, 204122 (2012).

[38]S. Y. Willow, M. R. Hermes, K. S. Kim, and S. Hirata, J. Chem. Theory Comput. **9**, 4396 (2013).

[39]R. Baer, D. Neuhauser, and E. Rabani, Phys. Rev. Lett. **111**, 106402 (2013).

[40]Our implementation does not require canonical orbitals, so we have the full freedom to choose any transformation of the single particle orbitals, for example the study in Ref. 50.

[41]G. H. Booth, A. J. W. Thom, and A. Alavi, J. Chem. Phys. **131**, 054106 (2009).

[42]D. Cleland, G. H. Booth, C. Overy, and A. Alavi, J. Chem. Theory Comput. **8**, 4138 (2012).

[43]J. J. Shepherd, G. Booth, A. Grüneis, and A. Alavi, Phys. Rev. B **85**, 081103 (2012).

[44]C. Overy, G. H. Booth, N. S. Blunt, J. J. Shepherd, D. Cleland, and A. Alavi, J. Chem. Phys. **141**, 244117 (2014).

[45]N. S. Blunt, S. D. Smart, G. H. Booth, and A. Alavi, J. Chem. Phys. **143**, 134117 (2015).

[46]N. S. Blunt, G. H. Booth, and A. Alavi, J. Chem. Phys. **146**, 244105 (2017).

[47]A. J. W. Thom, Phys. Rev. Lett. **105**, 263004 (2010).

[48]R. S. T. Franklin, J. S. Spencer, A. Zoccante, and A. J. W. Thom, J. Chem. Phys. **144**, 044111 (2016).

[49]C. J. C. Scott and A. J. W. Thom, J. Chem. Phys. **147**, 124105 (2017); e-print arXiv:1706.07017.

[50]V. A. Neufeld and A. J. W. Thom, "Exciting determinants in Quantum Monte Carlo: Loading the dice with fast, low memory weights," J. Chem. Theory Comput. (published online).

[51]J. E. Deustua, J. Shen, and P. Piecuch, Phys. Rev. Lett. **119**, 223003 (2017).

[52]G. H. Booth, S. D. Smart, and A. Alavi, Mol. Phys. **112**, 1855 (2014).

[53]J. S. Spencer, N. S. Blunt, and W. M. Foulkes, J. Chem. Phys. **136**, 054110 (2012).

[54]F. R. Petruzielo, A. A. Holmes, H. J. Changlani, M. P. Nightingale, and C. J. Umrigar, Phys. Rev. Lett. **109**, 230201 (2012).

[55]We set $p_{\text{size}}(s) = 1 - \sum_{i=0}^{s-1} p_{\text{size}}(i)$ in order to ensure a normalised probability.

[56]C. J. Umrigar, M. P. Nightingale, and K. J. Runge, J. Chem. Phys. **99**, 2865 (1993).

[57]J. M. Turney, A. C. Simmonett, R. M. Parrish, E. G. Hohenstein, F. Evangelista, J. T. Fermann, B. J. Mintz, L. A. Burns, J. J. Wilke, M. L. Abrams, N. J. Russ, M. L. Leininger, C. L. Janssen, E. T. Seidl, W. D. Allen, H. F. Schaefer, R. A. King, E. F. Valeev, C. D. Sherrill, and T. D. Crawford, Wiley Interdiscip. Rev.: Comput. Mol. Sci. **2**, 556 (2012).

[58]Q. Sun, T. C. Berkelbach, N. S. Blunt, G. H. Booth, S. Guo, Z. Li, J. Liu, J. D. McClain, E. R. Sayfutyarova, S. Sharma, S. Wouters, and G. K.-L. Chan, Wiley Interdiscip. Rev.: Comput. Mol. Sci. **8**, e1340 (2018).

[59]J. M. Foster and S. F. Boys, Rev. Mod. Phys. **32**, 300 (1960).

[60]H. Flyvbjerg and H. G. Petersen, J. Chem. Phys. **91**, 461 (1989).

[61]W. A. Vigor, J. S. Spencer, M. J. Bearpark, and A. J. W. Thom, J. Chem. Phys. **142**, 104101 (2015); e-print arXiv:1407.1753.

[62]T. E. Oliphant, *Guide to NumPy*, 2nd ed. (CreateSpace Independent Publishing Platform, USA, 2015).

[63]W. McKinney, in *Proceedings of the 9th Python in Science Conference*, edited by S. van der Walt and J. Millman (SciPy, 2010), pp. 51–56, available at http://conference.scipy.org/proceedings/scipy2010/mckinney.html.

[64]See https://github.com/jsspencer/pyblock for code.

[65]J. D. Hunter, Comput. Sci. Eng. **9**, 90 (2007).

[66]See https://doi.org/10.5281/zenodo.592845, https://github.com/mwaskom/seaborn for code.

[67]Booth *et al.* used a custom hash function based upon the list of occupied orbitals. We find hashing the bit string representation of the determinant simpler and computationally more efficient, whilst giving at least as good distribution over processors when a hash function of sufficient quality is used, and therefore use the MurmurHash2 function.

[68]A node may consist of a single processor or multiple processors. Within the MPI paradigm, we distribute over MPI ranks, where each rank contains one or more threads.

[69]R. M. Martin, *Electronic Structure* (Cambridge University Press, Cambridge, 2004), pp. 100–118.

[70]G. Giuliani and G. Vignale, *Quantum Theory of the Electron Liquid* (Cambridge University Press, Cambridge, 2005), pp. 1–68.

[71]P.-F. Loos and P. M. W. Gill, Wiley Interdiscip. Rev.: Comput. Mol. Sci. **6**, 410 (2016); e-print arXiv:1601.03544.

[72]L. K. Wagner and D. M. Ceperley, Rep. Prog. Phys. **79**, 094501 (2016); e-print arXiv:1602.01344.

[73]We note that available computational resources rarely grow polynomially with system size!.

[74]J. R. Lane, J. Chem. Theory Comput. **9**, 316 (2013).

[75]E. Papajak and D. G. Truhlar, J. Chem. Theory Comput. **7**, 10 (2011).

[76]Z. Rolik, L. Szegedy, I. Ladjnszki, B. Ladczki, and M. Kllay, J. Chem. Phys. **139**, 094105 (2013).

[77]A. A. Holmes, H. J. Changlani, and C. J. Umrigar, J. Chem. Theory Comput. **12**, 1561 (2016).

[78]T. H. Dunning, J. Chem. Phys. **90**, 1007 (1989).

[79]Using MRCC,[28] a single iteration was not completed within a week on a 32-core node on the Cambridge Service for Data Driven Discovery (CSD3) operated by the University of Cambridge Research Computing Service (http://www.csd3.cam.ac.uk/). This provides a lower estimate of 2 node-months, or 46000 core hours.

[80]Terminology from Ref. 50 has been used. Single excitations were sampled uniformly, double excitations with the heat bath excitation generator. Idea by and mentioned in Holmes *et al.*[77]

[81]These approaches were conceived and implemented some time before the work in Ref. 49, and briefly referred to in the same, and are reported in fuller detail now for completeness.

[82]H. Jagode, A. Danalis, and J. Dongarra, Int. High Perform. Comput. Appl. **32**, 540 (2018).