Semistochastic importance sampling of second-quantized operators in determinant space: application to multireference perturbation theory

Adam A Holmes and Edward F Valeev

July 5, 2021

Abstract

We present an algorithm for semistochastically applying a second-quantized operator to a Slater determinant using importance sampling. Our method efficiently applies the largest-magnitude terms deterministically, and enables sampling of the remaining, smaller terms, with probability proportional to a function of the terms' magnitudes. The time complexity of the deterministic component scales as only the number of large-magnitude terms (rather than all terms), as in the heat-bath configuration interaction (HCI) paper, and the time complexity of each sample in the stochastic component scales only as the logarithm of the number of orbitals. We then use our algorithm to perform efficient, semistochastic multireference perturbation theory in semistochastic heat-bath configuration interaction (SHCI).

1 Introduction

Many quantum chemistry algorithms operate in the space of Slater determinants. These methods include deterministic algorithms such as selected configuration interaction (e.g., HCI), stochastic/semistochastic algorithms using classical statistics such as semistochastic multireference perturbation theory (e.g., in SHCI), and projector Monte Carlo algorithms such as full configuration interaction quantum Monte Carlo (FCIQMC).

In all of these determinant-space algorithms, one of the key steps is applying a second-quantized operator (usually the Hamiltonian) to a Slater determinant (or linear combination of them):

- In selected CI, a variational wavefunction is obtained iteratively by 'selecting' new determinants each iteration using a criterion that is a function of the Hamiltonian times the previous iteration's wavefunction. Once the new determinants are selected, their coefficients are variationally optimized using an algorithm such as Davidson, in which the Hamiltonian is diagonalized in a space of Krylov vectors, each of which is a function of the Hamiltonian times other Krylov vectors.

- In semistochastic multireference perturbation theory as used in SHCI, contributions to the perturbation theory expression are computed by applying the Hamiltonian to determinants sampled from the variational Hamiltonian.

- In FCIQMC, the power method is simulated in the full set of Slater determinants by repeatedly stochastically (or semistochastically) applying a projector operator – a linear function of the Hamiltonian – to a stochastic representation of the wavefunction. The energy is estimated using a mixed estimator, which also makes use of the Hamiltonian times a trial wavefunction in determinant space.

Since the application of the Hamiltonian to a Slater determinant is such a key step in all of these algorithms, we explore methods to evaluate it. Broadly, there are three main approaches:

1. **Deterministic:** Simply evaluating all single and double excitations from the initial Slater determinant. Time complexity: $\mathcal{O}(N^2M^2)$, space complexity: $\mathcal{O}(N^2M^2)$, where N is the number of electrons and M (assumed to be $>> N$) is the number of orbitals.

2. **Stochastic:** Sampling excitations according to some distribution. Reduces the time and space complexity, but introduces a stochastic uncertainty (and in projector methods, the fermion sign problem!).

3. **Semistochastic:** Evaluating the largest-magnitude components, and sampling the remaining, smaller components. Has a much reduced time and space complexity relative to the deterministic approach, but with a greatly reduced stochastic uncertainty relative to fully stochastic methods. In projector methods, it also mitigates the bias incurred in taming the fermion sign problem, such as the initiator bias in FCIQMC.

Whether deterministic, stochastic, or semistochastic, we define an *efficient* algorithm for applying a second-quantized operator to a Slater determinant as one which does not spend too much time on components of the operator with small coefficients.

- In the case of deterministic methods, an efficient method would loop over only the large-magnitude terms in the operator, to avoid wasting time on the vast number of small-magnitude terms. HCI, introduced by one of us, made use of this approach to efficiently select new determinants in Selected CI.

- In the case of stochastic methods (or the stochastic component of semistochastic methods), rather than skipping the small-magnitude terms entirely, an efficient approach simply means not over-sampling small-magnitude terms. That is, the small-magnitude terms are importance-sampled, with probability a function of their magnitude. [1]

In this paper, we describe a unified approach to semistochastic importance sampling. Our approach extends the deterministic HCI algorithm to semistochastic methods, in a way that (unlike SHCI) is consistent with the underlying philosophy of HCI: we importance-sample the small-magnitude terms that were not already treated deterministically, thus avoiding wasteful over-sampling of these terms entirely. [2] We focus on the nonrelativistic

[1] While a semistochastic extension of HCI (called SHCI) has already been introduced by one of us, it did not have this efficient property; small-magnitude contributions were vastly over-sampled, since the sampling was done according to a distribution that did not take the matrix element magnitude into account.

[2] It should be noted that a crude version of semistochastic importance sampling has been used before in the context of FCIQMC. However, in that case, the deterministic component was pre-computed, and the stochastic component was not disjoint from it: there was some probability of sampling an excitation that already existed in the deterministic component and therefore had to be discarded.

quantum chemistry Hamiltonian, but note that the techniques could easily be generalized to alternative Hamiltonians or other second-quantized operators such as cluster operators.

2 Background

Here we describe the most important building blocks to the semistochastic importance sampling algorithm: deterministic evaluation of the screened sum over large-magnitude terms, and importance sampling of the remaining, small-magnitude terms.

2.1 Deterministic screening

The deterministic component of the algorithm has already been described in the original HCI paper. By storing excitations in sorted order by magnitude, we can generate all excitations exceeding a threshold without wasting any time on excitations lower than the threshold.

In the original HCI paper, only double excitations were treated efficiently, since they are simpler to compute and much more numerous than single excitations. However, in this paper, we also improve upon that method, in that we also pre-compute upper bounds on the single excitation magnitudes, so that they can be efficiently treated as well.

For double excitations, the matrix elements are given by:

$$|H(pq \to rs)| = \begin{cases} |g_{pqrs}|, & \text{opposite spin;} \\ |g_{pqrs} - g_{pqsr}|, & \text{same spin,} \end{cases} \tag{1}$$

and for single excitations, the matrix elements are:

$$|H(p \to r)| = \left| f_{pr} + \sum_{q \in \text{occ}} g_{pqqr} \right|. \tag{2}$$

Unlike double excitations, single excitation matrix element magnitudes depend on the orbital occupancies of the exciting determinant. Since we are interested in storing both types of excitations in a determinant-independent data structure, we compute the following upper bounds to the single excitation magnitudes:

$$\begin{aligned} |H(p \to r)| &= \left| f_{pr} + \sum_{q \in \text{occ}} g_{pqqr} \right| & (3) \\ &\leq \max\left(\left| f_{pr} + \sum_{q \in [N-1 \text{ largest}]} g_{pqqr} \right|, \left| f_{pr} + \sum_{q \in [N-1 \text{ smallest}]} g_{pqqr} \right| \right), & (4) \end{aligned}$$

where the sums in the last line above are over the $N-1$ distinct spin-orbitals $q \notin \{p, r\}$ (of the correct total spin) for which g_{pqqr} is largest or smallest (the values, not the magnitudes), respectively. All of these upper bounds for the single excitations can be trivially precomputed in $\mathcal{O}(M^3 \log M)$ time.

Once all matrix element magnitudes (or upper bounds in the case of single excitations) are computed, we then group them (along with their target orbital(s)) by exciting orbital(s) (i.e., (p, q) for double excitations, p for single excitations), and sort each group in decreasing order by magnitude. Thus, for double excitations, for each exciting orbital pair (p, q), we have a sorted list of all valid excitations $\{[(r, s), |H(pq \to rs)|]\}$ that those orbitals could excite to. Similarly, for single excitations, for each exciting orbital p, we have a list of all corresponding valid excitations $\{[r, |H(p \to r)|_{\max}]\}$ sorted by upper bounds on their magnitudes. This setup phase can be performed in $\mathcal{O}(M^4 \log M)$ time and has a storage requirement of size $\mathcal{O}(M^4)$, same as the two-body integrals.

Then, all excitations whose magnitudes exceed a given threshold ϵ can be computed efficiently as follows: Loop over each electron and each pair of electrons; for each, traverse its corresponding sorted list of candidate excitations. Once an excitation whose magnitude does not exceed ϵ is found, exit the inner for-loop and go to the next electron or electron pair. In the case of single excitations, each candidate excitation's matrix element must be computed to see whether its magnitude exceeds ϵ, since only an upper bound on its magnitude was stored.

Using this algorithm, no time is wasted on the double excitations that do not meet the threshold, but some time must be wasted on computing the magnitudes of candidate single excitations that end up being discarded. However, this algorithm is an improvement compared to the original HCI algorithm, in which all single excitations were considered candidates.

2.2 Discrete sampling

The sampling method presented in this paper makes use of two well-known approaches to discrete sampling: Alias sampling and binary-searching a discrete cumulative distribution function (CDF). We describe them both here.

2.2.1 Alias sampling

The idea of Alias sampling is to sample an arbitrary discrete distribution in two steps. First, sample an element with uniform probability. Then, if a low-probability element was selected, it has some probability of 'aliasing' to another, high-probability element.

For an N-element discrete distribution, Alias sampling requires $\mathcal{O}(N)$ time to generate a sampling data structure of size $\mathcal{O}(N)$, and each sample can be collected in constant time.

2.2.2 CDF searching

A simpler sampling algorithm is binary-searching the CDF. First, sample a real number r between 0 and 1 with uniform probability. Then, binary-search the CDF for r. Specifically, we are looking for the smallest index i for which $r < \text{CDF}_i$.

This sampling algorithm requires $\mathcal{O}(N)$ time and storage to set up (assuming that the CDF was not already known!), and each sample can be collected in $\mathcal{O}(\log N)$ time.

2.2.3 Both sampling algorithms are useful

Most of the time, for values of N that are not trivially small, Alias sampling would be the preferred choice of these two sampling algorithms as it has a lower time complexity per sample.

However, as we will describe below, CDF searching has an intriguing use case: Suppose we want to be able to sample one of the first n elements of a stored discrete distribution, where n is not known ahead of time. In that case, we could modify the sampling algorithm to sample only one of the first n elements by simply multiplying r by CDF_n before binary-searching, so the setup time is negligible! Alias sampling does not have an analogous property, so we need both sampling algorithms to perform the sampling described in this paper.

3 Semistochastic importance sampling algorithm

We are now in position to describe the semistochastic importance sampling algorithm we have developed. Recall that our goal is to construct an algorithm that has two components:

1. Efficiently find all large-magnitude excitations.
2. Sample remaining, small-magnitude excitations, with probability proportional to a function of their magnitudes.

Step 1 was already described in section 2.1. Here we describe step 2.

After performing step 1, each exciting electron or electron pair has a corresponding list of remaining target electrons or electron pairs whose excitation magnitudes are lower than the threshold. We divide the sampling into two stages:

1. Sample an exciting electron or electron pair with probability proportional to the sum of remaining (small-magnitude) targets that it could excite to.
2. Sample a target electron or electron pair with probability proportional to its excitation magnitude.

For sampling step 1, we create an Alias sampling data structure during the deterministic step. That is, for each exciting electron or electron pair, we iterate over sorted excitation magnitudes, and as soon as the threshold is reached, we record the sum of remaining excitation magnitudes as that electron or electron pair's relative probability. Of course, if this sum had to be computed on the fly, the total time would then scale as the total number of excitations, defeating the purpose of going beyond a deterministic algorithm in the first place. Therefore, we modify the initial setup routine in which the excitations are sorted, to include not just the matrix element magnitudes, but also the cumulative (in reverse order) sums of remaining matrix element magnitudes for each exciting electron or electron pair. That is, for double excitations, for each exciting orbital pair (p, q), we now store

$$\left\{ \left[(r,s), \quad |H(pq \to rs)|, \quad \sum_{rs \in \text{remaining}} |H(pq \to rs)| \right] \right\}, \tag{5}$$

and the analogous expression for single excitations. These cumulative sums can be precomputed in $\mathcal{O}(M^4)$ time after the sorting step. Once these cumulative sums are precomputed, constructing the Alias distribution takes only $\mathcal{O}(N^2)$ time.

For sampling step 2, we now have to sample a target electron or electron pair, from the set of remaining excitations (i.e., the ones that were not already treated deterministically). We have stored the cumulative sums of all of the excitations starting from the exciting electron or electron pair in the setup stage, and we have identified the index corresponding to the first excitation we want to be able to sample (i.e., the first excitation whose magnitude did not meet the threshold in the deterministic step). As described in section 2.2.3, we can use CDF searching to sample this distribution in $\mathcal{O}(\log M)$ time without any additional time required to set up the distribution.

Thus, the algorithm presented above augments the efficient deterministic screening algorithm described in the original HCI paper by enabling importance sampling of the remaining, screened-out terms that are not treated deterministically, in $\mathcal{O}(\log M)$ time per sample, with negligible overhead to the original (deterministic) algorithm.

3.1 Details

We now flesh out some minor but important details in the algorithm.

3.1.1 Sampling a multideterminant reference

The algorithm described above treats excitations from a single Slater determinant semistochastically. However, the generalization to a multideterminant reference is straightforward: the deterministic component is the same as in the original HCI paper, and the stochastic component can be performed by first sampling a determinant with the correct probability, then sampling an excitation using the single-determinant algorithm above. The correct probability will be a function of the determinant coefficient magnitude, times the sum of remaining excitations that don't meet the deterministic threshold.

3.1.2 Single excitations

Recall that for single excitations, the matrix element magnitudes cannot be known *a priori*, so instead we use upper bounds on their magnitudes. During the deterministic stage, each candidate single excitation matrix element must be computed and compared to the screening threshold. Thus, the remaining single excitations that have not been treated deterministically include candidates that are discarded, in addition to the remaining, untouched single excitations. Double excitations don't require checking and potentially discarding candidates, since their magnitudes are known *a priori*.

Therefore, the algorithm presented above is not correct for single excitations. We therefore give two possible modifications that incorporate single excitations correctly.

1. Treat single excitations fully stochastically. The simplest approach is simply to treat single excitations stochastically, and only use the semistochastic approach for the (much more numerous) double excitations. This can be done by simply using a very large ϵ for the single excitations, so that none of them exceed the threshold.

2. Save the discarded candidate single exciations as an auxiliary sampling distribution. In this case, each stored list of remaining target electrons is augmented by the list of discarded target electrons. This method appears to be complex and it is not obvious how to perform it efficiently.

In addition to the above modifications for correctness, single excitations are still over-sampled due to the fact that their relative probability is given by the maximum matrix element among all exciting determinants that could perform that excitation, rather than the matrix element itself. This over-sampling can be counteracted by estimating the amount of over-sampling (e.g., during the deterministic stage), and reducing the relative probability of single excitations by that factor.

3.1.3 Target orbitals may already be occupied

The semistochastic algorithm depends on the fact that double excitation matrix elements are independent of the other occupied orbitals in the determinant. However, this feature also presents a downside: while the exciting orbitals are always chosen from the set of occupied orbitals, there is no guarantee that the target orbitals are unoccupied. So, the above sampling algorithm will sometimes produce invalid excitations (into already occupied orbitals), whose matrix elements are zero. While the algorithm is still correct, it is not ideal in that the importance sampled probability distribution has an additional delta function centered at zero.

The simplest way to solve this problem is to simply collect new samples (new determinant, exciting orbitals, target orbitals), until a valid excitation is obtained. Since the relative probabilities among the valid excitations are unchanged, this does not introduce a bias.

However, it is not easy to efficiently compute the probability in that case. The fastest I can come up with is $\mathcal{O}(N^3)$ time per determinant.

4 Application to multireference perturbation theory

We now describe the application of our algorithm to multireference Epstein-Nesbet perturbation theory.

1. Compute the deterministic component of the σ-vector, $\sigma^{(\epsilon)} = \sum_i^{(\epsilon)} H_{ai}c_i$, as well as its energy,

$$\Delta E_{\text{determ.}} = \sum_a \sigma_a^{(\epsilon)} \frac{1}{E_0 - E_a} \sigma_a^{(\epsilon)} \tag{6}$$

2. Importance-sample the remaining components of the σ-vector

5 Conclusion and future work

We have presented an efficient, unified approach to treating second-quantized operators semistochastically using importance sampling for the stochastic component. It uses an initial setup stage that takes $\mathcal{O}(M^4 \log M)$ time and creates a data structure of size $\mathcal{O}(M^4)$ (the same scaling as the setup stage in the original HCI algorithm). After the setup, the Hamiltonian (or other second-quantized operator) can be applied semistochastically, by generating all excitations exceeding an arbitrary threshold ϵ, and importance sampling the remaining excitations that don't meet the threshold. The time it takes to do so scales as $\mathcal{O}(N^2 + N_{|H|>\epsilon})$ for the deterministic component, and $\mathcal{O}(\log M)$ time per sample for the stochastic component.

We have used our semistochastic importance sampling algorithm to accelerate multireference Epstein-Nesbet perturbation theory in the context of SHCI.

As there are many determinant-space algorithms in which a key step is efficiently acting on a Slater determinant with a second-quantized operator, we are now in the process of exploring the application of semistochastic importance sampling to other algorithms. In particular, we are exploring its use in FCIQMC, where it will enable a semistochastic projection, in which the division between the deterministic and stochastic components is dynamic, rather than fixed throughout the run as in the original S-FCIQMC paper. We expect that this approach will greatly ameliorate the fermion sign problem and reduce stochastic fluctuations.