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

Adam A Holmes

June 1, 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.

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.

In this paper, we describe a unified approach to semistochastic importance sampling, in which the deterministic component is computed efficiently on the fly, and the stochastic component efficiently samples only the remaining terms left out of the deterministic component. We focus on the nonrelativistic 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:

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

where the sums in the last line above are over the $N-1$ distinct 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 $\{[|H(pq \to rs)|, (r, s)]\}$ that those orbitals could excite to. Similarly, for single excitations, for each exciting orbital p, we have a list of all corresponding valid excitations $\{[|H(p \to r)|_{\max}, r]\}$ 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 $r < \mathrm{CDF}_i$.

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.

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. In that case, the setup time would be negligible! Alias sampling does not have an analogous property.

3 Semistochastic importance sampling algorithm

We are now in position to describe the semistochastic importance sampling algorithm we have developed.