Monte Carlo determination of multiple extremal eigenpairs

T. E. Booth

Applied Physics Division, Los Alamos National Laboratory, Los Alamos, New Mexico 87545, USA

J. E. Gubernatis

Theoretical Division, Los Alamos National Laboratory, Los Alamos, New Mexico 87545, USA

We present a Monte Carlo algorithm that allows the simultaneous determination of a few extremal eigenpairs of a very large matrix without the need to compute the inner product of two vectors or store all the components of any one vector. The algorithm, a Monte Carlo implementation of a deterministic one we recently benchmarked, is an extension of the power method. In the implementation presented, we used a basic Monte Carlo splitting and termination method called the comb, incorporated the weight cancellation method of Arnow *et al.*, and exploited a sampling method, the sewing method, that does a large state space sampling as a succession of small state space samplings. We illustrate the effectiveness of the algorithm by its determination of the two largest eigenvalues of the transfer matrices for variously sized two-dimensional, zero-field Ising models. While very likely useful for other transfer-matrix problems, the algorithm is however quite general and should find application to a larger variety of problems requiring a few dominant eigenvalues of a matrix.

PACS number(s): 02.70.Tt, 05.10.Ln

I. INTRODUCTION

A common problem in computational physics is computing the eigenpairs of large matrices. We will present a Monte Carlo algorithm that allows the simultaneous determination of a few extremal eigenpairs of a very large matrix without the need to orthogonalize pairs of vectors to each other or store all the components of any vector. This algorithm is a Monte Carlo implementation of deterministic one we recently benchmarked [1]. The newly benchmarked algorithm is based on a refinement of the power method recently developed by Booth [2,3] and does not require, as does the standard Ritz estimator [4,5], the explicit computation of the inner product of two vectors. This feature is of special importance for Monte Carlo use because Monte Carlo sampling provides only successive estimates of eigenvectors represented by very small subsets of their possible components. For such a situation, the explicit computation of an inner product is impossible.

The basic power method is the traditional starting point for a Monte Carlo determination of the eigenpair associated with the eigenvalues of largest absolute value λ_1. While various versions of the Monte Carlo power method often compute this dominant eigenvalue very well, computing subdominant eigenvalues $\lambda_2, \lambda_3, \ldots$ has often proven much more difficult and is much less frequently attempted. Our Monte Carlo power method computes multiple extremal eigenpairs simultaneously. The particular algorithm presented uses a basic Monte Carlo splitting and termination technique called the comb [6,7], incorporates the weight cancellation method of Arnow *et al.* [8], and exploits a sampling method, the sewing method [9], that does a large state space sampling as a succession of small state space samplings.

In refining the power method [1], we were targeting its use on matrices so large that they are unassailable deterministically because no single vector can be stored in memory. As the system size increases, finding a few extremal eigenpairs of the transfer matrix of the two-dimensional Ising model becomes such a problem. We will illustrate the effectiveness of the algorithm by determining the two largest eigenvalues of the transfer matrices for variously sized two-dimensional zero-field Ising models, exploiting Onsager's exact results [10,11] for their values as benchmarks. We comment that two extremal eigenvalues of this matrix are of significant physical interest: the logarithm of λ_1 is proportional to the free energy, and the logarithm of the ratio λ_2/λ_1 is proportional to the reciprocal of the correlation length that controls long-range spin correlations near the critical point. Although our algorithm is extendable to finding more than two extremal eigenpairs, we will focus on finding just λ_1 and λ_2 [10,11].

In the next section, Sec. II, we summarize the basic features of the transfer matrix of the two-dimensional Ising model in a zero magnetic field. In subsequent sections we will reference these features to make our algorithm presentation more concrete. In Sec. III, we summarize our extension of the power method for the determination of the two eigenpairs corresponding to the two eigenvalues of largest absolute value. Then, in Sec. IV, we discuss the basics of our Monte Carlo implementation of this algorithm. We first use the algorithm for medium-sized matrices for which computer memory is adequate to store the eigenvectors and then use it for much larger-sized matrices for which it is not. For the Monte Carlo sampling of states in latter case, we use the sewing algorithm [9] that facilitates sampling of states from a large space from a smaller space. Results presented are for the determination of the two largest eigenvalues of the Ising transfer matrix for various lattice sizes. In the last section, we summarize our work and comment on its likely application to other systems.

We note that the intent of the present application is presenting and benchmarking a relatively general numerical method and not numerically studying the finite-size scaling of the eigenvalues of the matrix used for the benchmarking. Because we reproduce the exact eigenvalues to satisfactory accuracy, our estimates will enjoy the same scaling as the

well-known exact results for this problem. When the magnetic field is not zero and the eigenvalues are not exactly known, various researchers have calculated up to the four largest eigenvalues by a deterministic version of the power method and have made extensive studies of the scaling of these eigenvalues [12]. With the algorithm to be presented we have reproduced the first two of these eigenvalues and hence would reproduce the basic scaling. A more extensive study of the eigenvalues of the field dependent Ising model will be presented elsewhere [13]. The Monte Carlo approach increases the lattice sizes accessible. We note that these deterministic calculations required multiple processors. All our calculations were done on a single processor.

II. ISING MODEL

In his much celebrated work Onsager calculated many of the properties of the two-dimensional Ising model exactly [10]. His calculations started with the expression of the partition function in terms of its transfer matrix [10,14,15]. He then found all the eigenvalues of this matrix analytically and showed that the scaling the dominant eigenvalue with the area in the thermodynamic limit (the area of the model approaching infinity) implied the onset of long-range ordering among the spin variables. The ratio of the second largest eigenvalue to the first is associated with the spatial behavior of this ordering. At the critical temperature, this ratio approaches unity as the area approaches infinity.

We will consider an $m \times n$ Ising model with periodic boundary conditions. Its energy is

$$E\{\mu\} = -J\sum_{i=1}^{m}\sum_{j=1}^{n}\mu_{i,j}\mu_{i+1,j} - J\sum_{i=1}^{m}\sum_{j=1}^{n}\mu_{i,j}\mu_{i,j+1}. \quad (1)$$

Here, (i,j) are the coordinates of a lattice site. The Ising spin variable $\mu_{i,j}$ on each site has the value of ± 1, the exchange constant $J > 0$, $\mu_{m+1,j} = \mu_{1,j}$, and $\mu_{i,m+1} = \mu_{i,1}$. The symbol

$$\sigma_j = (\mu_{1,j}, \mu_{2,j}, \ldots, \mu_{m,j}) \quad (2)$$

denotes a column configuration of Ising spins and there are 2^m possible configurations for each column.

The transfer matrix $A(\sigma, \sigma')$ follows from a re-expression of the partition function [11],

$$\begin{aligned} Z(m,m) &= \sum_{\{\mu\}} \exp[-\nu E(\{\mu\})] \\ &= \sum_{\sigma_1,\ldots,\sigma_m} A(\sigma_1,\sigma_2)A(\sigma_2,\sigma_3)\cdots A(\sigma_{m-1},\sigma_m)A(\sigma_m,\sigma_1) \\ &= \sum_{\sigma_1} A^n(\sigma_1,\sigma_1) \end{aligned} \quad (3)$$

where $\nu = J/k_B T$, k_B is Boltzmann's constant, and T is the temperature, and $A(\sigma, \sigma')$ is a $2^m \times 2^m$ matrix whose elements are

$$A(\sigma, \sigma') = \exp\left(\nu\sum_{k=1}^{m-1}\mu_k\mu_{k+1}\right)\exp\left(\nu\sum_{k=1}^{m}\mu_k\mu'_k\right) \quad (4)$$

As customary for Ising-model simulations, we represent a configuration σ by the first m bits of an integer between 0

and $2^m - 1$, with a set bit being a $+1$ Ising spin and an unset bit being a -1 spin. This convention maps the matrix element $A(\sigma, \sigma')$ between configurations to an element A_{ij} between integers.

We note that $A(\sigma, \sigma')$ is asymmetric and its elements are greater than zero so the matrix is maximally dense and hence irreducible. Because of the positivity and irreducibility the Perron-Frobenius Theorem [16] says that dominant eigenvalue is real, positive, and nondegenerate and all components of the corresponding eigenstate are real and have the same sign. The two largest eigenvalues of A, for *finite* m, are [17]

$$\lambda_1 = (2 \sinh 2\nu)^{m/2}\exp\left[\frac{1}{2}(\eta_1 + \eta_3 + \cdots + \eta_{2m-1})\right],$$

$$\lambda_2 = (2 \sinh 2\nu)^{m/2}\exp\left[\frac{1}{2}(\eta_2 + \eta_4 + \cdots + \eta_{2m})\right], \quad (5)$$

where

$$\cosh \eta_k = \cosh 2\nu \coth 2\nu - \cos\frac{\pi k}{m}. \quad (6)$$

The transfer matrix of the Ising model can be symmetrized [11] but we saw no computational advantage for using this form.

III. POWER METHOD

For some real-valued $M \times M$ matrix A, not necessarily symmetric, we will be concerned with the M eigenpairs $(\lambda_\alpha, \psi_\alpha)$ satisfying

$$A\psi_\alpha = \lambda_\alpha\psi_\alpha. \quad (7)$$

In the simplest application of the power method [4], an iteration is started with some ψ, normalized in a convenient, but otherwise relatively arbitrary, manner and consists of iterating the two steps

$$\phi = A\psi,$$

$$\psi = \phi/\|\phi\|, \quad (8)$$

until some convergence criterion is met. If we write

$$\psi = \sum_{\alpha=1}^{M}\omega_\alpha\psi_\alpha \quad (9)$$

and if $|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \cdots \geq |\lambda_N|$, then after n iterations

$$A^n\psi = \lambda_1^n\left[\omega_1\psi_1 + \sum_{\alpha=2}^{M}\omega_\alpha\left(\frac{\lambda_\alpha}{\lambda_1}\right)^n\psi_\alpha\right] \quad (10)$$

Accordingly, as $n \to \infty$,

$$\psi \to \psi_1/\|\psi_1\|,$$

$$\|\phi\| \to \lambda_1. \quad (11)$$

Thus, the dominant eigenpair is simultaneously determined. For the norm of the vector ϕ whose components are ϕ_i, a frequent choice is

$$\|\phi\| \equiv \|\phi\|_\infty = \max_i |\phi_i|. \tag{12}$$

For deterministic calculations of a few dominant eigenpairs, say N, one of two approaches are typically tried. One approach is to use the power method to determine the dominant eigenpair, use deflation to project out this state out of the matrix, and then reuse the power method on the deflated matrix. To determine several eigenpairs simultaneously, the power method can be generalized to

$$\Phi = A\Psi, \tag{13}$$

where Φ and Ψ are $M \times N$ matrices whose columns are orthogonalized to each other. This orthogonality needs maintenance throughout the computation or else all N vectors, represented by the columns of the initial Ψ, will converge to the one associated with the dominant eigenvalue [5]. This approach is called orthogonal [5] or simultaneous iteration [18].

For Monte Carlo calculations of a few dominant eigenpairs, we are proposing a quite different approach based on Booth's proposed refinement of the power method [2,3]. This refinement uses the observation that for any eigenpair (λ, ψ) and for each nonzero *component* of the eigenvector, the eigenvalue equation $A\psi = \lambda\psi$ can be rewritten as

$$\lambda = \frac{\sum_j A_{ij}\psi_j}{\psi_i} \tag{14}$$

and that similar equations can also be written for any number of groupings of components,

$$\lambda = \frac{\sum\limits_{i \in R_1}\sum\limits_j A_{ij}\psi_j}{\sum\limits_{i \in R_1} \psi_i} = \frac{\sum\limits_{i \in R_2}\sum\limits_j A_{ij}\psi_j}{\sum\limits_{i \in R_2} \psi_i} = \cdots = \frac{\sum\limits_{i \in R_L}\sum\limits_j A_{ij}\psi_j}{\sum\limits_{i \in R_L} \psi_i} \tag{15}$$

where the R_i are rules for different groupings. The groupings are quite flexible: they can overlap and their union need not cover the entire space. In addition, any two groupings, say 1 and 2, imply

$$\sum_{i \in R_2} \psi_i \sum_{i \in R_1}\sum_j A_{ij}\psi_j = \sum_{i \in R_1} \psi_i \sum_{i \in R2}\sum_j A_{ij}\psi_j. \tag{16}$$

From L groupings of the components, Booth constructed L estimators for the L largest eigenvalues and forced them to become equal by adjusting certain parameters at each iteration step. For the dominant two eigenvalues, we will do something similar.

First we note that for almost any starting point $\psi = \Sigma_\alpha \omega_\alpha \psi_\alpha$, the power method will converge to (λ_1, ψ_1). To find the two extremal eigenvalues, we need two normalized, but not necessarily orthogonal, starting points $\psi' = \Sigma_\alpha \omega'_\alpha \psi_\alpha$ and $\psi'' = \Sigma_\alpha \omega''_\alpha \psi_\alpha$ [1]. At each step, we will apply A to them individually. Without any intervention both will project the same dominant eigenfunction so at each step we adjust the relationship between their *sum* to direct one to the dominant state and the other to the next dominant one.

Formally, we start the iteration with $\psi = \psi' + \eta\psi''$. Suppose at the nth step, ψ' and ψ'' have iterated to $\hat\psi'$ and $\hat\psi''$, then at the $(n+1)^{th}$ step we invoke Eqs. (15) and (16) and require that

$$\frac{\sum\limits_{i \in R_1}\sum\limits_j A_{ij}\hat\psi'_j + \eta \sum\limits_{i \in R_1}\sum\limits_j A_{ij}\hat\psi''_j}{\sum\limits_{i \in R_1}\hat\psi'_i + \eta\sum\limits_{i \in R_1}\hat\psi''_i}$$
$$= \frac{\sum\limits_{i \in R_2}\sum\limits_j A_{ij}\hat\psi'_j + \eta \sum\limits_{i \in R_2}\sum\limits_j A_{ij}\hat\psi''_j}{\sum\limits_{i \in R_2}\hat\psi'_i + \eta\sum\limits_{i \in R_2}\hat\psi''_i}, \tag{17}$$

which leads to

$$q_2\eta^2 + q_1\eta + q_0 = 0 \tag{18}$$

with

$$q_2 = \sum_{i \in R_2}\hat\psi''_i \sum_{i \in R_1}\sum_j A_{ij}\hat\psi''_j - \sum_{i \in R_1}\hat\psi''_i \sum_{i \in R_2}\sum_j A_{ij}\hat\psi''_j,$$

$$q_1 = \sum_{i \in R_2}\hat\psi''_i \sum_{i \in R_1}\sum_j A_{ij}\hat\psi'_j - \sum_{i \in R_1}\hat\psi''_i \sum_{i \in R_2}\sum_j A_{ij}\hat\psi'_j$$
$$+ \sum_{i \in R_2}\hat\psi'_i \sum_{i \in R_1}\sum_j A_{ij}\hat\psi''_j - \sum_{i \in R_1}\hat\psi'_i \sum_{i \in R_2}\sum_j A_{ij}\hat\psi''_j,$$

$$q_0 = \sum_{i \in R_2}\hat\psi'_i \sum_{i \in R_1}\sum_j A_{ij}\hat\psi'_j - \sum_{i \in R_1}\hat\psi'_i \sum_{i \in R_2}\sum_j A_{ij}\hat\psi'_j. \tag{19}$$

The algorithm is to apply A repeatedly until two real solutions η_1 and η_2 for Eq. (18) exist. One solution will be then used to guide further iterations to (λ_1, ψ_1); the other, to (λ_2, ψ_2).

The power method becomes [1]: choose two starting points ψ' and ψ'', which need not be orthogonal, then for each iteration step, compute

$$\psi' \leftarrow \psi'/\|\psi'\|,$$

$$\psi'' \leftarrow \psi''/\|\psi''\|, \tag{20}$$

and if the roots of Eq. (18) are real [19], update using

$$\psi' \leftarrow A\psi'' + \eta_1 A\psi',$$

$$\psi'' \leftarrow A\psi'' + \eta_2 A\psi', \tag{21}$$

otherwise use

$$\psi' \leftarrow A\psi',$$

$$\hat\psi'' \leftarrow A\hat\psi''. \tag{22}$$

Eigenvalues can be estimated from

$$\lambda_1 = \frac{\sum_{i \in R_1} \sum_j A_{ij} \psi'_j + \eta_1 \sum_{i \in R_1} \sum_j A_{ij} \psi''_j}{\sum_{i \in R_1} \psi'_i + \eta_1 \sum_{i \in R_1} \psi''_i},$$

$$\lambda_2 = \frac{\sum_{i \in R_1} \sum_j A_{ij} \psi'_j + \eta_2 \sum_{i \in R_1} \sum_j A_{ij} \psi''_j}{\sum_{i \in R_1} \psi'_i + \eta_2 \sum_{i \in R_1} \psi''_i}, \quad (23)$$

where η_1 and η_2 generate the largest and next largest eigenvalue estimates.

IV. MONTE CARLO IMPLEMENTATION

The basic operation of a power method is a matrix-vector multiplication. Here, we now describe how we used the Monte Carlo method to estimate the repetition of such multiplications.

In the basis defining the matrix elements of A, we write

$$\psi' = \sum_i \omega'_i |i\rangle,$$

$$\psi'' = \sum_i \omega''_i |i\rangle. \quad (24)$$

We will call the amplitudes ω'_i and ω''_i *weights* even though they are not necessarily all positive nor are the sums of their absolute values unity. We will also assume that the elements A_{ij} of the $M \times M$ matrix A are easily generated, as we are ultimately interested in cases where M is so large that this matrix must be generated on the fly as opposed to being stored. Next, we imagine we have N particles distributed over the M basis states defining A. Generally, $N \ll M$. Then, at each iteration step, we interpret A_{ij} as the weight of a particle arriving in state $|i\rangle$ per unit weight of a particle in state $|j\rangle$. The action of A_{ij} on a ψ thus causes all particles currently in state $|j\rangle$ to jump to $|i\rangle$, carrying to $|i\rangle$ their current weight ω_j, modified by A_{ij}.

The jumps will be executed probabilistically. To do this we let the total weight leaving state $|j\rangle$ be

$$W_j = \sum_i A_{ij} \quad (25)$$

and define the transition probability from $|j\rangle$ to $|i\rangle$ be

$$T_{ij} = A_{ij}/W_j. \quad (26)$$

The number W_j is called the state weight multiplier. How we use these densities will depend on the size and types of the matrices under consideration.

A. Medium matrices

If M is sufficiently small so we can store all components of our vectors, then a Monte Carlo procedure for jumping is easily constructed. Instead of always (i.e., with probability 1) moving weight A_{ij} from state $|j\rangle$ to state $|i\rangle$, we will instead sample a $|i\rangle$ from T_{ij} and multiply the transferred weight by

the ratio of the true probability (1.0) to the sampled probability (T_{ij}); that is, if state $|i\rangle$ is sampled, the weight arriving in state $|i\rangle$ from $|j\rangle$ is multiplied by

$$A_{ij} \frac{1.0}{T_{ij}} = A_{ij} \frac{1.0}{A_{ij}/W_j} = W_j. \quad (27)$$

For many Monte Carlo simulations, as is the case for the transfer matrix of the Ising model, the particle weights defining the eigenvector associated with the largest eigenvalue can be made all positive. The second eigenfunction however must be represented by some particles of negative weight and some particles of positive weight. These negative and positive weights must for some jumps at least partially cancel to maintain a correct estimation of the second eigenfunction. When $N \ll M$, this cancellation does not occur often enough in a Monte Carlo simulation without proper design: because the number of states vastly out numbers the number of particles, the probability that a negatively and a positively weighted particle randomly arrive in the same state becomes trivially small.

There are several ways to arrange the cancellation [2]. We found the Arnow *et al.* [8] algorithm effective and convenient. First, we consider two particles of weights w_1 and w_2 in states $|j_1\rangle$ and $|j_2\rangle$, and then let T_{ij_1} and T_{ij_2} be the probabilities of their reaching state $|i\rangle$. Next we use the weight multiplier W_j for jumping from state $|j\rangle$ to state $|i\rangle$ and suppose that states $|i_1\rangle$ and $|i_2\rangle$ are sampled from the density $T_{ij_1} + T_{ij_2}$. This density can be sampled by sampling a $|i_1\rangle$ from T_{ij_1} and a $|i_2\rangle$ from T_{ij_2}. The true probability that particle 1 jumps to $|i_1\rangle$ is $T_{i_1 j_1}$ and the true probability that particle 2 jumps to $|i_1\rangle$ is $T_{i_1 j_2}$. The ratio of the true density to the sampled density for particle 1 is

$$\frac{T_{i_1 j_1}}{T_{i_1 j_1} + T_{i_1 j_2}} \quad (28)$$

so that the weight arriving at state $|i_1\rangle$ from particle 1 is

$$\frac{w_1 W_{j_1} T_{i_1 j_1}}{T_{i_1 j_1} + T_{i_1 j_2}}. \quad (29)$$

The true probability that particle 2 arrives at $|i_1\rangle$ is $T_{i_1 j_2}$. The ratio of the true density to the sampled density for particle 2 is

$$\frac{T_{i_1 j_2}}{T_{i_1 j_1} + T_{i_1 j_2}} \quad (30)$$

so that the weight arriving at $|i_1\rangle$ from particle 2 is

$$\frac{w_2 W_{j_2} T_{i_1 j_2}}{T_{i_1 j_1} + T_{i_1 j_2}} \quad (31)$$

Thus the total weight arriving at $|i_1\rangle$ is

$$\frac{w_1 W_{j_1} T_{i_1 j_1} + w_2 W_{j_2} T_{i_1 j_2}}{T_{i_1 j_1} + T_{i_1 j_2}}. \quad (32)$$

By similar arguments, the total weight arriving at $|i_2\rangle$ is

$$\frac{w_1 W_{j_1} T_{i_2 j_1} + w_2 W_{j_2} T_{i_2 j_2}}{T_{i_2 j_1} + T_{i_2 j_2}}. \tag{33}$$

We note that to get meaningful cancellation, say between particle 1 with weight $w_1 < 0$ and particle 2 with weight $w_2 > 0$, the transition probabilities must overlap somewhat. For example, if in (33) $T_{i_2 j_1} = \epsilon \ll 1$, then the total weight arriving at i_2 is essentially just the weight arriving of particle 2 alone.

$$\frac{w_1 W_{j_1} \epsilon + w_2 W_{j_2} T_{i_2 j_2}}{\epsilon + T_{i_2 j_2}} \approx w_2 W_{j_2}. \tag{34}$$

How one arranges better overlap is problem dependent. For our Ising simulations, we sorted the particles into state order (a state is represented by the bits of an integer). Particles 1 and 2 were then sampled together according to the Arnow *et al.* scheme, then particles 3 and 4 are sampled together, and so forth. The fact that the list is ordered means that there are (typically) many nearby states $|i\rangle$ that are accessible from both particles ℓ and $\ell + 1$ with nontrivial transition probabilities.

As the iteration progresses, the absolute value of the weights of some particles becomes very large, and those of some others, very small. As standard for Monte Carlo methods with weighted particles, particles with weights of small magnitude are stochastically eliminated and those with large magnitudes are stochastically split. To do this we used a procedure called the comb [6,7]. It is described in Appendix A.

The steps of the algorithm are: First, we initialize the states and weights of two vectors. For the Ising simulation, each vector had the same states but different weights. We selected the states uniformly and randomly over the interval $(0, 2^m - 1)$ and selected the ω_i' uniformly and randomly over the interval $(0, 1)$ and the ω_i'' uniformly and randomly over the interval $(-0.5, 0.5)$. Then, for a fixed number of times we iterate. At each iteration we execute the jump procedure for each particle, place the particle list in state order, effect cancellations, estimate the eigenvalues from Eq. (23), update ψ' and ψ'', and then comb. R_1 consisted of the states for which more than half of its m bits were 0s and R_2 consisted of the states for which more than half of its m bits were 1's. Additional algorithmic details are given in Appendix B.

Table I shows the computed λ_1 and λ_2 for $m = 12$. A computer program was written to make 20 independent (different random number seeds) calculations using N particles per iteration, for various values of N. Note that even when $N = 100 \ll M = 2^{12} = 4096$ the method still separates the eigenfunctions. The simulations were run at the critical temperature of the infinite lattice; that is, $\nu = 0.440\,686\,793\,509\,771\,5$ [11]. The last line in the table gives the eigenvalues deterministically obtained by using Eq. (5). From the last few lines, we also see that when the number of particles floods the number of states, that is, when all the basis states are being used multiple times, exceptional accuracy is obtained.

Also in the table is the timing for each run. The runs were done on a single 1.5 GHz Mac PowerPC processor. We note

TABLE I. Particle number N dependence of individual runs for the $m = 12$ Ising model. The λ_i and $\sigma_i (i = 1, 2)$ are run's eigenvalue averages and error estimates. The row labeled "Onsager" are the eigenvalues obtained from Eq. (5). Also given is the wall clock time for each run. The order of the transfer matrix is 4096.

N	λ_1	σ_1	λ_2	σ_2	Min.
100	71415.164	65	67021.909	103	5.4
1000	71527.110	17	67023.420	31	7.9
10000	71553.325	5.3	66956.314	9.1	22.8
100000	71557.854	2.0	67005.486	3.2	150.0
1000000	71557.129	0.36	67010.989	0.58	1474.0
Onsager	71557.047		67010.869		

that the runs appear to scale sublinearly with the number of particles. This scaling is deceiving. For a small number of particles, the time required to set up the transition matrix A_{ij} is significant compared to the rest of the calculation. The *Monte Carlo* part is dominated by the state order sort which scales as $N \log N$, so as N becomes very large, the run time scaling should eventually be slightly superlinear.

B. Large matrices

For $M \le 2^{12}$, sampling from the cumulative probability $C_i = \Sigma_{k=0}^{i} T_{kj}$ works well. If the number of states gets too large ($M = 2^{12}$ was our limit), then C_i cannot be sampled directly because it cannot fit in the computer's memory. In this case, we could just randomly pick from any state $|j\rangle$ any state $|i\rangle$ with probability $1/M$ instead of always picking a state $|i\rangle$ (i.e., picking it with probability 1). Then, if state $|i\rangle$ is sampled, the weight of particles arriving in state $|i\rangle$ is

$$A_{ij} \frac{1.0}{1/M} = A_{ij} M. \tag{35}$$

The problem with this approach is that the A_{ij} can have immense variation so that this simple sampling scheme is unlikely to work well as a Monte Carlo method. This situation is especially true for the Ising problem. A large part of such variations however can be removed by sampling the state in stages and then sewing the stages together.

To explain the sewing procedure [9], we will first assume that we can write any state $|i\rangle$ in our basis as a direct product of the states in a smaller basis, for example,

$$|i\rangle = |i_2\rangle |i_1\rangle$$

Instead of transferring weight

$$W_j = \sum_k A_{kj} \tag{36}$$

from state $|j\rangle$ to state $|i\rangle$ with probability

$$T_{ij} = A_{ij}/\sum_k A_{kj} = A_{ij}/W_j, \tag{37}$$

we will use the a_{ij} that would apply to the smaller set of states and then make an appropriate weight correction.

TABLE II. Estimates of the two dominant eigenvalues λ_1 and λ_2 and their errors of the transfer matrix for variously sized two-dimensional Ising models. Each estimate was based on 20 independent simulations. Also given for each lattice size is value for λ_1 computed via Onsager's exact result [10,11].

Matrix size	λ_1 (Onsager)	λ_1	λ_2 (Onsager)	λ_2
$2^{16} \times 2^{16}$	2.93297×10^6	$2.93307 \pm 0.00008 \times 10^6$	2.79225×10^6	$2.79482 \pm 0.00010 \times 10^6$
$2^{24} \times 2^{24}$	4.95473×10^9	$4.95480 \pm 0.00020 \times 10^9$	4.79510×10^9	$4.79502 \pm 0.00029 \times 10^9$
$2^{32} \times 2^{32}$	8.39316×10^{12}	$8.39311 \pm 0.00049 \times 10^{12}$	8.18959×10^{13}	$8.18807 \pm 0.00061 \times 10^{12}$
$2^{40} \times 2^{40}$	1.42333×10^{16}	$1.42334 \pm 0.00007 \times 10^{16}$	1.39565×10^{17}	$1.39558 \pm 0.00008 \times 10^{16}$
$2^{48} \times 2^{48}$	2.41504×10^{19}	$2.41522 \pm 0.00019 \times 10^{19}$	2.37584×10^{19}	$2.37481 \pm 0.00054 \times 10^{19}$

For each smaller set of states, we rewrite the analogous transition probability from state $|j\rangle$ to state $|i\rangle$ as

$$t_{ij} = a_{ij}/w_j, \qquad (38)$$

and the analogous weight multiplier as

$$w_j = \sum_k a_{kj}. \qquad (39)$$

We thus will sample $|i_1\rangle$ and $|i_2\rangle$ from the probability function

$$t_{i_2 j_2} t_{i_1 j_1}. \qquad (40)$$

Now, we define C_{ij} to be the weight correction necessary to preserve the expected weight transfer from state $|j\rangle = |j_2\rangle |j_1\rangle$ to state $|i\rangle$. It satisfies

$$A_{ij} = C_{ij} t_{i_2 j_2} t_{i_1 j_1}. \qquad (41)$$

Using Eq. (38) in Eq. (41), we find

$$A_{ij} = C_{ij} \frac{a_{i_2 j_2}}{w_{j_2}} \frac{a_{i_1 j_1}}{w_{j_1}}. \qquad (42)$$

Thus

$$C_{ij} = w_{j_1} w_{j_2} \frac{A_{ij}}{a_{i_1 j_1} a_{i_2 j_2}}. \qquad (43)$$

This sewing method generalizes easily. For k sets of states, (42) and (43) become

$$A_{ij} = C_{ij} \prod_{n=1}^{k} t_{i_n j_n} \qquad (44)$$

and

$$C_{ij} = A_{ij} \prod_{n=1}^{k} \frac{w_{j_n}}{a_{i_n j_n}}. \qquad (45)$$

For the Ising problem, the weight correction is

$$C_{ij} = \exp(\nu D_i) \prod_{n=1}^{k} w_{j_n} \qquad (46)$$

where νD_i is the energy difference per $k_B T$ between calculating with the bits together and the bits separately [9].

Using this sewing algorithm for the sampling of states, we computed the first and second eigenvalues and their standard deviations for variously sized two-dimensional Ising models by sewing sets of 8 bits. The results are shown in Table II. Twenty independent runs were done for each size with 500 iterations per run. There were 1 million particles per iteration for $m = 16$, 24, and 32 and 5 million for $m = 40$ and 48. Only the second half of the iterations for each run used in the estimation process. This choice for "burn-in" was arbitrary and excessive. Typically the iteration converges to its fixed point in about 10 or fewer steps. Clearly, the estimated mean of the eigenvalues are consistent with the analytic result of Onsager. The $m = 48$ run took about 2.15 h on a single 1.25 GHz Alpha EV6 processor. $\nu = 0.440\,686\,793\,509\,771\,5$, the bulk critical value. R_1 consisted of the states for which more than half of its m bits were 0s and R_2 consisted of the states for which more than half of its m bits were 1's. In Table III, we present the run averages for the $M = 2^{48}$ case. We note that $2^{48} \approx 2.8 \times 10^{14}$.

TABLE III. Run dependence of λ_1 and λ_2 for the 48×48 Ising model. Each run had 5 million particles and 500 iterations. Only the last 250 were used to computed the stated run averages.

Run	λ_1	λ_2
1	$2.415237261676499 \times 10^{19}$	$2.376562826996199 \times 10^{19}$
2	$2.416053055304024 \times 10^{19}$	$2.376357592801913 \times 10^{19}$
3	$2.414208101931427 \times 10^{19}$	$2.373861137045189 \times 10^{19}$
4	$2.414240257597889 \times 10^{19}$	$2.374603969362707 \times 10^{19}$
5	$2.415129736468956 \times 10^{19}$	$2.375668969910789 \times 10^{19}$
6	$2.414540994327886 \times 10^{19}$	$2.367761547315434 \times 10^{19}$
7	$2.415931928295190 \times 10^{19}$	$2.371678054324882 \times 10^{19}$
8	$2.416760807133448 \times 10^{19}$	$2.380200781153834 \times 10^{19}$
9	$2.413931047178054 \times 10^{19}$	$2.374313310822579 \times 10^{19}$
10	$2.414288017604171 \times 10^{19}$	$2.374183202851979 \times 10^{19}$
11	$2.415907427261585 \times 10^{19}$	$2.374280853036757 \times 10^{19}$
12	$2.415718175365685 \times 10^{19}$	$2.374828531467586 \times 10^{19}$
13	$2.415613030894034 \times 10^{19}$	$2.374305559747647 \times 10^{19}$
14	$2.414272502002316 \times 10^{19}$	$2.374687824232260 \times 10^{19}$
15	$2.416636076568460 \times 10^{19}$	$2.376864186656784 \times 10^{19}$
16	$2.414631286368427 \times 10^{19}$	$2.376097488856122 \times 10^{19}$
17	$2.415202515765169 \times 10^{19}$	$2.376875902411582 \times 10^{19}$
18	$2.415187019163818 \times 10^{19}$	$2.373320142080186 \times 10^{19}$
19	$2.414692528984318 \times 10^{19}$	$2.374355064690523 \times 10^{19}$
20	$2.416154446732387 \times 10^{19}$	$2.375485037469832 \times 10^{19}$

V. CONCLUDING REMARKS

We have presented a Monte Carlo algorithm that enables the determination of two extremal eigenvalues of a very large matrix. The explicit demonstration of the power of our algorithm was the determination of the two largest eigenvalues of the transfer matrix of the two-dimensional Ising model in a zero magnetic field. The convenience of this matrix was the existence of exact expressions for its eigenvalues for any finite-sized system. We were able to reproduce the exact values for the two dominant eigenvalues to within the statistical error of our simulations. In a relatively straightforward manner, the algorithm extends to the concurrent computation of more than two, but this extension has so far received only preliminary, but encouraging, investigation.

Physics and chemistry provide numerous problems where obtaining several extremal eigenvalues is important. One important problem class is finding the ground state [20] and a few excited states of atoms, molecules, solids, and nuclei. A number of deterministic and classical Monte Carlo methods have been re-expressed as quantum Monte Carlo techniques, and quantum Monte Carlo researchers have generated some additional approaches. Some of these techniques are discussed in [21–23]. Multiple investigators [24] have adopted and refined the various multifermion methods to find the dominant eigenvalue, and in some cases a few dominant eigenvalues, of various classical systems. When applied to transfer matrices, the approach is often called transfer-matrix Monte Carlo [23]. In some of these transfer-matrix Monte Carlo works, as in most of the quantum work, in order to obtain good results, an importance function is used to guide the sampling and a variational Monte Carlo simulation is first performed to optimize it. (The techniques for this optimization are still evolving [25].) For the application at hand, these complexities were not needed, though they are not necessarily excluded. Some might be useful if much higher accuracy is essential.

We note Nightingale and Blöte's [26] use of these transfer Monte Carlo methods and their exploitation of knowing the dominant eigenpair of a Markov chain transition matrix satisfying detailed balance to find its second, and few higher, eigenvalues. With the knowledge of the first eigenpair, we can transform our approach into a simpler Monte Carlo method for finding the just second eigenvalue [27]. Further, using a very similar way of estimating an eigenpair, Booth has presented an algorithm for determining a single eigenvalue without the need to know or concurrently determine any other one. We also note our transfer matrix was fully dense. In many prior works, for example, reference [26], the matrix was sparse because of the use of helical boundary conditions instead of periodic boundary ones. To handle megasized dense matrices, we created the sewing algorithm [9]. This exact procedure just samples the transfer matrix and is otherwise independent of our extension of the power method. If the matrix is sparse, the sewing algorithm is generally not needed, and other options for selecting the state to jump to exist, avoid the need to store the cumulative distribution, improve efficiency, and reduce variance [28].

A quick look at these prior multiple eigenvalues algorithms clearly reveals several distinct differences from ours.

One key difference is our way of estimating an eigenvalue. It is our exploitation of the *definition* of an eigenpair, namely, the product of a matrix times one of its eigenvectors scales *each component* of the vector by the eigenvalue. Because a similar scaling holds for sums of subsets of these components, we observed that this intrinsic relation among the components places a distinct restriction on what linear combinations of vectors can represent an eigenvector. To compute multiple eigenvalues (or even a single one), we embedded this restriction into the power method in a way that only adds to it the relatively simple task of finding the roots of a low-order polynomial.

Another difference lies in what our Monte Carlo needs to do and does. Our extension of the power method adds to it the potentially, extremely difficult task of properly canceling signed walkers. We are directly sampling from the intended eigenvectors. In particular, our algorithm requires sampling from the subdominant eigenvector which must have positive and negative components. Many of the single and multiple eigenvalue quantum Monte Carlo methods are for fermion problems where a significant impediment is the need to sample from the fermionic ground state which generally has a very complicated unknown nodal surface. In these applications, this surface is approximated. In fact, it was for fermion ground-state problems that the Arnow *et al.* [8] method to cancel signed walkers used here was first proposed [22]. Our successful cancellation of signed walkers, and the slightly superlinear scaling of our computation time with system complexity as opposed to an exponential scaling, demonstrates that we have "solved" the sign problem [29] for our application. The only time we used any nodal information was in specifying the starting points for the projection: the dominant eigenvector cannot have nodes so we initialized its components all positive; all subdominant eigenvectors however must have components of mixed sign so we simply initialize half the components of the second eigenvector as negative. These actions were in some cases more helpful than necessary.

Our transfer matrix was positive, asymmetric, and dense. How is our algorithm changed if the matrix lacks one or more of these properties? For simple test cases, we have successfully constructed deterministic procedures for matrices whose elements are complex valued. Also for simple test cases, we have had success for real asymmetric matrices whose eigenvalues are complex valued. Devising Monte Carlo algorithms for real, symmetric, sparse matrices has however received more of our attention [28].

For an indefinite but symmetric matrix, the symmetry and reality assure the reality of the eigenpairs, but do not fix the sign of the dominant eigenvalue or the components of its eigenvector. Such matrices can also have the dominant eigenvalue being degenerate. In our algorithm a signature of the dominant eigenvalue being degenerate is the quantity q_1 [Eq. (19)] in our quadratic Eq. (18) becoming extremely small. In cases where we knew the degeneracy *a priori*, we saw our estimate of λ_1 being very close to λ_2, that is, we saw our estimates of the dominant degenerate eigenvalue being slightly but artificially separated. This type of difficulty is inherent to a power method. If the matrix elements are of mixed sign, the modifications to the algorithm are quite

simple: we take the absolute value of the A_{ij} and multiply the weight multiplier by the sign of A_{ij}. We also have to account for the sign of the weights ω_i' and ω_i'' appropriately. Weight cancellation must be done with care.

Here, we focused principally on the determination of eigenvalues, noting the rapid convergence to their values, but what about the determination of the eigenvectors? If we can store the vectors in computer memory we can accurately determine them. For a problem in a continuum, for example, finding the dominant eigenvalues of an integral equation such as those occurring for the transport of neutrons, we have accurately estimated the two dominant eigenfunctions with an efficiency enhanced over the basic power method for determining just the dominant eigenfunction [30].

The dynamics of determining eigenvectors is as follows: Because the standard power method for the dominant eigenpair converges to ψ_1 as λ_2/λ_1 (10), the dominant eigenvector becomes increasingly difficult to determine as λ_1 and λ_2 become very close in magnitude. Because we are projecting to the first two dominant eigenpairs, our method converges to ψ_1 as λ_3/λ_1. This is usually a gain relative to the standard power method. Convergence to ψ_2 goes as λ_3/λ_2. This ratio illustrates the fact that in principle we can accelerate convergence to ψ_i by seeking the $L>i$ highest eigenfunctions and thus converging to ψ_i as λ_{L+1}/λ_i. To do this, we need to use L groupings and L starting states and adjust the sum of the iterated states to guide them to convergence to L different eigenvalues. We have observed accelerated convergence for the $L=3$ case on simple test cases. We comment that finding the eigenvectors of the transfer matrix of the Ising and other models is a problem for which finding the eigenvectors becomes increasingly difficult as the system size increases, if we could in fact store them. For this type of matrix, the eigenvalue spectra is expected to become quasicontinuous starting with the second eigenvalue and the first and second eigenvalues become asymptotically degenerate at a critical point [31].

In closing, we believe that the algorithm presented here is accurate, easy to implement, and applicable to many other problems. Different problems afford the opportunity to improve its efficiency by modifying some of its details. Wider use of the algorithm will define more crisply its strengths and limitations than is possible by just the present application.

APPENDIX A

The comb is a stochastic procedure for selecting M particles, not necessarily all different, from a list of N unevenly weighted particles and preserving the total weight. If $\omega_i>0$ is the weight of an individual particle in the original list, its weight in the new list becomes W_T/M where $W_T=\Sigma_i^N \omega_i$ is the total weight. To effect this procedure, we construct the cumulative sums of the original weights

$$C_j = \sum_{i=1}^{j} \omega_i \qquad (A1)$$

where $j=0,1,2,\ldots,N$ and $C_0=0$ and $C_N=W_T$, draw a random number ξ uniformly distributed over $(0,1)$, and then for $k=1,2,\ldots,M$, select particle j for the new list if

$$C_{j-1} < (k-1+\xi)\frac{W_T}{M} \leq C_j. \qquad (A2)$$

Dependent of the size of the difference $C_{j-1}-C_j$ relative to W_T/M, particle j is selected zero, one, or more times. The procedure calls the random number generator only once, and the new particle list is always of a predetermined size.

In the present simulations we have a list of particles and to each particle i is associated a state $|i\rangle$ and two weights, ω_i' and ω_i''. If we were to comb the two lists of weights separately, we would in general produce two lists of states. We wanted one such list. Accordingly, we combed in the following manner: we formed

$$p_i' = \frac{|\omega_i'|}{\displaystyle\sum_{i=1}^{N} |\omega_i'|}, \qquad (A3)$$

$$p_i'' = \frac{|\omega_i''|}{\displaystyle\sum_{i=1}^{N} |\omega_i''|}, \qquad (A4)$$

and then generated the cumulative distribution.

$$C_j = \sum_{i=1}^{j} \frac{1}{2}(p_i' + p_i''). \qquad (A5)$$

Now we combed the C_j as before, noting that we have chosen a normalization such that $W_T=1$. Instead of giving the selected particle j a weight W_T/M for both w_j' and w_j'', we instead assign

$$\omega_j' = \frac{p_i'\,\text{sign}(\omega_i')}{p_i' + p_i''}, \qquad (A6)$$

$$\omega_j'' = \frac{p_i''\,\text{sign}(\omega_i'')}{p_i' + p_i''}, \qquad (A7)$$

APPENDIX B

The following are some useful particulars of our Monte Carlo implementation of our modified power method. First, instead of updating via Eq. (21) we update via

$$\psi' \leftarrow A\psi' + \eta_1 A\psi'',$$

$$\psi'' \leftarrow \frac{1}{\eta_2}A\psi' + A\psi''. \qquad (B1)$$

This form is more symmetric and makes more explicit that we are trying to remove ψ'' from ψ' and vice versa. To avoid

overcorrecting we have sometimes found it useful to update via

$$\psi' \leftarrow A\psi' + \eta_1' A\psi'',$$

$$\psi'' \leftarrow \frac{1}{\eta_2'} A\psi' + A\psi'', \tag{B2}$$

where

$$\eta_1' = \text{sign}(\eta_1)\min(\alpha, |\eta_1|), \tag{B3}$$

$$\eta_2' = \text{sign}(\eta_2)\min(\alpha, |\eta_2|), \tag{B4}$$

and α is some small positive number.

Besides the eigenvalue estimators [Eq. (23)], there are others [Eq. (15)] equally valid that provide multiple cross checks to [Eq. (23)]. These include

$$\lambda_1 = \frac{\sum\limits_{i \in R_1} \sum\limits_j A_{ij}\psi_j'}{\sum\limits_{i \in R_1} \psi_i'} = \frac{\sum\limits_{i \in R_2} \sum\limits_j A_{ij}\psi_j'}{\sum\limits_{i \in R_2} \psi_i'}, \tag{B5}$$

$$\lambda_2 = \frac{\sum\limits_{i \in R_1} \sum\limits_j A_{ij}\psi_j''}{\sum\limits_{i \in R_1} \psi_i''} = \frac{\sum\limits_{i \in R_2} \sum\limits_j A_{ij}\psi_j''}{\sum\limits_{i \in R_2} \psi_i''}, \tag{B6}$$

all of which involve sums needed for Eq. (23) so they cost nothing extra to compute. Another use of these estimators is monitoring the utility of the regions R_1 and R_2, as an estimator will perform inconsistently if too few particles occupy a region or if the sum of the weights in a given region is small. The most useful regions are those that are a major subset of the positive and negative regions of the second eigenvector.

For the Ising model we obtained some additional useful weight cancellation by exploiting the fact that most of the particles are in fully magnetized states and before using the comb replacing all the particles in the "up" state by a single particle whose weight is the sum of the "up" weights of these particles and doing similarly for the particles in the "down" state.

Because the ω_i' should be positive, after combing we explicitly set each ω_i' equal to its absolute value. The step is useful because even though we initialize ψ' to have all positive components, the weight cancelation procedure can make some of them negative. Once so, they might project out rapidly and may adversely affect the eigenpair estimation until they do.

[1] J. E. Gubernatis and T. E. Booth, J. Comput. Phys. **227**, 8508 (2008).

[2] T. E. Booth, Nucl. Sci. Eng. **143**, 291 (2003).

[3] T. E. Booth, Nucl. Sci. Eng. **154**, 48 (2006).

[4] J. H. Wilkinson, *The Algebraic Eigenvalue Problem* (Oxford University Press, Oxford, 1965).

[5] G. H. Golub and C. F. van Loan, *Matrix Computations* (Johns Hopkins University Press, Baltimore, 1989).

[6] Donald H. Davis, Lawrence Radiation Laboratory Report No. UCRL-6707, Dec 12, 1961.

[7] T. E. Booth, in *1996 Radiation Protection and Shielding Division Topical Meeting*, American Nuclear Society, N. Falmouth, MA, April 1996.

[8] D. M. Arnow, M. H. Kalos, Michael A. Lee, and K. Schmidt, J. Chem. Phys. **77**, 5562 (1982).

[9] T. E. Booth and J. E. Gubernatis, Comput. Phys. Commun. **180**, 509 (2009).

[10] L. Onsager, Phys. Rev. **65**, 117 (1944).

[11] C. J. Thompson, *Mathematical Statisical Mechanics* (Princeton University Press, Princeton, 1972).

[12] For example, H. L. Richards, M. A. Novotny, and P. A. Rikvold, Phys. Rev. B **48**, 14584 (1993); M. Caselle and M. Hasenbusch, Nucl. Phys. B **579**, 667 (2000); P. Grinza and A. Rago, *ibid.* **651**, 387 (2003); M. Caselle, P. Grinza, and A. Rago, J. Stat. Phys.: Theory Exp. (2004), P10009; J. Kaupužs, Int. J. Mod. Phys. C **17**, 1095 (2006).

[13] Y. Kamiya, N. Kawashima, T. E. Booth, and J. E. Gubernatis (unpublished).

[14] E. W. Montroll, J. Chem. Phys. **9**, 706 (1941).

[15] W. J. Camp and M. E. Fisher, Phys. Rev. B **6**, 946 (1972).

[16] H. S. Wilf, *Mathematics for the Physical Sciences* (Dover, New York, 1978).

[17] B. Kaufman, Phys. Rev. **76**, 1232 (1949).

[18] S. Pissanetsky, *Sparse Matrix Technology* (Academic Press, London, 1984).

[19] If complex arithmetic is used, the need to wait for the reality of these roots can be obviated.

[20] If the magnitude of the ground-state eigenvalue is smaller than that of the dominant eigenvalue, convergence to the ground state is reasonably accomplished by appropriately shifting the eigenvalue spectrum, $A \rightarrow A - \sigma I$ [4].

[21] For example, D. M. Ceperley and B. Bernu, J. Chem. Phys. **89**, 6316 (1988); M. P. Nightingale and V. Melik-Alaverdian, Phys. Rev Lett. **87**, 043401 (2001); M. P. Nightingale and P.-N. Roy, J. Phys. Chem. A **110**, 5391 (2006).

[22] B. L. Hammond, W. A. Lester, Jr., and P. J. Reynolds, *Monte Carlo Methods in Ab Initio Chemisry* (World Scientific, Singapore, 1994).

[23] M. P. Nightingale and C. J. Umrigar, in *Monte Carlo Eigenvalue Methods in Quantum Mechanics and Statistical Mechanics*, Advances in Chemical Physics, edited by D. M. Ferguson, J. I. Siepmann, and D. G. Truhlar (John Wiley, New York, 1999), Vol. 105.

[24] For example, D. R. Stump, Nucl. Phys. B **265**, 113 (1986); M. P. Nightingale and H. W. J. Blöte, Phys. Rev. B **33**, 659 (1986); Phys. Rev. Lett. **60**, 1562 (1988); J. M. Thijssen and

H. J. F. Knops, Phys. Rev. B **42**, 2438 (1990); E. Granato and M. P. Nightingale, *ibid.* **48**, 7438 (1993); T. Koma, J. Stat. Phys. **71**, 269 (1993); M. P. Nightingale, and H. W. J. Blöte, Phys. Rev. B **54**, 1001 (1996); M. Hasenbusch, K. Pinn, and K. Rummukainen, Phys. Rev. D **49**, 495 (1994).

[25] M. P. Nightingale and C. J. Umrigar, in *Recent Advances in Quantum Monte Carlo Methods*, edited by W. A. Lester, Jr., (World Scientific, Singapore, 1996); J. Toulouse and C. J. Umrigar, J. Chem. Phys. **126**, 084102 (2007).

[26] M. P. Nightingale and H. W. J. Blöte Phys. Rev. Lett. **76**, 4548 (1996); Phys. Rev. B **62**, 1089 (2000).

[27] B. Rubenstein, J. E. Gubernaits, and J. D. Doll (unpublished).

[28] J. E. Gubernatis and T. E. Booth (unpublished).

[29] E. Y. Loh, J. E. Gubernatis, R. T. Scalettar, S. R. White, D. J. Scalapino, and R. L. Sugar, Phys. Rev. B **41**, 9301 (1990).

[30] T. E. Booth and J. E. Gubernatis, *Improved Criticality Convergence Via a Modified Monte Carlo Power Interation*, in 2009 International Conference on Mathematics, Computational Methods and Reactor Physics (American Nuclear Society, La Grange Park, IL, 2009), CD-ROM.

[31] V. Privman and M. E. Fisher, J. Stat. Phys. **33**, 385 (1983).