BASICS, QUANTUM MONTE CARLO AND STATISTICAL MECHANICS

M.P. NIGHTINGALE
Department of Physics, University of Rhode Island,
Kingston, Rhode Island, 02881.

Abstract. In the first part of these lectures, we review basic Monte Carlo techniques such as the Metropolis algorithm and the estimation of uncertainties. The second part presents Monte Carlo techniques designed to deal with eigenvalue problems. The emphasis here is on general principles.[1] The final, third part of the lectures contains a discussion of an application of these methods to the computation of correlation times of a system at its critical temperature. In this context we also discuss in detail the optimization of variational trial states for the excited states problem.

1. Introduction and basics

1.1. METROPOLIS METHOD

A Monte Carlo process that samples the relative probability distribution $\rho(\mathbf{S})$ of states or configurations $\mathbf{S}$ is usually generated by means of the generalized Metropolis algorithm.[1] This algorithm, which we include here for completeness, can be formulated as follows.

Suppose a configuration $\mathbf{S}$ is given at some time t of the Monte Carlo process. A new configuration $\mathbf{S}'$ at time $t+1$ is generated by means of a stochastic process that consists of two steps. These steps are defined in terms of proposal and acceptance transition probabilities $\mathcal{P}$ and $\mathcal{A}$. Suppose for now that the probabilities $\mathcal{P}$ and $\mathcal{A}$ are given, then:

1. An intermediate configuration $\mathbf{S}''$ is proposed with probability $\mathcal{P}(\mathbf{S}''|\mathbf{S})$;

[1]These lectures are based in part on: M.P. Nightingale and C.J. Umrigar, *Monte Carlo Eigenvalue Methods in Quantum Mechanics and Statistical Mechanics,* Advances in Chemical Physics, Vol. 105, Monte Carlo Methods in Chemistry, edited by David M. Ferguson, J. Ilja Siepmann, and Donald G. Truhlar, series editors I. Prigogine and Stuart A. Rice, Chapter 4 (In press, Wiley, NY 1998)

2. $\mathbf{S}' = \mathbf{S}''$ with probability $p \equiv \mathcal{A}(\mathbf{S}''|\mathbf{S})$, *i.e.,* the proposed configuration is accepted;
3. $\mathbf{S}' = \mathbf{S}$ with probability $q \equiv 1 - \mathcal{A}(\mathbf{S}''|\mathbf{S})$, *i.e.,* the proposed configuration is rejected and the old configuration $\mathbf{S}$ is promoted to time $t+1$.

More explicitly, the Monte Carlo sample is generated by means of a Markov matrix P with elements $P(\mathbf{S}'|\mathbf{S})$ of the form

$$P(\mathbf{S}'|\mathbf{S}) = \begin{cases} \mathcal{A}(\mathbf{S}'|\mathbf{S})\mathcal{P}(\mathbf{S}'|\mathbf{S}) & \text{for } \mathbf{S}' \neq \mathbf{S} \\ 1 - \sum_{\mathbf{S}''\neq\mathbf{S}} \mathcal{A}(\mathbf{S}''|\mathbf{S})\mathcal{P}(\mathbf{S}''|\mathbf{S}) & \text{for } \mathbf{S}' = \mathbf{S}, \end{cases} \tag{1}$$

if the states are discrete, while on the other hand

$$P(\mathbf{S}'|\mathbf{S}) = \mathcal{A}(\mathbf{S}'|\mathbf{S})\mathcal{P}(\mathbf{S}'|\mathbf{S}) + \left[1 - \int d\mathbf{S}''\mathcal{A}(\mathbf{S}''|\mathbf{S})\mathcal{P}(\mathbf{S}''|\mathbf{S})\right]\delta(\mathbf{S}' - \mathbf{S}) \tag{2}$$

if the states are continuous.

The Markov matrix P is designed to satisfy detailed balance

$$P(\mathbf{S}'|\mathbf{S})\rho(\mathbf{S}) = P(\mathbf{S}|\mathbf{S}')\rho(\mathbf{S}'), \tag{3}$$

so that, if the process has a unique stationary distribution, this will be $\rho(\mathbf{S})$, as desired. In principle, one has great freedom in the choice of the proposal matrix $\mathcal{P}$, but it is necessary to satisfy the requirement that transitions can be made between (almost) any pair of states with non-vanishing probability (density) in a finite number of steps.

The choice of the proposal matrix $\mathcal{P}$ is limited only by the requirement that step 1 of the algorithm can be executed efficiently. More complicated choices of $\mathcal{P}$ that are less efficient for each individual step, may still yield a more efficient sampling process, since they may generate a less strongly correlated time series — for more details on the optimization of the proposal transition see Cyrus Umrigar's lectures. Once a proposal matrix $\mathcal{P}$ is selected, an acceptance matrix is defined so that detailed balance, Eq. (3), is satisfied

$$\mathcal{A}(\mathbf{S}'|\mathbf{S}) = \min\left[1, \frac{\mathcal{P}(\mathbf{S}|\mathbf{S}')\rho(\mathbf{S}')}{\mathcal{P}(\mathbf{S}'|\mathbf{S})\rho(\mathbf{S})}\right]. \tag{4}$$

Given a stochastic variable X, the Metropolis algorithm can be used to estimate its expectation value, or ensemble average, $\langle X \rangle$ as the time average

$$\langle X \rangle \equiv \frac{\sum_{\mathbf{S}} X(\mathbf{S})\rho(\mathbf{S})}{\sum_{\mathbf{S}} \rho(\mathbf{S})} \approx \frac{1}{L}\sum_{t=1}^{L} X(\mathbf{S}_t). \tag{5}$$

Alternatively, one can time-average the expectation value conditional on $\mathbf{S}_t''$ having been proposed:

$$\langle X \rangle \approx \frac{1}{L} \sum_{t=1}^{L} \left[p_t X(\mathbf{S}_t'') + q_t X(\mathbf{S}_t) \right], \tag{6}$$

where $p_t = 1 - q_t$ is the probability of accepting $\mathbf{S}_t''$. The latter estimator has the advantage of reducing the statistical error somewhat, since $X(\mathbf{S}_t'')$ contributes to the average even for rejected moves, and this will increase efficiency if $X(\mathbf{S}_t'')$ is readily available.

A useful trick is to keep in mind that it is not necessary to sample the distribution ρ to compute $\langle X \rangle$; any distribution that has sufficient overlap with ρ will do. More explicitly, the following relation shows that the desired results can be obtained by reweighting, *i.e.*,

$$\langle X \rangle_\rho = \frac{\langle X\rho/\rho' \rangle_{\rho'}}{\langle \rho/\rho' \rangle_{\rho'}}, \tag{7}$$

where the subscript of the expectation value indicates the probability distribution with respect to which it is defined. In other words, to compute $\langle X \rangle_\rho$, one can sample any distribution ρ', as long as the ratio $\rho(\mathbf{S})/\rho'(\mathbf{S})$ does not fluctuate too wildly. This is particularly useful for calculation of the difference of expectation values with respect to two closely related distributions. An example of this [2, 3] is the calculation of the energy of a molecule as a function of the inter-nuclear distance. Also, Eq. (7) forms the basis of the histogram method.[4]

The Metropolis method is a general and flexible method to sample an arbitrary distribution. All that is required for its use is that one can compute efficiently the value assumed by the distribution for any configuration. The limitation of the method is that it generates a time series of configurations with serial correlations. Consequently, the number of independent observations contained in a time series $\mathbf{S}_1, \ldots, \mathbf{S}_L$ is not given by L, but roughly by $L_{\text{eff}} \approx L/\tau$ where τ is an appropriately defined auto-correlation time of the series. In other words, the variance of the finite-L time average featured in Eq. (5) is not given by $\text{var}(X)/L$ but approximately by $\text{var}(X)/L_{\text{eff}}$. Of course, the effective number of independent observations L_{eff} is not a priori known, and this introduces some problems in the computation of error bars, as will be discussed next.

1.2. MONTE CARLO AVERAGES AND UNCERTAINTIES

In *Guidelines for evaluating and expressing the uncertainty of NIST measurement results,* the authors state[5]:

> In general, the result of a measurement is an approximation of the value of the specific quantity subject to measurement, that is the **measurand,** and thus the result is complete only when accompanied by a quantitative statement of its uncertainty.

In some circles Monte Carlo methods are frowned upon, and at least in part this is because the preceding statement is all too often ignored. Thus it seems appropriate to review briefly how averages and error bars are computed, and what ideally they should mean.

In the simplest case, the measurand X is estimated from L independent and equally distributed observations X_k. Under those circumstances, with the sample mean given by

$$x_L = \frac{1}{L} \sum_{k=1}^{L} X_k, \tag{8}$$

the *standard uncertainty* $u(x_L)$ is the estimated standard deviation of the mean

$$u(x_L) = \sigma(x_L) = \sqrt{\frac{1}{L(L-1)} \sum_{k=1}^{L} (X_k - x_L)^2}. \tag{9}$$

In terms of these definitions, one can summarize that $X = x_L \pm u(x_L)$, but the trouble with the statement "$X = x_L \pm u(x_L)$" is that it is not always clear what is meant. The casual interpretation is that $X \approx x_L$ with an error on the order of $u(x_L)$. The desired interpretation of this statement in a Monte Carlo context is that L is large enough and that the distribution of the X_k is sufficiently well-behaved that the Central Limit Theorem applies. More explicitly, if we introduce the expanded uncertainty $U_p(x_L) = k_p u(x_L)$, "$X = x_L \pm u(x_L)$" means "$x_L - U_p(x_L) < X < x_L + U_p(x_L)$ is believed to have an approximate level of confidence $p = \mathrm{erf}(k_p/\sqrt{2})$," for reasonable choices of k_p, such as $k_{0.68} \approx 1$ or $k_{0.95} \approx 2$. In other words, the probability of being wrong by one or two error bars, is respectively 32% and 5%.

1.3. BLOCKING

In Monte Carlo practice there are several problems. First of all, the observations are rarely independent. This means that the standard uncertainty $u(x_L)$ cannot be obtained by means of Eq. (9), which usually produces an underestimate. Blocking is a commonly used method to compute the standard deviation of the mean $\sigma(x_L)$ in this case.

Blocking works as follows — see Ref. [6] for a more detailed discussion. Given a stationary series $X_i^{(1)}$ with $i = 1, \ldots, L^{(1)} \equiv 2^n$, compute recursively the non-overlapping moving averages, or block averages

$$X_{i'}^{(b+1)} = \frac{1}{2}(X_{2i'-1}^{(b)} + X_{2i'}^{(b)}) \tag{10}$$

with $i' = 1, \ldots, L^{(b+1)}$, where $L^{(b+1)} = L^{(b)}/2$. For simplicity we assume that $L^{(1)}$ is a power of 2; typically, $L \approx 2^{10}$, which in practice means that most likely the $X_i^{(1)}$ themselves are already block averages. One then proceeds on basis of the observation that, in all but pathological cases, as the blocking level b increases, the $X_i^{(b)}$ approach independent Gaussian stochastic variables. Clearly, the sample mean is invariant under the blocking transformation. The standard deviation of the mean $\sigma(x^{(b)})$ can be computed by applying Eq. (9) to the block variables, and for independent variables also the expectation value of the variance is invariant under blocking. In that case, one obtains as an estimate of the uncertainty

$$u(x_{L^{(b)}}^{(b)}) = \sigma(x_{L^{(b)}}) \pm \Delta^{(b)}, \tag{11}$$

where

$$\Delta^{(b)} = \frac{\sigma(x_{L^{(b)}}^{(b)})}{\sqrt{2(L^{(b)} - 1)}} \tag{12}$$

For correlated data, blocking typically yields increasingly uncertain estimates of the standard deviation of the mean. The best estimate is obtained for the smallest blocking level beyond which there is a plateau and no further changes occur in excess of the estimated uncertainty of the $u(xL^{(b)})$, as illustrated by the fake error estimates and their error bars in Fig. 1. A simple algorithm to find this value is the following: Compute how many error bars to the right of each error bar fail to differ significantly from the one under consideration. This yields a sequence of numbers

$$c_b = \text{number of } b' \text{ s.t. } b < b' \leq n \text{ and } |u(x^{(b)}) - u(x^{(b')})| < \Delta^{(b)} \tag{13}$$

for $b = 1, \ldots, n-1$. If the block averages $X_i^{(b)}$ have no significant serial correlations for larger values of the blocking level b then the numbers c_b will have a maximum, at that value of b that marks the edge the plateau, beyond which the numbers c_b keep decreasing as a function of b. It is an ominous sign, indicative of insufficient statistics, if that value is close to the maximum blocking level n.

If the Monte Carlo run is long enough, the preceding procedure yields all the desired results, but in practice, one frequently deals with problems close to the limit of what can be done reliably. Under those circumstances, it is not always clear that the results can be trusted. It can then be helpful

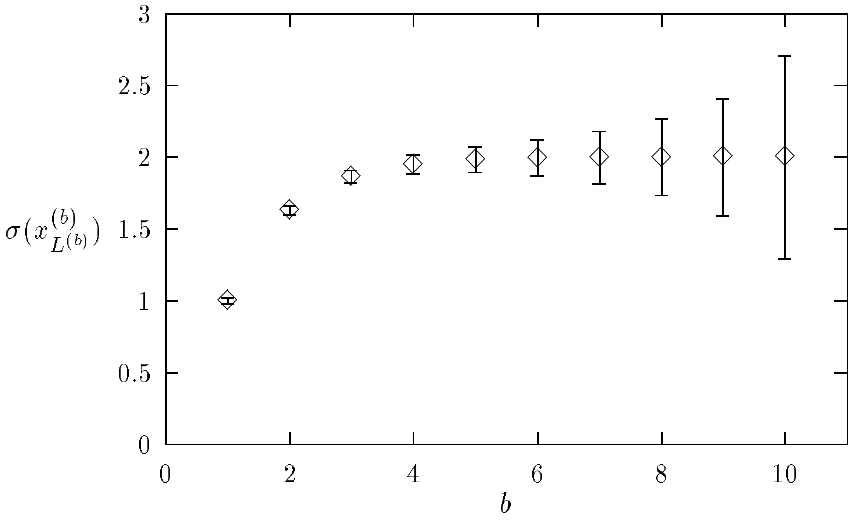


Figure 1. Plateau of fake block variances and their error bars.

to have additional indicators of impending doom. A simple quantity that can be used in this context is the correlation coefficient r, which yields an estimate of the magnitude of the serial correlations of the observations

$$r = \frac{\sum_{i=2}^{L} \Delta X_{i-1} \Delta X_i}{\sum_{i=2}^{L} \Delta X_{i-1}^2}, \tag{14}$$

where $\Delta X_i = X_i - x_L$. The uncertainty in r can be estimated as

$$u(r) \approx \sqrt{\frac{\sum_{i=2}^{L} (\Delta x_i - r\Delta x_{i-1})^2}{(L-3)\sum_{i=2}^{L} \Delta X_{i-1}^2}}. \tag{15}$$

1.4. NON-LINEARITY BIAS

Sometimes the quantity to be estimated is a non-linear function of one or more averages. For example, one computes a weighted average and the weights have an unknown normalization constant, as in Eq. (7). In that case, the quantity of interest is a ratio of two averages. Or one might be interested in the eigenvalues of a matrix the elements of which are estimated as Monte Carlo averages. In such cases involving non-linearity there is a minor complication. Suppose,*e.g.*, that an estimate of a function f of the expectation value $\bar{x}$ is required. An unbiased estimator of x is given by

Eq. (8), but the expectation value of $f(x_L)$ has a bias

$$\langle f(x_L)\rangle \approx f(\langle x_L\rangle) + \frac{1}{2L}\mathrm{var}(x_L) f''(x_L). \tag{16}$$

Clearly, the estimate of $f(\langle x\rangle)$ obtained by applying f to the overall average x_L is the least biased estimate available, since it has a smaller bias than the estimate obtained by averaging $f(X_i^{(b)})$ over all blocks. Yet, useful estimates of variance can still be obtained from the variance of the latter even though they are biased.

1.5. THE FINAL LEAP FROM ART TO SCIENCE

Now that we have discussed how to obtain an estimate of the standard deviation of the mean, all that remains to be done before one can make a "quantitative statement of the uncertainty of the measurand" is to verify that the distribution of the block averages $X_i^{(b)}$ is Gaussian on the plateau. For this purpose one can, *e.g.*, use a variant of the Kolmogorov-Smirnov test, which is a standard test used to determine if a given set of data points $x_1, \ldots, x_L$ consisting of independent observations of a real-valued stochastic variable is likely to have been sampled from a given probability distribution.[7] Suppose that the latter has cumulative distribution $P(x)$. Define

$$S(x) = \frac{\text{number of } i \text{ such that } x_i \le x}{L}. \tag{17}$$

The Kolmogorov-Smirnov statistic D is

$$D = \max_{-\infty<x<\infty} |P(x) - S(x)|. \tag{18}$$

If indeed P is given a priori, the probability distribution of D is known, at least to good approximation, so that one can determine whether any observed value of D is compatible with the presumed distribution of the data points. The trouble is that in our case the mean and the variance of the Gaussian are not known a priori, and have to be estimated from the data; likewise, P is not completely determined. This means that the text-book distribution of D mentioned above is not applicable.

A particularly simple variant of the standard procedure is the following popular application of Monte Carlo:

1. Compute mean and variance in the standard way,[8] or use any of the alternatives discussed below.
2. Compute the Kolmogorov-Smirnov statistic D with Eq. (18).
3. Repeat the same procedure with Gaussian random numbers and use Monte Carlo to estimate the probability to that the value of the Kolmogorov-Smirnov statistic exceeds the value D obtained in step 2.

As usual, the estimate obtained in step 3 can be used to determine if the observed value of D is compatible with the hypothesis of a Gaussian distribution. A problem with this procedure is that it is not likely to be sensitive to dangerous outliers. Under the best circumstances, this is a problem of the Kolmogorov-Smirnov statistic, but here it is exacerbated by the fact that mean and variance are estimated from the very data under investigation.

The goal of the test is to detect evidence of outliers. However, if one uses the usual estimators of mean and variance of the distribution whose normality is being tested, outliers are given considerable weight in estimating these parameters of the distribution, which makes detection less likely. A reasonable alternative might be to use the median as an estimate of the mean, and in the same spirit, to estimate the variance one may compute the median $\Delta x_{\rm med}$ of $|x_i - x|$ in terms of which asymptotically $\sigma \approx 0.674 \Delta x_{\rm med}$.

Of course, the Kolmogorov-Smirnov statistic is only one of many statistics that can be computed to measure the difference of the distributions P and S. See Ref.[7] for alternative statistics and references.

2. The power method

Before discussing technical details of Monte Carlo methods to compute eigenvalues and expectation values, we introduce the mathematical ideas and the types of expressions for which statistical estimates are sought. We formulate the problem in terms of an operator G of which one wants to compute the dominant eigenstate and eigenvalue, $|\psi_0\rangle$ and λ_0.

We assume throughout that we are dealing with states represented by real numbers and that also X is represented by a real, symmetric matrix. In many cases, generalization to complex numbers is trivial, but for some physical problems, while formal generalization may still be possible, the resulting Monte Carlo algorithms may be too noisy for practical use. Mathematically, *dominant* may mean dominant relative to eigenstates of a given symmetry only. In the context of Monte Carlo, this distinction again may have significant impact on the utility of the algorithm.

The methods we shall discuss are variations of the power method. This method relies on the fact that for a generic initial state $|u^{(0)}\rangle$ of the appropriate symmetry, the states $|u^{(t)}\rangle$ defined by

$$|u^{(t+1)}\rangle = \frac{1}{c_{t+1}} G |u^{(t)}\rangle \tag{19}$$

converge to the dominant eigenstate $|\psi_0\rangle$ of G, if the constants c_t are chosen so that the $|u^{(t)}\rangle$ assume any convenient standard form, in which case the constants c_t converge to the dominant eigenvalue. One possible standard form to define the constant c_t in Eq. (19) might be that, in some representation, the component of $|u^{(t)}\rangle$ largest in magnitude equals unity.

This convergence property of the power method follows immediately by expanding the initial state $|u^{(0)}\rangle$ in unnormalized eigenstates $|\psi_k\rangle$ of G. Then, for general t one has

$$\begin{aligned} |u^{(t)}\rangle &= \sum_k \lambda_k^t |\psi_k\rangle \\ &= \lambda_0^t |\psi_0\rangle [1 + \mathcal{O}(|\frac{\lambda_1}{\lambda_0}|^t)], \end{aligned}$$

if $|\lambda_0| > |\lambda_1| \geq \ldots$. Note that the expansion coefficients at $t = 0$ have been absorbed in the definition of the states $|\psi_k\rangle$.

For quantum mechanical systems, one applies the power method to the operator G which frequently is the imaginary-time evolution operator, $\exp(-\tau\mathcal{H})$, where $\mathcal{H}$ is the Hamiltonian. A technical problem in that case is that an explicit expression is known only asymptotically for short times τ. In practice, this asymptotic expression is used for a small but finite τ and this leads to systematic, time-step errors.

The exponential operator $\exp(-\tau\mathcal{H})$ is only one of various alternatives that can be employed to compute the ground-state properties of the Hamiltonian. If the latter is bounded from above, which frequently is the case for lattice systems, one may be able to use $\mathbb{1} - \tau\mathcal{H}$, where τ should be small enough that $\lambda_0 \equiv 1 - \tau E_0$ is the dominant eigenvalue of $\mathbb{1} - \tau\mathcal{H}$. In this case, there is no time-step error and the same holds for yet another method of inverting the spectrum of the Hamiltonian, *viz.* the Green function Monte Carlo method. There one uses $(\mathcal{H} - E)^{-1}$, where E is a constant chosen so that the ground state becomes the dominant eigenstate of this operator. In a Monte Carlo context, matrix elements of the respective operators are proportional to transition probabilities and therefore have to be non-negative, which, if one uses either of the last two methods, may impose further restrictions on the values of τ and E.

In the last part of these lectures, we discuss a statistical mechanical application of these methods to compute eigenvalues of the Markov matrix governing critical dynamics. The operator that plays the role of G is indeed an evolution operator defining dynamics of the system. We note in passing that the transfer matrix[9], which will not be discussed here in detail, evolves the physical system in a spatial rather than time direction, but this spatial direction corresponds to time from the point of view of a Monte Carlo time series.

Now, suppose that X is an operator of which one wants to compute an expectation value. Particularly simple to deal with are the cases in which the operators X and the evolution operator G are the same or commute.

We introduce the following notation. Suppose that $|u_\alpha\rangle$ and $|u_\beta\rangle$ are two states, then $X_{\alpha\beta}^{(p',p)}$ denotes the somewhat unusual matrix element

$$X_{\alpha\beta}^{(p',p)} = \frac{\langle u_\alpha|G^{p'} X G^{p}|u_\beta\rangle}{\langle u_\alpha|G^{p'+p}|u_\beta\rangle}. \tag{20}$$

The above definition is chosen, since it contains special cases of practical interest, but matrix elements used in physical applications of course are not usually of this form. We chose this form for simplicity of presentation, and since some form of normalization is required. The methods discussed here can be generalized immediately to more usual normalization.

The Monte Carlo methods to be discussed are designed to estimate particular instances of $X_{\alpha\beta}^{(p',p)}$, and often the ultimate goal is to compute the expectation value in the dominant eigenstate

$$X_0 = \frac{\langle\psi_0|X|\psi_0\rangle}{\langle\psi_0|\psi_0\rangle}, \tag{21}$$

which reduces to an expression for the dominant eigenvalue of interest if one chooses for X, the Hamiltonian, transfer or Markov matrix.

The simplest method is variational Monte Carlo, discussed in the next section. Here an approximate expectation value is computed by employing an approximate eigenstate of G. Typically, this is an optimized trial state, say $|u_{\rm T}\rangle$, in which case variational Monte Carlo yields $X_{\rm TT}^{(0,0)}$, which is simply the expectation value of X in this trial state. Clearly, this variational Monte Carlo estimate of X_0 has both systematic and statistical errors.

The variational error can be removed asymptotically by projecting out the dominant eigenstate, *i.e.,* by reducing the spectral weight of sub-dominant eigenstates by means of the power method. The simplest case is obtained if one applies the power method only to the right on the state $|u_\beta\rangle$ but not to the left on $\langle u_\alpha|$ in Eq. (20). Mathematically, this is the essence of diffusion and transfer matrix Monte Carlo, and in this way one obtains the desired result X_0 if the operator X commutes with the evolution operator G.

In our notation, this means that X_0 is given by the statistical estimate of $X_{\rm TT}^{(0,\infty)}$. In principle, this yields an unbiased estimate of X_0, but in practice one has to choose p finite but large enough that the estimated systematic error is less than the statistical error. In practical situations it can be quite difficult to ascertain that this indeed is the case. If one is interested in the limiting behavior for infinite p or p', the states $|u_\alpha\rangle$ and $|u_\beta\rangle$ need not be available in closed form, and may simply be whatever the Monte Carlo algorithm happens to generate, as long as the required symmetry is respected. This freedom translates into the flexibility in algorithm design exploited in diffusion and transfer matrix Monte Carlo.[9]

If G and X do not commute, the mixed estimator $X_{\rm TT}^{(0,\infty)}$ is not the desired result, but the residual systematic error can be reduced by combining the variational and mixed estimates by means of the expression

$$X_0 = 2X_{\rm TT}^{(0,\infty)} - X_{\rm TT}^{(0,0)} + \mathcal{O}[(|\psi_0\rangle - |u_{\rm T}\rangle)^2]. \tag{22}$$

To remove the variational bias systematically, if G and X do not commute, the power method must be used to both the left and the right in Eq. (20). Thus one obtains from $X_{\rm TT}^{(\infty,\infty)}$ an exact estimate of X_0, subject only to statistical errors. Of course, one has to pay the price of the implied double limit in terms of loss of statistical accuracy. In the context of the Monte Carlo algorithms discussed below this double projection technique to estimate $X_{\rm TT}^{(\infty,\infty)}$ is called *forward walking* or *future walking*, although personally I prefer *backward saving,* for reasons that will become clear.

We end this section on the power method with a brief discussion of the computational complexity of Monte Carlo methods for eigenvalue problems. In general, one can distinguish operations of three levels of computational complexity, depending on whether the operations have to do with single particles or lattice sites, the system as a whole, or state space summation or integration. The first typically involves a fixed number of elementary arithmetic operations, whereas this number clearly is at least proportional to the system size in the second case. Exhaustive state-space summation grows exponentially in the total system size, and for these problems Monte Carlo is often the only viable option.

Next, the convergence of the power method itself comes into play. The number of iterations required to reach a certain given accuracy is proportional to $\log|\lambda_0/\lambda_1|$, where the λ_0 and λ_1 are the eigenvalues of largest and second-largest magnitude. If one is dealing with a single-site transfer matrix of a critical system, that spectral gap is proportional to L^{-d} for a system in d dimensions with a cross section of linear dimension L. In this case, a single matrix multiplication is of the complexity of a one-particle problem. In contrast, both for the Markov matrix mentioned above, and the quantum mechanical evolution operator, the matrix multiplication itself is of system-size complexity. Moreover, both of these operators have their own specific problems. The quantum evolution operator of $G(\tau)$ has a gap on the order of τ, which means that τ should be chosen large for rapid convergence, but one does not obtain the correct results, because of the time-step error, unless τ is small. Finally, the spectrum of the Markov matrix may be determined by critical slowing down. This means that the gap of a single spin-flip matrix is on the order of $L^{-(z+d)}$, where z is typically a little bigger than two.[10] These convergence properties are well understood in terms of the mathematics of the power method. Not well understood, however, are problems that are specific to the Monte Carlo implementation of this method, which in some form or another introduces multiplicative fluctuating weights that are correlated with the quantities of interest.[11, 12]

3. Single-Thread Monte Carlo

In the previous section we have presented the mathematical expressions that can be evaluated with the Monte Carlo algorithms to be discussed next. The first algorithm we discuss is designed to compute an approximate statistical estimate of the matrix element X_0 by means of the variational estimate $X_{\mathrm{TT}}^{(0,0)}$. We write $\langle \mathbf{S}|u_{\mathrm{T}}\rangle = \langle u_{\mathrm{T}}|\mathbf{S}\rangle \equiv u_{\mathrm{T}}(\mathbf{S})$ and for non-vanishing $u_{\mathrm{T}}(\mathbf{S})$ define the *configurational eigenvalue* $X_{\mathrm{T}}(\mathbf{S})$ by

$$X_{\mathrm{T}}(\mathbf{S})u_{\mathrm{T}}(\mathbf{S}) \equiv \langle \mathbf{S}|X|u_{\mathrm{T}}\rangle = \sum_{\mathbf{S}'}\langle \mathbf{S}|X|\mathbf{S}'\rangle\langle \mathbf{S}'|u_{\mathrm{T}}\rangle. \tag{23}$$

This yields

$$X_{\mathrm{TT}}^{(0,0)} = \frac{\sum_{\mathbf{S}} u_{\mathrm{T}}(\mathbf{S})^2 X_{\mathrm{T}}(\mathbf{S})}{\sum_{\mathbf{S}} u_{\mathrm{T}}(\mathbf{S})^2}, \tag{24}$$

which shows that $X_{\mathrm{TT}}^{(0,0)}$ can be evaluated as a time average over a Monte Carlo time series of states $\mathbf{S}_1, \mathbf{S}_2, \ldots$ sampled from the probability distribution $u_{\mathrm{T}}(\mathbf{S})^2$, *i.e.*, a process in which $\mathrm{Prob}(\mathbf{S})$, the probability of finding a state $\mathbf{S}$ at any time, is given by

$$\mathrm{Prob}(\mathbf{S}) \propto u_{\mathrm{T}}(\mathbf{S})^2. \tag{25}$$

For such a process, the ensemble average in Eq. (24) can be written in the form of a time average

$$X_{\mathrm{TT}}^{(0,0)} = \lim_{L\to\infty} \frac{1}{L}\sum_{t=1}^{L} X_{\mathrm{T}}(\mathbf{S}_t). \tag{26}$$

which is precisely an example of a problem for which the Metropolis algorithm can be applied [*cf.* Eq. (5)].

For the time average in Eq. (26) to be of practical use, it has to be assumed that the configurational eigenvalue $X_{\mathrm{T}}(\mathbf{S})$ can be computed efficiently, which is the case if the sum over states $\mathbf{S}'$ on the right-hand side of Eq. (23) can be performed explicitly. For discrete states this means that X should be represented by a sparse matrix; if the states $\mathbf{S}$ form a continuum, $X_{\mathrm{T}}(\mathbf{S})$ can be computed directly if X is diagonal, or sparse, or *near-diagonal*, *i.e.,* it involves no or only low-order derivatives in the representation used. The more complicated case of an operator X with arbitrarily non-vanishing off-diagonal elements will be discussed at the end of this section.

An important special case, relevant for example to electronic structure calculations, is to choose for the operator X the Hamiltonian $\mathcal{H}$ itself, and for $\mathbf{S}$ the $3N$-dimensional real-space configuration of the system. In that special case, the quantity X_{T} is called the *local energy,* denoted by $\mathcal{E}_{\mathrm{T}}$. Clearly, in the ideal case, $|u_{\mathrm{T}}\rangle$ is an exact eigenstate of the evolution operator G. If in addition X commutes with G, then the configurational eigenvalue $X_{\mathrm{T}}(\mathbf{S})$ is a constant independent of $\mathbf{S}$ and equals the true eigenvalue of X. In this ideal case, the variance of the Monte Carlo estimator in Eq. (26) vanishes, which is an important zero-variance principle satisfied by variational Monte Carlo. The practical implication is that the efficiency of the Monte Carlo computation of the energy can be improved arbitrarily by improving the quality of the trial function. Of course, usually the time required for the computation of $u_{\mathrm{T}}(\mathbf{S})$ increases as the approximation becomes more sophisticated. For the energy, the optimal choice minimizes the product of variance and time; no such optimum is defined for an operator that does not commute with G, or if one makes the commonly employed fixed-node approximation, since in those cases the results have a systematic error that depends on the quality of the trial wave function.

3.1. PROJECTOR MONTE CARLO AND IMPORTANCE SAMPLING

As we saw, the generalized Metropolis method is a very powerful way to sample an arbitrary *given* distribution, and it allows one to construct infinitely many Markov processes with the desired distribution as the stationary state. None of these processes, however, may be appropriate to design a Monte Carlo version of the power method to solve eigenvalue problems. In such applications, the evolution operator G is *given,* possibly in approximate form, and its dominant eigenstate may *not* be known.

To construct an appropriate Monte Carlo process, the first problem is that G itself is not a Markov matrix, *i.e.*, it may violate one or both of the properties $G(\mathbf{S}'|\mathbf{S}) \geq 0$ and $\sum_{\mathbf{S}'} G(\mathbf{S}'|\mathbf{S}) = 1$. This problem can be solved in principle if we can find a factorization of the evolution matrix G into a Markov matrix P and a weight matrix g with non-negative elements such that

$$G(\mathbf{S}'|\mathbf{S}) = g(\mathbf{S}'|\mathbf{S})P(\mathbf{S}'|\mathbf{S}). \tag{27}$$

The weights g must be finite, and this almost always precludes use of the Metropolis method for continuous systems, as can be understood as follows. Since there is a finite probability that a proposed state will be rejected, the Markov matrix $P(\mathbf{S}'|\mathbf{S})$ will contain terms involving $\delta(\mathbf{S} - \mathbf{S}')$, as present in Eq. (2), but generically G will not have the same structure, and will not allow the definition of finite weights g according to Eq. (27).

As a comment on the side we note that violation of the condition that the weight g be positive results in the notorious *sign problem* in one of its guises. This problem is in most cases unavoidable in the treatment of fermionic or frustrated systems. The reader is referred to Mal Kalos's lectures for further details. In other words, we restrict ourselves to the case of an evolution operator G with nonnegative matrix elements only.

We resume our discussion of the factorization given in Eq. (27). Suppose for the sake of argument that the left eigenstate $\hat{\psi}_0$ of G is known and that its elements are positive,

$$\sum_{\mathbf{S}'} \hat{\psi}_0(\mathbf{S}')G(\mathbf{S}'|\mathbf{S}) \;=\; \lambda_0 \hat{\psi}_0(\mathbf{S}). \tag{28}$$

If in addition, the matrix elements of G are nonnegative, the following matrix $\hat{P}$ is a Markov matrix

$$\hat{P}(\mathbf{S}'|\mathbf{S}) \;=\; \frac{1}{\lambda_0}\hat{\psi}_0(\mathbf{S}')G(\mathbf{S}'|\mathbf{S})\frac{1}{\hat{\psi}_0(\mathbf{S})}. \tag{29}$$

Unless one is dealing with a Markov matrix from the outset, the left eigenstate of G is seldom known, but it is convenient in any event, to perform a so-called *importance sampling* transformation on G. For this purpose we introduce a guiding function $u_{\rm g}$ and define

$$\hat{G}(\mathbf{S}'|\mathbf{S}) \;=\; u_{\rm g}(\mathbf{S}')G(\mathbf{S}'|\mathbf{S})\frac{1}{u_{\rm g}(\mathbf{S})}. \tag{30}$$

We shall return to the issue of the guiding function, but for the time being the reader can think of it either as an arbitrary, positive function, or as an approximation to the dominant eigenstate of G. From a mathematical point of view, anything that can be computed with the original evolution operator G, can also be computed with $\hat{G}$, since the two represent the same abstract operator in a different basis, differing from the original one only in terms of normalization of the basis states.

We continue our discussion in terms of the transform $\hat{G}$ and replace Eq. (27) by the factorization

$$\hat{G}(\mathbf{S}'|\mathbf{S}) \;=\; \hat{g}(\mathbf{S}'|\mathbf{S})\hat{P}(\mathbf{S}'|\mathbf{S}). \tag{31}$$

We assume that $\hat{P}$ has the explicitly known distribution $u_{\rm g}^2$ as its stationary state, used to compute quantities of physical interest. Of course, in the latter the guiding function $u_{\rm g}$ might not appear. Yet, even then the *estimator* contains $u_{\rm g}^2$, as follows from the reweighting procedure given in Eq. (7). One can also construct stochastic processes for which the stationary state is not known explicitly, but the fact that the stationary state may not be known is immaterial if one is interested in the infinite projection limit $p \to \infty$. Clearly, to be applicable to such cases, the expressions below should be rewritten so that the unknown distribution does not appear in estimators. A guiding function $u_{\rm g}$ may still appear, but only as a transformation known in closed form and no longer as the stationary state of $\hat{P}$. On the other hand, a process for which the distribution is not known in closed form cannot be used to compute the matrix elements $X_{\alpha\beta}^{(p',p)}$ for finite p and p' and *given* states $|u_\alpha\rangle$ and $|u_\beta\rangle$. We shall return to this issue and shall be in a better position to discuss it more explicitly once the Monte Carlo process will have been described in more detail.

One possible choice for $\hat{P}$ that can be used if G is sufficiently sparse, avoids the Metropolis method, and produces finite weights, is the following *generalized heat-bath* transition matrix

$$\hat{P}(\mathbf{S}'|\mathbf{S}) = \frac{\hat{G}(\mathbf{S}'|\mathbf{S})}{\sum_{\mathbf{S}_1} \hat{G}(\mathbf{S}_1|\mathbf{S})}. \tag{32}$$

If $G(\mathbf{S}'|\mathbf{S})$ is symmetric, this transition matrix has a known stationary distribution, *viz.,* $G_{\rm g}(\mathbf{S})u_{\rm g}^2(\mathbf{S})$, where $G_{\rm g}(\mathbf{S}) = \langle \mathbf{S}|G|u_{\rm g}\rangle/u_{\rm g}(\mathbf{S})$, the configurational eigenvalue of G in state $\mathbf{S}$. If one insists on having a zero-variance principle —which means that the statistical noise vanishes in the limit of importance sampling with the exact dominant eigenstate— $\hat{P}$ must be chosen so that the corresponding transitions can be sampled directly. This is usually not feasible, unless $\hat{P}$ is sparse or near-diagonal, or can be transformed into a form involving non-interacting degrees of freedom. We note that if $\hat{P}$ is defined by Eq. (32), the weight matrix $\hat{g}$ depends only on $\mathbf{S}$.

3.2. MATRIX ELEMENTS

We now address the computation of the matrix elements $X_{\alpha\beta}^{(p',p)}$, assuming that the stationary state $u_{\rm g}^2$ is known explicitly and that the weight matrix $\hat{g}$ has finite elements. We shall discuss the following increasingly complex possibilities: (a) $[X, G] = 0$ and X is near-diagonal in the $\mathbf{S}$ representation; (b) X is diagonal in the $\mathbf{S}$ representation; (c) $X(\mathbf{S}|\mathbf{S}')$ is truly off-diagonal. In the discussion of case (c), we introduce the concept of *side walks.* These can be used to compute matrix elements of a more general nature than discussed up to that point. After deriving the expressions, we shall discuss some of the practical problems, and ways to reduce the variance of the statistical estimators.

3.2.1. $[X, G] = 0$ *and* X *Near-Diagonal*

In this case, $X_{\alpha\beta}^{(0,p'+p)} = X_{\alpha\beta}^{(p',p)}$ and it suffices to consider the computation of $X_{\alpha\beta}^{(0,p)}$. By repeated insertion in Eq. (20) of the resolution of the identity in the $\mathbf{S}$-basis, one obtains the expression

$$X_{\alpha\beta}^{(0,p)} = \frac{\sum_{\mathbf{S}_p,\ldots,\mathbf{S}_0} u_\alpha(\mathbf{S}_p) X_\alpha(\mathbf{S}_p) [\prod_{i=0}^{p-1} G(\mathbf{S}_{i+1}|\mathbf{S}_i)] u_\beta(\mathbf{S}_0)}{\sum_{\mathbf{S}_p,\ldots,\mathbf{S}_0} u_\alpha(\mathbf{S}_p) [\prod_{i=0}^{p-1} G(\mathbf{S}_{i+1}|\mathbf{S}_i)] u_\beta(\mathbf{S}_0)}. \tag{33}$$

The pivotal property that allows one to transform the previous ensemble average into a time average is that in the steady state, a series of subsequent states $\mathbf{S}_t, \mathbf{S}_{t+1}, \ldots, \mathbf{S}_{t+p}$ occurs with known probability

$$\mathrm{Prob}(\mathbf{S}_t, \mathbf{S}_{t+1}, \ldots, \mathbf{S}_{t+p}) \propto [\prod_{i=0}^{p-1} \hat{P}(\mathbf{S}_{t+i+1}|\mathbf{S}_{t+i})] u_{\mathrm{g}}(\mathbf{S}_t)^2. \tag{34}$$

To deal with factors that do appear in products of the factors G but in products of not factors of P, it is convenient to introduce the following definitions

$$\hat{W}_t(p, q) = \prod_{i=q}^{p-1} \hat{g}(\mathbf{S}_{t+i+1}|\mathbf{S}_{t+i}). \tag{35}$$

Also, we define

$$\hat{u}_\omega(\mathbf{S}) = \frac{u_\omega(\mathbf{S})}{u_{\mathrm{g}}(\mathbf{S})}, \tag{36}$$

where ω can be any of a number of subscripts. With these definitions, combining Eqs. (27), (33), and (34), one finds

$$X_{\alpha\beta}^{(0,p)} = \lim_{L\to\infty} \frac{\sum_{t=1}^{L} \hat{u}_\alpha(\mathbf{S}_{t+p}) X_\alpha(\mathbf{S}_{t+p}) \hat{W}_t(p, 0) \hat{u}_\beta(\mathbf{S}_t)}{\sum_{t=1}^{L} \hat{u}_\alpha(\mathbf{S}_{t+p}) \hat{W}_t(p, 0) \hat{u}_\beta(\mathbf{S}_t)}. \tag{37}$$

Sometimes, as mentioned before, one can drop the requirement that one sample a distribution u_{g}^2 known in closed form. For instance, if the $\hat{u}_\alpha$ in Eq. (37) is replaced by u_α and if one defines u_β such that $u_\beta/u_{\mathrm{g}}^2 \equiv 1$, then one can use the resulting expression to compute mixed ground state expectation values. This is precisely what is done in diffusion Monte Carlo.

3.2.2. *Diagonal X*

The preceding discussion can be generalized straightforwardly to the case in which X is diagonal in the $\mathbf{S}$ representation. By repeated insertion of the resolution of the identity in the $\mathbf{S}$-basis in the Eq. (20) for $X_{\alpha\beta}^{(p',p)}$, one obtains the identity

$$X_{\alpha\beta}^{(p',p)} = \frac{\sum_{\mathbf{S}_{p'+p},\ldots,\mathbf{S}_0} u_\alpha(\mathbf{S}_{p'+p})[\prod_{i=p}^{p'+p-1} G(\mathbf{S}_{i+1}|\mathbf{S}_i)]X(\mathbf{S}_p|\mathbf{S}_p)[\prod_{i=0}^{p-1} G(\mathbf{S}_{i+1}|\mathbf{S}_i)]u_\beta(\mathbf{S}_0)}{\sum_{\mathbf{S}_{p'+p},\ldots,\mathbf{S}_0} u_\alpha(\mathbf{S}_{p'+p})[\prod_{i=0}^{p'+p-1} G(\mathbf{S}_{i+1}|\mathbf{S}_i)]u_\beta(\mathbf{S}_0)}. \tag{38}$$

Once again by virtue of Eq. (34), we find

$$X_{\alpha\beta}^{(p',p)} = \lim_{L\to\infty} \frac{\sum_{t=1}^{L} \hat{u}_\alpha(\mathbf{S}_{t+p'+p})\hat{W}_t(p'+p,p)X(\mathbf{S}_{t+p}|\mathbf{S}_{t+p})\hat{W}_t(p,0)\hat{u}_\beta(\mathbf{S}_t)}{\sum_{t=1}^{L} \hat{u}_\alpha(\mathbf{S}_{t+p'+p})\hat{W}_t(p'+p,0)\hat{u}_\beta(\mathbf{S}_t)}, \tag{39}$$

where the factors of G that appear on either side of the operator X in the matrix element Eq. (20) are partially absorbed in the probabilities, while the remainder appears in the reweighting factors $\hat{W}$.

3.2.3. *Nondiagonal X*

In the special case that the matrix elements of G vanish only if those of X do, the preceding method can be generalized immediately to the final case in which X is non-diagonal. First introduce the transform

$$\hat{X}(\mathbf{S}'|\mathbf{S}) = u_{\mathrm{g}}(\mathbf{S}')X(\mathbf{S}'|\mathbf{S})\frac{1}{u_{\mathrm{g}}(\mathbf{S})}. \tag{40}$$

Then, the analog of Eq. (39) is

$$X_{\alpha\beta}^{(p',p)} = \lim_{L\to\infty} \frac{\sum_{t=1}^{L} \hat{u}_\alpha(\mathbf{S}_{t+p'+p})\hat{W}_t(p'+p,p+1)x(\mathbf{S}_{t+p+1}|\mathbf{S}_{t+p})\hat{W}_t(p,0)\hat{u}_\beta(\mathbf{S}_t)}{\sum_{t=1}^{L} \hat{u}_\alpha(\mathbf{S}_{t+p'+p})\hat{W}_t(p'+p,0)\hat{u}_\beta(\mathbf{S}_t)}, \tag{41}$$

where the x matrix elements are defined by

$$x(\mathbf{S}'|\mathbf{S}) = \frac{X(\mathbf{S}'|\mathbf{S})}{P(\mathbf{S}'|\mathbf{S})} = \frac{\hat{X}(\mathbf{S}'|\mathbf{S})}{\hat{P}(\mathbf{S}'|\mathbf{S})}. \tag{42}$$

Clearly, the preceding definition of $x(\mathbf{S}'|\mathbf{S})$ fails when $\hat{P}(\mathbf{S}'|\mathbf{S})$ vanishes but $\hat{X}(\mathbf{S}'|\mathbf{S})$ does not. Generically this happens, but a more complicated scheme can be employed in which one introduces *side-walks.* This is done by interrupting the continuing stochastic process at time $t+p$ by introducing a finite series of auxiliary states $\mathbf{S}'_{t+p+1}, \ldots, \mathbf{S}'_{t+p'+p}$. The latter are generated by a separate stochastic process so that in equilibrium, the sequence of subsequent states $\mathbf{S}_t, \mathbf{S}_{t+1}, \ldots, \mathbf{S}_{t+p}, \mathbf{S}'_{t+p+1}, \ldots, \mathbf{S}'_{t+p'+p}$ occurs with probability

$$\begin{aligned}
&\mathrm{Prob}[(\mathbf{S}_t, \mathbf{S}_{t+1}, \ldots, \mathbf{S}_{t+p}, \mathbf{S}'_{t+p+1}, \ldots, \mathbf{S}'_{t+p'+p})] \propto \\
&[\prod_{i=p+1}^{p'+p-1} \hat{P}(\mathbf{S}'_{t+i+1}|\mathbf{S}'_{t+i})]\hat{P}_X(\mathbf{S}'_{t+p+1}|\mathbf{S}_{t+p}) \times \\
&[\prod_{i=0}^{p-1} \hat{P}(\mathbf{S}_{t+i+1}|\mathbf{S}_{t+i})]\, u_{\mathrm{g}}(\mathbf{S}_t)^2
\end{aligned} \tag{43}$$

where $\hat{P}_X$ is a Markov matrix chosen to replace $\hat{P}$ in Eq. (42) so as to yield finite weights x. In this scheme, one generates a continuing thread identical to the usual Monte Carlo process in which each state $\mathbf{S}_t$ is sampled from the stationary state of $\hat{P}$, at least if one ignores the initial equilibration. Each state $\mathbf{S}_t$ of this backbone forms the beginning of a side walk, the first step of which is sampled from $\hat{P}_X$, while $\hat{P}$ again generates subsequent side steps. Clearly, with respect to the side walk, the first step disrupts the stationary state, so that the p' states $\mathbf{S}'_{t'}$, which form the side walk, do not sample the stationary state of the original stochastic process generated by $\hat{P}$, unless $\hat{P}_X$ coincidentally has the same stationary state as $\hat{P}$. Consequently, side walks have to be terminated, while the backbone is continued indefinitely.

A problem with the matrix elements with which we dealt up to now is that in the limit p' or $p \to \infty$ all of them reduce to matrix elements involving the dominant eigenstate, although symmetries might be used to yield other eigenstates besides the absolute dominant one. However, if symmetries fail, one has to employ the equivalent of an orthogonalization scheme, such as discussed in the next section, or one is forced to resort to evolution operators that contain, in exact or in approximate form, the corresponding projections. Examples of this are matrix elements computed in the context of the fixed-node approximation[13]. Within the framework of this approximation, one considers quantities of the form

$$X_{\alpha\beta}^{(p',p)} = \frac{\langle u_\alpha | G_1^{p'} X G_2^{p} | u_\beta \rangle}{\sqrt{\langle u_\alpha | G_1^{2p'} | u_\alpha \rangle \langle u_\beta | G_2^{2p} | u_\beta \rangle}}, \tag{44}$$

where the G_i are evolution operators combined with appropriate projectors, which in the fixed-node approximation are defined by the nodes of the states $u_\alpha(\mathbf{S})$ and $u_\beta(\mathbf{S})$. We shall describe how the preceding expression, Eq. (44), can be evaluated, but rather than writing out all the expressions explicitly, we present just the essence of the Monte Carlo method.

To deal with these expressions, one generates a backbone time series of states sampled from any distribution, say $u_{\mathrm{g}}(\mathbf{S})^2$, that has considerable overlap with the states $|u_\alpha(\mathbf{S})|$ and $|u_\beta(\mathbf{S})|$. Let us distinguish those backbone states by a superscript 0. Consider any such state $S^{(0)}$ at some given time. It forms the starting point of two side walks. We denote the states of these side walks by $\mathbf{S}_{t_i}^{(i)}$, where $i = 1, 2$ identifies the side walk and t_i labels the side steps. The side walks are generated from factorizations of the usual form, defined in Eq. (31), say $\hat{G}_i = \hat{g}_i \hat{P}_i$. A segment of the walk, *viz.*

$$\mathcal{S} = [\mathbf{S}^{(0)}, (\mathbf{S}_1^{(1)}, \mathbf{S}_2^{(1)}, \ldots), (\mathbf{S}_1^{(2)}, \mathbf{S}_2^{(2)}, \ldots)] \tag{45}$$

occurs with probability

$$\mathrm{Prob}(\mathcal{S}) = u_{\mathrm{g}}(\mathbf{S}^{(0)})^2\, \hat{P}_1(\mathbf{S}^{(0)}|\mathbf{S}_1^{(1)}) \ldots \hat{P}_2(\mathbf{S}^{(0)}|\mathbf{S}_1^{(2)}) \ldots . \tag{46}$$

We leave it to the reader to show that this expression for the probability suffices to compute by means of Monte Carlo all expressions appearing in numerator and denominator of Eq. (44), in the case that X is diagonal, and to generate the appropriate generalizations to other cases.

In the expressions derived above, the power method projections precipitate products of reweighting factors $\hat{g}$, and, as the projection times p and p' increase, the variance of the Monte Carlo estimators grows at least exponentially in the square root of the projection time. Clearly, the presence of the fluctuating weights $\hat{g}$ is due to the fact that the evolution operator $\hat{G}$ is not Markovian in the sense that it fails to conserve probability. The importance sampling transformation Eq. (30) was introduced to mitigate this problem. In Section 5, an algorithm involving branching walks will be introduced, which is a different strategy devised to deal with this problem. In diffusion and transfer matrix Monte Carlo, both strategies, importance sampling and branching, are usually employed simultaneously.

4. How to Avoid Reweighting

Before discussing the branching algorithms designed to deal more efficiently with the reweighting factors appearing in the expressions discussed above, we briefly mention an alternative that has surfaced occasionally without being studied extensively, to our knowledge. The idea will be illustrated in the case of the computation of the matrix element $X_{\alpha\beta}^{(0,p)}$, and we take Eq. (33) as our starting point. In statistical mechanical language, we introduce a reduced Hamiltonian

$$\mathcal{H} = -\ln u_{\mathrm{g}}(\mathbf{S}_p) - \sum_{i=0}^{p-1} \ln G(\mathbf{S}_{i+1}|\mathbf{S}_i) - \ln u_{\mathrm{g}}(\mathbf{S}_0), \tag{47}$$

and the corresponding Boltzmann distribution $\exp -\mathcal{H}(\mathbf{S}_p, \ldots, \mathbf{S}_0)$. One can now use the standard Metropolis algorithm to sample this distribution for this system consisting of $p+1$ layers bounded by the layers 0 and p. For the evaluation of Eq. (33) by Monte Carlo, this expression then straightforwardly becomes a ratio of correlation functions involving quantities defined at the boundaries. To see this, note that the denominator of Eq. (33) is the partition function

$$Z = \sum_{\mathbf{S}_p,\ldots,\mathbf{S}_0} e^{-\mathcal{H}(\mathbf{S}_p,\ldots,\mathbf{S}_0)}. \tag{48}$$

Note that in general, boundary terms involving some appropriately defined u_{g} should be introduced to ensure the non-negativity of the distribution. For the simultaneous computation of matrix elements for several values of the indices α and β, a guiding function u_{g} should be chosen that has considerable overlap with the corresponding $|u_\alpha|$ and $|u_\beta|$.

The Metropolis algorithm can of course be used to sample any probability distribution, and the introduction of the previous Hamiltonian illustrates just one particular point of view. If one applies the preceding idea to the case of the imaginary-time quantum mechanical evolution operator, one obtains the standard path-integral Monte Carlo method, in which case the layers are usually called time slices. The Hamiltonian in Eq. (47) corresponds to a system with boundaries; the corresponding version of path-integral Monte Carlo is associated with matrix elements, while the more common version with closed world line paths corresponds to the quantum statistical trace.

Clearly, this method has the advantage of suppressing the fluctuating weights in estimators. However, the disadvantage is that sampling the full, layered system yields a longer correlation time than sampling the single-layer distribution u_{g}^2. This is a consequence of the fact that the microscopic degrees of freedom are more strongly correlated in a layered system than in a single layer. Our limited experience suggests that for small systems reweighting is more efficient, whereas the path-integral style Metropolis approach tends to become more efficient as the system grows in size.[14] See Baroni and Moroni's lectures for a less abstract presentation of these ideas in the context of diffusion Monte Carlo with importance sampling. As they explain in detail, the latter suggests a proposal matrix for the Metropolis algorithm which proposes a new surface layer for one end of the stack of layers, while the old layers are pushed into the bulk and the dangling surface layer at the other side of the stack is removed.

5. Branching Monte Carlo

In Section 3 we discussed a method to compute Monte Carlo averages by exploiting the power method to reduce the spectral weight of undesirable, sub-dominant eigenstates. We saw that this leads to products of fluctuating weights associated with subsequent configurations sampled by a Monte Carlo time series. To suppress completely the systematic errors due to finite projection times, *i.e.,* the variational bias, one has to take averages of infinite products of weights. This limit would produce an "exact" method with infinite variance, which of course is of no practical use.

We have also discussed how optimized trial states can be used to reduce the variance of this method. The variance reduction may come about in two ways. Firstly, by starting with optimized trial states of higher quality, the variational bias, caused by the admixture of undesirable states, is smaller to begin with, so that fewer power method projections are required. In practical terms, this leads to a reduction of the number of factors in the fluctuating products. Secondly, a good estimate of the dominant eigenstate can be used to reduce the amount by which the evolution operator, divided by an appropriate constant, violates conservation of probability. This reduces the variance of the individual fluctuating weight factors in the factorization given in Eq. (31). All these considerations also apply to the branching Monte Carlo algorithm discussed in this section, which can be modified accordingly and in complete analogy with our previous discussion.

Before discussing the details of the branching algorithm, we mention that the algorithm presented here[9] contains the mathematical essence of both the diffusion and transfer matrix Monte Carlo algorithms. A related algorithm, *viz.*, Green function Monte Carlo, adds yet another level of complexity due to the fact that the evolution operator is known only as an infinite series. This series is stochastically summed at each step of the power method iterations. In practice this implies that even the time step becomes stochastic and intermediate Monte Carlo configurations are generated that do not contribute to expectation values. Neither Green function

Monte Carlo, nor its generalization designed to compute quantities at non-zero temperature[15], will be discussed in these lectures and we refer the interested reader to the literature for further details.[16, 17, 18]

Let us consider in detail the mechanism that produces large variance to understand what branching accomplishes if one has to compute products of many (ideally infinitely many) fluctuating weights. The time average over these products will typically be dominated by only very few large terms; the small terms are equally expensive to compute, but play no significant role in the average. This problem can be solved by performing many simultaneous Monte Carlo walks. One evolves a collection of walkers from one generation to the next, and the key idea is to eliminate as the process progresses the light-weight walkers that produce relatively small contributions to the time average. To keep the number of walkers reasonably constant, heavy-weight walkers are duplicated and the clones are subsequently evolved quasi-independently.

An algorithm designed according to this concept does not explicitly cut off the products over weights and therefore seems to correspond to infinite projection time. It would therefore seem that the time average over a stationary branching process corresponds to an average over the exact dominant eigenstate of the Monte Carlo evolution operator, but, as we shall see, this is rigorously the case only in the limit of an infinite number of walkers[19, 20]. For any finite number of walkers, the stationary distribution has a bias, the so-called *population control bias,* which is proportional to the inverse of the population size.[21] If the fluctuations in the weights are small and correlations, as discussed later, decay rapidly, this bias tends to be small. In many applications this appears to be the case and the corresponding bias is statistically insignificant. However, if these methods are applied to statistical mechanical systems at the critical point, significant bias can be present. We shall discuss a simple method of negligible computational cost to detect this bias and correct for it under all but the worst circumstances.

To discuss the branching Monte Carlo version of the power method, we continue to use the notation introduced above, and again consider the evolution operator $G(\mathbf{S}'|\mathbf{S})$. As above, the states $\mathbf{S}$ and $\mathbf{S}'$ will be treated here as discrete, but in practice the distinction between continuous and discrete states is a minor technicality, and generalization to the continuous case follows immediately by replacing sums by integrals and by replacing Kronecker δ's by Dirac δ functions.

To implement the power method iterations in Eq. (19) by a branching Monte Carlo process, $|u^{(t)}\rangle$ is represented by a collection of N_t walkers, where a walker by definition is a state-weight pair $(\mathbf{S}_i, w_i)$, $i = 1, \ldots, N_t$. As usual, the state variable $\mathbf{S}_i$ represents a possible configuration of the system evolving according to G, and w_i represents the statistical weight of walker i. These weights appear in averages and the efficiency of the branching Monte Carlo algorithm is realized by maintaining the weights in some range $\frac{1}{2} \approx w_{\rm l} < w_i < w_{\rm u} \approx 2$, where $w_{\rm l}$ and $w_{\rm u}$ are bounds introduced so as to keep all weights w_i of the same order of magnitude.

The first idea is to interpret a collection of walkers that make up generation t as a representation of the (sparse) vector $|\underline{u}^{(t)}\rangle$ with components

$$\langle \mathbf{S}|\underline{u}^{(t)}\rangle \equiv \underline{u}^{(t)}(\mathbf{S}) = \sum_{i=1}^{N_t} w_i \, \delta_{\mathbf{S}\mathbf{S}_i}, \tag{49}$$

where δ is the usual Kronecker δ-function. The under-bar is used to indicate that the $\underline{u}^{(t)}(\mathbf{S})$ represent a stochastic vector $|\underline{u}^{(t)}\rangle$. Of course, the same is true formally for the single thread Monte Carlo, but the new feature is that one can think of the collection of walkers as a reasonably accurate representation of the stationary state at *each* time step.

The second idea is to define a stochastic process in which the walkers evolve with transition probabilities such that the expectation value of $c_{t+1}|\underline{u}^{(t+1)}\rangle$, as represented by the walkers of generation $t+1$, equals $G|\underline{u}^{(t)}\rangle$ for any given collection of walkers representing $|\underline{u}^{(t)}\rangle$. It is tempting to conclude that, owing to this construction, the basic recursion relation of the power method, Eq. (19), is satisfied in an average sense, but this conclusion is not quite correct. The reason is that in practice, the constants c_t are defined on the fly. Consequently, c_{t+1} and $|\underline{u}^{(t+1)}\rangle$ are correlated random variables and there is no guarantee that the stationary state expectation value of $|\underline{u}^{(t)}\rangle$ is *exactly* an eigenstate of G, except in the limit of non-fluctuating normalization constants c_t, which, as we shall see, is tantamount to having an infinite number of walkers. More explicitly, the problem is that if one takes the time average of Eq. (19), and if the fluctuations of the c_{t+1} are correlated with $|\underline{u}^{(t)}\rangle$ or$|\underline{u}^{(t+1)}\rangle$, one does not produce the same state on the left- and right-hand sides of the time-averaged equation, and therefore the time-averaged state need not satisfy the eigenvalue equation. The resulting bias has been discussed by various authors.[11, 20, 21]

One way to define a stochastic process is to rewrite the power method iteration, Eq. (19), as

$$u^{(t+1)}(\mathbf{S}') = \frac{1}{c_{t+1}} \sum_{\mathbf{S}} P(\mathbf{S}'|\mathbf{S}) \, g(\mathbf{S}) u^{(t)}(\mathbf{S}), \tag{50}$$

where

$$g(\mathbf{S}) = \sum_{\mathbf{S}'} G(\mathbf{S}'|\mathbf{S}) \text{ and } P(\mathbf{S}'|\mathbf{S}) = G(\mathbf{S}'|\mathbf{S})/g(\mathbf{S}). \tag{51}$$

This is in fact how transfer matrix Monte Carlo is defined. Referring the reader back to the discussion of Eq. (27), we note that in diffusion Monte Carlo the weight g is defined so that it is not just a function of the initial state $\mathbf{S}$, but also depends on the final $\mathbf{S}'$. The algorithm given below can trivially be generalized to accommodate this by making the substitution $g(\mathbf{S}) \to g(\mathbf{S}'|\mathbf{S})$.

Equation (50) describes a process represented by a Monte Carlo run which, after a few initial equilibration steps, consists of a time series of M_0 updates of all walkers at times labeled by $t = \ldots, 0, 1, \ldots, M_0$. The update at time t consists of two steps designed to perform stochastically the matrix multiplications in Eq. (50). Following Nightingale and Blöte, [20] the process is defined by the following steps:

Let us consider one of these updates, say the one that transform the generation of walkers at time t into the generation at time $t+1$. We denote variables pertaining to times t and $t+1$ respectively by unprimed and primed symbols.

1. Update the old walker $(\mathbf{S}_i, w_i)$ to yield a temporary walker $(\mathbf{S}'_i, w'_i)$ according to the transition probability $P(\mathbf{S}'_i|\mathbf{S}_i)$, where $w'_i = g(\mathbf{S}_i)w_i/c'$, for $i = 1, \ldots, N_t$. Step two, defined below, can change the number of walkers, but to maintain their number close to a target number, say N_0, choose $c' = \hat{\lambda}_0(N_t/N_0)^{1/s}$, where $\hat{\lambda}_0$ is a running estimate of the eigenvalue λ_0 to be calculated, where $s \geq 1$ [see discussion after Eq. (53)].
2. From the temporary walkers construct the new generation of walkers as follows
   (a) Split each walker $(\mathbf{S}', w')$ for which $w' > b_{\rm u}$ into two identical walkers $(\mathbf{S}', \frac{1}{2}w')$.
   (b) Join pairs $(\mathbf{S}'_i, w'_i)$ and $(\mathbf{S}'_j, w'_j)$ with $w'_i < b_{\rm l}$ and $w'_j < b_{\rm l}$ to produce a single walker $(\mathbf{S}'_k, w'_i + w'_j)$, where $\mathbf{S}'_k = \mathbf{S}'_i$ or $\mathbf{S}'_k = S'_j$ with relative probabilities w'_i and w'_j.
   (c) Walkers for which $b_{\rm l} < w'_i < b_{\rm u}$, or left single in step 2b, become members of the new generation of walkers.

Note that, if the weights $g(\mathbf{S})$ fluctuate on average more than by a factor of two, it may be necessary to implement split-and-join operations consisting of more than two walkers at a time.

It may help to explicate the adverse effect of wildly fluctuating weight on the efficiency of the algorithm. Under those circumstances, some walkers will have multiple descendents, whereas others will have none. This leads to an inefficient algorithm, since at any given time there will be several walkers that are either identical or closely related. This will produce strongly correlated contributions to the time averages. In its final analysis, this is the same old problem that we encountered in a single-thread algorithm, where averages would be dominated by few terms with relatively large, explicitly given statistical weights. Branching mitigates this problem since walkers descendanting from a given walker ultimately decorrelate, but, as discussed in Section 3, the best cure is importance sampling.

The algorithm described above was constructed so that for any given realization of $|\underline{u}^{(t)}\rangle$, in accordance with Eq. (19) the expectation value of $c_{t+1}|\underline{u}^{(t+1)}\rangle$, satisfies

$$\mathrm{E}\left[c_{t+1}|\underline{u}^{(t+1)}\rangle\right] = G\,|\underline{u}^{(t)}\rangle, \tag{52}$$

where $\mathrm{E}(\cdot)$ denotes the conditional average over the transitions defined by the preceding stochastic process. More generally, by p-fold iteration one finds [22]

$$\mathrm{E}\left[\left(\prod_{b=1}^{p} c_{t+b}\right)|\underline{u}^{(t+p)}\rangle\right] = G^p\,|\underline{u}^{(t)}\rangle. \tag{53}$$

The stationary state average of $|u^{(t)}\rangle$ is close to the dominant eigenstate of G, but, as mentioned above, it has a systematic bias when the number N_t of walkers is finite. If, as is the case in some applications, this bias exceeds the statistical errors, one can again rely on the power method by means of Eq. (53) to reduce this bias by increasing the number of projections p. If that is done, one is back to the old problem of having to average products of fluctuating weights, and, as usual, the variance of the corresponding estimators increases as their bias decreases. Fortunately, in practice the population control bias of the stationary state is quite small, if at all detectable, but even in those cases, expectation values involving several values of p should be computed to verify the absence of population control bias. The reader is referred to Refs.[11, 21, 22, 23, 24] for a more detailed discussion of this problem. Suffice it to mention here, first of all, that s, as defined in the first step of the branching algorithm given above is the expected number of time steps it takes to restore the number of walkers to its target value N_0, and secondly, that strong population control ($s = 1$) tends to introduce a stronger bias than weaker control ($s > 1$).

With Eq. (53) one constructs an estimator [22] of the dominant eigenstate $|u^{(\infty)}\rangle$ of the evolution operator G:

$$|\check{u}^{(p)}\rangle = \frac{1}{M_0}\sum_{t=1}^{M_0}\left(\prod_{b=0}^{p-1} c_{t-b}\right)|\underline{u}^{(t)}\rangle. \tag{54}$$

For $p = 0$, in which case the product over b reduces to unity, this yields the stationary state of the branching Monte Carlo, which frequently is treated as the dominant eigenstate of G.

Clearly, this branching Monte Carlo algorithm can be used to compute the right-projected mixed estimators that were denoted by $X_{\rm TT}^{(0,\infty)}$ in Section 3. For example, suppose that $|u_{\rm T}\rangle$ is an approximation to the dominant eigenstate and that X is an arbitrary operator, as before. The mixed expectation value of X can be approximated as

$$\frac{\langle u_{\rm T}|X|u^{(\infty)}\rangle}{\langle u_{\rm T}|u^{(\infty)}\rangle} \approx \frac{\langle u_{\rm T}|X|\check{u}^{(p)}\rangle}{\langle u_{\rm T}|\check{u}^{(p)}\rangle}. \tag{55}$$

Averages for several values of p can be computed simultaneously and virtually for the price of one. We explicitly mention the important special case obtained by choosing for the general operator X the evolution operator G itself and by setting $\langle \mathbf{S}|u_{\rm T}\rangle \equiv 1$, which corresponds to complete ignorance of the dominant state. This yields the following estimator for the dominant eigenvalue λ_0 of G:

$$\lambda_0 \approx \frac{\sum_{t=1}^{M_0} \left(\prod_{b=0}^{p} c_{t-b}\right) \mathcal{W}^{(t)}}{\sum_{t=1}^{M_0} \left(\prod_{b=0}^{p-1} c_{t-b}\right) \mathcal{W}^{(t-1)}}, \tag{56}$$

where

$$\mathcal{W}^{(t)} = \sum_{i=1}^{N_t} w_i^{(t)}. \tag{57}$$

In diffusion Monte Carlo, this estimator can be used to construct the *growth estimate* of the ground state energy. That is, since in that special case $G \approx \exp(-\tau\mathcal{H})$, eigenvalues of the evolution operator and the Hamiltonian are related by

$$E_0 = -\frac{1}{\tau} \ln \lambda_0. \tag{58}$$

As mentioned several times before, it is important in practical applications to combine the raw branching algorithm defined above with importance sampling. Mathematically, this works in precisely the same way as in Section 3: one reformulates the same algorithm in terms of the similarity transform $\hat{G}$ with $u_{\rm g} = u_{\rm T}$ chosen to be an accurate, approximate dominant eigenstate [see Eq. (30)]. In the single-thread algorithm, the result is that the fluctuations of the weights g and their products $\mathcal{W}$ are reduced. In the context of the branching algorithm, this yields reduced fluctuations in the weight of walkers individually and in the size of the walker population. One result is that the population control bias is reduced. If we ignore this bias, a more fundamental difference is that the steady state of the branching algorithm is modified. That is, in the raw algorithm the walkers sample the dominant eigenstate of G, *i.e.,* $\psi_0(\mathbf{S})$, but, if the trial state $|u_{\rm T}\rangle$ is used for importance sampling, the distribution is $u_T(\mathbf{S})\psi_0(\mathbf{S})$, which, of course, is simply the dominant eigenstate of $\hat{G}$.

So far, we have only discussed how mixed expectation values can be computed with the branching Monte Carlo algorithm, but this yields the desired result only if one deals with operators that commute with the evolution operator G. This algorithm can, however, also be used to perform power method projections to the left. In fact, most of the concepts discussed in Sections 3.2.1, 3.2.2, and 3.2.3 can be implemented straightforwardly. To illustrate this point, we shall show how one can compute the left- and right-projected expectation value of a diagonal operator X. Since the branching algorithm is designed to explicitly perform the multiplication by G including all weights, all that is required is the following generalization[25], called *forward* or *future walking.*

Rather than defining a walker to be the pair formed by a state and a weight, for forward walking we define the walker to be of the form $[\mathbf{S}, w, X(\mathbf{S}_{-1}), ..., X(\mathbf{S}_{-p'})]$, where $\mathbf{S}_{-1}, \mathbf{S}_{-2}, \ldots$ are previous states of the walker. In other words, each walker is equipped with a finite stack of depth p' of previous values of the diagonal operator X. In going from one generation of walkers to the next, the state and weight of a walker are updated just as before to $\mathbf{S}'$ and w'. The only new feature is that the value $X(\mathbf{S})$ is pushed onto the stack:

$$[\mathbf{S}, w, X(\mathbf{S}_{-1}), ..., X(\mathbf{S}_{-p'})] \to [\mathbf{S}', w', X(\mathbf{S}), X(\mathbf{S}_{-1}), ..., X(\mathbf{S}_{-p'+1})]. \quad (59)$$

In this way, the p' times left-projected expectation value of X is obtained simply by replacing $X(\mathbf{S})$ by $X(\mathbf{S}_{-p'})$. Note that one saves the history of X rather than the history of configurations only for the purpose of computational efficiency. Copying a walker of this sort, as required in step 2a of the algorithm, can become time consuming if X is a many-component object. In such cases, it may be advantageous to replace the history of X by a history of pointers to the original copy of X.

6. Excited States: a case study

6.1. SUBSPACE ITERATION

Given a set of basis states, excited or, more generally, sub-dominant states can be computed variationally by solving a linear variational problem, and the Metropolis method can be used to evaluate the required matrix elements. A variation of the power method, described above, can then be used to remove the variational bias systematically. [12, 26, 27] As was mentioned before, the price to be paid for reduction of the variational bias is increased statistical noise, a problem which appears in this context with a vengeance. Again, the way to mitigate this problem is the use of optimized trial vectors.

The variational problem to be solved for the computation of excited states is the following one. Given n basis functions $|u_i\rangle$, find the $n \times n$ matrix of coefficients $d_i^{(j)}$ such that

$$|\tilde{\psi}_j\rangle = \sum_{i=1}^{n} d_i^{(j)} |u_i\rangle \tag{60}$$

are the "best," variational approximations for the n lowest eigenstates $|\psi_i\rangle$ of some Hamiltonian $\mathcal{H}$. In this problem we shall, at least initially, use the language of quantum mechanical systems, where one has to distinguish the Hamiltonian from the evolution operator $\exp(-\tau\mathcal{H})$. In the statistical mechanical applications, one encounters only the equivalent of the latter. In the expressions to be derived below, the substitution $\mathcal{H}G^p \to G^{p+1}$ will produce the expressions required for the statistical mechanical applications, at least if we assume that the non-symmetric matrices that appear in that context have been symmetrized.

One seeks the "best" solution to the linear variational problem in Eq. (60) in the sense that for all i the Rayleigh quotient $\langle\tilde{\psi}_i|\mathcal{H}|\tilde{\psi}_i\rangle/\langle\tilde{\psi}_i|\tilde{\psi}_i\rangle$ is stationary with respect to variation of the coefficients of the matrix d. The solution is that the matrix of coefficients d has to satisfy the following generalized eigenvalue equation

$$\sum_{i=1}^{n} H_{ki}\, d_i^{(j)} = \tilde{E}_j \sum_{i=1}^{n} N_{ki}\, d_i^{(j)}, \tag{61}$$

where

$$H_{ki} = \langle u_k|\mathcal{H}|u_i\rangle, \tag{62}$$

and

$$N_{ki} = \langle u_k|u_i\rangle. \tag{63}$$

Before discussing Monte Carlo issues, we note a number of important properties of this scheme. Firstly, the basis states $|u_i\rangle$ in general are not orthonormal and this is reflected by the fact that the matrix N in general is not a unit matrix and not diagonal; its elements have to be computed. Secondly, it is clear that any nonsingular linear combination of the basis vectors will produce precisely the same results, obtained from the correspondingly transformed version of Eq. (61). The final comment is that the variational eigenvalues bound the exact eigenvalues from above, *i.e.*, $\tilde{E}_i \geq E_i$, where we assume $E_1 \leq E_2 \leq \ldots$. One recovers exact eigenvalues E_i and the corresponding eigenstates, if the $|u_i\rangle$ span the same space as the exact eigenstates, or in other words have no admixtures of other states.

The required matrix elements can be computed using the variational Monte Carlo method discussed in the previous sections. The power method can subsequently be used to reduce the variational bias. Formally, one simply defines new basis states

$$|u_i^{(p)}\rangle = G^p\,|u_i\rangle \tag{64}$$

and substitutes these new basis states for the original ones. The corresponding matrices

$$H_{ki}^{(p)} = \langle u_k^{(p)}|\mathcal{H}|u_i^{(p)}\rangle \tag{65}$$

and

$$N_{ki}^{(p)} = \langle u_k^{(p)}|u_i{}^{(p)}\rangle \tag{66}$$

can again be computed by applying the methods discussed above for the computation of general matrix elements. Monte Carlo yields these matrix elements up to an overall normalization constant, which is irrelevant as can be seen from Eq. (61).

As an explicit example illustrating the nature of the Monte Carlo time-averages that one has to evaluate in this approach, we write down the expression for $N_{ij}^{(p)}$ as used for the computation of eigenvalues of the Markov matrix relevant to the problem of critical slowing down, discussed in detail in the next section. One estimates this matrix as

$$N_{ij}^{(p)} \propto \sum_t \frac{u_i(\mathbf{S}_t)}{\psi_{\rm B}(\mathbf{S}_t)}\frac{u_j(\mathbf{S}_{t+p})}{\psi_{\rm B}(\mathbf{S}_{t+p})}, \tag{67}$$

where the $\mathbf{S}_t$ are configurations forming a time series that is designed to sample the distribution of a system in thermodynamic equilibrium, *i.e.,* the Boltzmann distribution $\psi_{\rm B}^2$. It turns out that in this particular case, this distribution, the dominant eigenstate, has sufficient overlap with the magnitude of the sub-dominant states that it is not necessary to introduce a separate guiding function to allow the simultaneous computation of all matrix elements $N_{ij}^{(p)}$.

The expression given in Eq. (67) yields the $u/\psi_{\rm B}$-auto-correlation function at lag p. The expression for $H_{ij}^{(p)}$ is similar, and represents a cross-correlation function involving the configurational eigenvalues of the Markov matrix in the various basis states. Compared to the expressions derived in Section 3, Eq. (67) takes a particularly simple form in which products of fluctuating weights are absent, because one is dealing with a probability conserving evolution operator from the outset in this particular problem.

6.2. UNIVERSAL DYNAMIC AMPLITUDE RATIOS

Before continuing our general discussion, we temporarily change topics to introduce stochastic dynamics of critical systems. What make such systems interesting, is that one can distinguish universality classes in which behavior does not depend on many of the microscopic details. For static critical phenomena, it is known that universality classes can be identified by dimensionality, symmetry of the order parameter, and the range of the interactions. For dynamical phenomena, there are additional features such as whether or not the dynamics is local or subject to conservation laws.

On approach of a critical point, the correlation length ξ diverges. The dynamical exponent z governs the corresponding divergence of the correlation time τ by means of the relation $\tau \propto \xi^z$. Since the critical exponent z is one of the universal quantities, it has been used to identify universality classes. Unfortunately, z does not vary by much from one universality class to another, and this poses a serious computational problem in terms of the accuracy required to obtain significant differences. One of the outcomes of the work reviewed here is that there are other quantities within computational reach, namely universal amplitude ratios.[28] These ratios may serve as additional, and possibly more sensitive identifiers of universality classes. We shall consider various systems belonging to a single universality class, and we assume that the representatives of the class are parameterized by κ.

If a thermodynamic system is perturbed out of equilibrium, different thermodynamic quantities relax back at a different rates. More generally, there are infinitely many independent relaxation modes for a system in the thermodynamic limit. The Monte Carlo methods discussed in these lectures have been used to compute relaxation times of Ising models on square $L \times L$ lattices at the critical point.[28]

Let us denote by $\tau_{\kappa i}(L)$ the relaxation time of mode i of a system of linear dimension L. As indeed scaling theory suggests, it turns out that the relaxation time has the following factorization property

$$\tau_{\kappa i}(L) \approx m_\kappa A_i L^z, \tag{68}$$

where m_κ is a non-universal metric factor, while A_i is a universal amplitude which depends on the mode i, and z is the universal dynamical exponent introduced above.

Formulated as a computational problem, one has the following. Suppose $\mathbf{S} = (s_1, ..., s_{L^2})$, with $s_i = \pm 1$, is a spin configuration and $\rho_t(\mathbf{S})$ is the probability of finding $\mathbf{S}$ at time t. The probability distribution evolves in time according to

$$\rho_{t+1}(\mathbf{S}) = \sum_{\mathbf{S}'} P(\mathbf{S}|\mathbf{S}')\rho_t(\mathbf{S}'). \tag{69}$$

The detailed structure of the Markov matrix P is of no immediate importance for the current discussion. All that matters is that it satisfies detailed balance, has the Boltzmann distribution ψ_{B}^2 as its stationary state. Also, P

is a single-spin flip matrix, *i.e.* $P(\mathbf{S}|\mathbf{S}')$ vanishes if $\mathbf{S}$ and $\mathbf{S}'$ differ by more than a single spin. The desired relaxation time of mode i is given by

$$\tau_i(L) = -\frac{1}{L^2 \ln \lambda_i(L)}, \tag{70}$$

where λ_i is an eigenvalue of Markov matrix P.

6.3. NUMERICAL PROCEDURE

To verify Eq. (68), it is important to obtain estimates that are exact within the range of the estimated error. For this purpose we use a set of optimized variational basis functions, to which we subsequently apply the projection procedure described in Section 6.1 to remove the variational bias.

As mentioned, the Monte Carlo projection increases the statistical noise, and the solution to this problem is to improve the variational basis functions. We shall now discuss how this is done and we consider the problem using the language of the Schrödinger equation.

We first consider the ground state and review how one can optimize a many, say 50-100 parameter trial function $\psi_{\rm T}(R)$.[29] Recall that the *local energy* $\mathcal{E}(R)$ is defined by

$$\mathcal{H}\psi_{\rm T}(R) \equiv \mathcal{E}(R)\psi_{\rm T}(R). \tag{71}$$

The variance of the local energy is given by

$$\chi^2 = \langle(\mathcal{H} - \overline{\mathcal{E}})^2\rangle = \frac{\int |\psi_{\rm T}(R)|^2[\mathcal{E}(R) - \overline{\mathcal{E}}]^2\, dR}{\int |\psi_{\rm T}(R)|^2\, dR}. \tag{72}$$

A property that we shall exploit later is that $\chi^2 = 0$ for any eigenstate, not just the ground state.

The following sums up the Monte Carlo optimization procedure for a single state:

1. Sample $R_1, \ldots, R_s$ from ${\psi_{\rm T}}^2$ a typical sample size has $s \approx 3,000$.
2. Approximate the integrals in Eq. (72) by Monte Carlo sums.
3. Minimize χ^2 as follows, while keeping this sample *fixed*. For each member of the sample $R_1, \ldots, R_s$:
4. Compute $\psi_{\rm T}(R_1), \ldots, \psi_{\rm T}(R_s)$.
5. Compute $\mathcal{H}\psi_{\rm T}(R_1), \ldots, \mathcal{H}\psi_{\rm T}(R_s)$.
6. Find $\overline{\mathcal{E}}$ from least-squares fit of

$$\mathcal{H}\psi_{\rm T}(R_\sigma) = \overline{\mathcal{E}}\psi_{\rm T}(R_\sigma),\ \sigma = 1, \ldots, s. \tag{73}$$

7. Minimize the sum of squared residues of Eq. 73.†

†Once the parameters are changed from the values they had in step 1, one should use an appropriately weighted sum of squared residues. See Cyrus Umrigar's lectures.

This procedure can be generalized immediately to a set of basis functions, as required to implement Eq. (60). The only new ingredient is a guiding function ψ_{g}^2 that has sufficient overlap with all basis states used in the computation. A function that satisfies this requirement is:

$$\psi_{\mathrm{g}}(R)^2 = \sqrt{\sum_{i=1}^{n} \tilde{\psi}_i(R)^4}, \tag{74}$$

where the $\tilde{\psi}_i$ are the best current approximations to the *normalized* variational basis states. This yields the following algorithm to optimize basis states for n dominant eigenvalues:

1. Sample $R_1, \ldots, R_s$ from ψ_{g}^2.
2. Compute the arrays

$$\begin{pmatrix} u^{(1)}(R_1) \\ u^{(2)}(R_1) \\ \vdots \end{pmatrix}, \ldots, \begin{pmatrix} u^{(1)}(R_s) \\ u^{(2)}(R_s) \\ \vdots \end{pmatrix}. \tag{75}$$

3. Compute the arrays

$$\begin{pmatrix} \mathcal{H}u^{(1)}(R_1) \\ \mathcal{H}u^{(2)}(R_1) \\ \vdots \end{pmatrix}, \ldots, \begin{pmatrix} \mathcal{H}u^{(1)}(R_s) \\ \mathcal{H}u^{(2)}(R_s) \\ \vdots \end{pmatrix}. \tag{76}$$

4. Find the matrix elements $\overline{\mathcal{E}}_{ij}$ from the appropriately weighted least-squares fit to

$$\mathcal{H}u^{(i)}(R_\sigma) = \sum_{j=1}^{n} \overline{\mathcal{E}}_{ij} u^{(j)}(R_\sigma), \ \sigma = 1, \ldots, s. \tag{77}$$

5. Vary the parameters to optimize the fit, as explained below.

In case of a perfect fit, the eigenvalues of the truncated Hamiltonian matrix $\mathbf{E} = (\overline{\mathcal{E}}_{ij})_{i,j=1}^n$ are the required eigenvalues, but in real life one has to optimize the parameters of the basis functions, which can be done as follows:

1. Divide the sample in blocks and compute one Hamiltonian matrix $\mathbf{E}$ per block.
2. Minimize the variance of the $\mathbf{E}$-spectra over the blocks.

The variance vanishes if the basis functions $u^{(i)}$ are *linear combinations* of n eigenstates of $\mathcal{H}$. This gives rise to a computational problem, *viz.*, the variance is near-invariant under linear transformation of the $u^{(i)}$. This approximate "gauge invariance" gives rise to near-singular non-linear optimization problem. This can be avoided by simultaneously minimizing the variance of both the spectrum of the "local" Hamiltonian matrix $\mathbf{E}$ and the local energy $\mathcal{E}$ of the individual basis functions.

Finally, the variational bias of the eigenvalue estimates obtained with the optimized basis states is reduced by using Monte Carlo to make the substitution discussed previously

$$|u^{(i)}\rangle \to e^{-\mathcal{H}\tau}\,|u^{(i)}\rangle. \tag{78}$$

For this purpose, one has to use the short-time approximation of $\exp(-\mathcal{H}\tau)$, in quantum mechanical applications, as discussed in Lubos Mitas's lectures on diffusion Monte Carlo. To apply the preceding scheme to the problem of critical dynamics, all one has to do is to make use of the fact the analog of the quantum mechanical evolution is the symmetrized Markov $\hat{P}$ of stochastic dynamics, which is defined as

$$\hat{P}(\mathbf{S}|\mathbf{S}') = \frac{1}{\psi_{\mathrm{B}}(\mathbf{S})} P(\mathbf{S}|\mathbf{S}')\psi_{\mathrm{B}}(\mathbf{S}'), \tag{79}$$

in terms of which we have the correspondence

$$e^{-\mathcal{H}\tau} \to \hat{P}^t. \tag{80}$$

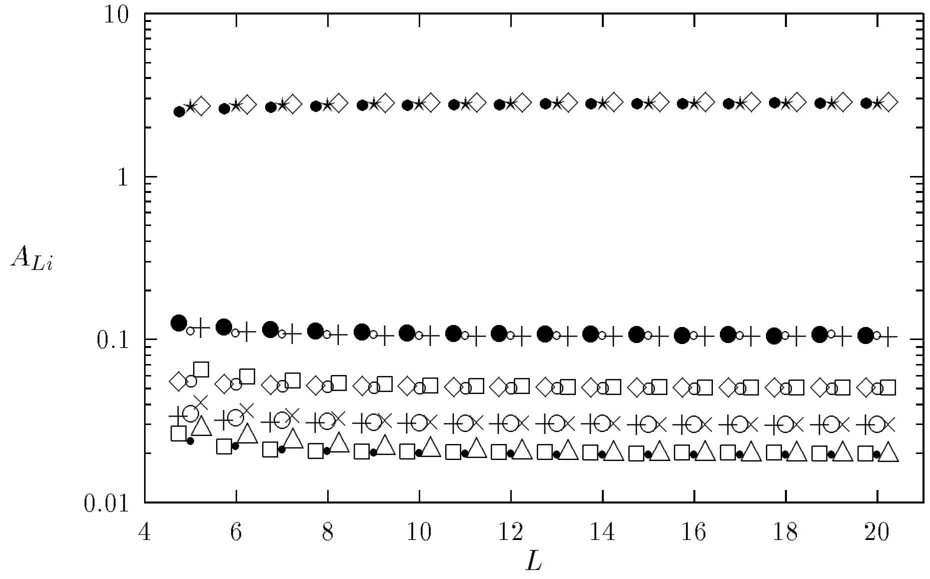
Next we briefly address the issue of the choice of trial functions for the eigenstates of symmetrized Markov matrix $\hat{P}$. We write

$$u(\mathbf{S}) = f(\mathbf{S}) \times \psi_{\mathrm{B}}(\mathbf{S}). \tag{81}$$

For the modes we considered, $f(\mathbf{S})$ was chosen to be a rotationally and translationally invariant polynomial in long-wavelength Fourier components of $\mathbf{S}$, the lowest-order one of which is simply the magnetization. Corresponding to the order parameter and energy-like modes, we considered polynomials either odd or even under the transformation $\mathbf{S} \to -\mathbf{S}$.

We briefly discuss some of the results that illustrate the validity of Eq. (68). Figure 2. shows plots of the effective amplitudes for the three dominant odd, and two dominant even modes of three different Ising models on $L \times L$ lattices. Of the three Ising models we studied, the first one, the NN model, had nearest-neighbor couplings only. The other two also had next-nearest-neighbor couplings. In one of them, the equivalent neighbor or EQN model, both couplings were of equal ferromagnetic strengths. In the third or NEQ model, the nearest-neighbor coupling was chosen ferromagnetic and of twice the magnitude of the antiferromagnetic next-nearest-neighbor coupling.

Figure 2. Universality of relaxation-time amplitudes, shown in a plot of the effective, size-dependent amplitudes A_{Li} on a logarithmic scale. To separate data points for the three models, the NEQ data were displaced to the left and the EQN data to the right. The data collapse predicted by Eq. (68) was produced by fitting the metric factors of the NN and NEQ models. Amplitudes of odd and even states alternate in magnitude.



To obtain estimates of the amplitudes of the relaxation modes, we fit the computed correlation times to expressions of the form

$$\tau_i(L) \approx L^z \sum_{k=0}^{n_c} \alpha_{ki} L^{-2k}. \tag{82}$$

In our computation of the non-universal metric factors, this quantity was set equal to unity by definition for the EQN model. Table 1 shows the metric factors computed for each mode separately as the ratio of the computed amplitudes. In agreement with the scaling prediction in Eq. (68), the computed metric factors depend only on the model but not on the mode.

Finally we mention that the spectral gaps of the Markov matrix vary over a considerable range

$$1 - \lambda_i(L) \approx L^{-(d+z)} \approx L^{-4.17}, \tag{83}$$

i.e. from approximately 3×10^{-3} for $L = 4$ to 3×10^{-6} for $L = 21$. For details of the numerical analysis based on Eq. (82) we refer the interested

TABLE 1. Non-universal metric factors m_κ, as defined in Eq. (68), computed for the NN and NEQ models. The modes indicated by o1, o2, and o3 are odd under spin inversion; the remaining two, e2 and e3, are even.

	NEQ	NN
o1	2.389(1)	1.5569 (5)
e2	2.394(2)	1.5569 (5)
o2	2.393(2)	1.5567 (6)
e3	2.391(2)	1.554 (2)
o3	2.385(4)	1.554 (2)

reader to Ref.[28]. Suffice it to mention that the value obtained for the universal dynamic critical exponent z featured in Eq. (68) is $z = 2.167 \pm 0.002$ which is indistinguishable from 13/6.

Acknowledgements

This work was supported by the (US) National Science Foundation through Grants DMR-9725080 and CHE-9625498.

References

1. M.H. Kalos and P.A. Whitlock, *Monte Carlo Methods,* Vol. 1, (Wiley, 1986).
2. P.J. Reynolds, R. N. Barnett, B. L. Hammond and W.A. Lester, Stat. Phys., **43**, 1017 (1986).
3. C.J. Umrigar, *Int. J. Quant. Chem. Symp.* **23**, 217 (1989).
4. R.H. Swendsen, Int. J. Mod. Phys. C, **7**, 281 (1996).
5. B. N. Tailor and Chris E. Kuyatt, *Guidelines for evaluating and expressing the uncertainty of NIST measurement result* (NIST Technical Note 1297, 1994 edition; http://physics.nist.gov/Pubs/guidelines).
6. H. Flyvbjeg and H.G. Petersen, *J. Chem. Phys.* **91**, 416 (1989).
7. W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling, *Numerical Recipes*, (Cambridge University Press, Cambrigde 1992), Section 14.3.
8. Statisticians call this the . See *e.g.* W.J. Conover, *Practical Nonparametric Statistics,* (John Wiley & Sons, 1980).
9. M. P. Nightingale and H.W.J. Blöte, *Phys. Rev. B* **54**, 1001 (1996).
10. M. P. Nightingale and H.W.J. Blöte, *Phys. Rev. Lett.* **76**, 4548 (1996).
11. J.H. Hetherington, *Phys. Rev. A* **30**, 2713 (1984).
12. D.M. Ceperley and B. Bernu, *J. Chem. Phys.* **89**, 6316 (1988).
13. R. N. Barnett, P. J. Reynolds and W. A. Lester, *J. Chem. Phys.* **96**, 2141 (1992).
14. M.P. Nightingale, Y. Ozeki, Y. Ye, unpublished.
15. P.A. Whitlock and M. H. Kalos, J. Comp. Phys. **30,** 361 (1979).
16. M. Kalos, D. Leveque and L. Verlet, *Phys. Rev.* **A9**, 2178 (1974).
17. D.M. Ceperley, M.H. Kalos, in *Monte Carlo Methods in Statistical Physics*, ed. by K. Binder, Topics Current Phys., Vol.7 (Springer, Berlin, Heidelberg 1979) Chap.4.

18. D. M. Ceperley, J. Comp. Phys., **51**, 404 (1983).
19. J.H. Hetherington, *Phys. Rev. A* **30**, 2713 (1984).
20. M.P. Nightingale and H.W.J. Blöte, *Phys. Rev. B* **33**, 659 (1986).
21. C.J. Umrigar, M.P. Nightingale, and K.J. Runge, J. Chem. Phys. **99**, 2865 (1993).
22. M.P. Nightingale and H.W.J. Blöte, *Phys. Rev. Lett.* **60**, 1662 (1988).
23. M.P. Nightingale, in *Finite-Size Scaling and Simulation of Statistical Mechanical Systems,* edited by Privman, (World Scientific, Singapore 1990), p.287-351.
24. K.J. Runge, *Phys. Rev. B* **45**, 12292 (1992).
25. M.H. Kalos, J. Comput. Phys. **1**, 257 (1966); the original idea of "forward walking" predates this paper [M.H. Kalos (private communication)]. For further references see Ref. 11 of Ref.[30].
26. B. Bernu, D.M. Ceperley, and W.A. Lester, Jr., *J. Chem. Phys.* **93**, 552 (1990).
27. W.R. Brown, W.A. Glauser, and W.A. Lester, Jr., *J. Chem. Phys.* **103**, 9721 (1995).
28. M. P. Nightingale and H.W.J. Blöte, *Phys. Rev. Lett.* **80**, 1007 (1998). Also see http://xxx.lanl.gov/abs/cond-mat/9708063
29. C.J. Umrigar, K.G. Wilson, and J.W. Wilkins, *Phys. Rev. Lett.* **60**, 1719 (1988); *Computer Simulation Studies in Condensed Matter Physics*, edited by D.P. Landau, K.K. Mon, and H.-B. Schüttler, Springer Proceedings in Physics Vol. 33 (Springer-Verlag, Berlin, 1988), p.185.
30. K.J. Runge, *Phys. Rev. B* **45**, 7229 (1992).