

RESERVE THIS SPACE

POPULATION CONTROL BIAS WITH APPLICATIONS TO PARALLEL DIFFUSION MONTE CARLO

JARON T. KROGEL^{*1,2}

DAVID M. CEPERLEY^{1,2}

¹Department of Physics, University of Illinois, Urbana-Champaign, Illinois

²NCSA, University of Illinois, Urbana-Champaign, Illinois

*jkrogel2@illinois.edu

The future of scientific computing will be driven by highly distributed parallel machines with millions of compute nodes. In order to take advantage of this already arriving wave of computing capability we must identify and remove the remaining barriers to parallel scaling in the Diffusion Monte Carlo algorithm. To address these scaling issues in a simple way, we propose that a time delay be introduced into the population control feedback. In order to assess this algorithm, we investigate the behavior of population fluctuations and the population control bias (which will emerge into greater relevance with larger physical systems and requirements of higher accuracy) in a model system for both the standard and time delayed DMC algorithms. We then condense our findings into a simple set of recommendations to improve the scaling of DMC while managing the population control bias.

RESERVE THIS SPACE

The Future of Parallel DMC

Modern parallel machines are composed of tens of thousands of SMP nodes which will rapidly increase into the millions in coming years. Keeping pace with such developments is crucial to the success of scientific computing. Monte Carlo methods are particularly well suited to make use of this ever increasing computing power since they contain low serial dependency. In Markov Chain Monte Carlo, a function is evaluated along a large chain of randomly generated particle configurations to obtain its mean. Conceptually, we think of the chain as a random walk, meaning that it is formed by a walker which steps from configuration to configuration. Typically, straightforward replication of Monte Carlo random walks allows near perfect scaling, however a central feature of any Monte Carlo algorithm will limit the practical efficiency gains realized when scaling to millions of processors: the equilibration time of the random walks.

The bottleneck due to equilibration time arises from the pursuit to simulate ever larger systems. Since the time to obtain new configurations is increased, the equilibration time consumes a larger fraction of the fixed serial run time. A direct way to minimize the ratio of serial equilibration time and total run time is to minimize the number of walkers per SMP node. For many Monte Carlo algorithms it is possible to achieve the exact minimum of just one walker per node; however this is not the case for the standard Diffusion Monte Carlo (DMC) method.

DMC is a Monte Carlo procedure to solve the many-body time independent Schrödinger equation (1, 2). It is currently the most widely used Quantum Monte Carlo algorithm for chemical and condensed matter systems. In the standard algorithm, the population of walkers fluctuates through a birth-death process known as branching and so load balancing challenges arise in the limit of few walkers per node. For example, a set of nodes each running with a single walker will quickly evolve into a situation where nodes contain zero, one, or two or more walkers each. In this circumstance, the overall efficiency can fall since all walkers must complete a step before the trial energy is collected and branching is performed. In typical implementations, current codes use on the approximately one hundred walkers per node to avoid load balancing inefficiency, however this leads to equilibration inefficiency as mentioned above. Though load balancing operations cannot perfectly redistribute the load for arbitrary population sizes, they will have to be performed frequently to prevent the situation from becoming even worse. Since most implementations of load balancing rely on operations within the Message Passing Interface (which impose synchronization at some level), parallel efficiency will further suffer with increasing machine size. At the same time, the branching process must be tightly regulated to prevent chronic shortages of walkers which could leave large numbers of nodes idle.

Demanding smaller fluctuations in the DMC walker population (greater stability) increases a systematic error within DMC as we detail below. However, with the advent of larger machines, a greater level of accuracy will inevitably be pursued, requiring parallel algorithms with greater robustness and less bias. Thus the population control bias, which has largely been neglected in the past, could become increasingly relevant to future applications.

In this work, we explore the relationship between the population control parameters and the stability and accuracy of the standard DMC method. Using a simple and efficient model problem, the simple harmonic oscillator, we obtain empirical formulas describing the behavior of population stability and the population control bias over a range of control parameters. Since we expect population dynamics (and hence stability and bias) to depend more strongly on the control method used than the particular details of the physical system, these results should apply broadly to current DMC practice.

Looking ahead to the rapidly approaching parallel efficiency difficulties outlined above, we move on to a possible solution to the scaling problem. We propose the introduction of a feedback delay in the population control mechanism which removes synchronization requirements from the algorithm and allows load balancing techniques that capitalize on keeping the time averaged load constant rather than the instantaneous load. These features ease the tradeoff between serial and parallel efficiency. As we must be cautious not to introduce unmanageable instability or bias into the simulation, we repeat the model analysis on cases with delayed feedback. These results are then consolidated into a brief set of practical recommendations to improve the parallel efficiency and reduce the population control bias in typical DMC simulations.

Other types of Quantum Monte Carlo, such as Reptation (3) and Path Integral Monte Carlo (4), do not suffer from population control bias. However, those methods have not been systematically applied to systems of many electrons, and in particular, there have been no studies of their relative efficiency, e.g. how their errors scale with computer time and number of electrons. In addition, these methods have other difficulties in scaling to very large number of processors. For these reasons, we limit the discussion to Diffusion Monte Carlo.

Review of the DMC Method

The Diffusion Monte Carlo algorithm (1, 2) can be viewed as a technique to iteratively refine the standard variational method (5). In the variational method, an *ansatz* for the many-body wavefunction (Ψ_T) is optimized to be as close as

possible to the true ground state, usually by minimizing a combination of its energy moments, $\mu_n = \langle \Psi_T | \hat{H}^n | \Psi_T \rangle$. The Hamiltonian operator, \hat{H} , is given by

$$\hat{H} = -\frac{\hbar^2}{2m} \nabla^2 + V \quad (1)$$

The variational energy obtained by integrating the local energy ($E_L = \Psi_T^{-1} \hat{H} \Psi_T$) over the trial probability density ($f_0 = |\Psi_T|^2$) is a strict upper bound on the ground state energy of the system.

$$E_V = \frac{\int dR f_0(R) E_L(R)}{\int dR f_0(R)} \geq E_0 \quad (2)$$

In DMC, the variational probability density is evolved forward in imaginary time through the iterative application of a short time Green's function.

$$f_{t+\tau}(R') = \int dR G(R', R; \tau) f_t(R) \quad (3)$$

The importance sampled (6,1,2,7) form of this propagator

$$G(R', R; \tau) = \psi_T(R') \langle R' | e^{-\tau(\hat{H} - E_T)} | R \rangle / \psi_T(R) \quad (4)$$

optimally guides the trial probability density toward the mixed distribution ($\Psi_T \Psi_0$) which is composed of the trial and exact ground state wavefunctions. In fact, the probability density evolves toward the mixed distribution exponentially fast in imaginary time.

$$f_t \xrightarrow[t \rightarrow \infty]{} \Psi_T \Psi_0 e^{-t(E_0 - E_T)} \quad (5)$$

The exponential factor on the right hand side of equation 5 provides the first glimpse of an instability in the DMC algorithm which will be addressed in the context of population control. The trial energy, E_T , is selected as close to the ground state energy (E_0) as possible to prevent the norm of the distribution from vanishing or diverging within the timescales of interest. The rapid evolution of the probability density similarly means that the DMC energy

$$E_{DMC}(t) = \frac{\int dR f_t(R) E_L(R)}{\int dR f_t(R)} \quad (6)$$

will also experience exponential convergence from the variational energy at zero time to the exact ground state energy at large imaginary times.

Thus far, we have considered only the formal outline of the DMC algorithm. The conversion of the formal method into a Monte Carlo sampling process introduces the practical issues of efficiency, stability, and accuracy. In practice, the short time Green's function is approximately factorized into a product of diffusion and branching terms.

$$\begin{aligned} G_d(R', R) &= \exp\left(-\frac{1}{4\tau}[R' - R - 2\tau\psi_\tau^{-1}(R)\nabla\psi_\tau(R)]^2\right) \\ G_b(R', R) &= \exp\left(-\frac{\tau}{2}[E_L(R') + E_L(R)]\right) \end{aligned} \quad (7)$$

Here the magnitude of the timestep, τ , is determined by the acceptable level of error in the breakup; later on τ will take on a meaning which differs from this standard usage. The diffusion term determines the spatial motion of the random walks, causing them to undergo Brownian motion which is biased toward regions of high probability as represented by the trial wavefunction. The branching term acts as a configurational weight, which is typically implemented as a birth-death process of the walkers.

The walker population at finite imaginary time is represented by the norm of the mixed distribution:

$$P_t = \int dR f_t(R) \quad (8)$$

Recalling the large time limit of the probability density in equation 5, it is clear that the walker population will perish or diverge, depending on the value of the trial energy. Yet, even if the trial energy is exact, local fluctuations in the branching weights (eq 7) will cause the dynamic population to vary in an unbounded fashion as imaginary time progresses. This is evident from the fact that the integral in equation 8 is approximated by a sum over all Monte Carlo trajectories (prior walker populations) which involve products of the branching weights up to time $t-\tau$. Thus the future weight of a single walker, represented by the number of its descendants, will exponentially disappear or diverge depending on the local energy along its future trajectory. The collective effect of unhindered branching is an instability in the walker population, which is clearly intolerable from a practical perspective.

Several procedures have been invented to restrict the fluctuations of the walker population at the expense of introducing additional bias into the calculation. Such procedures generally fall into one of two camps: fixed population methods, such as the “comb” (8, 9) and reconfiguration algorithms (10, 11), or the dynamic adjustment of the trial energy (1, 2, 12, 13). In this work, our attention will be restricted to the latter, as it is the most widely used method for population control in the DMC community. In the standard population control procedure, the trial energy is modified in the following way:

$$E_T(t) = \langle E_L \rangle_t - \frac{1}{g\tau} \log \frac{P_t}{P_0} \quad (9)$$

Here, the average indicates the current best estimate of the total energy and P_t and P_0 are the current and target populations, respectively.

The calculation of the total energy and population is what binds together the otherwise independent walkers, and causes a parallel bottleneck. The control parameter, g , is somewhat arbitrary and equals the typical number of generations (or branching steps) required for the population to return to its target value. In the absence of fluctuations, the population will return to its target value in exactly $g\tau$ units of imaginary time. These population control adjustments perturb the branching weights and hence introduce bias into any estimators. If the control parameter is too large, the population fluctuations will be hard to control; if it is too small, there could be a large bias on the estimated energy.

The population control bias has been addressed in only a handful of studies to date, possibly because its magnitude has been small for the typical range of physical system sizes (14). Though it has been known for some time that this bias ultimately scales exponentially with physical system size (13, 15), it seems that this fact is only becoming appreciated more recently (16, 17). It has been shown that the bias will decrease linearly with target population size (12, 13, 15, 18) and will decrease if the uncontrolled population fluctuations are small (10). The population control bias can be systematically reduced by updating the trial energy less often (12), increasing the target population size, loosening the dynamic control (13) (increasing g), or using importance sampling (12, 13, 18) with a better trial wavefunction. Attempts to completely correct for the bias include extrapolation in $1/P_0$ (12, 13) which requires many runs, and reweighting samples over a partial history (13, 15) in a single run, though this is not always more efficient than the extrapolation approach (19).

Delayed Feedback DMC

The pursuit of higher accuracy is one of the main reasons larger machines are developed for computational science. In the context of Quantum Monte Carlo this means that continual effort must be given to minimize statistical error and systematic biases. In order to meet these goals, we wish to reformulate Diffusion Monte Carlo in a simple way that removes the bottleneck to parallel efficiency and does not exacerbate the population control bias. Any modifications to the DMC algorithm should also maintain the stability of the algorithm, i.e. the population fluctuations must be controllable. Finally, we require that reproducibility be maintained, specifically that simulation results should not depend on machine conditions such as the number of nodes used or interruptions from background daemon processes.

A simple way of meeting these requirements is to introduce a time delay, or lag, in the population control feedback. Thus the trial energy used at time t only requires information from all walkers at time $t-L$. The opening of this window in “time” postpones any synchronization demands, and may remove them entirely provided that a load balancing can be done within this time window. This opens the possibility of near perfect parallel scaling even for a very small number of walkers per node. Introducing a feedback delay clearly satisfies the requirement of reproducibility since information from different imaginary times is not mixed. Its impact on stability and bias is not as obvious and requires further consideration and testing.

In general, the introduction of a feedback delay should decrease the stability of the simulation, meaning that fluctuations of the walker population should grow larger. Crudely speaking, the effect of the time delay is to increase the population control parameter (g) by the length of the delay (L), since it will take L more steps to reign in population fluctuations than it would otherwise. This loosening of control will allow larger fluctuations, increasing the probability that the population will explode, vanish, or simply become too volatile for effective load balancing.

The impact of a feedback delay on the population control bias is somewhat more subtle. To see its effect, consider walkers entering a region of very low local energy. Without a feedback delay, the population will immediately rise, causing a sharp response from the population control mechanism and hence a large bias. Introducing a time delay will allow walkers to diffuse back out of this biasing region before branching, thus smoothing out its effect and reducing the bias. The effect may or may not be large, but at the very least the introduction of delayed feedback should not increase the population control bias.

Dependence of Stability and Accuracy on DMC Parameters

The impact of delayed population feedback on stability and accuracy needs to be studied and quantified, with attention given to its interrelationship with other parameters used in DMC. Relevant parameters include the target population size, population control parameter, branching frequency, and trial energy update frequency. In order to probe these relationships empirically, we have developed a model that implements DMC for one of the simplest possible systems, the one-dimensional harmonic oscillator. This model is justified because the bias and population fluctuations should depend more strongly on the control method used than the actual details of the physical system.

Model system

The simple harmonic oscillator has the advantage that its imaginary time propagator is known exactly. This fact can be exploited to eliminate the timestep error typically incurred by the breakup into diffusion and branching terms (see eq 7). With the timestep error removed, the only bias remaining is due to the population control procedure, allowing us to study it in isolation and hence much more accurately. In order to mimic DMC, the importance sampled propagator is again factored into diffusion (G_d) and branching (G_b) terms:

$$\begin{aligned} G(R', R) &= G_d(R', R)G_b(R') \\ G_b(R') &\equiv \int dR G(R', R) \end{aligned} \quad (10)$$

In the limit of small imaginary time, the results in equation 7 are recovered.

Though the resulting DMC algorithm has no timestep error, the timestep used here still has an important connection to typical DMC simulation. In standard DMC, there are three timescales which are relevant to the discussion of population control bias. The first is the Trotter timestep, τ_t , which appears in the standard breakup given in eq 7. This timescale relates to the timestep error and it has been purposefully removed from our model. The other timescales are the branching period, τ_b , which is the time between branching operations, and τ_u , the time between updates of the trial energy (E_T). Branching more often without adjusting the trial energy should have no adverse effect on the bias, and so we set $\tau_b = \tau_u$ in our simulations. For the remainder of the work, we refer to this timescale as the timestep or the update period and will write it as τ . The bias will increase with more aggressive population control, and thus we expect to see a larger population control bias at small timesteps in our results.

The potential and trial wavefunction used for the model system are given by

$$V(x) = x^2 \quad \psi_T(x) = e^{-\frac{1}{2}ax^2} \quad (11)$$

with all quantities listed in dimensionless units ($\hbar = 1$, $m = 1/2$). The exact ground state energy for this system is equal to 1 and the ground state wavefunction is obtained when $a = 1$. Since a poor trial wavefunction will increase the branching rate, and thus more strongly manifest the population control bias, we use a trial function with $a = 0.1$ which badly oversamples low probability regions.

The exact importance sampled propagator, or Green's function, is given by

$$G(x', x; \tau) = (2\pi s)^{-\frac{1}{2}} e^{\tau E_T} e^{-\frac{1}{2}a(x'^2 - x^2)} e^{-\frac{1}{2s}[c(x'^2 + x^2) - 2xx']}] \quad (12)$$

Where c and s are defined as

$$c \equiv \cosh(2\tau) \quad s \equiv \sinh(2\tau) \quad (13)$$

The Green's function is then integrated, as in equation 10, to obtain the effective branching weight and diffusion transition probability.

$$\begin{aligned} G_d(x', x) &= \left(\frac{c + as}{2\pi s}\right)^{1/2} \exp\left(-\frac{c + as}{2s}\left[x' - \frac{x}{c + as}\right]^2\right) \\ G_b(x') &= e^{\tau E_T} (c + as)^{-1/2} \exp\left(-\frac{1}{2s}\left[c - as - \frac{1}{c + as}\right]x'^2\right) \end{aligned} \quad (14)$$

The transition probability can be sampled exactly using a Gaussian random number generator.

Definitions of stability and bias

Let us define exactly what we mean by stability and bias in DMC. Defining bias is straightforward: it is simply the deviation away from the exact ground state energy.

$$b \equiv \langle E \rangle - E_0 \quad (15)$$

Stability is most easily quantified in terms of its opposite: the volatility of the walker population. Here we define volatility as

$$v \equiv \langle f^2 \rangle - \langle f \rangle^2 \quad f \equiv \log(P_\tau/P_0) \quad (16)$$

which is the magnitude of population fluctuations.

Perhaps a more direct measure of stability is the probability that a simulation will not fail, meaning that the population remains within a practical range (we have used $P_0/10 < P < 10P_0$). Given the inexpensive nature of our calculations, we also explore this quantity by repeating runs many times in volatile areas of parameter space (which is spanned by g , P_0 , and τ). These quantities are linked in the sense that isosurfaces of the volatility closely match those of the stability. The reason for this close relationship is that both quantities are determined by the width of the population distribution. Thus conclusions drawn from the volatility are valid reflections of how viable, or stable, DMC simulations will be.

Results with zero feedback delay

A series of long simulations (3.34×10^8 samples) were performed over a range of control parameters ($g = 1, 2, 4, 8, 16, 32, 64, 128$), population sizes ($P_0 = 8, 16, 32, 64$), and update periods ($\tau = 0.0025, 0.005, 0.01, 0.02, 0.04$). These population targets are very small, but were chosen so that the volatility and bias would be increased and thus computed more accurately.

The relationship between volatility (v), control parameter (g), target population size (P_0), and update period (τ) can be inferred from figure 1. The data for various population sizes (differentiated by symbol) largely lie on top of one another, demonstrating the inverse relationship between volatility and population size. Simulations with $P_0 = 8$ were largely unstable and have been excluded from the fits (though the $1/P_0$ relationship still clearly held for these runs). The data also support a power law relationship between the volatility and $g\tau$. Slight deviations from power law behavior were found for $g < 4$ (data not shown). As g is taken below the critical threshold of 1, the population feedback overcorrects, causing the population to oscillate within an exponentially growing envelope. Thus, g can be regarded as an independent parameter only as this instability is approached (small g). To summarize, we find that:

$$v \propto \frac{g\tau}{P} \quad (17)$$

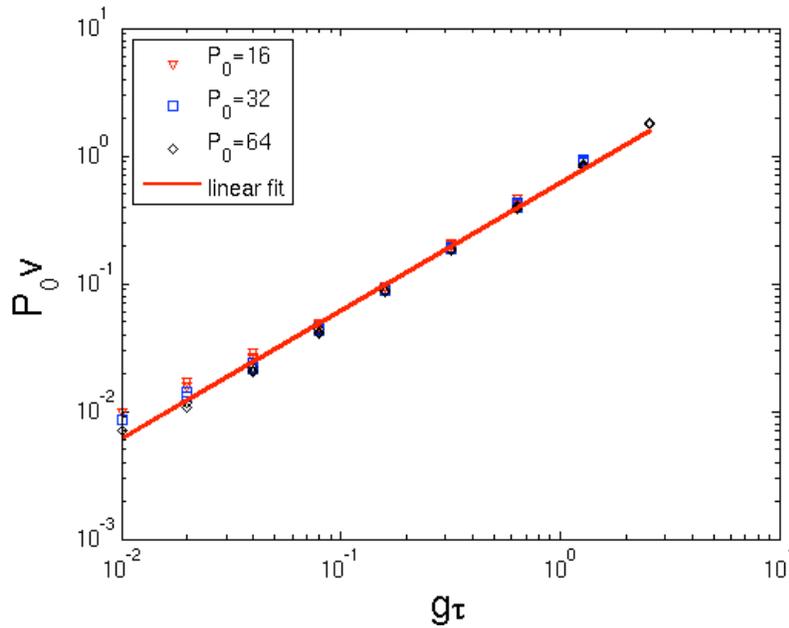


Figure 1. Population volatility vs. combined population control parameter and update period.

A similar relationship can be obtained for the population control bias. Figure 2 displays results for the bias over the same range of parameters in Figure 1. Again, it is immediately clear that the bias scales inversely with target population size. The bias is clearly not a power law in $g\tau$, but a logarithmic fit conforms to the data almost perfectly. The bias then approximately satisfies the empirical relation

$$b \propto \frac{1}{P} \log \frac{t_f}{g\tau} \quad (18)$$

where t_f is a constant roughly equal to 3.0. It should be noted that this form cannot hold in the limit of large $g\tau$ because the bias must strictly vanish in this

regime. A simple analysis of DMC using continuous weights rather than branching reveals that the bias must transition to an inverse relationship

$$b \xrightarrow{g\tau \rightarrow \infty} \frac{1}{Pg\tau} \quad (19)$$

in the large $g\tau$ limit.

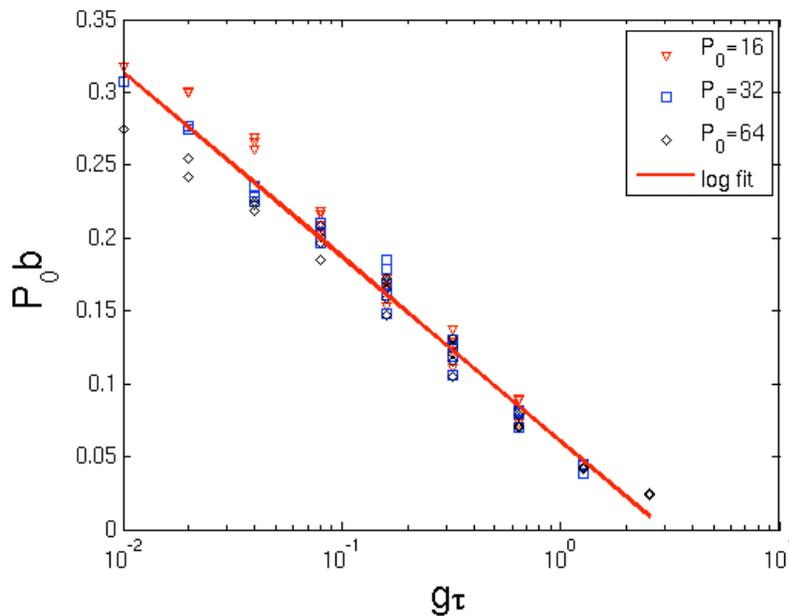


Figure 2. Population control bias vs. combined population control parameter and update period.

Although we have studied the bias in a simple model problem, our results should hold for other systems of typical interest involving singular potentials or the fixed node constraint. These complicating factors will indeed affect the bias, since they change the local energy distribution. However, the central limit theorem implies that these local energies, if averaged over enough steps, will converge to the same distribution as the model case, but with a different variance. Therefore the effects of nodes and singular potentials should be contained only within the fit constants in eq 18, and not in its functional form. Though this argument is fairly general, explicit tests of such systems should be made to fully confirm its validity.

Results with finite feedback delay

For the introduction a finite feedback delay (L), we expect that most of the prior results will simply be modified by replacing g with $g+L$. This expectation is largely borne out by the data, especially in the case of the volatility. Figure 3 shows filled stability contours of a 64 walker simulation with unstable regions becoming increasingly dark. Though the stability (probability of a successful run) is shown here, the essential features are identical to the inverse volatility. A dramatic effect of the feedback delay is seen in the region of tight control (small g) and finite delay.

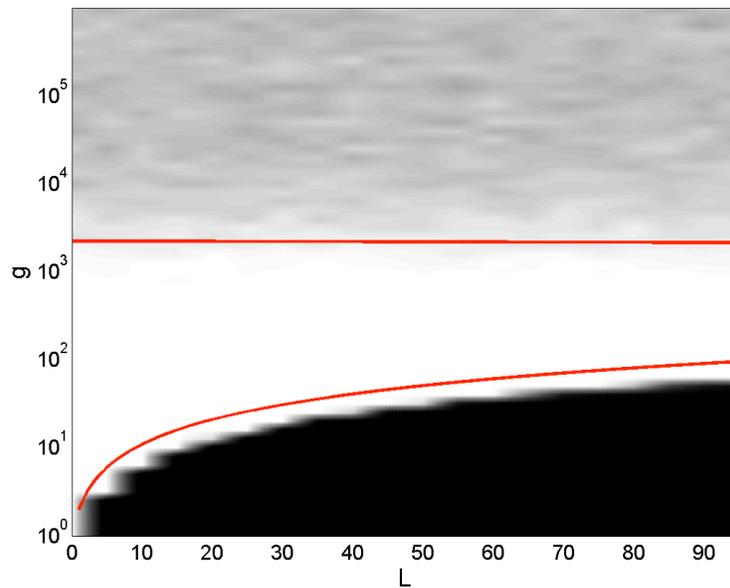


Figure 3. Stability vs. feedback delay and population control parameter shown with instability lines for $P_0=64$ and $\tau=0.01$. Brighter regions are more stable.

The strong instability seen in this region is a manifestation the critical instability discussed earlier, only here the system becomes unstable for all $g < L+1$. In this regime, population fluctuations are overcorrected, similar to the feedback experienced when a microphone is placed next to a loudspeaker. The upper solid line in the plot marks the expected isostability curve $g_{eff} = g + L$ shown along the boundary between stability and instability. For systems with smaller target populations, the stable (white) region becomes narrower and the predicted contour matches well with the observed stability until the critical instability is approached. Thus equation 17 is extended to obtain

$$v \propto \frac{(g + L)\tau}{P} \quad (20)$$

as long as the time delay satisfies

$$L < g - 1 \quad (21)$$

Provided L is chosen somewhat below this bound, the time delayed method should be as stable as the standard algorithm with a control parameter of $g + L$. Though the triangular stability region narrows as L increases, it should be noted that its size will increase proportional to the target population size.

Results for the population control bias demonstrate that a feedback delay does reduce the bias, though the beneficial effects are partially checked by an increase in volatility. Figure 4 illustrates this situation by showing the real decrease in the bias after the increased volatility is accounted for. Time delayed simulations were performed with a fixed control parameter ($g = 32$) over a range of delays (from 0 to 32). The time delayed data are plotted at an effective control parameter ($g+L$), so that data displayed at a particular value of g have the same volatility. It appears at first that adding a time delay equal to the control parameter halves the bias relative to the undelayed case. However, after the data are shifted to reflect the loss of stability, the actual gains are smaller. Still, increasing the feedback delay can be an effective means for reducing the population control bias provided the increased volatility is acceptable.

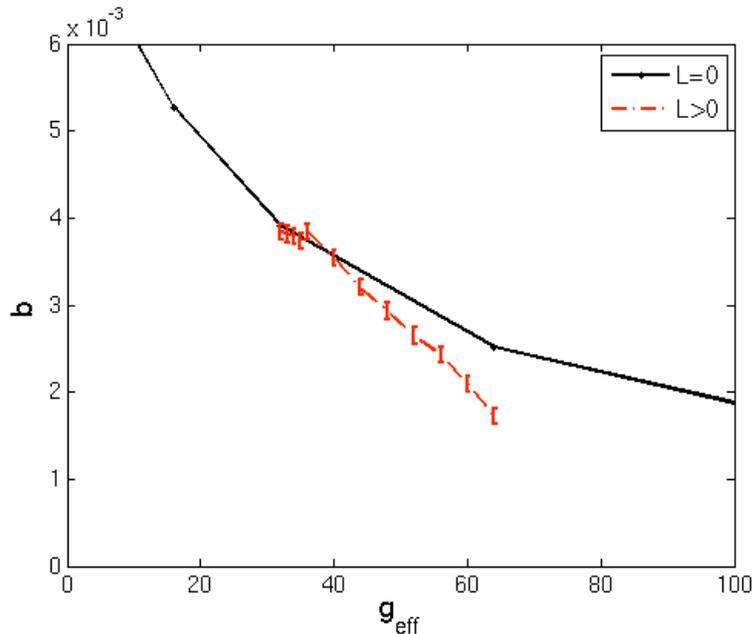


Figure 4. Population control bias vs. effective control parameter for standard (black) and time delayed (red) DMC at $\tau=0.01$ and $P_0=32$.

Practical Recommendations for DMC Simulations

Optimizing the balance between efficiency and accuracy is a significant goal of any discipline which employs large scale simulation. Finding the right tradeoffs in DMC is a case of selecting optimal values for the control parameters. Using the insight gained from the model system, we are now in a position to effectively navigate the parameter space toward the optimal solution.

A natural benefit of increasing machine size is that the target walker population must scale in the same way, providing an added reservoir of stability to the simulation and damping the bias. The increase in the size of the simulated physical system requires a decrease of the number of walkers per node in order to restore the serial efficiency lost from longer equilibration times. The low walker count per node increases the total idle time due to more interactions with synchronization barriers and strains load balancing algorithms, further reducing the parallel efficiency.

Introducing a population control feedback delay of sufficient length should push back the synchronization barriers inherent in standard DMC. Though the

time delay will increase the volatility of the walker population, and thus require frequent attention from the load balancer, there will be more time to get the balance right before the energy must be accumulated. The load should be defined in this case as the amount of time required to advance all walkers to a specific point in imaginary time, such as L steps beyond the average projection time of the walkers.

The population control parameter (g) should be selected so as to satisfy the lower bound in equation 21 while keeping it as small as is necessary to reign in the population fluctuations. If the population control bias is a primary concern, the population correction in the trial energy could be updated less often and/or standard correction methods such extrapolation (12) or reweighting (13) could be employed.

Conclusion

We have established empirical relationships which reveal the dependence of the population control bias and population volatility on the trial energy update period, population control parameter, and target population size. Though established by a thorough investigation of a simple model system, we expect that these relationships will apply generally to Diffusion Monte Carlo. We have proposed a simple modification to the DMC algorithm, namely the use of delayed feedback in the population control mechanism, which should increase the parallel performance of DMC simulations. This modification, which requires minimal change to existing codes, should function well with only a few walkers per node which will reduce the fraction of CPU time spent on walker equilibration. Model results have shown that the introduction of a feedback delay moderately reduces the population control bias after the increase in population volatility has been accounted for. Finally, we have provided general recommendations to improve serial and parallel efficiency while reducing bias by adjusting only the trial energy update frequency, population control parameter, and feedback delay. Future work will investigate the population control bias in real physical systems to assess the adequacy of conclusions drawn from the model system.

Acknowledgements

We gratefully acknowledge support from the U.S. Department of Energy under contract No. DE-F605-080R2335 and the National Science Foundation under grant OCI-0904572. Helpful comments and suggestions made by Jeongnim Kim and Bryan Clark have aided our work and we express our gratitude for their insights and encouragement.

References

1. Ceperley, D.M.; Alder, B.J. *Phys. Rev. Lett.*, **1980**, *45*, 566.
2. Reynolds, P.J.; Ceperley, D.M.; Alder, B.J., Lester, W.A. *J. Chem. Phys.*, **1982**, *77*, 5593-5603.
3. Baroni, S.; Moroni, S. *Phys. Rev. Lett.*, **1999**, *82*, 4745-4748.
4. Ceperley, D.M. *Rev. Mod. Phys.*, **1995**, *67*, 279-355.
5. McMillan, W.L. *Phys. Rev.*, **1965**, *138*, A442.
6. Kalos, M.H.; Levesque, D.; Verlet, L. *Phys. Rev. A*: **1974**, *9*, 2178.
7. Mentch, F.; Anderson, J.B. *J. Chem. Phys.*, **1981**, *74*, 6307-6311.
8. Davis, D.H. *Lawrence Radiation Laboratory Report No. UCRL-6707*, 1961.
9. Booth, T.E.; Gubernatis, J.E. *Phys. Rev. E.*, **2009**, *80*, 046704.
10. Buonauro, M.C.; Sorella, S. *Phys. Rev. B*: **1998**, *57*, 11446-11456.
11. Assaraf, R.; Caffarel, M.; Khelif, A. *Phys. Rev. E.*, **2000**, *61*, 4566.
12. Ceperley, D.M. *J. Comput. Phys.*, **1983**, *51*, 404-422.
13. Umrigar, C.J.; Nightingale, M.P.; Runge, K.J. *J. Chem. Phys.*, **1993**, *99*, 2865-2890.
14. Drummond, N.D.; Needs, R. J.; Sorouri, A.; Foulkes, W. M. C. *Phys. Rev. B*: **2008**, *78*, 125106.
15. Hetherington, J.H. *Phys. Rev. A.*, **1984**, *30*, 2713.
16. Warren, G.L.; Hinde, R.J. *Phys. Rev. E.*, **2006**, *73*, 056706.
17. Nemeč, N. *Phys. Rev. B*, **2010**, *81*, 035119.
18. Cerf, N.; Martin, O.C. *Phys. Rev. E.*, **1995**, *51*, 3679.
19. Drummond, N.D.; Radnai, Z.; Trail, J. R.; Towler, M. D.; Needs, R. J. *Phys. Rev. B* **2004**, *69*, 085116.