Exact Property Estimation from Diffusion Monte Carlo with Minimal Stochastic Reconfiguration

by

Ivana Bosá

B. Sc. (Physics) Faculty of Mathematics and Physics
Comenius University, Bratislava, Slovak Republic

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

in

The Faculty of Mathematics and Sciences

Department of Physics

BROCK UNIVERSITY

May 20, 2004

Abstract

Our objective is to develop a diffusion Monte Carlo (DMC) algorithm to estimate the exact expectation values, $\langle\Phi_0|\hat{A}|\Phi_0\rangle$, of multiplicative operators, such as polarizabilities and high-order hyperpolarizabilities, for isolated atoms and molecules.

The existing forward-walking pure diffusion Monte Carlo (FW-PDMC) algorithm which attempts this has a serious bias. On the other hand, the DMC algorithm with minimal stochastic reconfiguration provides unbiased estimates of the energies, but the expectation values $\langle\Phi_0|\hat{A}|\Psi\rangle$ are contaminated by Ψ, an user specified, approximate wave function, when $\hat{A}$ does not commute with the Hamiltonian. We modified the latter algorithm to obtain the exact expectation values for these operators, while at the same time eliminating the bias.

To compare the efficiency of FW-PDMC and the modified DMC algorithms we calculated simple properties of the H atom, such as various functions of coordinates and polarizabilities. Using three non-exact wave functions, one of moderate quality and the others very crude, in each case the results are within statistical error of the exact values.

Contents

List of Tables

List of Figures

Acknowledgements

I wish to express my gratitude for all support I received from Prof. Stuart M. Rothstein during my studies. It is a lifetime experience to work under his supervision, and I appreciate the opportunity he gave me.

I thank Dr. M. Caffarel for valuable correspondence on his algorithm. I also gratefully acknowledge generous grants of computer time at the University of Alberta and at the high performance computing facility located in Niagara College.

Also, I acknowledge support from the Thompson-Harrison Graduate Scholarship and the Edgar and Irmgard Penner Scholarship.

I would like to thank my husband Peter for all his patience and understanding during my studies and to thank my friends Prof. Julia Frankel, Stano, Zuzana and Vladimir for their support.

Chapter 1

Introduction

Quantum Monte Carlo (QMC) methods are powerful techniques used to solve the Shrödinger equation for atoms or molecules, specially suitable to calculate the ground-state energy and energy related properties.

Importance sampling in diffusion Monte Carlo (DMC) generates configurations distributed according to the mixed distribution $\Phi_0\Psi$ rather than the exact one Φ_0^2, where Φ_0 is the unknown exact wave function, and Ψ is an user specified guiding wave function. Therefore, the outputs are expectation values which are contaminated by Ψ: $\langle\Phi_0|\hat{A}|\Psi\rangle$. Exact expectation values (within the fixed-node approximation) are obtained only if $\hat{A}$ is the Hamiltonian or commutes with it. However, it is desirable to compute expectation values for operators which do not commute with Hamiltonian, i.e. those that are functions of coordinates or static moments of the charge distribution.

One of the widely-used methods to calculate expectation values for operators not commuting with Hamiltonian is the extrapolation method, [1]:

$$\langle\hat{A}\rangle_e \approx 2\frac{\langle\Phi_0|\hat{A}|\Psi\rangle}{\langle\Phi_0|\Psi\rangle} - \frac{\langle\Psi|\hat{A}|\Psi\rangle}{\langle\Psi|\Psi\rangle}$$

This rests on the approximation that the mixed distribution is "half-way" between the exact and variational one:

$$2\Phi_0\Psi - \Psi^2 \approx 2\Phi_0(\Phi_0 + \epsilon\chi) - (\Phi_0 + \epsilon\chi)^2 \approx \Phi_0^2 + O(\epsilon^2)$$

The accuracy of this approach is highly related to the quality of the trial wave function, and its bias is hard to assess. These uncertainties can be eliminated

by using truly pure estimators, such as provided by the forward walking method [2, 3, 4, 5]; the bilinear sampling [6, 7]; and the path integral method [8].

The most-explored algorithms are different implementations of the forward walking method. In general, forward walking methods estimate the ratio Φ_0/Ψ to convert the mixed distribution to the exact one. The ratio can be obtained from the number of asymptotic offspring of a walker [9]. This requires a tagging algorithm to determine at any time during the calculation which walker from a preceding configuration originated a present walker [3]. Alternatively the tagging process can be eliminated from the simulation by evaluation of weight proportional to the future progeny of every walker after each iteration [2, 4, 5].

Barnett *et al.* [3] describe two QMC algorithms to calculate exactexpectation values for coordinate operators. The first algorithm combines forward-walking with diffusion Monte Carlo. The configurations are drawn from the mixed distribution, and the estimated Φ_0/Ψ ratio converts the distribution to the exact one.

The second algorithm uses variational Monte Carlo (VMC) with DMC "side walks". Configurations are distributed according to the variational distribution Ψ^2 rather than the mixed one $\Phi_0\Psi$. The configurations drawn from Ψ^2 are then initial points for DMC side walks. After reaching the mixed distribution, the number of descendents is counted to estimate Φ_0/Ψ. Since the initial distribution of walkers is variational, the factor $|\Phi_0/\Psi|^2$ is necessary to obtain exact expectation values. This is accomplished by two independent samplings performed from each starting coordinate. The initial coordinates are stored and after the DMC side walk is finished the variational sampling may continue.

A different forward-walking procedure to obtain the exact expectation values is suggested by Casulleras and co-workers [4, 5]. Their method works only with the value of $A(\mathbf{R}_{i,j})$ in the present iteration. To sample the exact estimator of

A, they introduce an auxiliary variable P_j associated with each walker j. The simulation starts with DMC sampling giving as a result the mixed distribution of configurations: $\Phi_0\Psi$. At this point P_j is set to zero. As the walker evolves further, the variable P_j cumulates values of $A(\mathbf{R}_{i,j})$. If the walker is replicated the variable P_j is replicated as many times as the walker, without any changes. If the walker disappears the former contribution, stored in variable P_j, also does. After I iterations we have with M walkers, and the exact estimator of A is given by:

$$\langle A \rangle_e = \sum_{j=1}^{M} P_j/(I \times M).$$

Recently Saavedra and Kalos [6] reported bilinear diffusion quantum Monte Carlo a new method to calculate expectation values of the non-differential operators. Instead of sampling a function linear in the unknown exact wave function $\Psi\Phi_0$, the Schrödinger equation is transformed into a pair of integral equations whose solutions are bilinear, Φ_0^2. The random walk samples the ground state of the solution directly.

In this work we examine a new QMC algorithm of Dr. Caffarel [10] to test his claim that it improves the estimate of ground-state energy. His minimal stochastic reconfiguration method used for branching the walkers maximizes the efficiency of DMC with the absence of bias due population control as in Pure DMC [19]. We will extend its application by appending forward-walking to it to calculate exact non-differential properties of atoms, in particular, moments of electron-nucleus distance as well as the electrostatic properties for the H atom. To gauge the performance of the new algorithm we use three non-exact trial wave functions to complete this task, two very crude ones, although the exact solution is well known.

Previous work done in Dr.Rothstein's lab. [11, 12] using the exact wave function for H atom calculating the same quantities as we do gives us the possibility to compare previous simulation results with the new ones. Showing that the al-

gorithm is sufficiently robust to improve the calculations of the various properties would give us a tool for calculating properties of systems where the exact wave function is not known.

The following chapter provides the background of the DMC theory and the algorithms used for calculating the properties of isolated atoms and molecules. Therein also is given the description of the Dr. Caffarel's improved algorithm which we will adopt to calculate polarizabilities and high-order hyperpolarizabilities for H atom using a non-exact guiding wave function. In the third chapter appears a brief description of electrostatic theory relevant to this work. Our results and technical details of calculations are presented in fourth chapter. The last chapter contains a discussion of these results and conclusions.

Chapter 2

Diffusion Quantum Monte Carlo

2.1 Basic Theory

Diffusion Monte Carlo (DMC) is the simplest of the various quantum Monte Carlo (QMC) techniques available to solve the Schrödinger equation. The main idea behind DMC is similarity of time-dependent Schrödinger equation in imaginary time with a diffusion equation. The detail introduction to DMC can be found in a monograph [13] and a several reviews [14, 17].

The time-dependent Schrödinger equation can be written in atomic units ($\hbar = e = m_e = 1$) as:

$$i\frac{\partial\phi(\mathbf{R},t)}{\partial t} = -\frac{1}{2}\nabla^2\phi(\mathbf{R},t) + (V(\mathbf{R}) - E_T)\phi(\mathbf{R},t) \tag{2.1}$$

where $\mathbf{R}$ represents a 3N-dimensional configuration vector, N is the number of electrons, and corresponding Laplacian ∇ is in the form: $\nabla \equiv [\frac{\partial}{\partial x_1}, \ldots, \frac{\partial}{\partial x_{3N}}]$. E_T is an arbitrary energy shift.

Its solution can be written as:

$$\phi(\mathbf{R},t) = \sum_i C_i\Phi_i(\mathbf{R})e^{-it(E_i-E_T)} \tag{2.2}$$

where Φ_i and E_i are the eigenstates and eigenvalues of the Hamiltonian, respectively, and the coefficients C_i are defined as overlap of $\phi(\mathbf{R},0)$ with $\Phi_i(\mathbf{R})$.

This is a sum of exponentially decaying terms. Instead of a time-dependent superposition of oscillating states, *imaginary* time $\tau = it$ acts as a projector that

at large time extracts the lowest energy state.

$$\phi(\mathbf{R}, \tau \to \infty) \to C_0 \Phi_0(\mathbf{R}) e^{-\tau(E_0 - E_T)} \tag{2.3}$$

If we chose the trial energy E_T to be equal E_0 the time independence of (2.3) is achieved and the asymptotic form of $\phi(\mathbf{R}, \tau)$ will be a steady-state form.

By substituting imaginary time into equation (2.1) we can arrive at the imaginary-time Schrödinger equation:

$$-\frac{\partial \phi(\mathbf{R}, \tau)}{\partial \tau} = -\frac{1}{2}\nabla^2 \phi(\mathbf{R}, \tau) + (V(\mathbf{R}) - E_T)\phi(\mathbf{R}, \tau) \tag{2.4}$$

This equation without the second term on the right hand side describes the diffusion process of particles in the space driven by the density differences in the regions. Ignoring the first term on the right hand side, the second term results in a first-order rate equation with rate constant $(E_T - V(\mathbf{R}))$. This describes a situation when a source function is defined in the space, and the density of particles at given point $\mathbf{R}$ depends on the value of $V(\mathbf{R})$. In the region with $V(\mathbf{R}) < E_T$ the new particles are created; in the region with $V(\mathbf{R}) > E_T$ the particles are destroyed. Therefore equation (2.4) describes a diffusion process with time-dependent number of particles. In the Monte Carlo method we simulate both diffusion and the rate processes separately.

The imaginary time Schrödinger equation time can be written in the integral form

$$\phi(\mathbf{R}', \tau + \tau_a) = \int \mathcal{G}(\mathbf{R} \to \mathbf{R}', \tau_a)\phi(\mathbf{R}, \tau) d^{3N}\mathbf{R}, \tag{2.5}$$

using the Green's function $\mathcal{G}(\mathbf{R} \to \mathbf{R}', \tau_a)$, which obeys the same Schrödinger equation (2.4) as $\phi(\mathbf{R}, \tau)$ with the boundary condition:

$$\mathcal{G}(\mathbf{R} \to \mathbf{R}', 0) = \delta(\mathbf{R} - \mathbf{R}').$$

The Green's function consists of two parts, simulated by diffusion and branch-

ing:

$$\mathcal{G}(\mathbf{R} \rightarrow \mathbf{R}', \tau_a) = \frac{\exp[-(\mathbf{R}' - \mathbf{R})^2/2\tau_a]}{(2\pi\tau_a)^{3N/2}} \exp[-\tau_a(V(\mathbf{R}) - E_T)] \qquad (2.6)$$

The first part of Green's function is a multidimensional Gaussian spreading in time which represents diffusion of the configurations in configuration space. It can be simulated by random walk:

$$\mathbf{R} \rightarrow \mathbf{R} + \sqrt{\tau_a}\eta \qquad (2.7)$$

where η are random numbers drawn from a 3N-dimensional standard normal distribution with zero mean and unit variance. The second term is a branching term representing a source or sink of configurations. The branching process is simulated by creations and annihilations of the particles.

The process described above is inefficient due to the singularity in the potential energy which leads to large fluctuations in the population and large statistical uncertainties in expectation values. Importance sampling is used to reduce these fluctuations. Multiplying the equation (2.4) by trial function $\Psi(\mathbf{R})$, after some manipulations [11] we obtain the following equation:

$$\frac{\partial f(\mathbf{R},\tau)}{\partial \tau} = \frac{1}{2}\nabla^2 f(\mathbf{R},\tau) - \nabla(f(\mathbf{R},\tau)\frac{\nabla\Psi(\mathbf{R})}{\Psi(\mathbf{R})}) + [E_T - \frac{\hat{H}\Psi(\mathbf{R})}{\Psi(\mathbf{R})}]f(\mathbf{R},\tau) \qquad (2.8)$$

where

$$f(\mathbf{R},\tau) = \phi(\mathbf{R},\tau)\Psi(\mathbf{R}). \qquad (2.9)$$

Ψ is a good approximation to Φ_0, such as SCF, or small CI expansion, etc. This improves the efficiency (see below) of the simulation, but with the cost of biasing the random walk to produce the steady-state distribution $f(\mathbf{R},\infty)$ rather than $\phi(\mathbf{R},\infty)$.

$f(\mathbf{R},\infty)$ is a probability density; i.e., it must be normalizable and everywhere positive. This requires $\phi(\mathbf{R},\infty)$ $(= \Phi_0(\mathbf{R}))$ and $\Psi(\mathbf{R})$ to have the same nodes; that is, they have the same sign everywhere. DMC rests in this so-called "fixed-node" approximation [16, 18].

Eq. (2.8) describes the motion of particles in the external field with a time-dependent number of particles. The first term is a diffusion term. The second term is a term for drift, caused by an external field:

$$\mathbf{F}(\mathbf{R}) = \frac{\nabla\Psi(\mathbf{R})}{\Psi(\mathbf{R})}. \tag{2.10}$$

The drift moves particles from regions where Ψ^2 is small to the regions with large Ψ^2, improving the efficiency of the simulation. The third term is branching, with $(E^{loc}(\mathbf{R}) - E_T)$ in Eq. (2.8) replacing $(E_T - V(\mathbf{R}))$ in Eq. (2.4).
The local energy:

$$E^{loc}(\mathbf{R}) = \frac{\hat{H}\Psi(\mathbf{R})}{\Psi(\mathbf{R})} \tag{2.11}$$

is much smoother than the potential energy itself, which decreases the fluctuations in population. Again this improves the efficiency of the simulation.

Diffusion distributes the particles randomly in configuration space. The purpose of drift and branching is to give this random distribution the shape of the probability distribution function of the studied system.

The integral form of the imaginary time Schrödinger equation (2.8) can be rewritten as:

$$\begin{aligned} f(\mathbf{R}',\tau+\tau_a) &= \int \frac{\Psi(\mathbf{R}')}{\Psi(\mathbf{R})}\mathcal{G}(\mathbf{R}\rightarrow\mathbf{R}',\tau_a)f(\mathbf{R},\tau)d^{3N}\mathbf{R} \\ &= \int G(\mathbf{R}\rightarrow\mathbf{R}',\tau_a)f(\mathbf{R},\tau)d^{3N}\mathbf{R} \end{aligned} \tag{2.12}$$

The Green's function for equation (2.8) has the form [15, 16, 17]:

$$G(\mathbf{R}\rightarrow\mathbf{R}',\tau_a) \approx \frac{\exp[-(\mathbf{R}'-\mathbf{R}-\tau_a\mathbf{F}(\mathbf{R}))^2/2\tau_a]}{(2\pi\tau_a)^{3N/2}} \exp[-\tau_a(E^{loc}(\mathbf{R})-E_T)] \tag{2.13}$$

This Green's function is exact only in the limit $\tau_a \rightarrow 0$. This is a result of two approximations: First, we assume that one can separate the processes: diffusion, drift and branching in equation (2.8) and simulate these independently. Second, we assume that the magnitude and direction of drift during the move remains the

same, which, of course, is not the case. In order to ameliorate this time step bias, the calculation is performed using several different time steps, and the values are extrapolated to $\tau_a = 0$.

The Green's function (2.13) is simulated in DMC as follows:

- Distribute randomly a finite number of the walkers in the configuration space, each represents one copy of studied system (e.g. atom or molecule).
- Move each walker j at each iteration i through configuration space according to:

$$\mathbf{R}_{i,j} \rightarrow \mathbf{R}_{i,j} + \tau_a \frac{\nabla\Psi(\mathbf{R}_{i,j})}{\Psi(\mathbf{R}_{i,j})} + \sqrt{\tau_a}\eta_{i,j} \tag{2.14}$$

  where η are random numbers drawn from a 3N-dimensional normal distribution with zero mean and unit variance.

- Evaluate the branching factors:

$$b_{i,j} = \exp[-\tau_a(E^{loc}(\mathbf{R}_{i,j}) - E_T)], \tag{2.15}$$

  and make $\mathrm{int}(b_{i,j} + \xi)$ copies of walker at the same position, where ξ is random number uniformly generated from interval (0,1). If the number of copies is zero the walker is deleted.

After achieving a steady-state, the walkers are distributed according to the product of the trial function and the exact unknown function: $\Psi\Phi_0$, the so-called mixed distribution. The expectation value is calculated as the average value over each configuration through the iterations:

$$E_0 = \langle\Phi_0|\hat{H}|\Psi\rangle = \frac{\int \Phi_0\Psi\frac{\hat{H}\Psi}{\Psi}d\Omega}{\int \Phi_0\Psi d\Omega} \approx \frac{1}{M}\sum_{\{\mathbf{R}_{i,j}\}_{\Phi_0\Psi}} E^{loc}(\mathbf{R}_{i,j}) \tag{2.16}$$

and

$$\langle A\rangle_m = \langle\Phi_0|\hat{A}|\Psi\rangle = \frac{\int \Phi_0 A\Psi d\Omega}{\int \Phi_0\Psi d\Omega} \approx \frac{1}{M}\sum_{\{\mathbf{R}_{i,j}\}_{\Phi_0\Psi}} A(\mathbf{R}_{i,j}) \tag{2.17}$$

where 'm' denotes the mixed distribution. Note that in contrast to the energy, the expectation value of A, an operator that does not commute with Hamiltonian, is not exact.

2.2 Forward-Walking in Pure Diffusion Monte Carlo

In diffusion Monte Carlo the number of walkers can fluctuate due to the branching process. The population can either grow to infinity or entirely vanish; therefore, some population control is needed.

To simplify the computer codes by dealing with a fixed-size ensemble, the so-called pure diffusion Monte Carlo (PDMC) method was developed [19]. If we omit the branching part in the equation(2.3) we get a diffusion equation in the form:

$$\frac{\partial f(\mathbf{R},\tau)}{\partial \tau} = \frac{1}{2}\nabla^2 f(\mathbf{R},\tau) - \nabla(f(\mathbf{R},\tau)\frac{\nabla\Psi(\mathbf{R})}{\Psi(\mathbf{R})}) \tag{2.18}$$

which describes the diffusion and drift of a constant number of particles. As branching is absent, the PDMC algorithm samples from the variational distribution Ψ^2. By introducing weights for each walker j at each step of simulation i:

$$w^{(p)}(\mathbf{R}_{i,j}) = w_{i,j}^{(p)} = \prod_{k=i}^{i-L+1} b_{k,j} \propto \frac{\Phi_0}{\Psi} \tag{2.19}$$

we convert the variational distribution Ψ^2 to the mixed one $\Phi_0\Psi$. The weight of each walker represents its importance in the ensemble.

It is inefficient to cumulate the weights for the walkers for all past iterations, as the cumulative weights can become either very large or very small. Potentially only one walker can dominate the ensemble, which will give us a meaningless estimate for the ground-state properties. Therefore one truncates the number of weights in the past and uses only L previous values, and deals with this bias later.

The estimated ground-state energy is now a weighted average over the variational distribution of walkers:

$$E_0 = \langle \Phi_0 | \hat{H} | \Psi \rangle = \frac{\int \Psi^2 \frac{\Phi_0}{\Psi} \frac{\hat{H}\Psi}{\Psi} d\Omega}{\int \Psi^2 \frac{\Phi_0}{\Psi} d\Omega} \approx \approx \frac{\sum_{\{\mathbf{R}_{i,j}\}_{\Psi^2}} E^{loc}(\mathbf{R}_{i,j}) w^{(p)}(\mathbf{R}_{i,j})}{\sum_{\{\mathbf{R}_{i,j}\}_{\Psi^2}} w^{(p)}(\mathbf{R}_{i,j})} \tag{2.20}$$

This algorithm does not provide estimators of exact ground-state expectation values of operators which do not commute with Hamiltonian. To achieve the exact distribution of configurations we need to in effect "square" the weights for each walker. But to simply square the past weight is not correct. One needs two independent estimates of the weight at point $\mathbf{R}$ for each walker [2, 20]. This can be done by using a so-called "forward-walking algorithm".

For each configuration $\mathbf{R}_{i,j}$ and for each iteration we assign a "past" and a "future" weight using the previous L branching factors, as well as those L in the future. The future weight for j-th walker at i-th iteration is given by:

$$w^{(f)}(\mathbf{R}_{i,j}) = w^{(f)}_{i,j} = \prod_{k=i}^{i+L-1} b_{k,j} \propto \frac{\Phi_0}{\Psi} \tag{2.21}$$

and the "past-future" weights are given by:

$$w^{(p,f)}(\mathbf{R}_{i,j}) = w^{(p,f)}_{i,j} = w^{(p)}_{i,j} w^{(f)}_{i,j} \propto \left(\frac{\Phi_0(\mathbf{R}_{i,j})}{\Psi(\mathbf{R}_{i,j})} \right)^2 \tag{2.22}$$

The simulated distribution of configurations is the variational one, $\Psi^2(\mathbf{R}_{i,j})$, but the "past-future" weights effectively convert this distribution to the exact one: $\Phi_0^2(\mathbf{R}_{i,j})$.

Expectation values for a non-differential property represented by operator A, which does not commute with $\hat{H}$, are estimated by the "past-future" weighted averages:

$$\langle A \rangle_e = \langle \Phi_0 | \hat{A} | \Phi_0 \rangle = \frac{\int \Psi^2 \hat{A} (\frac{\Phi_0}{\Psi})^2 d\Omega}{\int \Psi^2 (\frac{\Phi_0}{\Psi})^2 d\Omega} \approx \frac{\sum_{\{\mathbf{R}_{i,j}\}_{\Psi^2}} A(\mathbf{R}_{i,j}) w^{(p,f)}(\mathbf{R}_{i,j})}{\sum_{\{\mathbf{R}_{i,j}\}_{\Psi^2}} w^{(p,f)}(\mathbf{R}_{i,j})} \tag{2.23}$$

The reason for truncation of the number of weights cumulated in the future is the same as of those in the past. Large values of L lead to an increase of variance of the weights as a function of the L stack-size. Unfortunately, as will be seen below, there is also a bias, most readily seen for crude trial functions.

2.3 Forward-Walking in Diffusion Monte Carlo

The variant of DMC with stochastic reconfiguration is presented by Assaraf, Caffarel and Khelif [10]. Their work contains a rigorous proof that any PDMC method is expected to diverge as the simulation time and the number of iterations increase, provided that Ψ is not exact.

The algorithm presented combined advantages of DMC's high efficiency without the bias in PDMC method. The reconfiguration process, which replaces the physical branching of the ensemble after drift and diffusion has been performed, is designed to minimize as much as possible the fluctuations of the weights at each step in the simulation.

For the i-th iteration of simulation process:

- A global weight associated with the entire population is calculated:

$$W_i = \frac{1}{M} \sum_{j=1}^{M} b_{i,j}, \tag{2.24}$$

  and the relative weight for the j-th walker is introduced:

$$\tilde{w}_{i,j} = \frac{b_{i,j}}{W_i}, \tag{2.25}$$

  where M is the number of walkers, and $b_{i,j}$ is branching factor for the j-th walker at the i-th iteration.

- The total population of M walkers is "reconfigured" by selecting with probability proportional to $\tilde{w}_{ij}$ the same number M of walkers. One divides the

population of walkers into two different sets: so-called "positive walkers" with $\tilde{w}_{i,j} \geq 1$, which can be potentially duplicated, and "negative walkers" with $0 \leq \tilde{w}_{i,j} < 1$, which can be potentially destroyed.

- The number of reconfigurations is defined as:

$$N_{\text{Reconf}} = \sum_{j+} |\tilde{w}_{i,j} - 1| = \sum_{j-} |\tilde{w}_{i,j} - 1| \tag{2.26}$$

where we sum over the set of positive $(j+)$ walkers or negative $(j-)$ walkers. To obtain the integer number of reconfigurations one computes $\text{int}(N_{\text{Reconf}} + \xi)$ where ξ is uniform random number drawn from the interval (0,1) Once the number of reconfigurations is known, N_{Reconf} walkers are removed from negative set of walkers and replaced by the same number of walkers from the positive set to keep the population constant.

The simulated distribution of configurations is the mixed one: $\Phi_0\Psi$. The ground-state energy is estimated by the weighted average of the local energy:

$$E_0 = \langle \Phi_0 | \hat{H} | \Psi \rangle = \frac{\int \Phi_0 \Psi \frac{\hat{H}\Psi}{\Psi} d\Omega}{\int \Phi_0 \Psi d\Omega} \approx \frac{\sum_{\{\mathbf{R}_{i,j}\}_{\Phi_0\Psi}} E^{loc}(\mathbf{R}_{i,j}) \omega^{(p)}(\mathbf{R}_{i,j})}{\sum_{\{\mathbf{R}_{i,j}\}_{\Phi_0\Psi}} \omega^{(p)}(\mathbf{R}_{i,j})} \tag{2.27}$$

where

$$\omega^{(p)}(\mathbf{R}_{i,j}) = \omega_i^{(p)} = \prod_{k=i}^{i-L+1} W_k \tag{2.28}$$

corrects the bias arising from maintaining the fixed ensemble size at each step of simulation. When the number of walkers goes to infinity the usual DMC and DMC with fixed number of walkers become equivalent due to constant average weight over an infinite population [21].

To sample an exact distribution of configurations we need to find an estimator for the "future" weights. Liu, Kalos and Chester [9] showed that the number of "descendents" of each configuration $\mathbf{R}_{i,j}$, "many" iterations later, is proportional to $\Phi_0(\mathbf{R}_{i,j})/\Psi(\mathbf{R}_{i,j})$. Thus by counting the number of "descendents" $n_{i,j} = n(\mathbf{R}_{i,j})$

many iterations in future (L) and using them as a weights, in effect we can convert the mixed distribution of configurations to the exact one.

The expectation value of property A taken over the exact distribution of configurations will be estimated as follows:

$$\langle A\rangle_e = \langle\Phi_0|\hat{A}|\Phi_0\rangle = \frac{\int \Phi_0 \hat{A}\Psi\frac{\Phi_0}{\Psi}d\Omega}{\int \Phi_0\Psi\frac{\Phi_0}{\Psi}d\Omega} \approx \frac{\sum_{\{\mathbf{R}_{i,j}\}_{\Phi_0\Psi}} A(\mathbf{R}_{i,j})\omega^{(p,f)}(\mathbf{R}_{i,j})n(\mathbf{R}_{i,j})}{\sum_{\{\mathbf{R}_{i,j}\}_{\Phi_0\Psi}} \omega^{(p,f)}(\mathbf{R}_{i,j})n(\mathbf{R}_{i,j})} \tag{2.29}$$

where now

$$\omega^{(p,f)}(\mathbf{R}_{i,j}) = \omega_i^{(p,f)} = \prod_{k=i}^{i-L+1} W_k \prod_{k=i}^{i+L-1} W_k \tag{2.30}$$

is correction factor for fixed size of ensemble in past and future.

2.4 Design of Simulation

In each of the algorithms described above rest on an approximate Green's function that becomes exact as τ_a approaches zero. This time step bias is ameliorated by extrapolating to zero from several runs performed at different time steps. Vrbik et al. [22] recommended equidistantly spaced time-step values. To obtain the most efficient fit the ensemble size is made proportional to $1/\tau_a^2$ as follows:

Number of walkers:

$$M_a = M_0\frac{\tau_0}{\tau_a} \tag{2.31}$$

Number of iterations:

$$I_a = I_0\frac{\tau_0}{\tau_a} \tag{2.32}$$

where τ_0 refers to a largest time step, the values of M_0 and I_0 are for the largest time step, and τ_a refers to a specific time step as well as the values of M_a and I_a.

In addition, we choose a time-step dependent stack size:

$$L_a = L_0(\frac{\tau_0}{\tau_a})^{1.5}. \tag{2.33}$$

Since we don't know the reasonable value of L_0 in advance, we perform the simulation using several initial values L_0.

The trial energy E_T is usually an estimate of the exact energy. The value of E_T doesn't affect the simulation process. Its only purpose is to avoid the problem with over- or underflowing range when calculating the branching factors.

2.4.1 Pure Diffusion Monte Carlo with Forward-Walking

- Initialize parameters:

  Choose n equidistantly distributed time steps (usually 6), with reasonable maximum time-step value, τ_0 [22]. Also choose values of M_0, I_0 and a set of L_0 values. The simulation is performed for each time-step separately, for all sets of L_0.

- For each time step:

  1. Update the values of M_a, I_a and L_a and initialize the position of walkers in configuration space.
  2. Equilibrate the ensemble, which means complete a sufficient number of iterations simulating the Green's function (2.13) by drift and diffusion of walkers to produce a variational distribution of walkers.
  3. Perform $2L_a - 1$ iterations, carrying out drift and diffusion of walkers. Calculate the local energy and branching factors for each walker at each iteration, as well as other properties of interest. Put $2L_a - 1$ values of branching factors and L_a values of energy and other properties into a stack.
  4. Perform the next I_a iterations, simulating the Green's function (2.13). At each iteration:

- Calculate values of branching factors, local energy $E_{k,j} = E^{loc}(\mathbf{R}_{k,j})$ and the properties of interest $A_{k,j} = A(\mathbf{R}_{k,j})$ for each walker j at each iteration k. Put these values at top of the stack.
- Calculate the weights for the set of walkers at the "present" iteration (the L_a-th most recent one) and the weighted averages of energy and the properties:

$$E_i = \frac{\sum_j^{M_a} E_{i,j} w_{i,j}^{(p)}}{\sum_j^{M_a} w_{i,j}^{(p)}} \tag{2.34}$$

$$A_i = \frac{\sum_j^{M_a} A_{i,j} w_{i,j}^{(p,f)}}{\sum_j^{M_a} w_{i,j}^{(p,f)}} \tag{2.35}$$

and repeat for all I_a iterations and each property.

5. Estimate the ground-state energy:

$$E_0 \approx \frac{1}{I_a} \sum_i^{I_a} E_i \tag{2.36}$$

and the ground-state expectation value of all sampled properties:

$$\langle A \rangle_e \approx \frac{1}{I_a} \sum_i^{I_a} A_i \tag{2.37}$$

2.4.2 Fixed Number of Walkers Diffusion Monte Carlo with Forward-Walking

- Initialize parameters:

  Choose n equidistantly distributed time steps (usually 6), with reasonable maximum time-step value, τ_0 [22]. Choose values of M_0, I_0 and a set of L_0 values. The simulation is performed for each time-step separately, for all sets of L_0.

- For each time step:

1. Update the values of M_a, I_a and L_a and initialize the position of walkers in the configuration space.

2. Equilibrate the ensemble, which means complete a sufficient number of iterations simulating the Green's function (2.13) by drift, diffusion and reconfiguration of walkers at each iteration base on the relative weights eq. (2.25) of each walker, to produce a mixed distribution of walkers.

3. Perform $2L_a - 1$ iterations, carrying out drift, diffusion and reconfiguration of walkers. Calculate the local energy and properties of interest for each walker at each iteration. Store the $2L_a - 1$ global weights and L_a values of energy and other properties into a stack.

4. Perform the next I_a iterations, simulating the Green's (2.13) function and at each iteration:

   - Calculate the global weights values of local energy $E_{k,j} = E^{loc}(\mathbf{R}_{k,j})$ and properties $A_{k,j} = A(\mathbf{R}_{k,j})$ for each walker j at each iteration k and put these values at top of the stack.
   - Count the number of descendents $n_{i,j}$ for each walker in the i-th iteration, backtracking from the most recent iteration. This is done using a tagging procedure, described in the Appendix A.
   - Calculate the average energy:

$$E_i = \frac{1}{M_a} \sum_j^{M_a} E_{i,j}^{loc} \tag{2.38}$$

     and the weighted-average of property A:

$$A_i = \frac{\sum_j^{M_a} A_{i,j} n_{i,j}}{\sum_j^{M_a} n_{i,j}} \tag{2.39}$$

     for each of the I_a iterations.

5. Estimate the ground-state energy:

$$E_0 \approx \frac{\sum_i^{I_a} E_i \omega_i^{(p)}}{\sum_j^{I_a} \omega_i^{(p)}} \tag{2.40}$$

and the ground-state value of all sampled properties:

$$\langle A \rangle_e \approx \frac{\sum_i^{I_a} A_i \omega_i^{(p,f)}}{\sum_i^{I_a} \omega_i^{(p,f)}} \tag{2.41}$$

Chapter 3

Charged Particles in Electrostatic Field

3.1 Multipole Moments Expansion

The purpose of this work is to test the performance of forward-walking DMC with minimal stochastic reconfiguration by doing calculations similar to those done previously in Dr.Rothstein's lab [11]. Therefore more details on calculating the atomic polarizabilties expectation values can be found elsewhere [11, 12]. Here we will give only a brief description of the theory relevant to this work.

When an atom, a molecule or a system of charged particles is placed in an electric field, the field distorts the electronic structure of the system. In classical electrodynamics the energy of such system E is given by:

$$E = E_0 + \sum_i q_i \phi(\mathbf{r_i}) \tag{3.1}$$

where E_0 is the energy of system without applied electric field, q_i represents charges placed at positions $\mathbf{r}_i$, and $\phi(\mathbf{r})$ is the potential of the external electrostatic field. The vector electric field generated by its potential ϕ is defined as follows:

$$\mathbf{E}(\mathbf{r}) = -\nabla\phi(\mathbf{r}) \tag{3.2}$$

The potential ϕ can be expanded into Taylor series in terms of spatial coordinates around the coordinate center using Einstein's summation convention:

$$\phi(\mathbf{r}) = \phi(O) + r_\alpha \frac{\partial}{\partial r_\alpha}\phi(O) + \frac{1}{2!} r_\alpha r_\beta \frac{\partial^2}{\partial r_\alpha \partial r_\beta}\phi(O) + \frac{1}{3!} r_\alpha r_\beta r_\gamma \frac{\partial^3}{\partial r_\alpha \partial r_\beta \partial r_\gamma}\phi(O) + \dots \tag{3.3}$$

where the Greek suffixes run through Cartesian coordinates x,y,z.

Now, using the components of electric field:

$$E_\alpha = -\frac{\partial}{\partial r_\alpha}\phi(O), \tag{3.4}$$

the electric field gradient:

$$E_{\alpha\beta} = \frac{\partial E_\alpha}{\partial r_\beta} = -\frac{\partial^2}{\partial r_\alpha \partial r_\beta}\phi(O), \tag{3.5}$$

and the electric field hyper-gradient:

$$E_{\alpha\beta\gamma} = \frac{\partial^2 E_\alpha}{\partial r_\beta \partial r_\gamma} = -\frac{\partial^3}{\partial r_\alpha \partial r_\beta \partial r_\gamma}\phi(O), \tag{3.6}$$

and introducing the multipole moments of the system:

$$q = \sum_i q_i \tag{3.7}$$

$$\mu_\alpha = \sum_i q_i r_{i\alpha} \tag{3.8}$$

$$Q_{\alpha\beta} = \sum_i q_i r_{i\alpha} r_{i\beta} \tag{3.9}$$

$$R_{\alpha\beta\gamma} = \sum_i q_i r_{i\alpha} r_{i\beta} r_{i\gamma} \tag{3.10}$$

we can rewrite the equation (3.3) in the *multipole expansion* of classical energy of the system

$$E = E_0 - q\phi_0 - \mu_\alpha E_\alpha - \frac{1}{2!}Q_{\alpha\beta}E_{\alpha\beta} - \frac{1}{3!}R_{\alpha\beta\gamma}E_{\alpha\beta\gamma} - \dots \tag{3.11}$$

The components of the electric field gradient, hyper-gradient, or higher order derivative of the electric field as well as the multipole moments tensors are symmetric with respect to interchange of any two suffixes.

We have many ways of expressing the multipole moments in the energy expansion without changing the physics of the interaction.

Using Laplace's theorem :

$$\nabla(\nabla\phi) = 0 \tag{3.12}$$

the traceless form of the multipole moments can be introduced:

$$\Theta_{\alpha\beta} = \frac{1}{2}\sum_i q_i(3r_{i\alpha}r_{i\beta} - r_i^2\delta_{\alpha\beta}) \tag{3.13}$$

$$\Omega_{\alpha\beta\gamma} = \frac{1}{2}\sum_i q_i(5r_{i\alpha}r_{i\beta}r_{i\gamma} - r_i^2(r_{i\alpha}\delta_{\beta\gamma} + r_{i\beta}\delta_{\alpha\gamma} + r_{i\gamma}\delta_{\alpha\beta})) \tag{3.14}$$

or, in the general case

$$M_{\alpha\beta\gamma\ldots\mu} = \frac{(-1)^m}{m!}\sum_i q_i r_i^{2m+1}\frac{\partial^m}{\partial r_{i\alpha}\partial r_{i\beta}\ldots\partial r_{i\mu}}(\frac{1}{r_i}) \tag{3.15}$$

where m is the rank of the multipole tensor $M_{\alpha,\beta\ldots\mu}$ [23]. The dipole moment is tensor of rank one; therefore the definition remains the same as in traced Cartesian form (3.8).

The multipole expansion in terms of the traceless multipole moments slightly differs from its traced Cartesian form

$$E = E_0 - \mu_\alpha E_\alpha - \frac{1}{3}\Theta_{\alpha\beta}E_{\alpha\beta} - \frac{1}{15}\Omega_{\alpha\beta\gamma}E_{\alpha\beta\gamma} - \ldots \tag{3.16}$$

We also omitted the monopole moment term $q\phi(\mathbf{r})$, because this vanishes for a neutral atom or molecule in their ground-state. Even for a non-neutral system this term is redundant due to gauge invariance of the electrostatic potential ϕ.

Summary of properties of traceless multipole moments and field tensors is as follows:

1. $\Theta_{\alpha\beta}$, $\Omega_{\alpha\beta\gamma}$, $E_{\alpha\beta}$, $E_{\alpha\beta\gamma}$, ... are symmetric in all their indices
2. $\nabla(\nabla\phi) = 0 \Rightarrow E_{\alpha\alpha} = 0, E_{\alpha\alpha\ldots} = 0$
3. $\Theta_{\alpha\alpha} = 0, \Omega_{\alpha\alpha\gamma} = 0, \ldots$

3.2 Perturbed Hamiltonian

The quantum system such as an atom or molecule has to be treated in the framework of quantum mechanics. The first step is replacing the classical Hamiltonian

with its operator version:

$$H = H_0 - \mu_\alpha E_\alpha - \frac{1}{3}\Theta_{\alpha\beta}E_{\alpha\beta} - \frac{1}{15}\Omega_{\alpha\beta\gamma}E_{\alpha\beta\gamma} - \frac{1}{105}\Phi_{\alpha\beta\gamma\delta}E_{\alpha\beta\gamma} - \frac{1}{945}M_{\alpha\beta\gamma\delta\epsilon}E_{\alpha\beta\gamma\delta\epsilon} - \frac{1}{10395}M_{\alpha\beta\gamma\delta\epsilon\zeta}E_{\alpha\beta\gamma\delta\epsilon\zeta} - \ldots \quad (3.17)$$

where the dipole, quadrupole, octupole, hexadecapole, dotriacontapole and tetrahexadecapole moments, respectively, are operators.

By solving the Shrödinger equation

$$\hat{H}\Phi = E\Phi \quad (3.18)$$

one can obtain the ground-state energy for such a system.

The perturbed ground-state energy of a general system with no symmetry can be Taylor-expanded in the electric field tensor components. There are two different notations used in the literature. Bishop and Pipin suggested notation in terms of X symbols [24]. We chose Buckingham's [25] instead:

$$E(E_\alpha, E_{\alpha\beta}, \ldots) = E_0 - \mu_\alpha^{(0)}E_\alpha - \frac{1}{3}\Theta_{\alpha\beta}^{(0)}E_{\alpha\beta} - \frac{1}{15}\Omega_{\alpha\beta\gamma}^{(0)}E_{\alpha\beta\gamma} - \frac{1}{2}\alpha_{\alpha\beta}E_\alpha E_\beta - \frac{1}{3!}\beta_{\alpha\beta\gamma}E_\alpha E_\beta E_\gamma - \frac{1}{4!}\gamma_{\alpha\beta\gamma\delta}E_\alpha E_\beta E_\gamma E_\delta - \frac{1}{3}A_{\alpha,\beta\gamma}E_\alpha E_{\beta\gamma} - \frac{1}{6}B_{\alpha,\beta,\gamma\delta}E_\alpha E_\beta E_{\gamma\delta} - \frac{1}{6}C_{\alpha\beta,\gamma\delta}E_{\alpha\beta}E_{\gamma\delta} - \frac{1}{15}E_{\alpha,\beta\gamma\delta}E_\alpha E_{\beta\gamma\delta} - \ldots \quad (3.19)$$

Here E_0 is the unperturbed ground-state energy, μ_α^0, $\Theta_{\alpha\beta}^0$, $\Omega_{\alpha\beta\gamma}^0$ are the components of permanent dipole, quadrupole and octupole of the system, respectively.

The polarizabilities satisfy several symmetry relationship due to the fact that each electric field tensor of rank larger than 2 is totally symmetric. Polarizabilities tensors of any system are required to be traceless in order to be defined in unique way.

Due to spherical symmetry of the atom all polarizabilities tensors with an odd number of suffixes are identically zero. The other polarizability tensors have symmetry relations between their different components as follows [11]:

$$\alpha_{xx} = \alpha_{yy} = \alpha_{zz} \tag{3.20}$$

$$\gamma_{xxxx} = \gamma_{yyyy} = \gamma_{zzzz} = 3\gamma_{xxyy} = 3\gamma_{yyzz} = 3\gamma_{xxzz} \tag{3.21}$$

$$\begin{aligned} B_{xx,xx} = \; & B_{yy,yy} = B_{zz,zz} = -2B_{xx,yy} = -2B_{xx,zz} \\ = \; & -2B_{yy,zz} = \tfrac{4}{3}B_{xy,xy} = \tfrac{4}{3}B_{xz,xz} = \tfrac{4}{3}B_{yz,yz} \end{aligned} \tag{3.22}$$

$$\begin{aligned} C_{xx,xx} = \; & C_{yy,yy} = C_{zz,zz} = -2C_{xx,yy} = -2C_{xx,zz} \\ = \; & -2C_{yy,zz} = \tfrac{4}{3}C_{xy,xy} = \tfrac{4}{3}C_{xz,xz} = \tfrac{4}{3}C_{yz,yz} \end{aligned} \tag{3.23}$$

Rothstein and co-workers [12] derived Monte Carlo estimators for these properties. They considered the electric field and field gradients as infinitesimal perturbations of the energy without correspondingly modifying the wave function, Ψ, which is considered fixed. Their formulas are the zero field and/or field gradients limits of Eq.(2.23), where from the Hellmann-Feynman theorem, A is equal to derivatives of the perturbed energy, $E - E_0$, with respect to the electric field and/or field gradients. The formulas were derived in the context of PDMC with forward-walking, but they are valid also for the fixed-ensemble-size DMC with forward-walking algorithm.

In the case of the α-polarizability:

$$\alpha_{xx} = -\frac{\partial^2 E}{\partial E_x^2}|_{E_\alpha=0, E_{\alpha\beta}=0,\ldots} = \tau(\langle\{\mu_x\}\mu_x\rangle_e - \langle\{\mu_x\}\rangle_e\langle\mu_x\rangle_e) \tag{3.24}$$

where we have the dipole moment operator for a single electron,

$$\mu_\alpha(\mathbf{R}_{i,j}) = -r_{\alpha\ i,j} \tag{3.25}$$

and

$$\{\mu_\alpha\}(\mathbf{R}_{i,j}) = -\sum_{k=0}^{L_a-1}(r_{\alpha\ i-k,j} + r_{\alpha\ i+k,j}) \tag{3.26}$$

in obvious notation. Eq.(3.24) is formally the covariance of $\{\mu_x\}$ and μ_x. It, and the others which follow below, involves only the serial correlation of known quantities, accumulated from iterations in the distant past through to the far future.

The second term vanishes for an atomic system, and finally the polarizability reduces to

$$\alpha_{xx} = \tau\langle\mu_x\{\mu_x\}\rangle_e \tag{3.27}$$

Estimators for the hyperpolarizabilities of atoms through the fourth power in the electric field are given as follows:

$$\gamma_{xxxx} = \quad \tau^3(\langle\mu_x\{\mu_x\}^3\rangle_e - 3\langle\{\mu_x\}^2\rangle_e\langle\mu_x\{\mu_x\}\rangle_e) \tag{3.28}$$

$$\gamma_{xxyy} = \quad \tau^3(\langle\mu_x\{\mu_x\}\{\mu_y\}^2\rangle_e - \langle\{\mu_y\}^2\rangle_e\langle\mu_x\{\mu_x\}\rangle_e) \tag{3.29}$$

$$B_{xx,xx} = \quad \tau^2\langle\mu_x\{\mu_x\}\{\Theta_{xx}\}\rangle_e \tag{3.30}$$

$$= \quad \tau^2\langle\Theta_{xx}\{\mu_x\}^2\rangle_e \tag{3.31}$$

$$B_{xx,yy} = \quad \tau^2\langle\mu_y\{\mu_y\}\{\Theta_{xx}\}\rangle_e \tag{3.32}$$

$$= \quad \tau^2\langle\Theta_{xx}\{\mu_y\}^2\rangle_e \tag{3.33}$$

$$B_{xz,xz} = \quad \tau^2\langle\mu_x\{\mu_z\}\{\Theta_{xz}\}\rangle_e \tag{3.34}$$

$$= \quad \tau^2\langle\Theta_{xz}\{\mu_x\}\{\mu_z\}\rangle_e \tag{3.35}$$

$$C_{xx,xx} = \quad \tau\langle\Theta_{xx}\{\Theta_{xx}\}\rangle_e \tag{3.36}$$

$$C_{xx,zz} = \quad \tau\langle\Theta_{xx}\{\Theta_{zz}\}\rangle_e \tag{3.37}$$

$$C_{xz,xz} = \quad \tau\langle\Theta_{xz}\{\Theta_{xz}\}\rangle_e \tag{3.38}$$

where for a single electron

$$\Theta_{\alpha\beta} = -\frac{1}{2}(3r_\alpha r_\beta - r^2\delta_{\alpha\beta}) \tag{3.39}$$

There are three estimators for $\alpha_{\alpha\beta}$ and nine for $\gamma_{\alpha\beta\gamma\delta}$: by permuting the x, y, and z indices one arrives at three of the form given in Eqs.(3.27) and (3.28), and six from Eq.(3.29). Similarly, there are twelve for $C_{\alpha\beta,\gamma\delta}$ and twenty seven for $B_{\alpha\beta,\gamma\delta}$.

It is important to note that these estimators avoid the well-used finite field approximation; for example [26].

Chapter 4

Technical Details

In this work we examine DMC with minimal stochastic reconfiguration by calculations of the ground-state energy for an atom, and later we extend this algorithm by introducing forward-walking to calculate non-differential properties: $\langle r \rangle_e$, $\langle r^{-1} \rangle_e$ and $\langle r^2 \rangle_e$. Then we apply this new algorithm to calculate more complex electronic properties of the atoms, such as static polarizabilities.

The best way to test a new algorithm is to choose the simplest system; therefore, we choose the hydrogen atom, where

- the exact wave function is known:

$$\Phi_0(\mathbf{r}) = \frac{1}{\sqrt{\pi}} exp(-r) \tag{4.1}$$

- the exact ground-state energy is known:

$$E_0 \equiv \frac{\hat{H}\Phi_0}{\Phi_0} = -0.5 \tag{4.2}$$

- and, being the simplest atomic system, using the DMC technique is very fast

Using the exact wave function makes weights Eq.(2.25) equal to one, and therefore the reconfiguration process is omitted. This is not an ideal situation, because our aim is to test the algorithm for an approximate guiding wave function, Ψ, where weights do not equal to one. Therefore we chose a non-exact trial wave function:

$$\Psi(\mathbf{r}) = \frac{\zeta^{3/2}}{\sqrt{\pi}} exp(-\zeta r) \tag{4.3}$$

and now the local energy has the form:

$$E_{i,j}^{loc} = \frac{\hat{H}\Psi(\mathbf{r}_{i,j})}{\Psi(\mathbf{r}_{i,j})} = -\frac{\zeta^2}{2} + \frac{\zeta}{r_{i,j}} - \frac{1}{r_{i,j}} \tag{4.4}$$

The difference between exact and non-exact wave function will introduce reconfigurations into the calculations.

To test the algorithm's sensitivity to the quality of the wave function, we use three orbital exponents, ζ: 0.9, which quality is typical of those routinely used in DMC simulations, and 0.4 and 0.1, which are very crude choices.

The local energy of this trial function has a singularity for an electron close to the nucleus. This will yield to a large branching factor for the walker and it will adversely affect the reconfiguration process. Thus a truncation of the exponent of the branching factor is used as follows:

$$-\tau_a(E_{i,j}^{loc} - E_T) = \begin{cases} -\tau_a(E_{i,j}^{loc} - E_T) & \text{if } |\tau_a(E_{i,j}^{loc} - E_T)| \leq 1.0 \\ -1.0 sign[(E_{i,j}^{loc} - E_T)] & \text{otherwise} \end{cases} \tag{4.5}$$

Furthermore, for some configurations the drift vector can be inappropriately too large. This will move the electron far from the region of reasonable probability. Therefore we choose for the drift the following truncation:

$$\tau_a\mathbf{F} = \begin{cases} \tau_a\mathbf{F} & \text{if } |\tau_a\mathbf{F}| \leq 1.0 \\ 1.0 sign[\mathbf{F}] & \text{otherwise} \end{cases} \tag{4.6}$$

The bias introduced by these schemes vanishes as $\tau_a \to 0$.

4.1 Design of Simulation for Polarizabilities

To calculate the polarizabilities we start with the multipole moments in their traceless form (Eq.3.8,3.13). There are 8 unique ones: $\mu_x, \mu_y, \mu_z, \Theta_{xx}, \Theta_{yy}, \Theta_{xy}, \Theta_{xz}, \Theta_{yz}$.

Now Θ_{zz} can be computed using Θ_{xx} and Θ_{yy}. Values of the $2L_a - 1$ recent ones are saved in the stack so the cumulative values (3.26) for each of those moments for each walker at the L_a-th most recent iteration can be calculated.

These cumulative values represent the past and the future for the set of walkers at the L_a-th most recent iteration. Therefore, it is very important when we reconfigure the walker at the recent iteration to reconfigure also its stack of multipole moments. This means, if the walker is deleted and replaced by another we insert the position of the walker and as well all previous values of its multipole moments.

The simulations otherwise follow the description given in Chapter 2 for fixed number of walkers DMC with forward-walking. Except for the $\gamma_{\alpha\beta\gamma\delta}$ hyperpolarizability the expectation values are estimated directly (Eq.3.27 and 3.30 - 3.38). For $\gamma_{\alpha\beta\gamma\delta}$ we have to first evaluate the estimators: $\langle\mu_x\{\mu_x\}\rangle_e, \langle\mu_y\{\mu_y\}\rangle_e, \langle\mu_z\{\mu_z\}\rangle_e, \ldots,$ $\langle\mu_x\{\mu_x\}\{\mu_y\}^2\rangle_e, \langle\mu_y\{\mu_y\}\{\mu_z\}^2\rangle_e, \langle\mu_z\{\mu_z\}\{\mu_x\}^2\rangle_e$; and after finishing the time step we combine them according to the Eqs.(3.28 and 3.29).

Up to the fourth-order there are four non-zero polarizabilities, each having one specifying constant [11]. In principle, only one polarizability component has to be evaluated to know all others components Eqs.(3.20) - (3.23). To decrease the statistical error and exploit the efficiency of doing so we evaluate in a single run all 51 polarizability components estimators Eqs.(3.27) - (3.38). The data for given polarizability with the same value of L_0 from all simulations were combined together and then extrapolated to zero time step (see Section 4.3).

4.2 Initial Values of Simulation Parameters

The first step of a DMC simulation is initialization of parameters such as number of walkers M_0, iterations I_0, values of time-steps and length of the stack L_0. In

our case these numbers are chosen base on the previous experience [11] and on the several preliminary runs.

We use six different time steps. This number allows us to explore the suitability of a different $\tau_a \rightarrow 0$ extrapolation models to our data. To decide upon the smallest and largest time step, we have to take in account that with increasing time step we increase the bias, but with decreasing time step we increase the variance of the final estimate. Therefore, with decreasing time step we have to increase the CPU time used to complete the simulation. This and the increasing ensemble size Eq.(2.31) - (2.33) are designed to maintain approximately equal variance for all time steps with the same value of L_0. The L_0 value has to be large enough to see vanishing L-bias of the estimates. The CPU time required for the simulation depends linearly on the number of particles N_a and the number of iterations I_a. The dependence on L_0 is also roughly linear.

The simulation is dependent on an initial random number generator seed; therefore to obtain uncorrelated estimates the simulation is run with several (ten) different values of seed. Simulation parameter values in table 4.1 are for one independent run. These take into account the hardware resources available and desired accuracy of the results.

4.3 Estimating the Ground-state Property

As the Green's function at the heart of DMC is exact only in the zero time-step limit [27, 28, 29], the simulated properties have a substantial time-step bias over the range of time steps employed here. It is particularly severe for the crude wave functions employed here.

To ameliorate this we extrapolate the simulated quantities to zero time-step, employing a quadratic- and, when the $\tau_a = 0$ intercept is consistent with it,

Table 4.1: Initial values of simulation parameters for H

Parameter	Value
Number of particles M_0	100
Number of iterations I_0	4000
Time steps	a) 0.1,...,0.6 [0.1]a.u.
	b) 0.05,...,0.3 [0.05]a.u.
Stack lengths L_0	a) 10,...,80[10]
	b) 20,...,50[10]

a) Values used to estimate the results shown in graphs 5.1 - 5.4

b) Values used to estimate the results shown in table 5.1

a cubic-polynomial model. (Polynomials are notoriously inflexible: there is a spread of intercepts, reflecting a model bias [30], each resulting from an assumed extrapolation model.) When two polynomial models are employed we use the mean value of the intercepts, weighted by the inverse of the squared standard error of the intercept. The statistical error is taken as the square root of the sum of the inverse of the squared standard errors [31].

This is done separately for simulations using stack-sizes proportional to the same value of L_0. Usually those which show no visible finite stack-size bias are the L_0 = 30, 40, 50 data sets. We compute the weighted mean and associated statistical error of these nominally unbiased values to arrive at our final estimate of the property.

The data analysis is illustrated for a typical case of $\langle r^2 \rangle$ as follows: Employing polynomial models, $\langle r^2 \rangle$ simulation data is extrapolated to zero time-step for each of the L_0 data sets; Table 4.2. Weighted means of the intercepts are computed, rejecting those with a visible stack-size bias; Table 4.3. Here we discard the L_0 = 20 data set estimate for the crudest wave function. The final estimate is the

weighted mean of the remaining, nominally unbiased data.

All used models for all data are shown in Appendix B.

Table 4.2: Regression models used to fit time-step biased $\langle r^2 \rangle$ diffusion Monte Carlo data for the ground state of the hydrogen atom. The guiding functions are 1s STOs. All entries are in atomic units.

ζ	L_0	model	τ-range	intercept
0.1	20	τ^2	0.05 - 0.2	2.872(48)
	20	τ^3	0.05 - 0.3	2.898(80)
	30	τ^2	0.05 - 0.25	2.979(43)
	40	τ^2	0.05 - 0.25	2.974(49)
	50	τ^2	0.05 - 0.3	2.968(49)
0.4	20	τ^2	0.05 - 0.2	2.993(34)
	20	τ^3	0.05 - 0.3	2.993(45)
	30	τ^2	0.05 - 0.2	2.986(38)
	30	τ^3	0.05 - 0.3	2.971(50)
	40	τ^2	0.05 - 0.2	2.973(43)
	50	τ^2	0.05 - 0.2	2.972(52)
0.9	20	τ^2	0.05 - 0.25	2.995(14)
	20	τ^3	0.05 - 0.3	2.983(27)
	30	τ^2	0.05 - 0.25	3.005(16)
	30	τ^3	0.05 - 0.3	2.994(30)
	40	τ^2	0.05 - 0.25	2.998(17)
	40	τ^3	0.05 - 0.3	2.981(32)
	50	τ^2	0.05 - 0.25	2.999(16)
	50	τ^3	0.05 - 0.3	2.980(32)

Table 4.3: Estimates of $\langle r^2 \rangle_e$ based on the data appearing in Table 4.2.

ζ	L_0	$\langle r^2 \rangle$
0.1	20	2.879(41)
	30	2.979(43)
	40	2.974(49)
	50	2.968(49)
final	30-50	2.974(27)
0.4	20	2.993(27)
	30	2.981(30)
	40	2.973(43)
	50	2.972(52)
final	20-50	2.984(17)
0.9	20	2.992(12)
	30	3.003(14)
	40	2.994(15)
	50	2.995(14)
final	20-50	2.996(7)

Chapter 5

Results and Conclusions

Our aim is to explore property estimation utilizing a DMC algorithm with a fixed number of walkers recently published by Caffarel *et al.* [10]. We compare this algorithm with the PDMC algorithm used previously in Dr.Rothstein lab. to calculate various non-differential properties of atoms [2, 12].

On graphs 5.1 we can see a dramatic failure of Rothstein's PDMC estimator Eq.(2.20) using as a guiding function a crude 1s STO with orbital exponent 0.1. For the more representative wave function with orbital exponent 0.9 the bias is barely visible. On the other hand, the DMC algorithm with fixed number of walkers shows no bias regardless of the wave function quality. These results suggest that his DMC algorithm with a fixed number of walkers improves the calculated energy as claimed by Caffarel *et al.* [10].

Extending the DMC algorithm to estimate non-differential properties we append the forward-walking to the existing algorithm. The graphs 5.2 - 5.4 show results from calculations of simple properties such as moments of electron-nucleus distance. Here we can see again that the PDMC algorithm for the crude wave function shows strong bias, whereas the DMC algorithm gives unbiased results for both wave functions.

Employing these approximate wave functions for for hydrogen, the simulated moments of electron-nuclear distance and the polarizabilities up to fourth order using the forward-walking DMC with fixed number of walkers are shown in table 5.1.

The Monte Carlo estimates agree with the analytical values, with a statistical error which increases substantially with decreasing overlap of the guiding function with the exact wave function. Naturally, our most precise results were obtained by employing the most accurate wave function, orbital exponent equal to 0.9. As measured by its overlap with the exact wave function, its quality is typical of those routinely used in DMC simulations; for example, an energy-optimized H_2 trial function, consisting of a linear combination of two STOs and a simple Jastrow-Pade electron correlation function [32].

Our other trial functions are much too crude to be representative of what is typically employed in the field. Nevertheless, the results suggest that the algorithm performs well, even for these cases.

In table 5.2 we now relate some of our results to those obtained from competing algorithms described in Chapter 1.

Casulleras and Boronat [4] use two wave functions for their simulations: a 1s STO with orbital exponent 0.9, and the product of the exact wave function with a Gaussian. The quality of the latter is worse than the first, and the results have larger statistical error. To compare the accuracy of these results with ours we have to take in account that Casulleras and Boronat [4] ran their simulation using only a single time step ($\tau = 0.05$), and the simulation was approximately 20 times longer then ours at the same time step. Each would reduce the variance relative to ours.

Barnett *et al.* [3] use for their calculations a 1s STO wave function with orbital exponent 0.95, a considerably better choice than ours. They explored the sensitivity of their algorithm to the time step used for the simulation and the convergence length when calculating the number of walkers. All of their results show high accuracy. We chose to display the one from DMC with forward-walking, albeit at a single time step without extrapolating to an infinite number

of iterations in the future.

The paper of Arias de Saavedra *et al.* [6], published last year, reports results from bilinear diffusion QMC. Their calculations are done for two cases. The first one is when no guiding function is used. The second uses as a guiding function one which obeys the cusp condition for all ζ:

$$\Psi(\mathrm{r}) = \exp\left(-\frac{\zeta r}{\zeta + r}\right)$$

where $\zeta = 0.1$. The results of both cases are very similar.

All simulation results reported in the table 5.2 are within statistical error of the analytical. We see no obvious advantage of our algorithm over those which use DMC, other than ours has been verified as unbiased, even when crude wave functions are employed.

In summary, we presented a forward-walking version of Caffarel *et al.*'s [10] DMC algorithm to provide unbiased estimates of expectation values of multiplicative operators. The algorithm combines the advantages of minimizing the variance arising from physically branching the walkers with the convenience of programming with a fixed-ensemble. We exploited the latter feature to estimate electrical properties, without recourse to the finite field approximation.

We tested the algorithm for the ground state of hydrogen atom using three guiding functions, two very crude. In each case we found satisfactory agreement with the analytical values of the moments of the electron-nuclear distance and for static polarizabilities and hyperpolarizabilities.

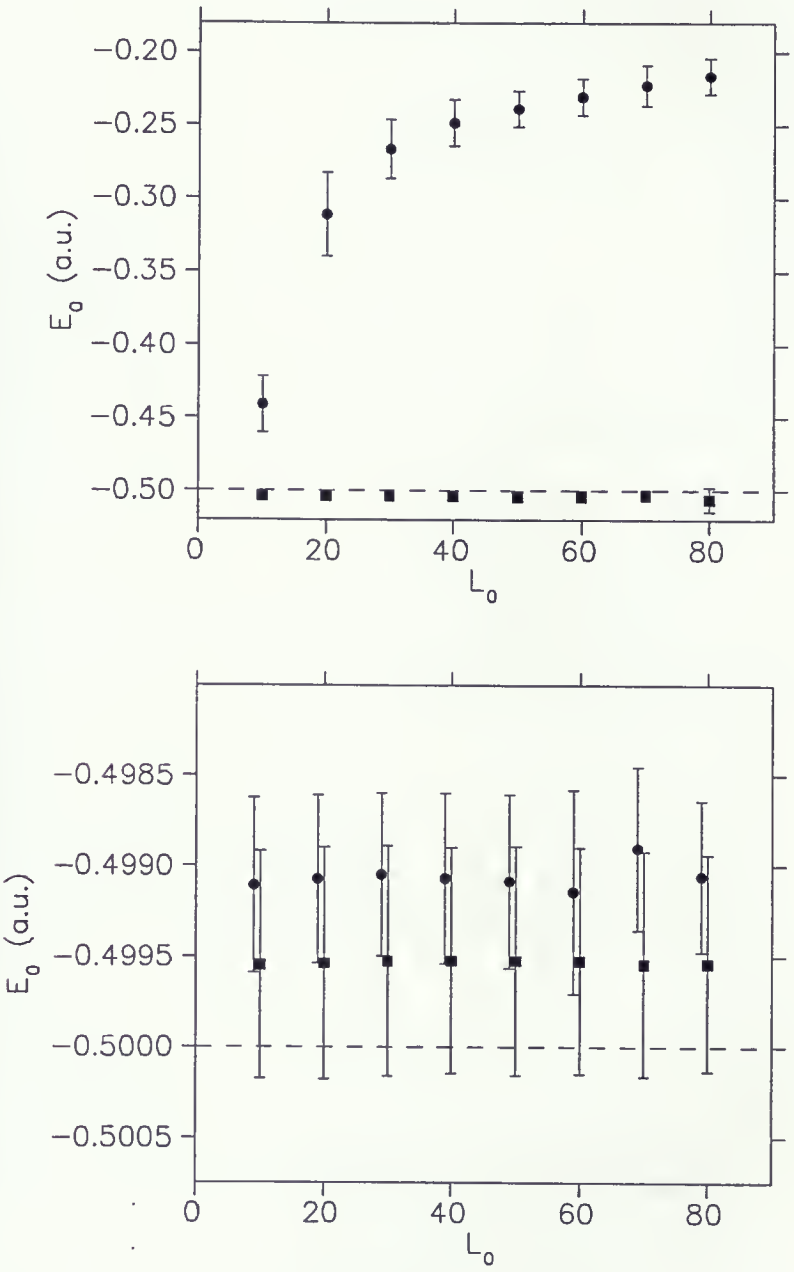

Figure 5.1: Estimated exact expectation values of energy for H atom for wave function with orbital exponent $\zeta = 0.1$ (top) and $\zeta = 0.9$ (bottom). Dashed line is the exact energy value; ● Pure diffusion Monte Carlo; ■ Diffusion Monte Carlo with fixed number of walkers.

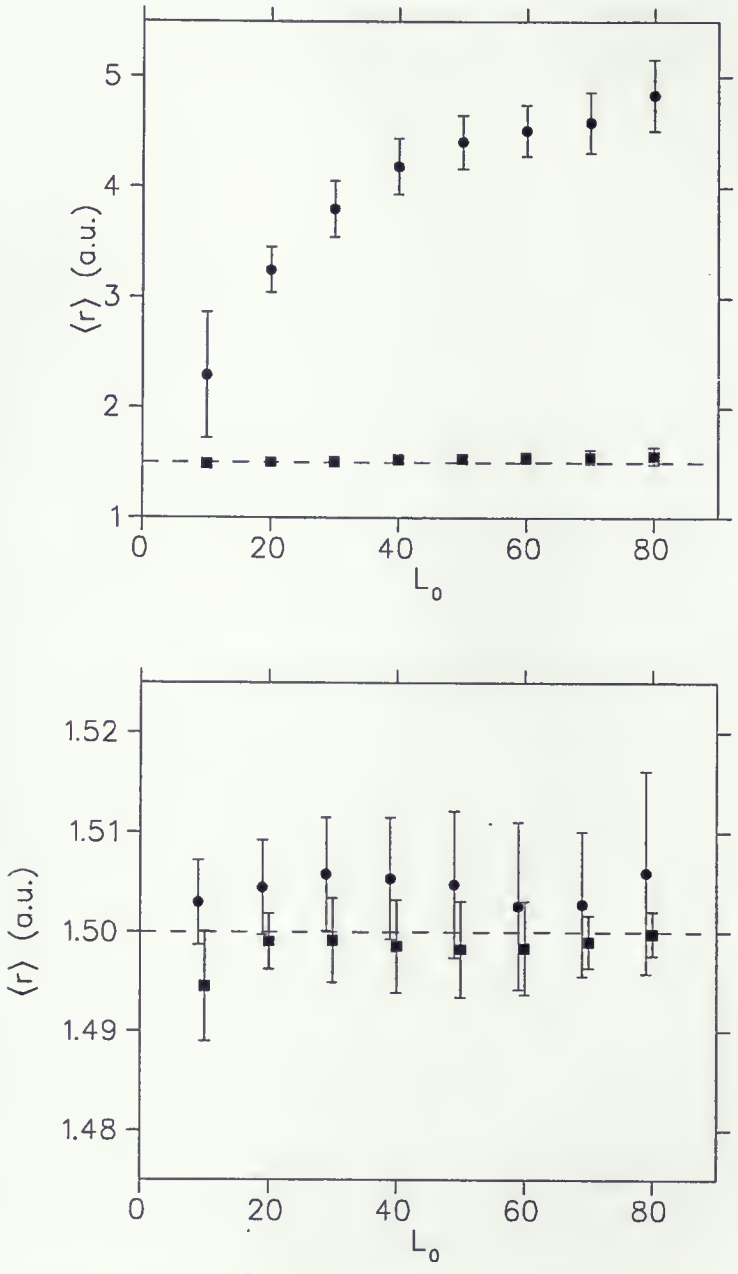

Figure 5.2: Estimated exact expectation values of $\langle r \rangle$ for H atom for wave function with orbital exponent $\zeta = 0.1$ (top) and $\zeta = 0.9$ (bottom). Dashed line is the exact value; ● Forward-walking pure diffusion Monte Carlo; ■ Forward-walking diffusion Monte Carlo with fixed number of walkers.

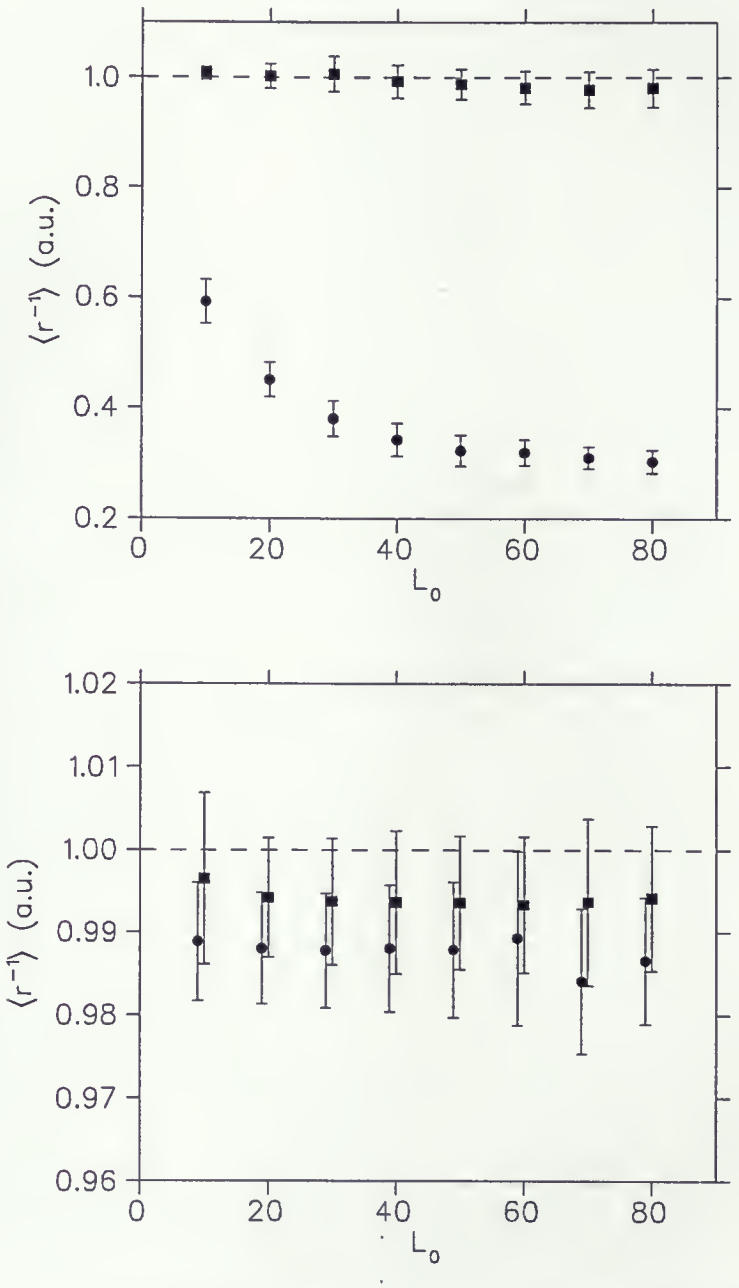

Figure 5.3: Estimated exact expectation values of $\langle r^{-1} \rangle$ for H atom for wave function with orbital exponent $\zeta = 0.1$ (top) and $\zeta = 0.9$ (bottom). Dashed line is the exact value; ● Forward-walking pure diffusion Monte Carlo; ■ Forward-walking diffusion Monte Carlo with fixed number of walkers.

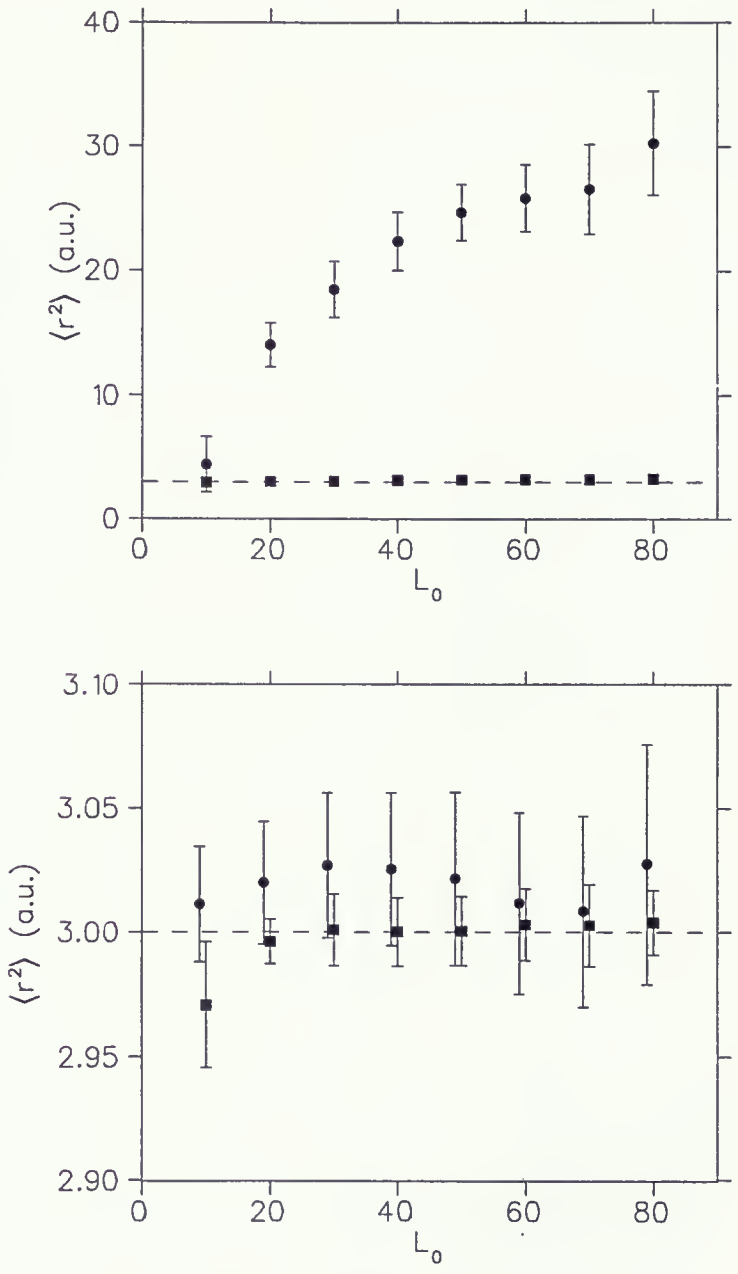

Figure 5.4: Estimated exact expectation values of $\langle r^2 \rangle$ for H atom for wave function with orbital exponent $\zeta = 0.1$ (top) and $\zeta = 0.9$ (bottom). Dashed line is the exact value; ● Forward-walking pure diffusion Monte Carlo; ■ Forward-walking diffusion Monte Carlo with fixed number of walkers.

Table 5.1: Expectation values from diffusion Monte Carlo simulations with a fixed number of walkers using 1s STO guiding functions for hydrogen atom ground state. All values are in atomic units.

ζ	S[a]	$\langle r \rangle_e$	$\langle r^{-1} \rangle_e$	$\langle r^2 \rangle_e$	α	γ	3C	B
0.1	0.190	1.4925(58)	1.010(4)	2.974(27)	4.65(15)	1335.(104)	14.24(81)	-118.5(81)
0.4	0.738	1.4970(29)	0.996(3)	2.984(17)	4.34(9)	1293.(89)	14.87(27)	-102.3(31)
0.9	0.996	1.5022(14)	0.995(2)	2.996(7)	4.57(3)	1341.(42)	15.11(12)	-105.9(14)
1.0[b]	1.000				4.499(2)	1327.(10)	15.06(3)	-106.9(3)
exact	1.	1.5	1.0	3.0	4.5[c]	1333.125[d]	15.[e,f]	-106.5[e]

[a] $\langle \Phi_0 | \Psi \rangle / \sqrt{\langle \Phi_0 | \Phi_0 \rangle \langle \Psi | \Psi \rangle}$

[b] Pure diffusion Monte Carlo; Ref. [12].

[c] Ref. [33], [d] Ref. [34], [e] Ref. [35, 36], [f] Ref. [37]

Table 5.2: Expectation values from quantum Monte Carlo simulations for hydrogen atom ground state. All values are in atomic units.

Ψ	$\langle r \rangle_e$	$\langle r^2 \rangle_e$	$\langle r^{-1} \rangle_e$	E_{var}
STO($\zeta = 0.9$) [a]	1.4999(10)	3.0002(36)	0.9987(10)	-0.495
STO($\zeta = 1.$)Gaussian [a]	1.4993(28)	2.995(14)	0.9975(14)	-0.4853
STO($\zeta = 0.95$) [b]	1.5023(15)	3.009(7)		-0.49875
none [c]	1.4995(26)	2.997(13)	1.0005(20)	n/a
rational function [c]	1.5001(36)	2.998(16)	0.9990(24)	
STO($\zeta = 0.9$) [d]	1.5022(14)	2.996(7)	0.995(2)	-0.495
exact	1.5	3.0	1.0	-0.5

[a] Casulleras and Boronat, DMC with auxiliary variables [4]

[b] Barnett *et al.*, DMC with forward-walking [3]

[c] Arias de Saavedra *et al.*, bilinear diffusion QMC [6]

[d] This work

Appendix A

Tagging Algorithm

We need to count the number of descendents $n_{i,j} \propto \Phi_0(\mathbf{R}_{i,j})/\Psi(\mathbf{R}_{i,j})$ "many" (L) iterations in "future" for each configuration sampled from $\Phi_0(\mathbf{R}_{i,j})\Psi(\mathbf{R}_{i,j})$ in the "present". In another words, for an arbitrary walker at time $i\tau$, we need to know which of the walkers at time $(i+L)\tau$ are its descendents. This information can be stored in a "family tree". As the walk progresses each walker is labeled, so its location in the tree is specified.

One way of tagging walkers during a DMC simulation was described previously by Lester Jr. and co-workers [38, 3]. Instead, we employ Vrbik's [39] tagging algorithm, which is tailored for simulations with a fixed ensemble of walkers and for applications where the walkers are branched with a high probability, as when crude guiding functions are employed. His algorithm is described as follows:

Each walker is labeled with the ensemble position number of its immediate "parent". This is illustrated by Fig. A.1 Here each column represents the ensemble positions of the walkers after one iteration. The position j of a walker in a given iteration is unique. Therefore, if this walker has offsprings in the succeeding iteration we can label them by its parent's position, j. The number in right upper corner is the walker's tag number, the position number of its parent from the preceding iteration. By knowing the parent from preceding iteration we can backtrack a particular walker at the $(i+L)$-th iteration to its ancestor at the i-th iteration.

The whole process of calculation number of descendents can be illustrated by

example of L = 4. Say we have six walkers, and we want to know how many descendents they will have three iterations later. For simplicity, we can assume that at each iteration there is only one reconfiguration: one walker is replaced by another, as we see in Fig. A.1

After the first iteration, the $(i+1)$-st one, walker number two was replaced by walker number one. Its position in the ensemble is again two, but its tag number is one. After the second iteration walker number three was replaced by walker two. Its position number again is three, but tag number is two. (Its direct ancestor has position number two at previous iteration). The same is repeated for next two iterations.

Now, we need to calculate the number of descendents at the $(i+3)$-rd iteration of walkers at the i-th iteration. To do this, we follow the tag numbers for each walker at the $(i+3)$-rd iteration back to the i-th one. This is shown on Fig. A.1 for walker number 4. Its tag number is 3, so its ancestor at the $(i+2)$-nd iteration was walker number 3. In turn, this walker has tag number 2, so its ancestor at the $(i+1)$-st iteration was walker number two. Similarly, this ancestor has tag number one. Thus the walker with position number one at i-th iteration will have increased its number of descendents by one.

We repeat this for all walkers at iteration $(i+3)$. Our example will give us for walkers at iteration i the following numbers of descendents three iterations in the future: $n_1 = 4$, $n_2 = 0$, $n_3 = 0$, $n_4 = 1$, $n_5 = 1$ and $n_6 = 1$.

We proceed in exactly the same manner when we move the "present" iteration to the next iteration, the $(i+1)$-st one. For this iteration, the numbers of descendents, three iterations in the future, $(i+4)$, are $n_1 = 1$, $n_2 = 3$, $n_3 = 0$, $n_4 = 0$, $n_5 = 1$ and $n_6 = 1$.

Of course, in practice the number of walkers at each iteration is larger, and the number of reconfigurations at each iteration will be more than one. Also, the

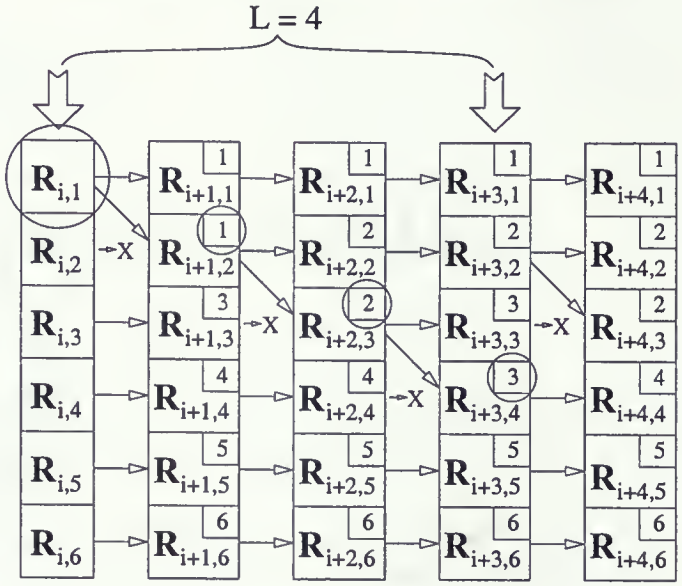

Figure A.1: Illustration of the tagging procedure for forward-walking algorithm with stack-size equal to 4.

replaced and duplicated walkers don't necessarily occupy ensemble positions next to each other, which we assumed for illustrative purposes. These don't present difficulties, as the key to the tagging algorithm is knowing a walker's parent from the preceding iteration, regardless of their mutual positions in the ensemble.

Appendix B

Extrapolation Models Used

Table B.1: Regression models used to fit the $\langle r \rangle$ data. All entries in atomic units.

ζ	L_0	model	τ-range	$\langle r \rangle(\tau = 0)$
0.1	20	τ^2	0.05 - 0.25	1.4713(88)
	30	τ^2	0.05 - 0.3	1.4926(91)
	40	τ^2	0.05 - 0.3	1.4953(100)
	50	τ^2	0.05 - 0.3	1.4886(117)
0.4	20	τ^2	0.05 - 0.3	1.4963(49)
	30	τ^2	0.05 - 0.3	1.5022(57)
	40	τ^2	0.05 - 0.3	1.4956(63)
	50	τ^2	0.05 - 0.3	1.4913(76)
0.9	20	τ^2	0.05 - 0.3	1.5008(28)
	30	τ^2	0.05 - 0.3	1.5031(30)
	40	τ^2	0.05 - 0.3	1.5027(33)
	50	τ^2	0.05 - 0.3	1.5036(34)

Table B.2: Regression models used to fit the $\langle r^{-1} \rangle$ data. All entries in atomic units.

ζ	L_0	model	τ-range	$\langle r^{-1} \rangle (\tau = 0)$
0.1	20	τ^2	0.05 - 0.3	1.0193(56)
	30	τ^2	0.05 - 0.3	1.0082(62)
	40	τ^2	0.05 - 0.3	1.0098(70)
	50	τ^2	0.05 - 0.3	1.0139(81)
0.4	20	τ^2	0.05 - 0.2	0.9923(47)
	20	τ^3	0.05 - 0.3	0.9921(67)
	30	τ^2	0.05 - 0.2	0.9960(62)
	30	τ^3	0.05 - 0.3	0.9972(84)
	40	τ^2	0.05 - 0.2	0.9995(71)
	40	τ^3	0.05 - 0.3	1.0039(93)
	50	τ^2	0.05 - 0.2	0.9997(78)
	50	τ^3	0.05 - 0.3	1.0046(102)
0.9	20	τ^2	0.05 - 0.2	0.9953(21)
	20	τ^3	0.05 - 0.3	0.9964(30)
	20	τ^3	0.05 - 0.25	0.9992(39)
	30	τ^2	0.05 - 0.2	0.9947(21)
	30	τ^3	0.05 - 0.3	0.9963(30)
	40	τ^2	0.05 - 0.2	0.9941(22)
	40	τ^3	0.05 - 0.3	0.9965(32)
	50	τ^2	0.05 - 0.2	0.9934(23)
	50	τ^3	0.05 - 0.3	0.9959(34)

Table B.3: Regression models used to fit the $\langle r^2 \rangle$ data. All entries in atomic units.

ζ	L_0	model	τ-range	$\langle r^2 \rangle(\tau = 0)$
0.1	20	τ^2	0.05 - 0.2	2.872(48)
	20	τ^3	0.05 - 0.3	2.898(80)
	30	τ^2	0.05 - 0.25	2.979(43)
	40	τ^2	0.05 - 0.25	2.974(49)
	50	τ^2	0.05 - 0.3	2.968(49)
0.4	20	τ^2	0.05 - 0.2	2.993(34)
	20	τ^3	0.05 - 0.3	2.993(45)
	30	τ^2	0.05 - 0.2	2.986(38)
	30	τ^3	0.05 - 0.3	2.971(50)
	40	τ^2	0.05 - 0.2	2.973(43)
	50	τ^2	0.05 - 0.2	2.972(52)
0.9	20	τ^2	0.05 - 0.25	2.995(14)
	20	τ^3	0.05 - 0.3	2.983(27)
	30	τ^2	0.05 - 0.25	3.005(16)
	30	τ^3	0.05 - 0.3	2.994(30)
	40	τ^2	0.05 - 0.25	2.998(17)
	40	τ^3	0.05 - 0.3	2.981(32)
	50	τ^2	0.05 - 0.25	2.999(16)
	50	τ^3	0.05 - 0.3	2.980(32)

Table B.4: Regression models used to fit the α polarizability data. All entries in atomic units.

ζ	L_0	model	τ-range	$\alpha(\tau = 0)$
0.1	20	τ^2	0.05 - 0.3	4.303(100)
	30	τ^2	0.05 - 0.25	4.408(169)
	40	τ^2	0.05 - 0.3	4.653(188)
	50	τ^2	0.05 - 0.3	4.649(232)
0.4	20	τ^2	0.05 - 0.2	4.840 (94)
	30	τ^3	0.05 - 0.3	4.272(184)
	30	τ^2	0.05 - 0.2	4.461(140)
	40	τ^2	0.05 - 0.2	4.316(185)
	50	τ^2	0.05 - 0.2	4.109(243)
0.9	20	τ^3	0.05 - 0.3	4.524 (68)
	20	τ^2	0.05 - 0.2	4.568 (46)
	30	τ^3	0.05 - 0.3	4.545 (88)
	30	τ^2	0.05 - 0.25	4.604 (52)
	40	τ^3	0.05 - 0.3	4.527(104)
	40	τ^2	0.05 - 0.2	4.604 (75)
	50	τ^3	0.05 - 0.3	4.537(115)
	50	τ^2	0.05 - 0.20	4.616 (85)

Table B.5: Regression models used to fit the γ hyperpolarizability data. All entries in atomic units.

ζ	L_0	model	τ-range	$\gamma(\tau = 0)$
0.1	20	τ^2	0.05 - 0.3	1635(153)
	20	τ^2	0.05 - 0.2	1125(231)
	30	τ^2	0.05 - 0.3	840(225)
	40	τ^2	0.05 - 0.3	953(244)
	50	τ^2	0.05 - 0.3	1587(274)
0.4	20	τ^2	0.05 - 0.3	1646 (63)
	30	τ^2	0.05 - 0.3	1279(111)
	40	τ^2	0.05 - 0.3	1308(176)
	50	τ^2	0.05 - 0.3	1338(274)
0.9	20	τ^2	0.05 - 0.2	1455(51)
	30	τ^2	0.05 - 0.3	1365(57)
	40	τ^2	0.05 - 0.3	1310(80)
	50	τ^2	0.05 - 0.3	1317(100)

Table B.6: Regression models used to fit the C hyperpolarizability data. All entries in atomic units.

ζ	L_0	model	τ-range	$3C(\tau = 0)$
0.1	20	τ^2	0.05 - 0.3	6.44(101)
	30	τ^2	0.05 - 0.2	13.78(125)
	40	τ^2	0.05 - 0.2	14.41(144)
	50	τ^2	0.05 - 0.2	14.78(161)
0.4	20	τ^2	0.05 - 0.3	15.06(41)
	30	τ^2	0.05 - 0.3	15.35(50)
	40	τ^2	0.05 - 0.3	14.51(63)
	50	τ^2	0.05 - 0.3	13.43(82)
0.9	20	τ^2	0.05 - 0.3	15.20(20)
	30	τ^3	0.05 - 0.3	15.13(46)
	30	τ^2	0.05 - 0.2	14.94(31)
	40	τ^3	0.05 - 0.3	14.71(51)
	40	τ^2	0.05 - 0.25	15.21(30)
	50	τ^3	0.05 - 0.3	14.64(55)
	50	τ^2	0.05 - 0.25	15.27(33)

Table B.7: Regression models used to fit the B hyperpolarizability data. All entries in atomic units.

ζ	L_0	model	τ-range	$B(\tau = 0)$
0.1	20	τ^2	0.05 - 0.3	-57.8(65)
	30	τ^2	0.05 - 0.3	-92.9(87)
	40	τ^2	0.05 - 0.3	-117.7(92)
	50	τ^2	0.05 - 0.2	-121.4(170)
0.4	20	τ^2	0.05 - 0.3	-111.6(28)
	30	τ^2	0.05 - 0.3	-102.8(41)
	40	τ^2	0.05 - 0.3	-101.1(58)
	50	τ^2	0.05 - 0.3	-102.6(86)
0.9	20	τ^3	0.05 - 0.3	-109.4(27)
	20	τ^2	0.05 - 0.25	-112.7(17)
	30	τ^3	0.05 - 0.3	-103.8(40)
	30	τ^2	0.05 - 0.25	-108.0(24)
	40	τ^3	0.05 - 0.3	-99.7(49)
	40	τ^2	0.05 - 0.25	-106.6(29)
	50	τ^3	0.05 - 0.3	-99.15(56)
	50	τ^2	0.05 - 0.25	-107.4(33)

Bibliography

[1] D.M. Ceperley, and M.H. Kalos in *Monte Carlo Methods in Statistical Physics*, edited by K.Binder, (Springer-Verlag, Berlin 1979).

[2] P. Langfelder, S.M. Rothstein, and J. Vrbik, J.Chem.Phys. **107**, 8525 (1997).

[3] R.N. Barnett, P.J. Reynolds, and W.A. Lester, Jr, J.Comp.Phys. **96**, 258 (1991).

[4] J. Casulleras, and J. Boronat, Phys. Rev. B **52**, 3654 (1995).

[5] A. Sarsa, J. Boronat, and J. Casulleras, J.Chem.Phys. **116**, 5956 (2002).

[6] F. Arias de Saavedra, and M.H.Kalos, Phys.Rev. E **67**, 026708 (2003).

[7] Shiwei Zhang, and M.H. Kalos, J.Stat.Phys. **70**, 515 (1993).

[8] A. Sarsa, K.E. Schmidt, and W.R. Margo, J.Chem.Phys. **113**, 1366 (2000).

[9] K.S. Liu, M.H. Kalos, and G.V. Chester, Phys.Rev. A **10**, 303 (1974).

[10] R. Assaraf, M. Caffarel, and A. Khelif, Phys.Rev. E **61**, 4566 (2000).

[11] M. Hornik, Master's thesis, Brock University, 2001.

[12] M. Hornik, and S.M.Rothstein in *Recent Advances in Quantum Monte Carlo Methods*, World Scientific 2002, pp. 71-94.

[13] B.L. Hammond, W.A. Lester Jr., and P.J. Reynolds, *Monte Carlo Methods in Ab Initio Quantum Chemistry*, volume 1 of *World Scientific Lecture*

and Course Notes in Chemistry, World Scientific Publishing Co. Pte. Ltd., Singapore, 1994.

[14] P.J. Reynolds, J. Tobochnik, and H. Gould, Comput.Phys. **6**, 682 (1990).

[15] S.M. Rothstein, and J. Vrbik, J.Chem.Phys. **87**, 1902 (1987).

[16] P.J. Reynolds, D.M. Ceperley, B.J. Alder, and W.A. Lester, Jr., J.Chem.Phys. **77**, 5593 (1982).

[17] B.H. Wells, Green's Function Monte Carlo Method in Methods in Computational Chemistry Vol 1. Plenum, New York, 1987.

[18] J.B. Anderson, J.Chem.Phys. **65**, 4121 (1976).

[19] M. Caffarel, and P. Claverie, J.Chem.Phys. **88**, 1088 (1988).

[20] P.J. Reynolds, J.Chem.Phys. **92**, 2118 (1990).

[21] M. Caffarel, private communication.

[22] J. Vrbik, and S.M. Rothstein, Int.J.Quant.Chem. **29**, 461(1986).

[23] D.E. Stogryn, and A.P. Stogryn, Mol.Phys. **11**, 371 (1966).

[24] D.M. Bishop, and J. Pipin, Chem.Phys.Lett. **236**, 15 (1995).

[25] A.D.Buckingham, Adv.Chem.Phys. **12**, 107 (1967).

[26] S.A. Alexander, R.L. Coldwell, G. Aissing, and A.J. Thakkar, Intern.J.Quantum Chem. S**26**, 213 (1992).

[27] S.M. Rothstein, and J. Vrbik, J. Chem. Phys. **87**, 1902 (1987).

[28] J.B. Anderson, and D.R. Garmer, J. Chem. Phys. **87**, 1903 (1987).

[29] P.J. Reynolds, R.K. Owen, and W.A. Lester, Jr., J. Chem. Phys. **87**, 1905 (1987).

[30] M.F. DePasquale, S.M. Rothstein, and J. Vrbik, J. Chem. Phys. **89**, 3629 (1988).

[31] W.C. Hamilton, *Statistics in Physical Science* (Roland, New York, 1964, pp. 41-42.

[32] M. Hornik, M. Snajdr, and S.M. Rothstein, J. Chem. Phys. **113**, 3496 (2000).

[33] I. Waller, Z.Physics **38**, 635 (1926).

[34] G.L. Sewell, Proc.Camb.Phil.Soc. **45**, 687 (1949).

[35] C.A. Coulson, Proc.Roy.Soc.Edin A **61**, 20 (1941).

[36] A.D. Buckingham, C.A. Coulson and J.T.Lewis, Proc.Phys.Soc.A **69**, 639 (1956).

[37] A. Dalgarno, Advan.Phys. **11**, 281 (1941).

[38] P.J. Reynolds, R.N. Barnett, B.L. Hammond, and W.A. Lester, Jr., J. Stat. Phys. **43**, 1017 (1986).

[39] J. Vrbik, private communication.

5196- 147