CHEMICAL APPLICATION OF DIFFUSION QUANTUM MONTE CARLO

PETER J. REYNOLDS
AND
WILLIAM A. LESTER, JR.

MATERIALS AND MOLECULAR RESEARCH DIVISION
LAWRENCE BERKELEY LABORATORY
UNIVERSITY OF CALIFORNIA

BERKELEY, CALIFORNIA

CHEMICAL APPLICATION OF DIFFUSION QUANTUM MONTE CARLO*

Peter J. Reynolds and William A. Lester, Jr.+
Materials and Molecular Research Division
Lawrence Berkeley Laboratory
University of California
Berkeley, California 94720

The diffusion quantum Monte Carlo (QMC) method gives a stochastic solution to the Schrodinger equation. This approach has recently been receiving increasing attention in chemical applications as a result of its high accuracy. However, reducing statistical uncertainty remains a priority because chemical effects are often obtained as small differences of large numbers. We give as an example the singlet-triplet splitting of the energy of the methylene molecule CH_2.

We have implemented the QMC algorithm on the Cyber 205, first as a direct transcription of the algorithm running on our VAX 11/780, and second by explicitly writing vector code for all loops longer than a crossover length C*. We discuss the speed of the codes relative to one another as a function of C*, and relative to the VAX. Since CH_2 has only eight electrons, most of the loops in this application are fairly short. The longest inner loops run over the set of atomic basis functions. We discuss the CPU time dependence obtained versus the number of basis functions, and compare this with that obtained from traditional quantum chemistry codes and that obtained from traditional computer architectures. Finally, we discuss some preliminary work on restructuring the algorithm to compute the separate Monte Carlo realizations in parallel--potentially allowing vectors of unlimited length.

*This work was supported in part by the Director, Office of Energy Research, Office of Basic Energy Sciences, Chemical Sciences Division of the U. S. Department of Energy under Contract No. DE-AC03-76SF00098, Director's Program Development Fund, Lawrence Berkeley Laboratory, and the Control Data Corporation.

+Also Department of Chemistry, University of California, Berkeley, California.

1. BACKGROUND

In recent years Monte Carlo methods have been increasingly applied to quantum-mechanical problems. Quantum Monte Carlo (QMC) methods fall into two major categories. Variational QMC[1] is a method of evaluating expectation values of physical quantities with a given (generally optimized) trial wave function Ψ_T. The procedure in effect amounts to evaluating a ratio of two integrals, although the actual Monte Carlo procedure is generally more sophisticated. The second major category of QMC is the "exact" type.[2] In these latter approaches the Schrödinger equation is actually "solved". It is not necessary to already have a highly accurate wave function in order to compute the expectation values. Properties of interest are in effect "measured" as the system evolves under the Schrödinger equation. When a stationary state is obtained, averages of the measured quantities give the desired expectation values.

Only recently have chemical calculations by exact QMC methods been carried out.[3,4] We will discuss here one such QMC method -- the fixed-node, diffusion QMC -- which we have been using in calculating molecular energies. In Section 2 we present the basic theory. Section 3 describes the algorithm. The implementation of this algorithm on the Cyber 205, its optimization, and results, are discussed in Section 4.

2. BASIC THEORY[4]

The Schrödinger equation may be rewritten in imaginary time, and with a constant shift in the zero of energy in the following form:

$$\frac{\partial \Psi(\underset{\sim}{R},t)}{\partial t} = [D\nabla^2 - V(\underset{\sim}{R}) + E_T]\ \Psi(\underset{\sim}{R},t) \quad . \tag{1}$$

Here $D \equiv \hbar^2/2m_e$, $\underset{\sim}{R}$ is the three-N dimensional coordinate vector of the N electrons, and $V(\underset{\sim}{R})$ is the potential energy (the Coulomb potential for a molecular system). Equation (1) is simply a diffusion equation combined with a first-order rate process, and thus may be readily simulated. The function $\Psi(\underset{\sim}{R},t)$ plays the role of the density of diffusing particles. These particles undergo branching (exponential birth or death processes) according to the rate term $[E_T - V(\underset{\sim}{R})]\ \Psi(\underset{\sim}{R})$. Thus, the number of diffusers increases or decreases at a given point in proportion to the density of diffusers already there.

The steady-state solution to Eq. (1) is the ground-state eigenfunction of the Schrödinger equation. Furthermore, the value of E_T at which the population of diffusers is asymptotically constant gives the energy eigenvalue E_0. The lowest eigenstate, however, is that of a Bose system. In order to treat a Fermi system, such as a molecule, we need to impose anti-symmetry on $\Psi(\underset{\sim}{R})$. A method which does this, and at the same time allows us to sample more efficiently (to reduce our statistical error), is importance sampling with an

anti-symmetrized importance function Ψ_I. The zeros (nodes) of Ψ_I become absorbing boundaries for the diffusion process, maintaining the anti-symmetry. A simple form for Ψ_I which gives the necessary anti-symmetry is a Slater determinant of molecular orbitals multiplied by a symmetric function of the coordinates.

To implement importance sampling, one simply multiplies Eq. (1) by Ψ_I and rewrites it in terms of a new probability density $f(\underset{\sim}{R},t)$ given by

$$f(\underset{\sim}{R},t) \equiv \Psi_I(\underset{\sim}{R})\,\Psi(\underset{\sim}{R},t). \tag{2}$$

The resultant equation for f can be written as

$$\frac{\partial f}{\partial t} = D\nabla^2 f + [E_T - E_L(\underset{\sim}{R})]f - D\nabla\cdot[fF_Q(\underset{\sim}{R})]\ . \tag{3}$$

The local energy $E_L(\underset{\sim}{R})$ and the "quantum force" $F_Q(\underset{\sim}{R})$ are simple functions of $\Psi_I(\underset{\sim}{R})$. Eq. (3), like Eq. (1), is a generalized diffusion equation, now with the addition of a drift term, due to the effect of F_Q. It is Eq. (3) that we solve stochastically. Using a Green's function approach, our diffusers are made to follow a "random walk" (Markov chain) in such a way that their asymptotic distribution is given by the steady-state solution, $f_\infty(\underset{\sim}{R})$, of Eq. (3). Properties of interest (such as the energy) are measured during the "walks", and are thus averages over the distribution $f_\infty(\underset{\sim}{R})$.

3. ALGORITHM

We present here an outline of the algoritnm for performing diffusion QMC. For more detail see Ref. 4. This algorithm is not structured specifically for the architecture of the Cyber 205. We will return to this point in the next section.

(0) Initialization. First generate an ensemble of N_c configurations of the N-electron system. Typically $N_c \approx 100\text{-}500$. These coordinates may be chosen randomly, or more efficiently from the distribution $|\Psi_I(\underset{\sim}{R})|^2$. This initial distribution is $f(\underset{\sim}{R}, t=0)$.

(1) Loop over blocks. In each block:

(2) Repeatedly loop over the ensemble until the time in each configuration has reached the chosen target time. For each member of the ensemble compute the inverse of the Slater matrix. Then:

(3) Loop over the electrons. Compute F_Q for the current electron. Move to

$$r' = r + D\tau F_Q + \chi \qquad (4)$$

where τ is the discrete time-step size, and χ is a 3-dimensional Gaussian random variable with a mean of zero

and a variance of $2D\tau$. This corresponds to the diffusive motion. If the electron crosses a node, eliminate the configuration from the ensemble and continue loop (2) over the ensemble. Otherwise update the Slater matrix and its inverse, and continue loop (3).

After all electrons in the current configuration have been moved, advance the time associated with this new configuration R' by τ. Calculate $E_L(\underset{\sim}{R}')$. Also calculate the branching factor, or multiplicity.

$$M = \exp\left(-\tau\{[E_L(\underset{\sim}{R}) + E_L(\underset{\sim}{R}')]/2 - E_T\}\right). \qquad (5)$$

Return M copies of this configuration to the ensemble. This branching, or birth and death process, corresponds to the rate term in Eq. (3). Weight all averages by M. Continue loop (2). After all members of the ensemble have reached the target time, the current block is finished. Use $\langle E_L \rangle$ to update E_T. Store $\langle E_L \rangle$ and the other averages. "Renormalize" the ensemble back to its original size N_c. (This is necessary because the population grows or shrinks exponentially. Although we have endeavored to make the exponent close to zero [cf Eq. (5)], asymptotically at large time the population will either vanish or overflow the allocated storage.) Reset all averages to zero. Continue loop (1) for the desired number of blocks.

(4) Average over blocks.

4. CYBER 205 IMPLEMENTATION.

The problem we chose to study is the singlet-triplet energy splitting of the methylene molecule, CH_2. CH_2 is fairly typical of the molecules we have been studying by QMC, in terms of the number of electrons and the number of nuclei. As a result, most of the inner loops in this application are quite short. The longest inner loop runs over the set of atomic basis functions. With this in mind, we present our results on the relative performance of the Cyber 205 and the VAX 11/780. To compare with the CDC 7600, we note that our code runs almost exactly ten times faster on the 7600 than on the VAX.

We have implemented the QMC algorithm on the Cyber 205, initially by simply transcribing our working program from the VAX to the Cyber. The major impediment at this stage was the lack of unformatted I/O on the Cyber and, even worse, its inability to handle logical records longer than 137 bytes. After rewriting these portions of the code, the program finally ran.

With automatic vectorization both on and off, the Cyber ran approximately 16 times the speed of the VAX. Apparently, any speed-up from vectorization of the longer loops was lost to the start-up time for vectorizing the short loops. It seemed clear that explicit vectorization was required. Thus, as our next step, all long inner loops of constant length were written explicitly in vector

syntax, while short constant-length loops were left as DO loops. Most loops in the code, however, are of variable length. These were all recoded in the form:

```
IF (length .GT. C*)      THEN

        [Vector code]

ELSE

        [Scalar code]

END IF.
```

We present in Figure 1 our performance results as a function of the crossover length C*. At values of C* greater than 26 the scalar section of code is always being executed, and thus the curve flattens out. For C* less tnan approximately 16, it appears that vector start-up time hinders performance. The optimum crossover point appears to be around 16. The lowest of the three curves corresponds to the implementation described above. Subroutine calls are quite costly on the Cyber 205. Thus in the middle curve we show the result of removing two short subroutines (both written in IF-THEN-ELSE form) and substituting vector code directly into the calling

programs. The speed-up is fairly dramatic, providing a peak speed of close to 20 times the VAX (up from 17).

Interestingly, although the compliler recognizes that A**2 should be replaced by A*A, inside of vector code A**2 calls the float-to-an-integer-power routine. Needless to say, this is costly. Essentially, changing one line of vector code from A**2 to A*A led to the improvement shown in the top curve. Clearly the improvement is most pronounced for small C*, where this line of code is being executed more frequently.

As mentioned earlier, the longest inner loop is over the number of atomic basis set functions, n. Traditional quantum chemistry codes scale as n^4 or n^5. Thus increasing the size of the basis set can be very costly. In our QMC approach, the algorithmic dependence on n is linear. In Fig. 2 we plot the relative run times as a function of basis set size on both the VAX (upper curve) and the Cyber 205 (lower curve). Both curves are indeed fairly linear in n. However, the slope for the Cyber is almost flat. This smaller slope is due to an increase in the vector length rather than an increase in the number of machine instructions being executed. The result is a speed enhancement of 30 over the VAX (up from 20) by n=50.

Although a factor of 30 over the VAX (or equivalently a factor of 3 over the 7600) is certainly good, it is nowhere near our hoped

for performance. This can be explained by the fact that even loops of length 50 are relatively short on the Cyber 205. Possibly more important, however, is that the relatively long inner loops constitute only a fraction of the code being executed. Thus, truly high speed for this kind of application requires on architectural rewrite of the code.

Looking over the algorithm (cf Sect. 3) it is clear that the entire structure is highly parallel. This is a fairly general characteristic of Monte Carlo codes. Thus, on a parallel processor the loop (1) over blocks can be eliminated, and each block can be computed independently on a separate processor. There is no communication required between processors until the very end, when [step (4)] the average over blocks is computed.

For a truly efficient Cyber 205 algorithm, however, loop (1) is too short to vectorize, generally ranging between 10 and 100. Loop (2) is much more desirable to vectorize, with $N_c \approx 100\text{-}500$. To do so, this loop must be made innermost in the new algorithm. In other words, the entire ensemble must be treated in parallel. Furthermore, the vector length is dynamic, since at each time-step the birth and death process modifies the ensemble size. We are currently developing this fully vector code for future implementation. This code appears to have great potential for fully exploiting the vector capabilities of the 205.

Finally in Table 1, we present our results on the singlet-triplet energy splitting of methylene, and compare these results with theory and experiment.

5. ACKNOWLEDGMENTS

We would like to thank the Control Data Corporation for a grant of computer time on the Cyber 205 at Colorado State University. We also thank the Institute for Computational Studies at Colorado State University and the Center for Advanced Vector Technology for their orientation program and assistance. Finally, we thank Steve McGrogan for many helpful comments on the code, and Robert Barnett for preparing the figures.

REFERENCES

1. W. L. McMillan, Phys. Rev. 138, A442 (1965); D. M. Ceperley, G. V. Chester, and M. H. Kalos, Phys. Rev. B 16, 3081 (1977).
2. See for example, M. H. Kalos, Phys. Rev. 128, 1791 (1962), and M. H. Kalos, D. Levesque, and L. Verlet, Phys. Rev. A 9, 2178 (1974).
3. J. B. Anderson, J. Chem. Phys. 63 1499 (1975); 65, 4121 (1976); 73, 3897 (1980); 74, 6307 (1981).
4. P. J. Reynolds, D. M. Ceperley, B. J. Alder, and W. A. Lester, Jr., J. Chem. Phys. 77, 5593 (1982).

TABLE 1.

The ground-state ($^3B_1$) and first-excited state ($^1A_1$) energies of methylene.

Method	$^3B_1$-energy (hartrees)	$^1A_1$-energy (hartrees)
Hartree-Fock	-38.9348	-38.8944
CI-SD	-39.1071	-39.0956
CI-SDQ (est.)	-39.122	-39.105
QMC	-39.128±0.004	-39.108±0.004
Experimental	-39.148	---

$^1A_1 - {}^3B_1$ energy (kcal/mole)	
CI	9.9-11.3
Expt	8.5-19.6
QMC	12.3±3.4

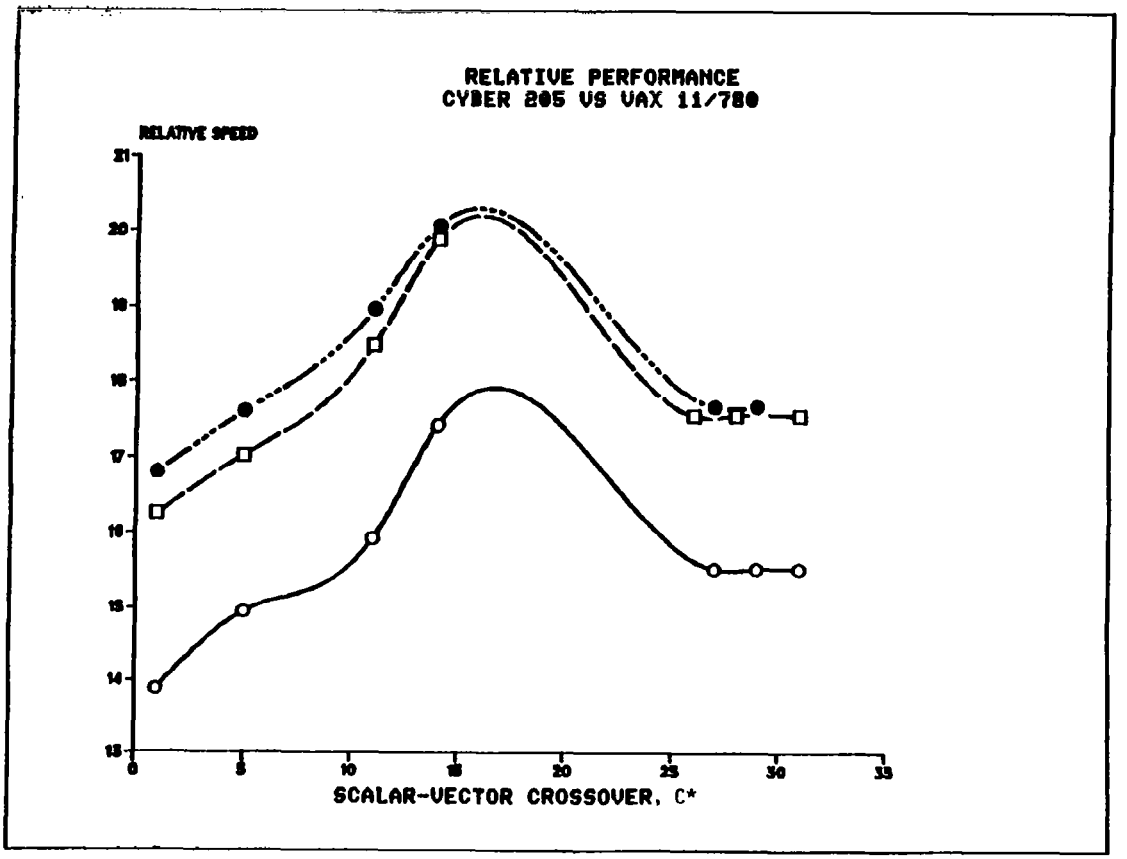

Figure 1. Relative speeds of the Cyber 205 and the VAX 11/780 for quantum Monte Carlo calculations of the ground-state energy of CH_2. The crossover point C* is the vector length below which variable-length loops are run in scaler mode. Thus, for large C* these loops are all run in scaler mode,whereas for very small C*, vector start-up time hinders performance. The three curves correspond to different degrees of hand-optimization of the code. See text for details. Note that the curves interpolating the data points are simply polynomial fits to the data. The actual curve for a particular molecule is a set of steps at the values of the various loop lengths that occur in the problem. The fits can be considered an "average" behavior for this type of calculation.

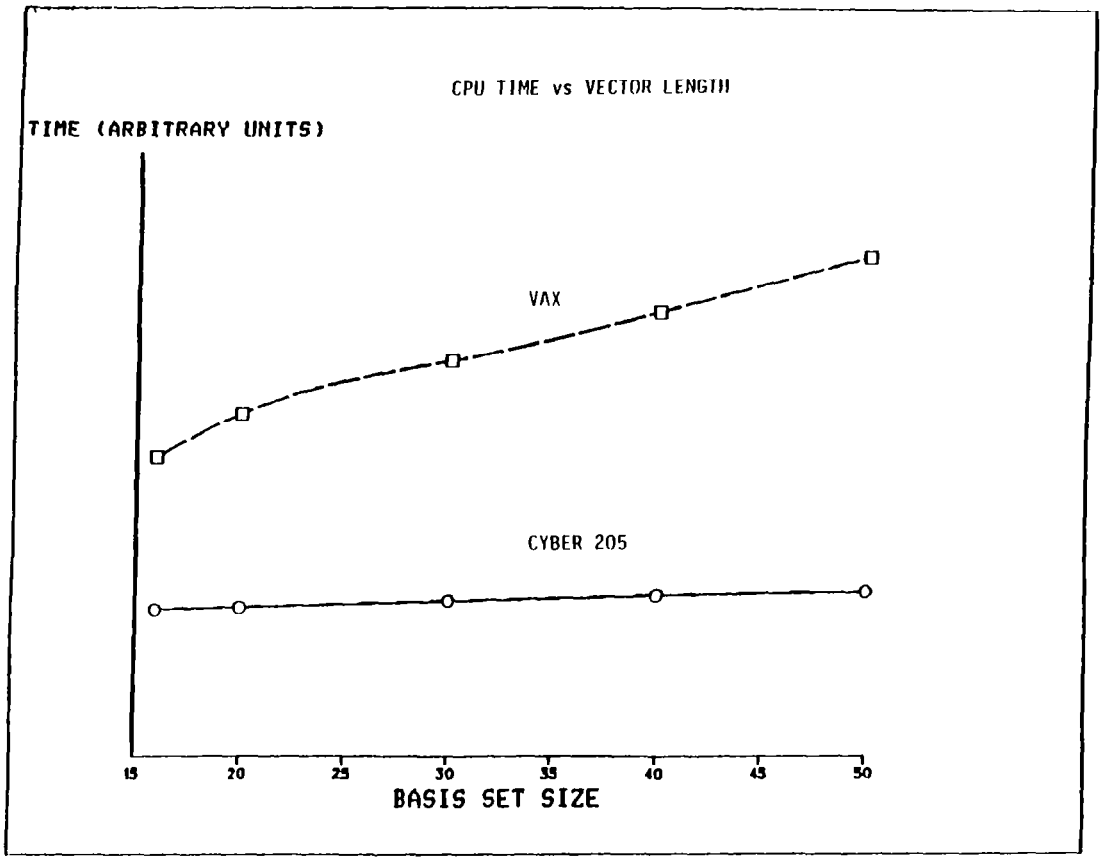

Figure 2. CPU time versus the number of atomic basis set functions, n. Conventional codes scale as n^{λ} with $\lambda \approx 4\text{-}6$ while QMC scales simply as n. Both the VAX and Cyber show this n dependence clearly. However, the slope for the Cyber is almost zero. At n=16 the Cyber is 20 times the speed of the VAX while at n=50 the Cyber is 30 times faster.