Quantum Monte Carlo lab notes

Lucas Wagner and Lubos Mitas

August 8, 2006

1 DMC for Gaussian Wavefunctions

Run the program by typing

```
~train18/mitas/dmc
```

Here we'll walk you through the usage of the simple DMC solver for the harmonic oscillator.

The potential is $\frac{1}{2}x^2$, so the exact energy is 0.5. The exact ground state wave function is proportional to $e^{-\frac{1}{2}x^2}$. By default, this program allows you to modify the exponent of the trial wave function and the timestep of the DMC simulation. No matter what you choose for the trial wave function, in the limit of zero time step, DMC will obtain 0.5 for the energy. The trial function only affects the efficiency of the simulation. This is in constrast to the many-fermion case, where fixed-node DMC will produce different results depending on the nodal surface of the trial function.

We will not give you a tutorial, but more a list of things to try:

- Try entering 0.5 for the trial function parameter and any timestep (try 1.0). The average will be exactly 0.5 with zero error bar and there will be no timestep error.
- For a more realistic calculation, enter 0.6 for the trial function and several different timesteps (for example, 0.1, 0.05, 0.02, and 0.01). Plot the results and note the dependence on timestep and the extrapolation to zero.
- For those interested in the method, look into the code at

  ```
  ~train18/mitas/src/dmc_simple/dmc_simple.cpp
  ```

There are many comments that explain how the method is implemented, and you can try an absolute value potential, the primitive propagator(which does not use a trial wave function), and many other things. If you are more interested in the practical use of a full-fledged code, look at the QWalk examples below, which show how we can calculate the total energy of chemical systems extremely accurately.

2 QMC calculations on methane

Note: for this section, the binaries are located at

```
~train18/mitas
```

and the data files are located at

```
~train18/mitas/ch4_qwalk
```

We're going to walk through the calculation of the fixed-node Diffusion Monte Carlo energy of methane. While this calculation is doable (and was done) on a normal PC, we've gone ahead and done some of them so you don't have to. If you're interested in how long it takes, the wall clock times are at the end of each output file. Keep in mind that the calculations were done on a four year old laptop (Pentium III 1133 MHz).

We start the calculation with a GAMESS input deck (`ch4.inp`) to prepare the one-particle orbitals, which we use to make the Hartree-Fock many particle wave function:

$$\Psi_{HF} = \det\left[\phi_i, \phi_2, \ldots, \phi_n\right]$$

We use pseudopotentials for both carbon and hydrogen here, which speed up the QMC calculation quite a bit (it is possible to do full-potential QMC calculations, but the scaling is approximately $Z^{5.5}$). The punch file and output files are `.pun` and `.out`, respectively.

Next, we use the `gamess2qmc` converter, which creates several files that describe the wave function and system in QWalk format. Those are `ch4.basis` (basis functions), `ch4.orb` (orbital coefficients), `ch4.slater` (determinant information), `ch4.sys` (nuclear positions and pseudopotential), and `ch4.jast2` (Jastrow correlation factor). We will not worry about the format of the files for the moment. First we'll see how they're used.

Open `ch4.hf`. Here we include the `ch4.sys` section, and define the trial function as the Slater determinant. The method is VMC, which generates electron configurations according to the probability distribution Ψ_T^2. The energy (and any other properties) is evaluated as $\langle E\rangle = \int \Psi_T^2 \frac{H\Psi_T}{\Psi_T}$, which is approximately equal to a Monte Carlo average over a finite number of samples of $\left\langle \frac{H\Psi_T}{\Psi_T} \right\rangle$. Run the command

```
gosling ch4.hf.log
```

to get a summary of the results. The total energy should match GAMESS, except with a stochastic uncertainty(the error bar), since we used exactly the same trial function.

Evaluating the Hartree-Fock wave function via Monte Carlo gives nothing but a lot of wasted computer cycles, since GAMESS is much faster. Now we'll start to exploit the advantages that doing the many-body integrals explicitely gives us. We add a so-called Jastrow factor (defined in `ch4.jast2`), so that the new wave function is:

$$\Psi_{VMC} = \det\left[\phi_1, \phi_2, \ldots, \phi_n\right] e^U$$

where

$$U = \sum_{iIk} c_{kI} a_k\left(r_{iI}\right) + \sum_{ijn} d_n b_n\left(r_{ij}\right)$$

(i and j are electron numbers, I is nucleus number, and k and n are indices over a basis set expansion). The b_n functions depend explicitely on the electron-electron distances, which makes the integrals impossible to do analytically while compactly describing electron-electron correlation. The a_k functions are mostly technical (but important!) terms that allow the one-particle density to readjust to the introduction of the correlation, since we only optimize the terms in the Jastrow factor.

Open `ch4.vmc` for the QWalk input file. We attach the Jastrow function to the Slater determinant and run another VMC. This has two goals. The first is to evaluate the energy of the new trial function, which has changed since even our starting Jastrow factor includes the correct behavior of the wave function as two electrons approach each other(known as the Kato cusp conditions). You will see that even that small modification improves the energy. The second, and more important effect of the first VMC is to generate a set of configurations according to Ψ_{VMC}^2 and store them in the file `ch4.config`. The next method, `OPTIMIZE2`, then takes these configurations and optimizes the variational parameters in the Jastrow factor (the c_{kI}s, the d_ns, and the shape of the a_ks and b_ns). It provides an approximation of the energy during the optimization. We then run another VMC to get the true energy of the new trial function. For a summary of 'before and after', run

```
gosling ch4.vmc.log
```

There will be two sections here, which correspond to the before optimization wave function and the after optimization wave function. The large energy decrease you see is from optimizing only 10 parameters, as opposed to the millions of determinants and large basis sets that it would take to obtain such a low energy in Configuration Interaction. More complicated Jastrow factors are also possible, and do improve the results somewhat, although we will not explore them here.

Variational Monte Carlo can be quite powerful when used by itself, but it suffers from variational bias–the choice of the functional form can describe one system better than another. It also uses stochastic optimization, which can give results that can be somewhat unpredictable. These problems can be overcome by carefully choosing the variational parameters and optimizing them, but Diffusion Monte Carlo can do this automatically as well as providing more accurate results. Hopefully Dr. Mitas has explained this to relative satisfaction in the lecture, and you can also look at the simple DMC implementation in the other part of the lab. Thus we move to the input file `ch4.dmc`. It uses the new trial function produced by the VMC procedure, and has an almost identical syntax to VMC, although it is around five times slower. Type

```
gosling ch4.dmc.log
```

to get a summary of the DMC results. It may be instructive to plot the Hartree-Fock, VMC, and DMC energies to get a feel for what the differences are.