united nations
educational, scientific
and cultural
organization

international atomic
energy agency

the
abdus salam
international centre for theoretical physics

ICTP 40th Anniversary

SMR 1595 - 3

Joint DEMOCRITOS - ICTP School on

CONTINUUM QUANTUM MONTE CARLO METHODS

12 - 23 January 2004

COMPUTER LABORATORY SESSION

Introduction to the CASINO code

Mike TOWLER and Neil DRUMMOND
TCM Group, Cavendish Laboratory, Cambridge University
Cambridge CB3 OHE, U.K.

These are preliminary lecture notes, intended only for distribution to participants.

Introduction to the CASINO code: computer laboratory session

Mike Towler and Neil Drummond

Friday 16 January, 2004

Welcome to CASINO!

In this worksheet, we demonstrate how to use the CASINO code by leading the user through some simple examples.

It is hoped that by the end of the session, users will be sufficiently familiar with the basic input variables to carry out the following for both molecules and solids: Hartree-Fock VMC calculations; Jastrow factor optimisations using variance minimisation; and DMC calculations. Users should also know how to perform reblocking analysis of the data and be aware of the `casinohelp` utility.

Activity 1: A simple Hartree-Fock VMC calculation

Let's begin by calculating the Hartree-Fock energy of a hydrogen atom using VMC. This activity should take about 20 minutes.

- Go to the directory ∼/`CASINO/examples/atom/hydrogen` and list its contents. There should be a `gwfn.data` file holding the orbitals generated by a GAUSSIAN94 calculation[1] and a CASINO `input` file. No pseudopotential file is supplied, so CASINO will assume you wish to do an all-electron calculation. No `jasfun.data` is supplied, because there is no need of a Jastrow factor for a one-electron atom! All of the geometry information is contained in the `gwfn.data` file.

  (Note: it would probably be a good idea to create a backup copy of the `hydrogen` directory before you start work in it, so that there won't be a problem if you e.g. accidentally delete a file. The same applies to the directories used in the subsequent activities.)

- Read through the `input` file and observe that:

  - The lines in `input` consist of a parameter name, then a colon, then the parameter value. The order of the lines doesn't matter and many of the parameters are optional.
  - **neu** and **ned** (the numbers of spin-up and spin-down electrons) are 1 and 0 respectively, as one would expect for a hydrogen atom.
  - The **irun** parameter is set to 1, meaning that this will be a VMC calculation with no Jastrow factor.
  - The **btype** parameter is set to 2, which implies that a Gaussian basis set is to be used for the orbitals. (CASINO therefore expects a `gwfn.data` file to exist.)
  - The **inew** parameter is set to 1, indicating that this is going to be a new run.

[1] See the bottom of `gwfn.data` for the GAUSSIAN output.

- We are simulating a finite system, so **isperiodic** is set to F.
- The number of equilibration[2] configuration moves **nequil** is set to 1000.
- The **nmove** parameter (the number of moves whose energies are written out) is set to 50000.
- The "correlation period" **corper**[3] is set to 10, which should eliminate any serial correlation in the energy.
- The timestep[4] **dtvmc** is set to a value (0.57 in this case) which ensures that about 50% of the proposed electron moves are accepted; this roughly maximises the efficiency with which configuration space is sampled[5].

To obtain information on an input parameter, the *casinohelp* utility can be used. For example, to obtain information on the **btype** parameter, type *casinohelp btype*. This will write out information about the different values **btype** can take. Try repeating this for some of the other parameters in `input`.

- Type *runvmc* to run the VMC simulation.
- When it has finished, three files, called `out`, `vmc.hist` and `vmc.posout`, should have been produced. Type *ve*. It should say something like:

```
File out:
Total energy : -0.500007666230 +/- 0.000074699169 Time : 13.8300 seconds
```

(Note that Hartree atomic units are used everywhere in CASINO, unless otherwise stated.)

For comparison, the true Hartree-Fock energy of the hydrogen atom is exactly -0.5 a.u.; so VMC gives the correct answer—with an error bar.

- Next, glance through the `out` file. This contains a complete report of the calculation. The total energy and its components are given at the end. Just before the total energy, we see the "acceptance ratio", which is the fraction of electron moves accepted. This should be about 50%.
- The file `vmc.hist` contains all the energy data produced during the simulation. In order to obtain an accurate estimate of the error bar on the energy, you must analyse the data in `vmc.hist` using the utility *reblock*.
  - Type *reblock* in the directory containing the `vmc.hist` file.
  - It will ask you what units you want. Choose Hartree atomic units per molecule (option 1).

[2] It is not actually necessary to equilibrate the electron distribution in this case because there is only one electron. This is exceptional, however!

[3] The local energy is only evaluated once every **corper** moves because it is inefficient to perform energy evaluations every move.

[4] The VMC single-electron transition probability distribution in CASINO is Gaussian and the VMC timestep is the variance of that Gaussian.

[5] If the timestep is too short then the correspondingly short electron moves will usually be accepted, but configuration space will be sampled slowly; if the timestep is too large then the majority of proposed moves will be rejected, again leading to poor statistical efficiency. If the electron move acceptance ratio is too low then **dtvmc** should be decreased; if it is too high then **dtvmc** should be increased.

- Then it will print the results of a reblocking analysis of the total energy. The error bar (in the "Std err of mean" column) is generally underestimated for small values of the block length, but should rise to a roughly constant value for larger blocks. For the largest block lengths it will oscillate because the error bar on the error estimate becomes significant. The constant value in the middle "plateau" is the required error bar on the QMC energy.
- In this case (and in the case of VMC simulations generally, provided **corper** is greater than about 4), there is almost no serial correlation present in the energy data[6]. So the "plateau" region starts at block length 1. Choose a block length of 32. The energy and energy components, together with their reblocked error bars, are displayed. Take a note of the total energy and its error bar.
- A file called `reblock.plot` is produced by *reblock*. It holds the standard error in the mean energy plotted against the reblocking transformation number (the block length is given by two to the power of the reblocking transformation number). To view it, type *plot_reblock*. It is clear that there is no serial correlation in these data; compare your plot with the graph shown further down in these notes that was obtained using strongly correlated DMC data.

- The file `vmc.posout` contains the final positions of the electrons and the current state of the random number generator so that the VMC run may be continued if desired. For example, if the error bar is still too large after a certain number of moves then you can make it smaller by running the simulation for longer. To do this:

  - Set the `input` keyword **inew** to 0.
  - Rename the `vmc.posout` file to `vmc.posin`.
  - Run the calculation for another **nmove** moves by typing *runvmc*.

  All the extra data will be appended to `vmc.hist`. Type *reblock* and repeat the reblocking process, making your own choice of block length. You have doubled the amount of data, so the error bar on the total energy should be smaller by a factor of $\sqrt{2}$ from what you found before. Is it?

- Finally, type *clearup* to get rid of all the output files. Also, remember to change **inew** back to 1 in the `input` file.

If you have time, try carrying out a VMC simulation for a silane molecule.
(This can be found in `~/CASINO/examples/molecule/silane/ground_state/`.)

Note that a Jastrow factor is present this time, so **irun** is set to 2. Remember to check that the acceptance ratio ("levels 1 and 2") is close to 50% and to perform reblocking analysis afterwards.

Activity 2: A variance minimisation calculation

Let's optimise the Jastrow factor of an homogeneous electron gas of density parameter $r_s = 1$ and 54 electrons per cell. This exercise should take about 30 minutes.

- Go to `~/CASINO/examples/electron_phases/3D_fluid`.
- Copy `~/CASINO/examples/generic/jasfun.data_blank` to this directory (and rename it as `jasfun.data`, thus getting rid of the old `jasfun.data`).

[6] Note that the error bars in the `out` file are calculated assuming there is no serial correlation in the energy data; this should always be checked by reblocking analysis, however.

- Open the `jasfun.data` file using a text editor.
  - The first section (the seven lines starting at "Non-linear rij term") relates to the term in the Jastrow factor that permits the electron-electron cusp condition to be satisfied. The A parameter in the blank file is given as 4.6, and we will use this as our initial guess[7]. Note that optimisation of the A parameter is enabled.
  - The second section relates to the $S_1(r_{ij})$ Jastrow term described in the manual. The parameters arising in this term are listed underneath "Spin parallel and antiparallel components". Because the "Enable spin polarization" flag is set, there are separate parameter sets for spin-parallel and antiparallel electron pairs. Our initial guess for all the parameters other than A is zero: this is the safest choice. Please set "Enable optimization" to T, so that we can optimise these parameters, and reduce the "No of parameters per spin" to 3. You will need to make sure there are 3 rows of parameters by deleting 3 of the 6 lines underneath "Spin parallel and antiparallel components".
  - The third section contains information about atom-centred functions (the $S_2(r_{iI})$ Jastrow term). There are no atoms in the electron gas, so please change the "No of sets of atom-centred functions" to 0 and delete the lines from "SET 1" to "END SET 1" inclusive.

- Now examine the `input` file. Observe that:
  - The **nwrcon** parameter is the number of configurations to be written out[8].
  - **corper** is given the relatively large value of 10 in order to minimise serial correlation effects[9].

- Type *runvarmin -n 2 -v &*. The *-n* flag means "do two iterations of the configuration generation/optimisation cycle"; the *-v* flag means "when you've finished these two iteration, do a final VMC calculation with the optimised Jastrow factor so we can check that it really does improve the wavefunction". When *runvarmin* has finished, a `VARMIN_FINISHED` file will appear.

- The *runvarmin* script creates a directory `io` in which all the input and output files from the different stages of the calculation are placed (with a numerical suffix indicating which iteration of configuration generation/optimisation they are associated with). The most important things to look at are `vmc.out.1`, `vmc.out.2` and `vmc.out.3`. These are the VMC output files with, respectively, the unoptimised Jastrow factor, the optimised Jastrow factor after the first iteration, and the final optimised Jastrow factor (obviously there will be even more of these if you increase the argument of the *-n* flag). If you type *ve vmc.out.** then you can see whether the optimisation has worked at each stage. In this case we see:

```
> ve vmc.out.*
File vmc.out.1:
Total energy : 1.060985722685 +/- 0.007516920925 Time : 200.1400 seconds
File vmc.out.2:
Total energy : 0.600856683188 +/- 0.000593843417 Time : 202.3700 seconds
File vmc.out.3:
Total energy : 0.601205435248 +/- 0.000524099703 Time : 145.3000 seconds
```

[7] A better first guess for A would be the value suggested by the random phase approximation.
[8] We normally set **nwrcon**=**nmove**.
[9] Even larger values of **corper** are often used for variance minimisation in practice.

Optimising the parameters has lowered both the error bar and the total energy significantly. Essentially all of the improvement is in the first iteration.

- Please look at the output of the variance minimisation runs in the files `varmin.out.1` etc. A report of the progress of the actual variance minimisation can be found there, including the value of the variance at each iteration.
- The `jasfun.in` and `jasfun.out` files hold the Jastrow function before and after each variance minimisation run. The final Jastrow function is also placed in the `jasfun.data` file in the directory in which the run took place.

Activity 3: A DMC calculation

A DMC calculation consists of three basic steps:

1. **VMC configuration generation**;
2. **DMC equilibration**;
3. **DMC statistics accumulation**.

Fortunately, the *rundmc* script chains these three stages together, so we can run a DMC simulation with a single command.

Let's calculate the DMC energy of a hydrogen molecule. This exercise should take about 30 minutes.

- Go to ∼`/CASINO/examples/molecule/h2/RHF/dmc` and look in the `input` file.
  - An equilibrated VMC run is set to go for **nmove**$= 250$ moves in order to produce **nwrcon**$= 250$ configurations.
  - For DMC equilibration, we will run **nblock_dmc_equil**$= 100$ blocks of **nmove_dmc_equil**$= 10$ moves.
  - For statistics accumulation we will run **nblock_dmc_stats**$= 125$ blocks of **nmove_dmc_stats**$= 200$ moves.
  - The **nconfig** parameter (the target configuration population) is set to 250: the same as **nwrcon**[10].
  - The **dtdmc** parameter is the DMC timestep (set to 0.004); note that it is much smaller than the VMC timestep because the DMC Green's function is only exact in the limit that the timestep goes to zero[11].
  - **irun** is set to 2 because the DMC process starts off with a VMC run.
- Type *rundmc &* to start the run. When it has finished, the file `DMC_FINISHED` will appear in the directory.

[10] The only reason **nconfig** and **nwrcon** would ever differ is on a parallel machine; configuration generation is very fast, and can be done quickly on far fewer processors than the DMC calculation. Running configuration generation on batch queues with small numbers of processors thus usually reduces time spent waiting in queues; one should increase **nwrcon** in proportion to the reduction in the number of processors to maintain the same total configuration number.

[11] For accurate work, you may need to consider extrapolating to zero timestep.

- Every now and again, while CASINO is running, type *graphit* (you need to be in the directory where you started *rundmc*). Two plots should appear. The top graph shows the configuration population plotted against move number; the bottom one shows the average local energy (black), the reference energy E_T (red) and the "best estimate" of the ground state energy (green) plotted against move number. The energy falls off almost immediately at the start of the equilibration phase. The simulation has "settled down" by the start of the statistics accumulation phase[12]. It should always be ensured that the fall-off has finished by the end of the equilibration phase.

- The *rundmc* script creates an `io` directory in which the input and output files from each stage of the calculation are placed. Please have a look at these, but note that the `config.in` files are unformatted, so there is no point in trying to look at them. Observe that the *rundmc* script has automatically changed the **irun** parameter in the `input` file to drive the different phases of the calculation.

- The DMC may take a while to run. However, the output files for a DMC calculation of $C_{29}H_{36}$ can be found in the `~/CASINO/examples/C29H36_example` directory.

- Please go to this directory and type *reblock* to analyse the data in the `dmc.hist` and `dmc.hist2` files[13]. *Reblock* will read these files and then starting asking you questions:

  - It will ask for your choice of units. Choose atomic units.
  - It will tell you the number of moves you carried out in the "DMC equilibration" run and then ask you which line of `dmc.hist` and `dmc.hist2` you would like to regard as the start of statistics accumulation. Normally one chooses the next move after the end of the equilibration data to be the starting point, although you are free to choose a different number if, for example, you think it hasn't quite settled down by the start of the "statistics accumulation" run. Choose line 1251 to be the starting point.
  - It will ask how many lines you want to average over. Enter -1 to select all the remaining lines.
  - *Reblock* will show you a reblocking analysis, as it did for the VMC calculation, and ask you to choose a block length. The error bar appears to be showing a plateau for block lengths of more than 256, so choose 4096 which is in the middle of this plateau.
  - *Reblock* will then print out the final DMC energies and error bars, together with an analysis of the population fluctuations, effective time steps, and acceptance ratios. In the case of $C_{29}H_{36}$, the final energy is $-186.0959(15)$ a.u..
  - Again, a `reblock.plot` file is produced. Type *plot_reblock* to view it: it should be the same as that shown below. The variation of the standard error with block length and the existence of the plateau are much more pronounced with the DMC data than was the case for VMC.

[12] Note that a different algorithm is used for updating the "best estimate" of the energy in the equilibration and statistics accumulation phases. During equilibration we only average over data from the last two blocks whereas during statistics accumulation we average over all data since the start of that phase.

[13] *reblock* also reads `input` to determine some input parameters it needs to know about; it will ask you about these if it can't find `input`.

- Return to ∼/CASINO/examples/molecule/h2/RHF/dmc.
- When the run has finished, try carrying out a reblocking analysis. The run isn't really long enough, so the plateau is only just starting to appear. What final energy and error bar do you obtain?
- Type *clearup* when you are finished.

Activity 4: A solid state calculation

Let's apply what we have learned and carry out a solid state calculation. This exercise should take about 40 minutes.

- Go to directory ∼/CASINO/examples/crystal/pw/silicon/222G/ and list its contents. Note the presence of the silicon pseudopotential file (si_pp.data) and the files required for the use of the "modified periodic Coulomb" interaction (density.data and eepot.data).
- Look in the input file.
  - **btype** is equal 1, meaning that we are using a plane-wave basis.
  - **isperiodic** is T because we are simulating a periodic system.
  - The input block shown below is supplied, indicating that the simulation cell consists of $2 \times 2 \times 2$ primitive cells. (This corresponds to the 8 k points used in the DFT calculation.)

    ```
    %block npcell
    2 2 2
    %endblock npcell
    ```
- First of all, you should carry out a Hartree-Fock VMC calculation to check you can reproduce the kinetic energy from the DFT calculation that generated the pwfn.data file (3.247806 a.u.). This is an important check, and should always be carried out when a new wavefunction file is generated.

  - Try to set up the VMC calculation by yourself. Remember to make sure the Jastrow factor *isn't* used at this stage. You should also check that the (second-level) move acceptance ratio is about 50%. It's a good idea to use more than 1000 equilibration steps and you will probably need to carry out a few hundred energy-calculation moves. However don't make **nequil** and **nmove** too large, or you will have to wait a long time for the run to finish!
  - When the run has ended, carry out a reblocking analysis. Check that the kinetic energy estimator KEI is in statistical agreement with the DFT kinetic energy. If you want smaller error bars then you will have to run for longer.

- Now try optimising a Jastrow factor. Open the `jasfun.data` file in a text editor.
  - Make sure that optimisation of A is enabled.
  - Choose the number S_1 parameters per spin-pair. (Optimising more than about 3 parameters per spin-pair may be difficult because of time constraints). Decide whether you want separate parameter sets for parallel and antiparallel spin-pairs, then replace the parameters below "Spin parallel and antiparallel components" with the appropriate number of parameters, all equal to zero. Make sure that optimisation is enabled.
  - This time we have a set of "$S_2(r_{iI})$" atom-centred functions in the Jastrow factor. In general we have a different set of these parameters for each species of atom present; so, in this case we just have one set. These functions are centred on each of the 16 silicon atoms in the simulation cell. Choose a number of $S_2(r_{iI})$ parameters; less than 3 is probably advisable just now. This number should be entered in the first column of the line underneath "No of parameters in ri,rirj,ririj(2),rirjrij(2) terms". It should be followed by five zeroes. Make sure that spin-polarisation is disabled. Then edit the parameters themselves underneath the line "Components". Enter the appropriate number of lines, with each parameter set to zero.
  - Now try to set up the input file for a variance minimisation calculation. Because of time constraints, it is only feasible to generate about 200 configurations. Remember to adjust **irun** and **nwrcon** and to increase **corper** (again, time constraints mean that **corper** will have to be about 6). It should be emphasised that these parameters would take larger values under normal circumstances.
  - Run the variance minimisation calculation.
  - When it is finished, go to the `io` directory and check that the energy has fallen over the course of the variance minimisation.

- The optimised Jastrow factor could now be used in a DMC simulation. Carrying out a reasonable DMC simulation is not likely to be feasible in the alloted time, but you could set one up anyway if you have time.

Activity 5: Generating localised orbitals

Let's now look at the procedure for carrying out a calculation with localised orbitals using the silane molecule (SiH_4) as an example. The required input files have been set up appropriately in the directory `~/CASINO/examples/molecule/splines/silane_localized/`. This exercise should take about 40 minutes.

- **Carrying out the transformation from Bloch to Wannier orbitals**. The first stage requires the `pwfn.data` (in this case generated using the output of the GP plane-wave DFT code) and

input_wannier[14] files. Type *xwannier* to run the wannier conversion utility. The output consists of a new plane-wave file pwfn.data.wannier and a file wannier_centers.dat that contains a list of the Wannier function centres.

- Type *mv pwfn.data pwfn.data.bloch* and then type *mv pwfn.data.wannier pwfn.data*. From now on, we will work with localised orbitals.

- **Generation of a spline representation of the orbitals**. We could run CASINO using the localised orbitals represented in the plane wave basis; however, it is faster and more memory efficient to represent the localised orbitals numerically. The next stage requires a pwfn.data file (and corresponding wannier_centers.dat file if this exists) and a generate.dat[15] file. Type *generate_spline* to run the utility for producing the splined orbitals. The output of this code is a swfn.data file[16].

- **Carrying out a QMC calculation.** Now, finally, we can carry out QMC calculations using the localised orbitals. Type *runvmc* to carry out a HF VMC calculation. Note the following in the input file:

  - **btype**$= 6$ for the splined orbitals;
  - We have set isperiodic to false because we are simulating a molecule. Now that we have generated truncated localised orbitals, we can dispense with periodicity;
  - We have provided lists of the occupied states for spin-up and spin-down electrons (the statelist_up and statelist_down blocks).

- Let's attempt to visualise the localised orbitals.

  - Add the following lines to the bottom of the input file:

```
%block lineplot
1
4
1 2 3 4
1 1 1 1
500
-10.0 0.0 0.0
 10.0 0.0 0.0
%endblock lineplot
```

[14] The only parameter the user is ever likely to want to change in input_wannier is the number of states included in the Wannier transformation (all the states in the pwfn.data file from 1 up to this number are transformed; any remaining states are unchanged). In the present case, we are including all the states in the transformation.

[15] For this example, we shall use localised grids for all the splined states; if, however, some states were non-localised then it would be advisable to represent those states over the whole simulation box. The "multiplication factor" in the generate.dat file controls the fineness of the spline grid. Generally, this needs to be set to at least about 2 before the kinetic energy of the splined orbitals is the same as that of the orbitals in the plane-wave basis to a high degree of accuracy. The orbitals have a spherical truncation surface such that 99.7% of the square of the orbital norm lies inside this radius. A further shell of width at least 0.05 a.u. is saved from truncation: the orbital can be brought smoothly to zero over this shell region.

[16] The swfn.data file is unformatted. If you wish to look at its contents then use the format_spline utility to generate a formatted swfn.data.formatted file. This is also useful if you want to transfer a swfn.data file from one platform to another, since the unformatted file can usually only be read by the machine that generated it. Run format_spline in the presence of swfn.data.formatted file to produce the corresponding unformatted file.

This means we will be plotting orbitals from $(-10, 0, 0)$ to $(10, 0, 0)$, where the silicon atom is located at the origin. Note that these coordinates are expressed in Ångstroms.

- Type *runvmc*.
- Type *xmgrace lineplot.dat*.

Type *casinohelp lineplot* to get information on **lineplot** block, and try plotting in different directions.

- Take a note of the total and kinetic energies (use *reblock* and the CPU time (in `out`) obtained with the localised orbital. Then *clearup* the directory and move `pwfn.data.bloch` back to `pwfn.data`. Now try the following:

  - Carry out a Hartree-Fock VMC calculation using the Bloch orbitals expanded in the plane-wave basis. (Note that you will need to change **btype** and **isperiodic**, and you will have to remove the **lineplot** block.) Compare the energies and CPU timings with the results obtained using Wannier orbitals. Are the results what you expect?
  - Plot some of the orbitals and compare them with the localised ones.

Conclusions

We have seen how the CASINO program is used to carry out its three main functions, and we have carried out calculations for molecules and solids using orbitals expanded in a variety of bases.

For more detailed information, please use the *casinohelp* facility or refer to the manual. In addition, there are many other small systems to be experimented with in the `~/CASINO/examples` directory.

We wish you the best of luck with CASINO!