Variational and Fixed Node Diffusion Monte Carlo simulation of the electron gas in 2D

Use the code `qmc.f` to reproduce some of the results of Ref. [1][1]

The electron gas is a fundamental model in condensed matter physics, and one of the most successful applications of QMC. The basic trial function is of the Slater-Jastrow (SJ) form, with explicit two-body correlations in the Jastrow factor and plane-wave orbitals in the Slater determinant. We follow Ref. [1] [Y. Kwon, D.M. Ceperley, and R.M. Marin, Phys. Rev. B **48**, 12037 (1993)] to investigate the effects of three-body correlations in the Jastrow factor and backflow correlations in the determinant on ground-state properties of the two-dimensional electron gas. We use energy units of Rydberg and length units of $r_s a_0$, with a_0 the Bohr radius.

Files:

- `doc.pdf`: this file
- `2degas.pdf`: Ref. [1]
- `setup.f`: a setup fortran code
- `qmc.f`: the QMC fortran code
- `qmc.h`: a file included by `qmc.f` at compile time
- `statfor.f`: fortran code for statistical analysis (see Appendix C)
- `statforw.f`: fortran code for weighted averages and statistical errors (see Appendix D)

Step by step:

Compile all four codes `setup.f`, `qmc.f`, `statfor.f` and `statforw.f` with the gfortran compiler (use the option -w to suppress warning messages and the option -O3 to get a faster executable). If you get an error message such as "Rank mismatch in argument [...]" try the option "-Wno-argument-mismatch".

1. Run the setup code: when prompted, enter a "run id", the r_s parameter, the number of up- and down-spin electrons, a flag to turn on/off the two-body term, a flag to turn on/off the three-body term, and a flag to turn on/off the backflow correlations. The run id is a string of your choice which identifies the name of the subsequent QMC run. In the following, we assume that this string is `xxx`, and that you have chosen $r_s = 5$, 13 up- and 13 down-spin electrons (for the unpolarized electron gas, choose equal numbers for up- and down-spin electrons).[2] We also assume that

[1] In Ref. [1] the Diffusion Monte Carlo is called Green's Function Monte Carlo.

[2] Since we are going to use periodic boundary conditions, you should choose closed shells, i.e. 5, 9, 13, 21, 27, ...

you have chosen flags to turn the two-body term on, the three-body term off and the backflow term off: in this case, we have the SJ parameter-free trial function of Ref. [1], with long-range two-body correlations and plane-waves in the up- and down-spin determinants.

The setup will create a number of files:

- `xxx.sy` is a list of items `keyword value [value [...]]` which describe the system. A list of keywords is given in the Appendix A
- `xxx.up.x`: initial coordinates of the up electrons
- `xxx.down.x`: initial coordinates of the down electrons
- `xxx.v2`: table for the e-e pair potential
- `xxx.u2`: table for the e-e Jastrow factor
- `xxx.k`: a list of $\mathbf{k}$ points[3].

2. Create an input file `xxx.in` with the line

```
vmc 50 500 1.0
```

This input line tells the QMC code to do a VMC simulation with 50 blocks of 500 steps each with time step 1.0. A list of keywords for the input file is given in the Appendix B.

3. Run a VMC simulation using the QMC code, possibly redirecting the standard output to `xxx.vmc`. [4] The code asks for the run id: you can either type the run id as prompted, or write the run id on a file and redirect the standard input. [5] The code will open all the needed `xxx.`*whatever* files to perform the assigned task(s). With the input file specified above it prints on the standard output [4] the 50 block averages (first column), block weight[6] (second column), cumulative average(third column), and estimated statistical errors (fourth column) for some quantities, as shown below:

[3] The $\mathbf{k}$ points are used both for the plane waves in the determinants and for the Fourier components of the long-range part of the Coulomb potential and of the two-body correlations, and the two-body Jastrow factor has no adjustable parameters. Note that the `xxx.k` contains only one entry for each pair $(\mathbf{k}, -\mathbf{k})$. The QMC code will use orbitals $\sin(\mathbf{k}\cdot\mathbf{r})$ and $\cos(\mathbf{k}\cdot\mathbf{r})$ instead of $\exp(-i\mathbf{k}\cdot\mathbf{r})$ and $\exp(i\mathbf{k}\cdot\mathbf{r})$

[4] You can redirect the standard output of an executable `./executable` to file `outfile` either with

```
./executable > outfile
```

or with

```
./executable | tee outfile
```

The latter command will send the output of the code both to the standard output and on file `outfile`.

[5] You can redirect the standard input as `./executable < infile`

[6] In VMC the weight is just the number of steps per block, but in DMC it does fluctuate.

[.....]

```
===>> vmc block       50
-0.29322385191E+00  0.500E+03  -0.29382181229E+00  0.18E-03  elocal
 0.47212706540E+00  0.500E+03   0.49744701041E+00  0.23E-02  acc.rate
-0.37407353673E+00  0.500E+03  -0.37655495354E+00  0.36E-03  epot
 0.80849684816E-01  0.500E+03   0.82733141252E-01  0.42E-03  ekin
 0.86197686480E-01  0.500E+03   0.86505741271E-01  0.11E-03  e2
```

Most of the quantities shown have an obvious name. The quantity `e2` is the average of the square of the local energy, and it can be used to calculate the variance of the local energy.

You may plot the block averages of the local energy using

```
gnuplot> plot '< grep elocal xxx.out' u 0:1 with lines
```

and similarly for other quantities. The statistical errors calculated in the QMC code assume independent blocks. If you want to check for possible correlations of, say, the local energy, use the `statfor.f` code (see Appendix C):

```
grep elocal xxx.out | ./statfor.x
```

The energy should be in agreement with the entry $E_{\rm V}^{\rm SJ}$ of Table III of Ref. [1] (-0.2935(1) for $N = 26$ and $r_s = 5$). If you want a smaller statistical error, modify the `xxx.in` file as follows:

```
restart
vmc 100 500 1.0
```

and run the QMC code again, possibly appending the standard output to `xxx.out`. [7]

With the `restart` keyword, the code starts where the previous run left, and it adds more blocks as specified (i.e. from 51 to 100, in this example).

4. Introduce and optimize three-body and backflow correlations. Run the setup, choose a different run id `yyy` and this time turn the three-body and backflow options on. The file `yyy.sy` will now contain lines with the keywords `u3` and `backflow`; the setup code will also create files `yyy.u3` and `yyy.b` with unoptimized but not unreasonable three-body and backflow

[7] You can append the standard output of an executable `./executable` to file `outfile` either with

```
./executable >> outfile
```

or with

```
./executable | tee -a outfile
```

The latter command will send the output of the code both to the standard output and on file `outfile`.

functions (cfr. Eqs. 9, 11, 12, 13, 15 of Ref. [1]; equation (13) makes sure that the trial function and its first derivatives are continuous at $L/2$, where L is the side of the simulation cell). At difference with Ref. [1] we use gaussians for both the trhee-body and the backflow correlations $\xi(r)$ and $\eta(r)$.

Create an input file `yyy.in` containing the lines:

```
vmc 20 250 1.0 10
optimize 500 yyy.u3 yyy.b
```

The last number in the `vmc` line tells the code to store a configuration every 10 step. The line `optimize` tells the code to use 500 stored configurations to optimize the functions contained in files `yyy.u3` and `yyy.b` (i.e. the three-body and backflow terms). The optimization is performed through correlated sampling minimizing a linear combination of energy and variance.

Run the QMC code, redirecting the standard output to (say) `yyy.opt`. After the 20 VMC blocks, the standard output shows the progress of the optimization, as follows:

```
                        [ ..... ]
f2, effpt = 1.1909049280117050E-003 465.02044057808058
f2, effpt = 1.1824127664439050E-003 457.68583199729284
f2, effpt = 1.1685014305766307E-003 443.62015803473531
f2, effpt = 1.1503879060493270E-003 420.82726910609472
f2, effpt = 1.1340924358576520E-003 400.58115712883392
f2, effpt = 1.1340924358576520E-003 400.58115712883392
lmdif2 exited, info = 1
final variational parms:
0.33695E+00 0.11690E+01 0.47609E+00 0.11354E+01
```

`f2` is the quantity that gets minimized, and `effpt` is the effective number of points in the correlated-sampling weighted averages. `info` is an exit code: a value of 1 (or 2, or 3) signals convergence to a (possibly local) minimum, a value of -1 signals that `effpt` has dropped below 1/2 of the number of configurations used. The files `yyy.u3` and `yyy.b` are overwritten with the three-body and backflow functions calculated with the new variational parameters; the original functions are saved in files `yyy.u3.old yyy.b.old`, which can be retrieved if the optimization step fails.

Rerun (appending the standard output to `yyy.opt`) until nothing[8] changes,

[8] Particularly the VMC energy.

within statistical uncertainties.

Now get rid of the `optimize` line in the input file, and run a (possibly longer) VMC simulation, redirecting the output to a new file, such as `yyy.vmc`.

The gain in variational energy that you have achieved over the SJ result by including three-body and backflow correlations should match the difference between the SJ+3BD+BF and the SJ entries for $r_s = 5$ of Table II of Ref. [1].[9].

5. Run a FN-DMC simulation. Modify the file `yyy.in` as follows:

```
dmc 10 50 0.1 20 -0.297
```

This tells the code to do a DMC run with 10 blocks, 50 steps per block, time step 0.1, 20 walkers[10] and -0.297 as initial trial energy (taken from previous VMC runs). Run the QMC code and redirect, for instance, to `yyy.eq` (eq for "equilibration"). The output looks like this:

[.....]

```
===>> dmc block       10
-0.29735178135E+00  0.245E+04  -0.29888072858E+00  0.26E-03  elocal
 0.96844578578E+00  0.245E+04   0.97199683932E+00  0.77E-03  acc.rate
-0.37347312700E+00  0.245E+04  -0.37386471366E+00  0.60E-03  epot
 0.76121345656E-01  0.245E+04   0.74983985076E-01  0.49E-03  ekin
 0.88496588944E-01  0.245E+04   0.89405495549E-01  0.15E-03  e2
max.multiplicity 20 exceeded 0 times
end
end of run
```

Add the `restart` keyword on the first line of `yyy.in`, increase the number of blocks and rerun, appending the output to `yyy.eq`. Keep increasing the number of blocks until the energy is stationary (equilibration). This can be monitored with a plot of the block averages (see e.g. item 3 above).

Once the equilibration is done, get rid of the `restart` keyword in the input file and run a new DMC simulation to calculate the energy, redirecting the output to a new file, say, yyy.dmc. In DMC both the equilibration time and the autocorrelation time are longer than in VMC, because of

[9] This table has been calculated with $N = 58$. However both the three-body and the backflow terms have effects mainly at short range, therefore the gain should be similar with $N = 26$.

[10] Make sure you have at least 20 configurations in the files `yyy.up.x` and `yyy.down.x`. If not, run a short VMC run to store a sufficient number of configurations.

the smaller time step (typically $\sim$ 10 times smaller). The statistical errors shown in the output assume uncorrelated block averages. This should be checked by performing a statistical analysis of the data. You can use `statforwf` (see Appendix D):

```
grep elocal yyy.dmc | ./statforw.x
```

If you want a smaller statistical error, add the keyword `restart` in the input file, increase the number of blocks and rerun the QMC code, appending the output to `yyy.dmc`.

You may want to try different time steps and/or number of walkers[11],as the FN-DMC solution is attained in the limit of zero time step and infinite number of walkers.

[11] If you increase the number of walkers you need more initial configurations; run a VMC to store them and perform a new equilibration.

Appendix A: Keywords for the file `xxx.sy` used in this calculation

ndim `d`

`d` (integer): dimensions of the physical space

type `name number hbs2m file`

define a type of particles

`name` (string): name of this type of particles

`number` (integer): number of particles of this type

`hbs2m` (real): value of $\hbar^2/2m$ for this type of particles

`file` (string): file with initial configuration(s) of the particles

v2 `name_a name_b file` iexp

pair potential between particles of type `name_a` and `name_b`

`name_a` (string): name of a type of particles

`name_b` (string): name of a (possibly different) type of particles

`file` (string): name of the file with the tabulated potential

`iexp` (integer): the short-range part of the tabulated potential is multiplied by a factor $r^{-\mathrm{iexp}}$

u2 `name_a name_b file`

two-body Jastrow factor for particles of type `name_a` and `name_b`

`name_a` (string): name of a type of particles

`name_b` (string): name of a (possibly different) type of particles

`file` (string): name of the file with the tabulated Jastrow factor

v0 `value`

`value` (real): a constant term in the potential [12]

pbc `L_x [L_y [...]]`

periodic boundary conditions

`L_x` (real): side of the simulation cell in the x direction

`[...]` (real): side of the simulation cell in the other direction(s)

kspace `file`

[12] This term comes from the splitting of the Coulomb potential into a short-range and a long-range part. See V. Natoli and D.M. Ceperley, J. Comput. Physics 117, 171 (1995).

`file` (string): name of a file with a list of **k** points [13]

u3 `name_a name_b file`

three-body Jastrow factor for particles of type `name_a` and `name_b`

`name_a` (string): name of a type of particles

`name_b` (string): name of a (possibly different) type of particles

`file` (string): name of the file with the tabulated three-body function

backflow `name_a name_b file`

backflow correlations for particles of type `name_a` and `name_b`

`name_a` (string): name of a type of particles

`name_b` (string): name of a (possibly different) type of particles

`file` (string): name of the file with the tabulated backflow function

plane-wave `name`

determinant of plane waves for particles of type `name`

`name` (string): name of a type of particles

rhok `name`

turn on the calculation of the density fluctuation $\rho_{\mathbf{k}}$ [14] for particles of type `name`.

[13] Needed for the orbitals in the plane-wave or backflow determinants, and/or for the `rhok` keyword

[14] The density fluctuations $\rho_{\mathbf{k}} = \sum_j \exp(-i\mathbf{k} \cdot \mathbf{r}_j)$ are needed to calculate the long-range part of the potential and of the pair Jastrow factor in reciprocal space

Appendix B: Keywords for the file `xxx.in` used in this calculation

vmc `nblocks nstep tstep [istore]`

define a VMC run

`nblocks` (integer): number of blocks

`nsteps` (integer): number of steps per block

`tstep` (real): time step (size of the move)

`istore` (integer, optional): store a configuration every `istore` step

optimize `nconf file_1` [`file_2` [...]]

define an optimization run

`nconf` (integer): number of configurations for correlated sampling

`file_1` (string): name of a file with a function to be optimized [15]

`[...]` more files with other functions to be optimized

dmc `nblocks nstep tstep nwalkers etrial`

define a DMC run

`nblocks` (integer): number of blocks

`nsteps` (integer): number of steps per block

`tstep` (real): time step

`nwalkers` (integer): number of walkers

`tstep` (real): initial value of the trial energy

restart

restart a previous run; this keyword must be followed by the same `vmc` or `dmc` keyword of the previous run, with a larger `nblocks` and the same `nsteps` and `nwalkers`

[15] This file contains (i) number of grid points and grid step; (ii) a tabulated function with three derivatives for a spline; (iii) the name of the routine used to calculate the function; (iv) the number of variational parameters in it; (v) a list with the value of each parameter and a flag 0 or 1 which toggles the optimization of that parameter. See the files `xxx.u3`, `xxx.b`.

Appendix C: Statistical analysis using the code `statfor.f`

compile the code, e.g. `gfortran statfor.f -o statfor.x`

the code reads a stream of numbers from standard input

writes number of data, mean, variance, correlation time, number of independent data and statistical error on standard output

also writes: a histogram of data on file 'histo.out', and the autocorrelation of the data on file 'corr.out'

to analyze (say) the block averages of elocal from file `qmc.out`, use the command:

```
grep elocal qmc.out | ./statfor.x
```

Appendix D: Statistical analysis using the code `statforw.f`

compile the code, e.g. `gfortran statforw.f -o statforw.x`

the code reads two streams of numbers (data and weights) from standard input

writes number of effective data in the weighted average ,mean, variance, correlation time, number of independent data and statistical error on standard output

also writes: a histogram of data on file 'histo.out', and the autocorrelation of the weighted data on file 'corr.out'

to analyze (say) the block averages of elocal from file `qmc.out`, use the command:

```
grep elocal qmc.out | ./statforw.x
```