Basic VMC/DMC code for educational purposes

Jindřich Kolorenč, May 2018

1 Configuration file

There is a separate executable for each of the physical systems to be simulated, paraHe_run, othoHe_run, H2_run (sec. 2). Each of these programs performs a VMC simulation followed by a DMC simulation. It is controlled by a configuration file qmc_run.cfg. It contains name–value pairs (the order of these pairs does not matter). The number of walkers and the number of Monte Carlo steps are specified as powers of two. The number of walkers is given as

```
NW          10
```

which means that there will be $2^{\texttt{NW}} = 2^{10} = 1024$ walkers (strictly in VMC and on average in DMC).

1.1 VMC

The parameters of VMC calculation are specified as

```
VMCtherm   11
VMCsteps   15
VMCtstep    0.1
```

which defines the number of thermalization/equilibration steps as $2^{\texttt{VMCtherm}}$ and the number of step when the energy is measured as $2^{\texttt{VMCsteps}}$. The "time step" VMCtstep has a different meaning for the simple Metropolis moves [1] and for the diffusion-drift moves. The Metropolis moves are performed as

$$\mathcal{R} = \mathcal{R}' + \xi(\texttt{VMCtstep} - 0.5) \tag{1}$$

where ξ is a uniform random number on the interval $[0, 1)$. In the case of the diffusion-drift moves the meaning is the same as the time step in DMC. The value of VMCtstep should to be chosen such that the acceptance ratio of the proposal moves is not too close to either 1 or 0.

1.2 DMC

The parameters of the DMC simulation are specified analogously,

```
DMCtherm   12
DMCsteps   15
DMCtstep    0.01
DMCruns     1
```

Setting the number of DMC runs, DMCruns, larger than 1 is intended to perform the time-step extrapolation to zero. The code runs sequential DMC simulations with decreasing time steps

$$\texttt{DMCtstep}, \frac{\texttt{DMCruns} - 1}{\texttt{DMCruns}}\texttt{DMCtstep}, \frac{\texttt{DMCruns} - 2}{\texttt{DMCruns}}\texttt{DMCtstep}, \dots, \frac{1}{\texttt{DMCruns}}\texttt{DMCtstep}\,. \tag{2}$$

The data for the extrapolation are written to tstep_extrap.dat.

1.3 System-specific data

Finally, the parameters of the simulated physical system need to be specified inside the system{ } group. Details are described in the next section.

2 Available systems

One can play with three predefined two-electron systems... or one can define his/her own system by implementing an appropriate qmc_input module, sec. 4.5.

2.1 paraHe_run: helium atom (spin singlet)

The singlet state of helium atom (or helium-like two-electron ions) is simulated with the trial wave function having a Hylleraas form [2] (English translation in [3])

$$\Psi_{\mathrm{T}}(\mathbf{r}_1,\mathbf{r}_2) = \left(1+\frac{1}{2}r_{12}\right)\left[1+a\left(r_1^2+r_2^2\right)\right]\mathrm{e}^{-Z(r_1+r_2)}\,. \tag{3}$$

It has a single variational parameter a and fulfills the electron-ion and electron-electron cusp conditions for any a. The `system{ }` group in the configuration file `qmc_run.cfg` reads as

```
system { Znuc 2.0  a 0.0 }
```

This example says that the nuclear charge is Z = `Znuc` = 2.0 (mandatory parameter) and the variational parameter in the Hylleraas ansatz is a = `a` = 0.0. If the parameter a is not given, the code takes the optimal value that minimizes the total energy.

The total energy E corresponding to Ψ_{T} from eq. (3) can be evaluated analytically [2, 4]. One arrives at

$$E = \frac{Z}{4}\,\frac{A}{B} \tag{4}$$

where

$$\begin{aligned} A &= -3a^2\Big\{4Z\big[Z(2176Z+6903)+4992\big]-10065\Big\} \\ &\quad -8aZ^2\Big\{32Z\big[8Z(3Z+7)+27\big]-567\Big\} \\ &\quad -8Z^4\Big\{4Z\big[Z(16Z+25)+4\big]-35\Big\}\,, \qquad (5)\\ B &= 9a^2\big[Z(896Z+3355)+3680\big] \\ &\quad +24aZ^2\big[Z(64Z+189)+168\big]+8Z^4\big[Z(16Z+35)+24\big]\,. \qquad (6) \end{aligned}$$

The derivative of the total energy with respect to a is $\partial E/\partial a = 8Z^3C/B^2$ where

$$\begin{aligned} C &= 9a^2\Bigg\{Z\Bigg[Z\Big[Z\big[4096Z(8Z+47)+413413\big]+304160\Big]-43776\Bigg]-98910\Bigg\} \\ &\quad +12aZ^2\Bigg\{2Z\Bigg[Z\Big[Z\big[8Z(256Z+1301)+16329\big]-1848\Big]-18912\Bigg]-18105\Bigg\} \\ &\quad -8Z^4\Bigg\{Z\Big[Z\big[Z(16Z+737)+2856\big]+3168\Big]+1008\Bigg\}\,. \end{aligned} \tag{7}$$

To find the optimal a that minimizes the total energy we need to solve $\partial E/\partial a = 0$ for a, which is straightforward since it is just a quadratic equation (only the final formula is too long to be explicitly written here).

2.2 orthoHe_run: helium atom (spin triplet)

The triplet state of the helium atom (or a helium-like two-electron ion) has an antisymmetric wave function and one can use it to illustrate the fixed-node constraint. The wave function is approximated by a Slater determinant constructed from 1s and 2s orbitals,

$$\Psi_{\rm T}(\mathbf{r}_1, \mathbf{r}_2) = \psi_{1\rm s}(r_1)\psi_{2\rm s}(r_2) - \psi_{1\rm s}(r_2)\psi_{2\rm s}(r_1)\,, \tag{8}$$

where

$$\psi_{1\rm s}(r) = \mathrm{e}^{-Z_1 r} \quad \text{and} \quad \psi_{2\rm s}(r) = \left(1 - \frac{Z_2 r}{2}\right)\mathrm{e}^{-Z_2 r/2}\,. \tag{9}$$

The system is so simple that the nodal hypersurface is entirely determined by symmetry and it is specified by a condition $r_1 = r_2$ [5, 6]. Note that the complete nodal surface is much larger than the obvious node defined by electron coincidence $\mathbf{r}_1 = \mathbf{r}_2$! Our trial function has the exact nodes even though it is certainly not the exact ground state.

In the system{ } group in the configuration file qmc_run.cfg one can specify three parameters

```
system {
  Znuc 2.0
  Z1   1.99364
  Z2   1.55094
}
```

Only the nuclear charge Znuc is mandatory, the charges in the wave function are optional. If they are not given, Z1 = Z2 = Znuc is used. For VMC one can treat Z_1 and Z_2 as variational parameters (the optimal values are shown in the example above) but for DMC it is better to satisfy the electron-nucleus cusp conditions by keeping Z_1 and Z_2 equal to the charge of the nucleus. For satisfying also the electron-electron cusp conditions we would need to include a correlation factor.

2.3 H2_run: hydrogen molecule (spin singlet)

The hydrogen molecule is simulated with the Heitler–London trial wave function [7] (English translation in [3])

$$\Psi_{\rm T}(\mathbf{r}_1, \mathbf{r}_2) = \mathrm{e}^{-|\mathbf{r}_1 - \mathbf{r}_A| - |\mathbf{r}_2 - \mathbf{r}_B|} + \mathrm{e}^{-|\mathbf{r}_2 - \mathbf{r}_A| - |\mathbf{r}_1 - \mathbf{r}_B|}\,, \tag{10}$$

where $\mathbf{r}_A$ is the position of one nucleus and $\mathbf{r}_B$ is the position of the other nucleus. This ansatz does not fulfill the electron-electron cusp conditions and it has only part of the electron-nucleus cusps.

The only parameter that can be (and must be) specified in the system{ } group is the distance between the nuclei $R = |\mathbf{r}_A - \mathbf{r}_B|$,

```
system { R  1.4 }
```

One can map out the bonding curve by running separate simulations for different values R.

3 Output

Apart from the results written to STDOUT, the program writes several files to the disk.

3.1 Evmc_trace.dat, Edmc_trace.dat

The average of the total energy across the walker population at each step of the VMC or DMC simulation. The thermalization period (VMC) and the initial-projection period (DMC) is not saved.

3.2 NWdmc_trace.dat

The size of the walker population at each step of the DMC simulation (the initial projection period is not saved).

3.3 Evmc_reblock.dat, Edmc_reblock.dat

Estimation of the error bar for the total energy in VMC and DMC simulations by the blocking method – two neighboring blocks are merged into one and the number of blocks is thus halved at each step. The file shows the error bar as a function of the number of remaining blocks.

3.4 E2vmc_reblock.dat, E2dmc_reblock.dat

Estimation of the error bar of the variance

$$\frac{\langle\Psi_{\mathrm{T}}|\hat{H}^2|\Psi_{\mathrm{T}}\rangle - \langle\Psi_{\mathrm{T}}|\hat{H}|\Psi_{\mathrm{T}}\rangle^2}{\langle\Psi_{\mathrm{T}}|\Psi_{\mathrm{T}}\rangle} \tag{11}$$

which measures the quality of the trial wave function.

3.5 tstep_extrap.dat

DMC energy as a function of the time step (if DMCruns > 1).

4 Source code

The source code (Fortran 95 + TR15581) is split to several files. There is a fair amount of comments in the code so it should be possible to figure out how the program works.

4.1 qmc_run.F90

The top-level driver that calls the VMC and DMC algorithms on the given physical system.

4.2 qmc.F90

Implementation of the VMC and DMC methods following [8]. The subroutines vmc_run() and dmc_run() are multithreaded.

4.3 `reblocking.f90`

Implementation of the blocking method [9].

4.4 `rnd.f90`

Pseudorandom number generator [10].

4.5 `paraHe_input.f90`, `orthoHe_input.f90`, `H2_input.f90`

Calculation of the wave function $\Psi_\mathrm{T}(\mathcal{R})$, the local energy $E_\mathrm{L} = \left[\hat{H}\,\Psi_\mathrm{T}(\mathcal{R})\right]/\,\Psi_\mathrm{T}(\mathcal{R})$ and the drift velocity $v_\mathrm{D} = \nabla \ln |\Psi_\mathrm{T}(\mathcal{R})|$. Each of these files implements an instance of `qmc_input` module.

4.6 `types_const.f90`

Constants and utility routines.

References

[1] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, *Equation of state calculations by fast computing machines*, J. Chem. Phys. **21**, 1087 (1953).

[2] E. Hylleraas, *Neue Berechnung der Energie des Heliums in Grundzustande, sowie des tiefsten Terms von Ortho-Helium*, Z. Phys. **54**, 347 (1929).

[3] H. Hettema, *Quantum Chemistry: Classic Scientific Papers* (World Scientific, 2000).

[4] C. Schwartz, *Experiment and theory in computations of He atom ground state*, Int. J. Mod. Phys. E **15**, 877 (2006).

[5] D. J. Klein and H. M. Pickett, *Nodal hypersurfaces and Anderson's random-walk simulation of the Schrödinger equation*, J. Chem. Phys. **64**, 4811 (1976).

[6] D. Bressanini and P. J. Reynolds, *Unexpected symmetry in the nodal structure of the he atom*, Phys. Rev. Lett. **95**, 110201 (2005).

[7] W. Heitler and F. London, *Wechselwirkung neutraler Atome und homopolare Bindung nach der Quantenmechanik*, Z. Phys. **44**, 455 (1927).

[8] P. J. Reynolds, D. M. Ceperley, B. J. Alder, and W. A. Lester, Jr., *Fixed-node quantum Monte Carlo for molecules*, J. Chem. Phys. **77**, 5593 (1982).

[9] H. Flyvbjerg and H. G. Petersen, *Error estimates on averages of correlated data*, J. Chem. Phys. **91**, 461 (1989).

[10] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in Fortran 90* (Cambridge University Press, 1996).