

TCCM THEORETICAL METHODS FOR SIMULATION OF MATERIALS - LTTC WINTER
SCHOOL HOMEWORK
PURE DIFFUSION QUANTUM MONTE CARLO METHOD

Professor : Anthony Scemama
Student : Matías Emanuel di Mauro Esteban
University : Univesity of Trieste

February 19, 2023

Index

| | | |
|-----|--|---|
| 1 | INTRODUCITON ... | 3 |
| 2 | PURE DIFFUSION MONTE CARLO ... | 3 |
| 3 | SYSTEMS UNDER STUDY ... | 4 |
| 3.1 | THE TRIAL WAVE FUNCTION ... | 4 |
| 3.2 | THE LOCAL ENERGY ... | 5 |
| 3.3 | THE DRIFT VECTOR ... | 5 |
| 4 | THE PDWMC CODE ... | 6 |
| 4.1 | THE FORTRAN CODE ... | 6 |
| 4.2 | THE INPUT FILE AND LAUNCHING THE CALCULATION ... | 6 |
| 4.3 | THE OUTPUT ... | 6 |

1 INTRODUCTION

This PDF aims to give a short explanation on the Pure Diffusion Quantum Montecarlo code written to solve the assignment of the LTTC Winter School 'From Theory to Implementation : Tutorials in Theoretical Chemistry'. It begins with a recall of the PDQMC algorithm. Afterwards it gives a few words on what it was implemented to compute the energy of the studied systems together with some instructions on how to run the code.

2 PURE DIFFUSION QUANTUM MONTE CARLO

The **Pure Diffusion Quantum Monte Carlo** is a Quantum Chemistry stochastic method that allows to study complex systems. It attempts to provide a solution to the many-body problem coming close to the exact energy with chemical accuracy. To do so the Time-Dependent Schrödinger Equation is solved in imaginary time $t \rightarrow -i\tau$.

$$-\frac{\partial \psi(\mathbf{r}, \tau)}{\partial \tau} = (\hat{H} - E_{REF}) \psi(\mathbf{r}, \tau) \quad (1)$$

$$= \left(-\frac{1}{2}\Delta + V(\mathbf{r}) - E_{REF} \right) \psi(\mathbf{r}, \tau) \quad (2)$$

so that $\psi(\mathbf{r}, \tau) = \Psi(\mathbf{r}, t)|_{t=-i\tau}$, where $\Psi(\mathbf{r}, t)$ is the wave function describing the system. E_{REF} introduces a shift in the energy and helps with the convergence of the algorithm.

At sufficiently large imaginary times, the imaginary-time-dependent wave function $\psi(\mathbf{r}, \tau)$ collapses to the ground state $\Phi_o(\mathbf{r})$. The system is propagated in time sampling the configurational space of all the spatial degrees of freedom $\{\mathbf{r}\}$ by generating *brownian paths*. These paths $\{\mathbf{r}^1, \mathbf{r}^2, \dots, \mathbf{r}^{NMC}\}$ simulate the *diffusion part* of eq. (2), connected to the kinetic energy component $\frac{1}{2}\Delta$, while the *branching term* associated with the potential is considered as a multiplicative weight along the trajectory $\prod_i \exp[-(V(\mathbf{r}_i) - E_{REF})\tau]$.

The trajectories are not generated blindly. The sampling is enhanced with the help of a trial wave function Ψ_T so that the actual propagation equation is solved for $\Pi(\mathbf{r}, \tau) = \Psi_T(\mathbf{r}) \cdot \psi(\mathbf{r}, \tau)$. The brownian paths are guided by a **drift vector** $\mathbf{b}(\mathbf{r}) = \frac{\nabla \Psi_T}{\Psi_T}$ that pushes the sampling 'walker' towards the high probability configurations. Moreover, it is not the full potential what is evaluated along the trajectory but instead, it is the local energy $E_L = \frac{\hat{H} \Psi_T}{\Psi_T}$, which is a much smoother function. This scheme allows to compute the energy of the system by sampling the local energy throughout the trajectory and taking its average by weighing its values at each configuration $\{\mathbf{r}^k\}$ with a proper weight:

$$W(\mathbf{r}^k) = \prod_{k=1}^n \exp[-(E_L(\mathbf{r}^k) - E_{REF})\delta\tau] \quad (3)$$

$\{\mathbf{r}^k\}$ represents a collective variable indicating the position of all the electrons of the system at a given configuration k along the path. $\delta\tau$ is the *time-step* variable.

The PDQMC algorithm follows:

1. Start with $W(\mathbf{r}^o) = 1$ and $\tau_o = 0$.
2. Evaluate the local energy at \mathbf{r}^k , $E_L(\mathbf{r}^k)$.
3. Compute the contribution to the weight $w(\mathbf{r}^k) = \exp[-(E_L(\mathbf{r}^k) - E_{REF})\delta\tau]$.
4. Update and accumulate the weight $W(\mathbf{r}^k) = W(\mathbf{r}^{k-1}) \times w(\mathbf{r}^k)$.
5. Accumulate the weighted energy $W(\mathbf{r}^k) \times E_L(\mathbf{r}^k)$, and the weight $W(\mathbf{r}^k)$ for normalization.
6. Update the imaginary time $\tau_k = \tau_{k-1} + \delta\tau$.
7. If $\tau_k > \tau_{max}$ (τ_{max} given as input parameter), the long projection time has been reached and a new trajectory is started from the current position setting $W(\mathbf{r}^k) = 0$ and $\tau_k = 0$.
8. Update the configuration through the stochastic equation $\mathbf{r}'_i = \mathbf{r}^k_i + \delta\tau \mathbf{b}_i(\mathbf{r}^k) + \sqrt{\delta\tau} \chi$ for all the electrons $i = 1, \dots, N$ of the system. The variable χ is a three-dimensional vector with random components generated following a Gaussian distribution.
9. Evaluate the trial wave function $\Psi_T(\mathbf{r}')$ and the drift $\mathbf{b}(\mathbf{r}') = \frac{\nabla \Psi_T(\mathbf{r}')}{\Psi_T(\mathbf{r}')}$ at the new configuration.
10. Metropolis Algorithm :

- (a) Compute the ratio $A = \frac{T(\mathbf{r}' \rightarrow \mathbf{r}^k)P(\mathbf{r}')}{T(\mathbf{r}^k \rightarrow \mathbf{r}')P(\mathbf{r}^k)}$ where $P(\mathbf{r}) = |\Psi_T(\mathbf{r})|^2$ and

$$T(\mathbf{r}^k \rightarrow \mathbf{r}') = \frac{1}{(2\pi\delta\tau)^{3/2}} \exp \left[-\frac{[\mathbf{r}' - \mathbf{r}^k - \delta\tau \mathbf{b}(\mathbf{r}^k)]^2}{2\delta\tau} \right] \text{ is the transition probability to go from } \mathbf{r}^k \text{ configuration to configuration } \mathbf{r}'.^1$$

- (b) Draw a uniform random number $v \in [0, 1]$.
(c) If $v \leq A$, then the move is accepted and \mathbf{r}' becomes the new configuration \mathbf{r}^{k+1} .
(d) Otherwise the move is rejected and the next step is again computed from \mathbf{r}^k .

Finally, the energy of the system is estimated as:

$$E = \frac{\sum_{k=1}^{NMC} E_L(\mathbf{r}^k) W(\mathbf{r}^k, k\delta\tau)}{\sum_{k=1}^{NMC} W(\mathbf{r}^k, k\delta\tau)} \quad (4)$$

where NMC is the total number of configurations that compose a trajectory [1].

3 SYSTEMS UNDER STUDY

The PDQMC code works to compute the ground state energy of systems like the ones in the following list in their singlet state.

- The Hydrogen atom ($NEL = 1$, $NUC = 1$).
- The Helium atom ($NEL = 2$, $NUC = 1$).
- The H_2^+ ion at a given internuclei distance R_{AB} ($NEL = 1$, $NUC = 2$).
- The H_2 molecule at a given internuclei distance R_{AB} ($NEL = 2$, $NUC = 2$).
- The H_3^+ ion at a given nuclei configuration $\{\mathbf{R}\}$ ($NEL = 2$, $NUC = 3$).

The variables NEL and NUC control the number of electrons and nuclei. A general electronic Hamiltonian describing these systems can be written in the following form [atomic units]:

$$H = \underbrace{-\frac{1}{2} \sum_{i=1}^{NEL} \Delta_i}_T + \underbrace{\frac{1}{2} \sum_{i=1}^{NEL} \sum_{j \neq i}^{NEL} \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|}}_{V_{ee}} - \underbrace{\sum_{i=1}^{NEL} \sum_{A=1}^{NUC} \frac{Z_A}{|\mathbf{r}_i - \mathbf{R}_A|}}_{V_{eN}} + \underbrace{\frac{1}{2} \sum_{A=1}^{NUC} \sum_{B \neq A}^{NUC} \frac{1}{|\mathbf{R}_A - \mathbf{R}_B|}}_{V_{NN}} \quad (5)$$

3.1 THE TRIAL WAVE FUNCTION

In order to sample correctly the configurational space of the electronic degrees of freedom a trial wave function $\Psi_T(\mathbf{r})$ has to be proposed. Considering that we are dealing with maximum two electrons in a singlet state, the antisymmetry of the total wave function for these fermionic systems is guaranteed by the spin component $|\sigma\rangle = \frac{1}{\sqrt{2}}(|\uparrow\downarrow\rangle - |\downarrow\uparrow\rangle)$. The latter leave us with a symmetric spatial wave function that, in principle, behaves as a 'bosonic' system.

In this context the symmetric spatial wave function reduces to the product of two Molecular Orbitals (MO's) $\varphi(\mathbf{r})$:

$$\Psi_T(\mathbf{r}_1, \mathbf{r}_2) = \varphi_a(\mathbf{r}_1) \times \varphi_a(\mathbf{r}_2) \quad (6)$$

The MO's are single-particle wave functions that host one electron each. In examples of the list these orbitals are doubly occupied at most. The subindex a indicate that they correspond to the orbitals of lowest energy. In the simplest case of a *Minimal Basis Set* in which each nuclei $\{\mathbf{R}_A\}_{A=1}^{NUC}$ contributes with one Atomic Orbital (AO) $\phi_A(\mathbf{r}) = \phi(\mathbf{r} - \mathbf{R}_A)$ centered at the nucleus position \mathbf{R}_A , the MO's can be expressed as a linear combination of AO's:

$$\varphi_a(\mathbf{r}) = \sum_{A=1}^{NUC} c_{Aa} \phi_A(\mathbf{r}) = c_{Aa} \phi_A(\mathbf{r}) + c_{Ba} \phi_B(\mathbf{r}) + c_{Ca} \phi_C(\mathbf{r}) + \dots \quad (7)$$

so that $\{\phi_A\}_{A=1}^{NUC}$ is the Basis Set we are working with. As a consequence, the trial wave function assumes the following form:

$$\Psi_T(\mathbf{r}_1, \mathbf{r}_2) = \sum_{A=1}^{NUC} c_{Aa} \phi_A(\mathbf{r}_1) \times \sum_{B=1}^{NUC} c_{Ba} \phi_B(\mathbf{r}_2) \quad (8)$$

¹When dealing with a multi-electronic system the transition probability at a given configuration $\{\mathbf{r}^k\}$ can be computed as the product of the individual transition probabilities of each one of the electrons. This is because the random component of the stochastic propagation equation makes their movement loosely correlated and independent. $T(\mathbf{r}^k \rightarrow \mathbf{r}') = \prod_{i=1}^{NEL} T(\mathbf{r}_i^k \rightarrow \mathbf{r}'_i)$.

For this problem the proposed AO's are 1s hydrogen-like wave function centered at some nuclei position:

$$\phi_A(\mathbf{r}) = Ne^{-a|\mathbf{r}-\mathbf{R}_A|} \quad (9)$$

where N is a normalization constant and a is a parameter related to the nuclear charge Z_A . The set of coefficients $\{c_{Aa}\}$ are set all to 1.

It is worth mentioning that since we are dealing with symmetric wave functions free of nodes, the systems under study do not suffer from the *Fixed Node Error*. The only source of errors come from the discretization of time, the *time-step error* $\delta\tau$.

The following subsections show the analytical form that the local energy and the drift vector assume for this type of trial wave function.

3.2 THE LOCAL ENERGY

The local energy E_L is defined as

$$E_L = \frac{\hat{H}\Psi_T}{\Psi_T} \quad (10)$$

For a trial wave function like the one in eq. (8), the local energy looks like

$$E_L = -\frac{1}{2} \sum_{i=1}^{NEL} \frac{\Delta_i \Psi_T}{\Psi_T} + \frac{1}{2} \sum_{i=1}^{NEL} \sum_{j \neq i}^{NEL} \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|} - \sum_{i=1}^{NEL} \sum_{A=1}^{NUC} \frac{Z_A}{|\mathbf{r}_i - \mathbf{R}_A|} + \frac{1}{2} \sum_{A=1}^{NUC} \sum_{B \neq A}^{NUC} \frac{1}{|\mathbf{R}_A - \mathbf{R}_B|} \quad (11)$$

considering that the potentials are local, multiplicative operators. The kinetic energy term can be reduced thanks to the linearity of the Laplacian operator (in the two-electron case):

$$-\frac{1}{2} \sum_{i=1}^2 \frac{\Delta_i \Psi_T}{\Psi_T} = -\frac{1}{2} \frac{\Delta_1 \varphi_a(\mathbf{r}_1) \varphi_a(\mathbf{r}_2)}{\varphi_a(\mathbf{r}_1) \varphi_a(\mathbf{r}_2)} - \frac{1}{2} \frac{\Delta_2 \varphi_a(\mathbf{r}_1) \varphi_a(\mathbf{r}_2)}{\varphi_a(\mathbf{r}_1) \varphi_a(\mathbf{r}_2)} = -\frac{1}{2} \frac{\Delta_1 \varphi_a(\mathbf{r}_1)}{\varphi_a(\mathbf{r}_1)} - \frac{1}{2} \frac{\Delta_2 \varphi_a(\mathbf{r}_2)}{\varphi_a(\mathbf{r}_2)} \quad (12)$$

As it can be seen, it results in the sum of the kintetic energies of each electron occupying one of the MO's. Moreover, the Laplacian of a MO is just the sum of the Laplacians of each AO appearing in the linear combination:

$$\Delta \varphi(\mathbf{r}) = \Delta \sum_{A=1}^{NUC} c_{Aa} \phi_A(\mathbf{r}) = \sum_{A=1}^{NUC} c_{Aa} \Delta \phi_A(\mathbf{r}) = c_{Aa} \Delta \phi_A(\mathbf{r}) + c_{Ba} \Delta \phi_B(\mathbf{r}) + c_{Ca} \Delta \phi_C(\mathbf{r}) + \dots \quad (13)$$

Finally, for AO's like the one proposed in eq. (9), the Laplacian can be written as:

$$\Delta \phi(\mathbf{r} - \mathbf{R}) = \Delta N e^{-a|\mathbf{r}-\mathbf{R}|} = \left(a^2 - \frac{2a}{|\mathbf{r} - \mathbf{R}|} \right) \phi(\mathbf{r} - \mathbf{R}) \quad (14)$$

3.3 THE DRIFT VECTOR

The drift vector of the electron i is defined as

$$\mathbf{b}_i(\mathbf{r}) = \frac{\nabla_i \Psi_T(\mathbf{r})}{\Psi_T(\mathbf{r})} \quad (15)$$

In the case of two electrons and a wave function given by eq. (8), the drift assumes the following form (taking electron **1** as reference):

$$\mathbf{b}_1(\mathbf{r}) = \frac{\nabla_1 \varphi_a(\mathbf{r}_1) \varphi_a(\mathbf{r}_2)}{\varphi_a(\mathbf{r}_1) \varphi_a(\mathbf{r}_2)} = \frac{\nabla_1 \varphi_a(\mathbf{r}_1)}{\varphi_a(\mathbf{r}_1)} \quad (16)$$

The same holds for electron **2**. Once again, thanks to the linearity of the Gradient operator, the Gradient of the MO's can be reduced to the vector sum of the Gradients of the corresponding AO's appearing in its linear combination.

$$\nabla \varphi_a(\mathbf{r}) = \nabla \sum_{A=1}^{NUC} c_{Aa} \phi_A(\mathbf{r}) = \sum_{A=1}^{NUC} c_{Aa} \nabla \phi_A(\mathbf{r}) = c_{Aa} \nabla \phi_A(\mathbf{r}) + c_{Ba} \nabla \phi_B(\mathbf{r}) + c_{Ca} \nabla \phi_C(\mathbf{r}) + \dots \quad (17)$$

The Gradient of a 1s function is given by the following expression:

$$\nabla \phi(\mathbf{r} - \mathbf{R}) = \nabla N e^{-a|\mathbf{r}-\mathbf{R}|} = -a \frac{\mathbf{r} - \mathbf{R}}{|\mathbf{r} - \mathbf{R}|} \phi(\mathbf{r} - \mathbf{R}) = -a \hat{\mathbf{u}} \phi(\mathbf{r} - \mathbf{R}) \quad (18)$$

where $\hat{\mathbf{u}} = \frac{\mathbf{r} - \mathbf{R}}{|\mathbf{r} - \mathbf{R}|}$ is a unit vector pointing in the $\mathbf{r} - \mathbf{R}$ direction.

4 THE PDQMC CODE

The code is provided in a **zip** archive with all the required files together with a directory called **examples**. This folder contains the input and output of some example calculations carried out with the systems described above.

4.1 THE FORTRAN CODE

The code is written in *Fortran90*. It implements the PDQMC scheme in a series of ***.F90** files containing all the required functions and subroutines to compute the energies of the example systems.

- **qmc.F90** : file containing the main calling programme. It runs a series of N_{runs} independent QMC calculations, and performs the associated statistical analysis.
- **pdmc.F90** : file containing the main PDMC subroutine that generates a path of NMC steps over which it samples the local energy appropriately.
- **utilities.F90** : file containing a collection of functions and subroutines to print outputs, calculate random numbers with Gaussian distribution (Box-Müller algorithm), etc.
- **WFC.F90** : file containing the function that calculates Ψ_T .
- **MOs.F90** : file containing the functions that calculate the MO's, their Laplacian and Gradients.
- **AOs.F90** : file containing the functions that calculate the AO's, their Laplacian and Gradients.
- **energies.F90** : file containing the functions and subroutines that calculate the local energy E_L .
- **drift.F90** : file containing the subroutine that computes the drift vector for a given electron.

The programme can be compiled with *gfortran* using the **Make File** typing the command `$ make` on the terminal.

4.2 THE INPUT FILE AND LAUNCHING THE CALCULATION

The input file is organized in a series of *name lists* containing all the parameters required by the programme. A space **must** be left in between lists.

```
& SIMULATION PARAMETERS
100                      !Number of total runs [Nruns].
100000                  !Number of total Monte Carlo iterations NMC.
0.05                   !Time step [dt].
100.0                  !Maximum projection time [tau_max].

& ELECTRONS
1                      !Number electrons in the system [NEL].

& NUCLEI [Z a Rx Ry Rz]
1                      !Number of Nuclei [NUC].
1 1.2 0.50 0.50 0.50   !Nuclear charge [Z] | Exponent of AO [a] | Nuclear coordinates [Angs.]

& ENERGY
-0.5                   !Reference energy [E_ref].
```

Figure 1: Example Input File.

The calculation can be launched by typing `$./PDMC.x < input.in > output.out` on the terminal.

4.3 THE OUTPUT

At the end of the calculation the programme gives the output file printing the input parameters, together with the final energy in atomic units (with and without the V_{NN} term), the acceptance rate A , and their corresponding statistics.

```

*****
|          PDMC CALCULATION          |
*****

-----SIMULATION PARAMETERS-----

Number of runs      :      100
Number of MC iterations :    100000
Time step           :     0.05000
Maximum projection time : 100.00000

-----ELECTRONS-----

Number of electrons :      1

-----NUCLEI-----

Number of nuclei    :      1

Z | a | Rx      Ry      Rz [a.u.]
  |---|
1 | 1.20 | 0.945  0.945  0.945

-----ENERGIES-----

Reference energy     :   -0.50000
*****

*****
|          PDMC RESULTS          |
*****

-----ELECTRONIC ENERGY-----

E =  -0.500 +/-   0.001 [Hartree]

-----ELECTRONIC ENERGY + VNN-----

E =  -0.500 +/-   0.001 [Hartree]

-----ACCEPTANCE RATE-----

A =   0.990 +/-   0.000

```

Figure 2: Example Output File.

An additional file called **energy.out** is generated. This file contains the evolution of the total energy and its components along the path, averaged over all the *Nruns* independent QMC calculations. An energy plot can be generated by typing *\$ make plot* on the terminal.

Finally, the command *\$ make clean* can be used to remove all the outputs.

References

- [1] 'QMC-LTTC-2023' Lecture Notes.