Density-Functional-Monte-Carlo for calculating properties of nanosystems

K. Putteneers[1,2], F. Brosens[1]

[1] TQC, Universiteit Antwerpen, Universiteitsplein 1, 2610 Wilrijk, Belgium
[2] katrijn.putteneers@ua.ac.be

Universiteit Antwerpen

Theory of Quantum and Complex systems

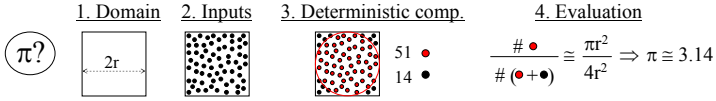
Introduction

We propose a transparent and easy-to-implement method for calculating the charge and potential distribution of nanosystems. The method consists of a Monte-Carlo simulation which forms a direct implementation of the Hohenberg-Kohn (H-K) theorems[1]. This "Density-Functional-Monte-Carlo method" is explained and applied to a nanoshell.

Monte-Carlo simulation

"Iteratively evaluating a deterministic model using sets of random numbers as inputs."

Mostly the calculations follow the next pattern:
1. Define a domain of possible inputs.
2. Generate inputs randomly from a probability distribution over the domain.
3. Perform a deterministic computation on the inputs.
4. Evaluate the results of the deterministic computation.



Hohenberg-Kohn (H-K) theorems

1. The energy E of a system is a unique functional of the density n(**r**) (if the system is non-degenerate)

$$E = E[n(\mathbf{r})]$$

2. If the number of particles N is conserved, this unique functional reaches its minimum at the correct ground state density $n_{GS}(\mathbf{r})$

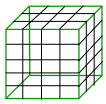
$$N \text{ constant} \Rightarrow E_{min} = E[n_{GS}(\mathbf{r})]$$

⟶ basis of Density Functional Theory (DFT)

Method

A Monte-Carlo simulation is performed in which the domain is a mesh in a nanosystem, the random inputs are "walkers" on the mesh and moves of the walkers to neighbouring mesh points, the deterministic computation is the calculation of the energy functional corresponding to a certain density and the evaluation is the comparison between the energy functionals corresponding to different electron densities. This is a way to implement the H-K theorems.

1. Domain



a mesh in a nanosystem

2. Inputs

a. initial density distribution of 'walkers' on the mesh
b. iterative random moves of walkers to a neighbouring mesh point

3. Deterministic computation

calculation of the energy functional

$$E[n(\mathbf{r})] = T[n(\mathbf{r})] + E_H[n(\mathbf{r})] + E_{XC}[n(\mathbf{r})]$$

4. Evaluation

$E^{[i]}[n(\mathbf{r})] < E^{[i-1]}[n(\mathbf{r})]$
$\Longleftrightarrow$ change $n_w^{[i]}(\mathbf{r})$ in next iteration

$E^{[i]}[n(\mathbf{r})] > E^{[i-1]}[n(\mathbf{r})]$
$\Longleftrightarrow$ change $n_w^{[i-1]}(\mathbf{r})$ in next iteration

$E^{[i]}[n(\mathbf{r})] = E^{[i-1]}[n(\mathbf{r})]$
$\Longleftrightarrow$ $n_{GS}(\mathbf{r})$ is found - stop

because

N_w is constant $\Rightarrow$ N_e is constant

$\Downarrow$

$E_{min} = E[n_{GS}(\mathbf{r})]$

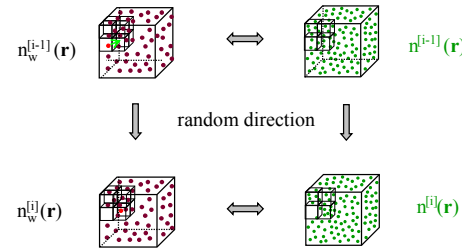
2a. walker density ↔ electron density

N_w walkers ⟷ N_e electrons

1 walker ⟷ $\frac{N_e}{N_w}$ electrons

$n_w = \frac{N_V \text{ walkers}}{\text{volume}}$ ⟷ $\frac{N_V \frac{N_e}{N_w} \text{ electrons}}{\text{volume}} = n$

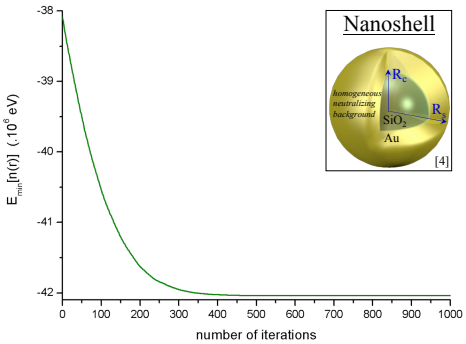
2b. moving a walker ↔ changing electron density distribution



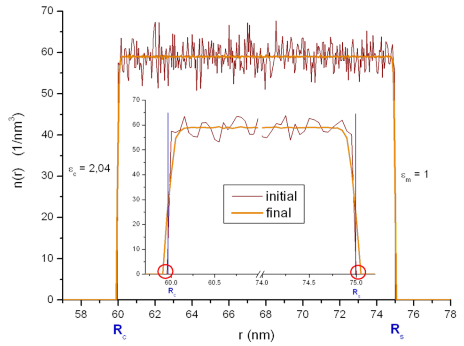
Example

Nanoshells[2] are spherical symmetric nanosize particles with alternating dielectric and metallic concentric layers. They have applications in biomedicine, opto-electronics, high-pressure measurements[3], … The Density-Functional-Monte-Carlo method is used to calculate the density and potential distribution of a nanoshell with a dielectric core and a metallic shell.

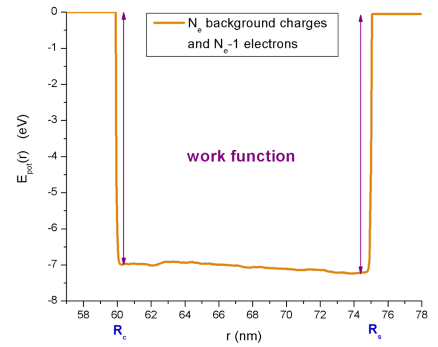
Energy functional



Electron density



Potential energy/Work function



Conclusion

The Density-Functional-Monte-Carlo method is presented as a transparant and easy-to-implement method. Applying it to a nanoshell shows that it can be used to calculate the density and potential distribution of a nanosystem.

References: [1] Hohenberg P., Kohn W., Inhomogeneous electron gas, PR, vol. 136, nr. 3B, p. B864-B871 (1964); [2] Halas N., Optical properties of nanoshells, Optics & Photonics news, p. 26-30 (August 2002); [3] poster of Nick Van den Broeck; [4] picture of nanoshell made by Nick Van den Broeck with Blender

Acknowledgement: Research funded by a Ph.D grant of the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen)