Proceedings of the 45th International School and Conference on the Physics of Semiconductors "Jaszowiec" 2016, Szczyrk

Exact Exchange Scheme in the Parallel *r*-Space Implementation of the Kohn–Sham Realization of the Density Functional Theory

M. Marchwiany[a,b], M. Birowska[a] and J.A. Majewski[a,*]

[a]Faculty of Physics, University of Warsaw, L. Pasteura 5, 02-093 Warszawa, Poland

[b]Interdisciplinary Centre for Mathematical and Computational Modelling (ICM), University of Warsaw, A. Pawińskiego 5a, 02-106 Warsaw, Poland

In this communication, we present the r-space implementation of the Kohn–Sham realization of the density functional theory with the exact exchange functional within the computational algorithm for computers of parallel architecture. In comparison to the standard approach employing the local density functional, the scheme with exact exchange functional requires roughly ten times larger computational burden. The developed parallelization procedure accelerates the computations by a factor of four and six for the exact exchange and the local density functional schemes, respectively. It brings us closer to the treatment of dispersive van der Waals interactions on the fully *ab initio* level in the large class of systems.

DOI: 10.12693/APhysPolA.130.1236

PACS/topics: 31.15.E–, 31.15.–p, 61.50.Ah, 61.50.Lt, 71.15.Mb

1. Introduction

The Kohn–Sham realization of the density functional theory (KS–DFT) [1] is nowadays the most common and widely used computational method for calculation of cohesive and electronic properties of the whole plethora of systems in physics, chemistry, and materials science. There are available numerous computer packages employing the standard approximations to the DFT energy functionals, mostly implementing the expansion of the electronic states in the Kohn–Sham (K–S) equations into a basis. Plane waves are employed in the codes such as VASP [2] or Quantum Espresso [3], whereas localized atomic-like functions (orbitals) are used in the SIESTA package [4]. All of these methodologies for solving K–S equations have their drawbacks. Solution of the K–S equations on the real space grid constitutes an alternative way to determine K–S wave-functions and cures some of the drawbacks in the basis based methods. For example, it allows for implementation of periodic boundary conditions only in the directions of the physical periodicity of the studied systems and eliminates the need of employing the so-called supercell geometry for systems lacking periodicity in some directions. This is of particular importance for zero, one, and two-dimensional systems, which plays extremely important role in materials science nowadays. However, the biggest advantage of the real space solution of the K–S equations is that this computational scheme allows for efficient implementation of K–S scheme on the computers of parallel architecture, which is a prerequisite of modern computational science.

Search for efficient numerical codes solving the K–S equations is of particular interest nowadays, when the reliable predictions of materials properties require higher accuracy of computations than provided by standard approximations to the exchange-correlation functionals, such as local density approximation (LDA), generalized gradient approximation (GGA), or/and hybrid functionals, employed routinely in the K–S method. For example, the correct description of many systems (such as carbon-based layered materials) requires the proper account of the dispersive long-range Van der Waals (VdW) interactions. These interactions are not provided by the standard functionals. The exact-exchange (EXX) KS-DFT with correlation energy calculated employing the random phase approximations (RPA) has proved to overcome many shortcomings of the standard KS–DFT approximations [5] and, in particular, correctly accounts for VdW interactions. However, the EXX scheme alone is computationally very demanding and requires large computational power and/or efficient numerical algorithms.

In this communication, we examine the accuracy and efficiency of the parallelization scheme for the K–S method with the EXX exchange functional implementing direct r-space formalism.

2. Methodology

Our studies are based on the newly developed method of the solution of the K–S equations directly on a discrete grid in r-space. This scheme has been previously used by other authors [6, 7], however, we have written our own computer code to make it suitable for parallelization. This has been done by implementing the finite difference method [8] and developing the efficient parallelization procedure. In our computational scheme, three layers of parallelization method have been implemented, as it is presented in Fig. 1. Following computational

[*]corresponding author; e-mail: Jacek.Majewski@fuw.edu.pl

steps have been performed in parallel: (i) the summation over k-points to get total charge density and energy, (ii) the solution of K–S equations for wave-functions of the occupied states, and (iii) for each k-point and the wave-function, the solution is searched for simultaneously on the subsets of r-points constituting the total mesh. The parallelization method is based on the standardized and portable message passing interface (MPI) [9].

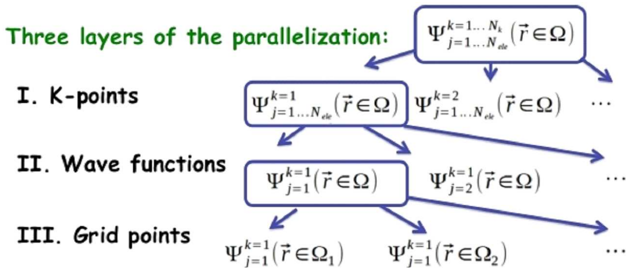
Three layers of the parallelization:
$\Psi^{k=1\ldots N_k}_{j=1\ldots N_{ele}}(\vec{r}\in\Omega)$
I. K-points
$\Psi^{k=1}_{j=1\ldots N_{ele}}(\vec{r}\in\Omega)$
$\Psi^{k=2}_{j=1\ldots N_{ele}}(\vec{r}\in\Omega)$
...
II. Wave functions
$\Psi^{k=1}_{j=1}(\vec{r}\in\Omega)$
$\Psi^{k=1}_{j=2}(\vec{r}\in\Omega)$
...
III. Grid points
$\Psi^{k=1}_{j=1}(\vec{r}\in\Omega_1)$
$\Psi^{k=1}_{j=1}(\vec{r}\in\Omega_2)$
...

Fig. 1. Three MPI parallelization levels are employed in the code, distributing the computational tasks over processors. The processors are organized in a hierarchy of groups. The first group is distributed over k-points (denoted by k). For each of the k-point group, the wave function subgroup (the number of wave function denoted by j) is distinguished. Moreover, for given k-point and wave function, the mesh parallelization is done.

In the developed code, we employ variational implementation of the EXX method, closely following Ref. [10].

3. Results

The efficiency tests of the developed computational scheme have been performed for the representative system consisting of two carbon atoms. We have done two tests. In the first one, we check how the computational time depends on the number of processors (cores) used for the parallelization. We have performed tests for two exchange-correlation functionals in the K–S scheme, the first one with the standard LDA functional, and the second one with the accurate EXX functional. To get some relation to the "supercell" calculations, periodic boundary conditions in three dimensions have been imposed on the calculated system, and 30 × 30 × 30 real-space grid have been used for the cubic supercell with the edge equal to 7.5 Å. All necessary Brillouin-zone (BZ) integrations have been carried out on a uniform mesh with 8 k-points, which corresponds to the $2\times2\times2$ grid. During the calculation, the self-consistent field (SCF) cycle has been iterated until the total energy changed by less than 10^{-3} eV. Our results show that the calculations with EXX formalism require usually at least 10 times longer CPU time than the standard LDA ones (see Fig. 2). Partly, it is caused by the fact that more SCF steps are needed in the EXX scheme than in the standard LDA method to reach the same accuracy level of the energy. It is also apparent that the highest acceleration of computations is reached for distribution of computational task over 16 cores, for both the LDA and the EXX exchange functional. For higher number of cores used in the parallelization procedure, the time of computations saturates in the LDA case and even increases in the EXX problems. This behavior is ascribed to the increasing communication burden among the critical number of processors. Through parallelization, the computation time can be decreased by factor 6 in the LDA case, and by factor 4 in the EXX case.

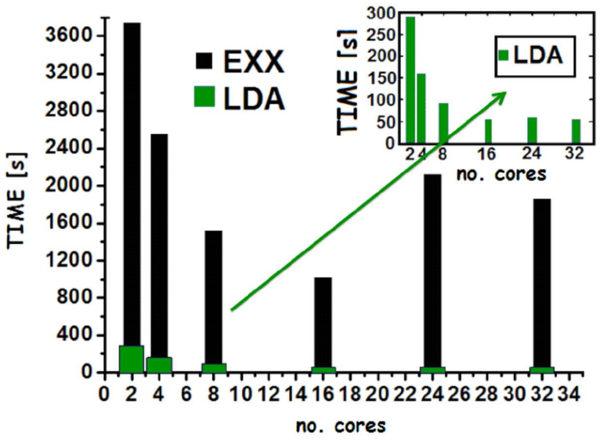

Fig. 2. Dependence of the time needed to reach the required convergence of the calculations versus the number of cores used for approaches with the LDA and the EXX functional in the K–S scheme. The EXX is roughly 10 times computationally more demanding than the standard LDA. The inset shows the zoom in of the scaling of the CPU time with the number of cores for the LDA method.

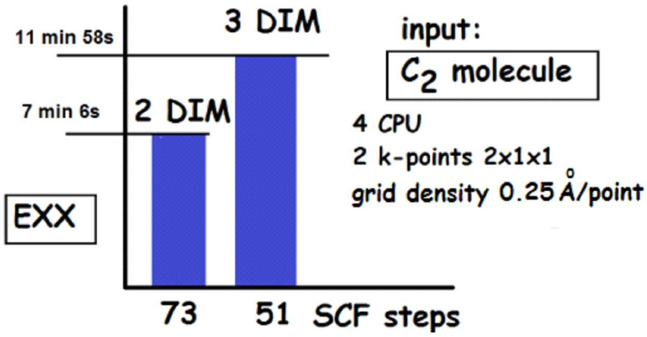

Fig. 3. Dependence of the computation time on the type of boundary conditions employed in the computations. In the 2DIM case (periodic boundary conditions in two dimensions) one can save up about 25% of computer time in comparison to the 3DIM case (periodic boundary conditions employed in all three dimensions), however 2DIM case requires more SCF steps.

In the second test, we take the advantage of the developed code that gives us opportunity not to use the periodic supercell in all three dimensions and consider the same system of C_2 molecules on the same 30 × 30 × 30 point grid as in the first test, but now employing periodic boundary conditions only in two dimensions and not in all three, as it has been done previously. For tests with periodic boundary conditions in three and two dimensions (indicated hereafter as 3DIM and 2DIM, respectively), we use more sophisticated EXX functional. We perform the integration over the Brillouin zone employing in both cases two k-points, and use four cores to parallelize the code. The results are depicted in Fig. 3. One can see that employing periodic boundary conditions in two instead of three dimensions can speed up the calculations by about 25%, even when the number of self-consistent steps required to reach assumed convergence increases considerably in the 2DIM case (73 in comparison to 51 in the 3DIM case). This is the result of the different Hamiltonian matrices (i.e., kinetic energy part of the Hamiltonian) used in 2DIM and 3DIM cases according to finite difference method based discretizing scheme.

4. Summary

Our calculations have revealed that the usage of the exact exchange scheme for solution of the Kohn–Sham equations costs approximately 10 times more computational time than employment of the standard local density approximation to the exchange and correlation functional. Three layers of parallelization employed in our code can accelerate computations by a factor of 4 and 6 in the case of the EXX and the LDA functionals employed to solve Kohn-Sham equations, respectively. The parallelization procedure gets more efficient and the calculations can be accelerated further, if one reduces the number of directions with the artificial boundary conditions employed within the supercell geometry to solve the K–S equations.

Acknowledgments

The support of The Foundation for Polish Science (FNP) through the TEAM project and of the National Research Council (NCN) through the grant HARMONIA DEC-2013/10/M/ ST3/00793 is gratefully acknowledged. Numerical calculations have been performed using the facilities of the Interdisciplinary Centre of Modeling, University of Warsaw.

References

[1] W. Kohn, L.J. Sham, *Phys. Rev.* **140**, A1133 (1965).

[2] G. Kresse, J. Hafner, *Phys. Rev. B* **47**, 558 (1993).

[3] S. Baroni, A. Dal Corso, S. de Gironcoli, P. Giannozzi, *J. Phys. Condens. Matter* **21**, 395502 (2009).

[4] J.M. Soler, E. Artacho, J.D. Gale, A. Garcia, J. Junquera, P. Ordejon, D. Sanchez-Portal, *J. Phys. Condens. Matter* **14**, 2745 (2002).

[5] H.V. Nguyen, G. Galli, *J. Chem. Phys.* **132**, 044109 (2010).

[6] M.M.G. Alemany, M. Jain, L. Kronik, J.R. Chelikowsky, *Phys. Rev. B* **69**, 075101 (2004).

[7] W. Mi, X. Shao, Ch. Su, Y. Zhou, S. Zhang, Q. Li, H. Wang, L. Zhang, M. Miao, Y. Wang, Y. Ma, *Comp. Phys. Commun.* **200**, 87 (2016); and references therein.

[8] T.L. Beck, *Rev. Mod. Phys.* **72**, 1041 (2000).

[9] E. Lusk, N. Doss, A. Skjellum, *Parallel Comput.* **22**, 789 (1996).

[10] T.W. Hollins, S.J. Clark, K. Refson, N.I. Gidopoulos, *Phys. Rev. B* **85**, 235126 (2012).