Journal of Computational Physics 307 (2016) 593–613

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

RESCU: A real space electronic structure method

Vincent Michaud-Rioux*, Lei Zhang, Hong Guo

Center for the Physics of Materials, Department of Physics, McGill University, Montreal, H3A 2T8, Canada

ARTICLE INFO

Article history:
Received 15 October 2015
Accepted 7 December 2015
Available online 14 December 2015

Keywords:
Density-functional theory
Methods of electronic structure calculations
Finite-difference methods
Self-consistent-field methods

ABSTRACT

In this work we present RESCU, a powerful MATLAB-based Kohn–Sham density functional theory (KS-DFT) solver. We demonstrate that RESCU can compute the electronic structure properties of systems comprising many thousands of atoms using modest computer resources, e.g. 16 to 256 cores. Its computational efficiency is achieved from exploiting four routes. First, we use numerical atomic orbital (NAO) techniques to efficiently generate a good quality initial subspace which is crucially required by Chebyshev filtering methods. Second, we exploit the fact that only a subspace spanning the occupied Kohn–Sham states is required, and solving accurately the KS equation using eigensolvers can generally be avoided. Third, by judiciously analyzing and optimizing various parts of the procedure in RESCU, we delay the $O(N^3)$ scaling to large N, and our tests show that RESCU scales consistently as $O(N^{2.3})$ from a few hundred atoms to more than 5000 atoms when using a real space grid discretization. The scaling is better or comparable in a NAO basis up to the 14,000 atoms level. Fourth, we exploit various numerical algorithms and, in particular, we introduce a partial Rayleigh–Ritz algorithm to achieve efficiency gains for systems comprising more than 10,000 electrons. We demonstrate the power of RESCU in solving KS-DFT problems using many examples running on 16, 64 and/or 256 cores: a 5832 Si atoms supercell; a 8788 Al atoms supercell; a 5324 Cu atoms supercell and a small DNA molecule submerged in 1713 water molecules for a total 5399 atoms. The KS-DFT is entirely converged in a few hours in all cases. Our results suggest that the RESCU method has reached a milestone of solving thousands of atoms by KS-DFT on a modest computer cluster.

1. Introduction

Density functional theory (DFT) [1] based numerical programs are nowadays the standard tool for predicting and understanding the structural and electronic properties of materials that involve many electrons. The idea of treating complicated many-body interactions in real materials by a self-consistent mean field theory appeared in the early days of quantum mechanics. In 1927, Thomas and Fermi proposed a semiclassical model [2,3] in which electrons in an external potential are described using only the electronic density. Subsequent calculations are simplified since the complicated many-body wavefunction is avoided. The Thomas–Fermi model was later improved by Dirac who included an exchange energy functional [4] and by von Weizsäcker who added a gradient correction to the kinetic energy functional [5]. Nearly four decades later, Hohenberg and Kohn (HK) put DFT on firm theoretical footing by proving that the ground-state expectation values are functionals of the density and that the ground-state density can be calculated by minimizing an energy functional [1].

* Corresponding author.
E-mail addresses: vincentm@physics.mcgill.ca (V. Michaud-Rioux), zhanglei@physics.mcgill.ca (L. Zhang), guo@physics.mcgill.ca (H. Guo).

http://dx.doi.org/10.1016/j.jcp.2015.12.014
0021-9991/© 2015 Elsevier Inc. All rights reserved.

Certain assumptions of the original HK theorems, such as the ground-state non-degeneracy, were later relaxed or eliminated [6]. These theories proved that the ground-state properties of any electronic system can in principle be calculated – if not necessarily understood – without using many-body wave functions. For practical applications, Kohn and Sham (KS) demonstrated that the problem of minimizing the total energy of a system with respect to the electronic density could take the form of a non-interacting electron problem [7]. In the KS formulation, the kinetic energy is evaluated via single particle wave functions which is more accurate than using kinetic energy functionals that depend explicitly on the density. KS-DFT allows one to analyze a variety of physical systems and performing a DFT calculation today is all but synonymous to solving the KS equation [7].

Various approaches for solving the KS equation have emerged such as the full potential all-electron methods [8,9] and the *ab initio* pseudopotential methods [10–14]. In KS-DFT solvers, several bases have been used to express quantum mechanical operators including real space Cartesian grids, finite elements, planewaves, wavelets, numerical atomic orbitals (NAO), Gaussian orbitals, muffin-tin orbitals and some others. The goal is to predict structural and electronic properties of real materials reaching the required accuracy for the given research topic and KS-DFT is playing a prominent role in materials physics and engineering.

At present, a major issue of practical DFT methods is their limited capability of solving material problems involving large number of atoms using a small computer cluster (e.g. 16 to 256 cores). For instance, algorithms implemented in state-of-the-art electronic packages such as VASP [12,13] and AbInit [15] can comfortably solve systems comprising of a few hundred atoms – but not many thousands on such small computer clusters. With the widening accessibility of supercomputers and the developments of advanced parallel computing algorithms, heroic KS-DFT calculations at the level of 10,000 atoms became possible in recent years, but at the expense of using thousands or even tens of thousands of computing cores [16–18]. Nevertheless, for practical material research and innovation, many small research groups in the world do not have access, cannot afford or simply wish not to use supercomputers. An urgent and very important task is to develop a KS-DFT method that can solve the KS equation without degrading the solution accuracy, at the level of several thousand atoms or more on a small computer cluster. It is the purpose of this work to report and describe such a KS-DFT solver and its associated software implementation.

To see why it is still possible to gain computational efficiency in traditional eigenvalue-based KS-DFT approaches, we note – as others had noted before us [19,20] – that the solution process of the KS-DFT is a self-consistent procedure where one numerically converges the Hamiltonian step by step by solving the KS equation repeatedly and accurately. However, it appears unclear why one has to solve accurately the KS equation for the not-yet-converged Hamiltonian in the intermediary steps. Another observation is that, in the eigensolver-based KS-DFT methods, different parts in the computation scale differently as a function of electron number N, some $O(N)$, others $O(N^2)$ and eventually these are dominated by the $O(N^3)$ parts. If one is able to *"delay"* the crossover to $O(N^3)$ scaling, larger systems can potentially be solved using small computers. It turns out that these computational gains can be realized as we present below.

Our KS-DFT method combines NAO and the real space finite-differences plus Chebyshev filtering (CF) technique introduced by Zhou et al. [19,20]. We found it is key to generate efficiently a proper initial subspace in the Chebyshev filtering framework, and this is achieved by the use a NAO basis. We advance efficient parallelization, a partial Rayleigh–Ritz (pRR) method for the computation of the density matrix and careful optimization of the solution process, and we have reached our goal of solving solid state physics problems consisting of thousands of atoms using 16 to 256 cores. Our code is called RESCU – which stands for Real space Electronic Structure CalcUlator – and it is implemented in the technical computing platform MATLAB. We use our own MPI and ScaLAPACK interfaces to harness efficiently the computational power of the cores. As such, RESCU combines the vocations of a prototyping code and a production code. In particular, the pRR allows us to compute the single particle density matrix in problems involving an exceedingly large number of electrons by taking advantage of the quasi-minimal property of basis sets built from CF. In the present paper, we will present the algorithmic and implementation advancements achieved during the development of RESCU. As practical examples, we demonstrate the following KS-DFT calculations: we simulate 5832 Si atoms (23,328 electrons) on a real space grid, converging the entire KS-DFT calculation using 256 cores for about 5.5 hours; we simulate 4000 Al atoms (12,000 electrons) on a real space grid, converging the entire KS-DFT calculation using 64 cores for about 5.1 hours; we simulate a supercell consisting of 13,824 Si atoms (55,296 electrons) using a NAO basis, converging the entire calculation using 64 cores for about 6.4 hours; we simulate a supercell consisting of 5324 Cu atoms (58,564 electrons) using a NAO basis, converging the entire calculation using 256 cores for about 12 hours. We also consider a disordered system consisting of a small DNA molecule submerged in 1713 water molecules, for a total of 5399 atoms (14,596 electrons), and converge the entire KS-DFT run in 9.6 hours on 256 cores. These results are compiled in Table 2 which is found in section 7. The scaling of the RESCU method is presented going from 16 cores to 256 cores for various tests. Finally, since RESCU is primarily a real space implementation of KS-DFT, it does not require periodicity when dealing with condensed phase materials and can thus easily treat problems involving interfaces, surfaces, defects, disordered materials, etc.

The paper is organized as follow. In section 2, we briefly state the fundamentals of DFT and introduce the single particle density matrix theoretical framework which is used throughout this article. In section 3, we review the state-of-the-art numerical methods for solving the KS equations and recount their advantages and disadvantages. In section 4, we describe in some detail the Chebyshev filtering method. In section 5, we present a computational complexity analysis of the Chebyshev filtering method and introduce the partial Rayleigh–Ritz algorithm. We explain how it takes advantage of the Chebyshev filtered basis sets to improve on the standard Rayleigh–Ritz algorithm. In section 6, we describe the implementation of the