Society of Engineering Science 51st Annual Technical Meeting

1–3 October 2014

Purdue University, West Lafayette, Indiana, USA

Coarse-Graining KS-DFT

Ponga, Mauricio, mponga@caltech.edu; Ortiz, Michael; Bhattacharya, Kaushik, California Institute of Technology, United States

ABSTRACT

In this study, we develop and implement a Coarse-Grained version of the Kohn-Sham Density Funtional (CG-KS-DFT) method to predict the evolution of crystal defects. The CG-KS-DFT method used in this study is based on the Linear Scaling Spectral Gauss Quadrature (LSSGQ) method, which proposes a reformulation of the traditional DFT equations. One of the main advantages of the LSSGQ method is that eliminates the need to explicitly compute orbitals. This property is achieved by using integral representation of the electronic quantities over the spectrum of the linear Hamiltonian operator. In addition, the evaluation of these integrals can be performed using spectral Gaussian quadrature rules. Therefore, the spectral nodes and weights of the quadrature rule are obtained using an efficient Lanczos type iteration, which are computed independently for each point in the domain. This property allows us to apply a systematic coarse graining description of the LSSGQ method, for example, using the Quasi-Continuum (QC) framework. Within this technique, a systematically coarsening of the domain is performed by applying judicious kinematic constraints, reducing the total number of degrees of freedom of the system. Finally, the combination of the LSSGQ method and the coarse graining approximation of the QC method enables the analysis of defects at a fraction of the original computational cost, without any significant loss of accuracy. In this study, we introduce the fundamental equations to develop a coarse graining description. Then, we compute the formation energies of different defects in Mg, such us vacancies, divacancies, dislocations, and twin boundaries for crystals with large number of atoms. In addition, the parallel performance of the implementation and some experiment for massively large parallel programing are also going to be presented.