Computer Physics Communications 215 (2017) 1–6

Contents lists available at ScienceDirect

Computer Physics Communications

journal homepage: www.elsevier.com/locate/cpc

Optimizing kernel methods for Poisson integrals on a uniform grid

D. Gabay [a], A. Boag [a], A. Natan [a,b,*]

[a] Department of Physical Electronics, Tel-Aviv University, Tel-Aviv, 69978, Israel

[b] The Sackler Center for Computational Molecular and Materials Science, Tel-Aviv University, Tel-Aviv, 69978, Israel

ARTICLE INFO

Article history:
Received 13 November 2016
Received in revised form
23 January 2017
Accepted 24 January 2017
Available online 2 February 2017

Keywords:
Poisson solver
Density-Functional-Theory
Electrostatic potential
Ab-initio

ABSTRACT

We analyze the error and error propagation in the calculation of the Poisson integral on a uniform grid within Density Functional Theory (DFT) real-space calculations. We suggest and examine several schemes for near neighbors' interaction correction for the Green's function kernel to improve the accuracy. Finally, we demonstrate the effect of the different kernels on DFT eigenvalues and Hartree energy accuracy in systems such as C_{60} and $C_{40}H_{82}$.

1. Introduction

The solution of the Poisson equation plays an important role in first principles methods such as Density Functional Theory (DFT) [1], time dependent DFT (TDDFT) [2], Hartree–Fock [3], GW [4], and others. In recent years this is becoming even more important as many DFT methods utilize hybrid functionals that include a fraction of Fock exchange or screened exchange [1,5–11]. The Poisson equation for the electrostatic potential, V, is given by:

$$\nabla^2 V(\mathbf{r}) = -4\pi\rho(\mathbf{r}) \tag{1}$$

where ρ is the electronic density. A common approach to solve Eq. (1) is to use iterative solvers such as the conjugate gradient (CG) method [12] after representing it in a given basis. For isolated systems, Dirichlet boundary conditions are used to supplement the equation, by calculating the electrostatic potential outside the domain. In this work, we focus on the solution of Eq. (1) on a uniform discrete grid. In such a representation, the density, ρ, is sampled on the grid and the differential Laplacian operator is replaced by a high order finite difference expression [13–15].

An alternative approach to solving Eq. (1) is to write the equivalent Poisson integral:

$$V(\mathbf{r}) = \int_D \frac{\rho(\mathbf{r}')}{|\mathbf{r}' - \mathbf{r}|} d\mathbf{r}' \tag{2}$$

where D is the computational domain. The advantage of Eq. (2) is that there is no need to calculate the boundary conditions. A disadvantage is that a direct calculation of the integral is too expensive, i.e. $O(N^2)$, where N is the number of grid points. This computational difficulty can be resolved by efficient schemes such as Fast Fourier Transform (FFT) [16,17], Fast Multipole Method (FMM) [18,19], auxiliary grids [20,21] and tensor methods [22,23] that achieve $\mathcal{O}(N \log N)$ or even $O(N)$ scaling.

Another potential problem in the numerical evaluation of Eq. (2) is the correct discretization of the integral. A naive discretization would lead to a too large error (as is demonstrated later). Mathematically this is because a high-order integration and smooth representation of the charge are needed, Physically — the Green's function, $1/|\mathbf{r}' - \mathbf{r}|$, is correct only in the continuum limit and a different function should be used when doing the calculation on a grid. There could be several ways to solve this problem, mathematically – higher order integration methods such as Gaussian quadrature can improve the result, physically – one can suggest Green functions (or kernels) that are more suitable for the discrete grid. In this manuscript, we analyze the error of different kernel methods in evaluating the integral of Eq. (2) and compare to the analytical solution for a Gaussian charge distribution and to the solution of Eq. (1) with high order finite difference methods. We use FFT for the analysis, but the conclusions can be applied to any other numerical integration method. We start with a description of the different kernel methods, we then analyze the errors and finally demonstrate how they propagate into errors in eigenvalues of DFT calculations.

An efficient approach which we do not compare here is the transformation of the integral in Eq. (2) into an equivalent

* Corresponding author at: Department of Physical Electronics, Tel-Aviv University, Tel-Aviv, 69978, Israel.
E-mail addresses: dor0gabay@gmail.com (D. Gabay), boag@eng.tau.ac.il (A. Boag), amirnatan@post.tau.ac.il (A. Natan).

http://dx.doi.org/10.1016/j.cpc.2017.01.016
0010-4655/

integral [24,25]:

$$\int_{\mathcal{R}^3} \frac{\rho(\mathbf{r}')}{|\mathbf{r}'-\mathbf{r}|}d\mathbf{r}' = \frac{2}{\sqrt{\pi}}\int_0^\infty dt \int_{\mathcal{R}^3} d\mathbf{r}' e^{-t^2|\mathbf{r}-\mathbf{r}'|^2}\rho(\mathbf{r}'). \tag{3}$$

The spatial integral in Eq. (3) can be easily separated to 1D integrals in each coordinate. This transformation can then be used, together with tensor decomposition methods for the density, to yield efficient and accurate calculations [22–29] with implementations for multi-resolution wavelets basis, uniform grids and more general mesh structures.

2. Discrete integration schemes

We assume a uniform discrete grid where $x_i = ih$, $y_j = jh$, $z_k = kh$, where i, j, k are integers and h is the grid spacing. We further assume that there is a reversible index function $n(i, j, k)$ that gives a unique and reversible integer index mapping for every point $\mathbf{r}_n = (x_i, y_j, z_k)$ in the grid. For simplicity, we assume a box shaped domain. With this, we can write the naive discrete approximation for Eq. (2) as:

$$V(\mathbf{r}_n) \simeq h^3 \sum_{m\neq n} \frac{\rho(\mathbf{r}_m)}{|\mathbf{r}_n - \mathbf{r}_m|}. \tag{4}$$

However, Eq. (4) has several problems. First it does not include the self interaction term, second, the near field potential of a cubic voxel with some charge density can deviate significantly from $1/r$ behavior due to the cube finite size. The summation in Eq. (4) can be written in a more general way as:

$$V(\mathbf{r}_n) = h^3 \sum_m \rho(\mathbf{r}_m) G(\mathbf{r}_n - \mathbf{r}_m). \tag{5}$$

It is easy to choose a G function that will reduce Eq. (5) to Eq. (4), but we can also choose different discrete Green functions that will make the discrete summation of Eq. (5) closer to the exact integral of Eq. (2). Furthermore, Eq. (5) retains the form of a convolution and so it is easy to efficiently calculate it with FFT as is further discussed in the next paragraph.

2.1. FFT formulation

A direct calculation of Eq. (5) can be too time consuming for large grids ($\mathcal{O}(N^2)$), we therefore use FFT for this calculation. Eastwood and Brownrigg [30] have shown that by zero padding of the density and doubling of the domain in each direction, Eq. (5) can be turned *exactly* into a cyclic convolution summation. This can be used with FFT, the convolution theorem and inverse FFT to yield a very efficient ($\mathcal{O}(N \log N)$) procedure for calculating the summation of Eq. (5). Such schemes have already been implemented in DFT codes [16,17,31]. We use it for a fast calculation of Eq. (5) to analyze the accuracy of different choices of the kernel function $G(\mathbf{r}_n)$.

2.2. Finite difference delta kernel

One possible way to define a corrected kernel is to numerically solve the Poisson equation with the CG method for a delta function density [21]. This is done once at the beginning of the simulation.

$$\nabla^2 G_{\mathrm{CG}}(\mathbf{r}) = -4\pi\delta(\mathbf{r}). \tag{6}$$

The solution of the continuous form of Eq. (6) is $G_{\mathrm{CG}}(\mathbf{r}) = 1/r$. In the discrete form, we replace the Laplacian with its high order finite difference equivalent [15] and for the density we have the Kronecker discrete delta function. The solution $G_{\mathrm{CG}}(\mathbf{r}_n)$, of the discrete high order finite difference equation, can now be used as the Green's function kernel in Eq. (5). The solution of the discrete form of Eq. (6) with CG can be done on a domain that is twice larger in each dimension, compared to the original domain, and in that case we can use the solution directly as $G(\mathbf{r}_n)$. Another possibility is to, instead, solve the equation on a smaller domain, Ω_{CG} and then extend the kernel according to:

$$G(\mathbf{r}_n) = \begin{cases} G_{\mathrm{CG}}(\mathbf{r}_n), & \mathbf{r}_n \in \Omega_{\mathrm{CG}} \\ 1/r_n, & \text{otherwise.} \end{cases} \tag{7}$$

2.3. Semi-analytical kernel corrections

Wilton et al. [32] have suggested an effective analytical approximation for the Green's function kernel on discrete grids. The method transforms the three dimensional integral into a single dimensional one by consecutively applying the Gauss integral theorem. This leads to a new analytical approximation for the Green's function kernel, $G(\mathbf{r}_n)$, of which the details are provided in the article by Wilton et al. [32]. We call this method the Wilton kernel in future references in this paper.

2.4. Numerically optimized kernel

While the Wilton's method has an elegant analytical form, the accuracy it achieves is typically not sufficient for DFT calculations. One approach, can be to start with a given kernel (which can be Wilton's) and use numerical methods to minimize the error relative to the known analytical solution of a given Gaussian charge density with a specific value of h/σ, where h is the grid step size and σ is the Gaussian standard deviation. We can then check the performance of such an optimized kernel with other Gaussian charge densities (different values of h/σ). We have performed this procedure with the Wilton kernel as a starting point. We minimized the total error, ϵ_V, as defined later in the text by Eq. (13), relative to the analytical solution for a Gaussian charge density with $h/\sigma = 0.5$. We have used as optimization parameters the values of the kernel function at points inside a cube of size of 1 to 3 neighbors in each direction. The optimization results in a set of correction factors, $F(\mathbf{r}_n)$, that give the new, numerically optimized, kernel:

$$G_{\mathrm{NOPT}}(\mathbf{r}_n) = F(\mathbf{r}_n) \cdot G_{\mathrm{Wilton}}(\mathbf{r}_n). \tag{8}$$

For example, the number of parameters in a cube of 1 neighbor in each direction is $3 \times 3 \times 3 = 27$, but it can be reduced to 4 independent parameters by symmetry considerations. It should be noted that the factors, $F(\mathbf{r}_n)$, are different from unity only within the defined cube of near neighbors.

2.5. High order spline basis functions

Another possible way to calculate the Poisson integral is to represent the charge density by known basis functions and calculate the Poisson integral for those basis functions. There can be many choices for such basis functions. In our analysis, we have chosen to use linear and cubic spline basis functions [33]. We can write:

$$\begin{aligned} V(\mathbf{r}_n) &= \int_D \frac{\rho(\mathbf{r})}{|\mathbf{r}_n - \mathbf{r}|}d\mathbf{r} \\ &= \sum_m \rho(\mathbf{r}_m) \int \frac{B(\mathbf{r}-\mathbf{r}_m)}{|\mathbf{r}-\mathbf{r}_n|}d\mathbf{r} \end{aligned} \tag{9}$$

where we represent $\rho(\mathbf{r})$ by:

$$\rho(\mathbf{r}) = \sum_m \rho(\mathbf{r}_m) B(\mathbf{r}-\mathbf{r}_m) \tag{10}$$