

## Efficient solution of Poisson's equation with free boundary conditions

Luigi Genovese<sup>a)</sup> and Thierry Deutsch<sup>b)</sup>*Département de Recherche Fondamentale sur la Matière Condensée, SP2M/L\_Sim, CEA-Grenoble, 38054 Grenoble Cedex 9, France*

Alexey Neelov and Stefan Goedecker

*Institute of Physics, University of Basel, Klingelbergstrasse 82, CH-4056 Basel, Switzerland*

Gregory Beylkin

*Department of Applied Mathematics, University of Colorado at Boulder, Boulder, Colorado 80309-0526*

(Received 15 May 2006; accepted 13 July 2006; published online 17 August 2006)

Interpolating scaling functions give a faithful representation of a localized charge distribution by its values on a grid. For such charge distributions, using a fast Fourier method, we obtain highly accurate electrostatic potentials for free boundary conditions at the cost of  $O(N \log N)$  operations, where  $N$  is the number of grid points. Thus, with our approach, free boundary conditions are treated as efficiently as the periodic conditions via plane wave methods. © 2006 American Institute of Physics. [DOI: 10.1063/1.2335442]

### I. INTRODUCTION

Solving Poisson's equation

$$\nabla^2 V = \rho \quad (1)$$

to find the electrostatic potential  $V$  arising from a charge distribution  $\rho$  is a basic problem that can be found in nearly any field of physics and chemistry. It is therefore essential to have efficient solution methods for it.

A large variety of methods has been developed for systems of point particles interacting by electrostatic forces. Formally this problem can be considered as the solution of Poisson's equation where the charge distribution is a sum of delta functions. The classical method for periodic boundary conditions is the Ewald method.<sup>1</sup> For large systems and free boundary conditions, the fast multipole method<sup>2</sup> is a powerful method due to its linear scaling with respect to the number of particles.

The fast multipole method (FMM) has been generalized for continuous systems where the charge density can be represented as a sum of Gaussians multiplied by a polynomial.<sup>3</sup> This generalization exhibits linear scaling with respect to the volume but not with respect to the number of Gaussians at constant volume. It works well in the context of quantum chemistry calculations with medium size Gaussian basis sets where the charge density is naturally obtained in the required form but it is not a general purpose method.

For periodic boundary conditions and smoothly varying charge densities, plane wave methods are simple and powerful because the Laplacian is a diagonal matrix in a plane wave representation. Given a Fourier representation of the charge density, one can therefore obtain the potential simply by dividing each Fourier coefficient by  $|\mathbf{k}|^2$ , where  $\mathbf{k}$  is the wave vector of the Fourier coefficient. If the charge density is originally given in real space, a first fast Fourier transform

(FFT) is required to obtain its Fourier coefficients and a second FFT is required to obtain the potential in real space from its Fourier coefficients. The overall scaling is therefore of order  $N \log N$ , where  $N$  is the number of grid points.

Because of the simplicity of these plane wave methods, various attempts have been made to generalize them to free boundary conditions. The most rudimentary method is just to take a very large periodic volume and to hope that the amplitude of the potential is nearly zero on the surface of the volume. Due to the long range of electrostatic forces, this condition is, however, not fulfilled for volume sizes that are affordable with plane waves. Such a scheme is only possible if adaptive periodic wavelets are used.<sup>4</sup> In addition periodic boundary conditions do not permit to treat systems with monopoles and dipoles because for such systems no well defined solution exists under periodic boundary conditions. The first method to attack the problem in a systematic way was by Hockney.<sup>5</sup> He proposed a Fourier approximation to the kernel

$$K(r) = \frac{1}{r} \quad (2)$$

of the Poisson equation. The method was intended for applications in plasma physics where no high accuracy is required. In other application such as electronic structure calculations, high accuracy is required and the method is not optimal. For a spherical geometry, the Fourier coefficients of the  $1/r$  kernel can be calculated analytically. This is the basis of the simple and powerful method by Füsti-Molnar and Pulay.<sup>6</sup> Its obvious restriction is that it is efficient only for spherical geometries. Another method was proposed by Martyna and Tuckerman. This method gives high accuracy only in the center of the computational volume. It requires therefore artificially large simulation boxes which are numerically very expensive. All the above discussed methods use FFTs at some point and have therefore an  $O(N \log N)$  scaling.

<sup>a)</sup>Electronic mail: luigi.genovese@cea.fr

<sup>b)</sup>Electronic mail: thierry.deutsch@cea.fr

In this paper we will describe a new Poisson solver for free boundary conditions on a uniform mesh. Contrary to Poisson solvers based on plane wave functions, our method is using interpolating scaling functions to represent the charge density. It is therefore from the beginning free of long-range interactions between supercells that falsify results if plane waves are used to describe nonperiodic systems. Due to the convolutions we have to evaluate our method that has an  $N \log N$  scaling instead of the ideal linear scaling. Due to its small prefactor the method is, however, most efficient when dealing with localized densities such as those that can be found, for example, in the context of *ab initio* pseudopotential electronic structure calculations using finite differences,<sup>7</sup> finite elements,<sup>8</sup> or plane waves for nonperiodic systems.

## II. INTERPOLATING SCALING FUNCTIONS

Scaling functions arise in wavelet theory.<sup>9</sup> A scaling function basis set can be obtained from all the translations by a certain grid spacing  $h$  of the mother wavelet centered at the origin. What distinguished scaling functions from other basis functions with compact support such as finite elements or Lagrange functions is the refinement relation. The refinement relation establishes a relation between a scaling function  $\phi(x-i)$  and the same scaling functions compressed by a factor of 2, or, equivalently, between the scaling functions on a grid with grid spacing  $h$  and another one with spacing  $h/2$ . For the scaling function centered at the origin, it reads

$$\phi(x) = \sum_{j=-m}^m h_j \phi(2x-j), \quad (3)$$

where the  $h_j$ 's are the elements of a filter that characterize the wavelet family. An interpolating wavelet has in addition the property that it is equal to 1 at the origin and zero at all other integer points. Because of this property, it is very simple to find the scaling function expansion coefficients of any function. The coefficients are just the values of the function to be expanded on the grid.  $m$ th order interpolating scaling functions are generated by  $(m-1)$ th order recursive interpolation.<sup>10</sup> Figure 1 shows an 14th order and 100th order interpolating scaling functions. Three-dimensional scaling functions can be obtained as the product of their one-dimensional counterparts,

$$\begin{aligned} \Phi_{i_1, i_2, i_3}(\mathbf{r}) &= \phi_{i_1}(x) \phi_{i_2}(y) \phi_{i_3}(z) \\ &= \phi(x-i_1) \phi(y-i_2) \phi(z-i_3), \end{aligned} \quad (4)$$

where  $\mathbf{r}=(x,y,z)$ . The points  $i_1$ ,  $i_2$ , and  $i_3$  are the nodes of a uniform three-dimensional mesh, with  $i_p=1, \dots, n_p$ ,  $p=1, 2, 3$ .

Continuous charge distributions are represented in numerical work typically by their values  $\rho_{i,j,k}$  on a grid. It follows from the above described properties of interpolating scaling functions that the corresponding continuous charge density is given by

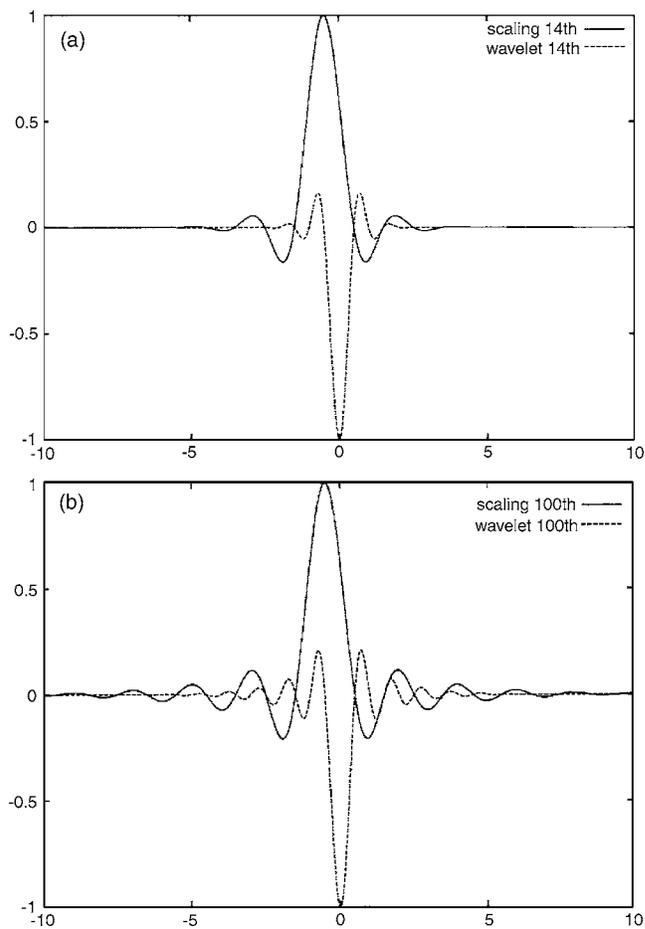


FIG. 1. Plots of interpolating scaling functions and wavelets of 14th (a) and 100th (b) orders.

$$\rho(\mathbf{r}) = \sum_{i_1, i_2, i_3} \rho_{i_1, i_2, i_3} \phi(x-i_1) \phi(y-i_2) \phi(z-i_3). \quad (5)$$

The mapping of Eq. (5) between the discretized and continuous charge distribution ensures that the first  $m$  discrete and continuous moments are identical for a  $m$ th order interpolating wavelet family, i.e.,

$$\sum_{i,j,k} i^{\ell_1} j^{\ell_2} k^{\ell_3} \rho_{i,j,k} = \int dx x^{\ell_1} dy y^{\ell_2} dz z^{\ell_3} \rho(\mathbf{r}), \quad (6)$$

if  $\ell_1, \ell_2, \ell_3 < m$ . The proof of this relation is given in the Appendix. Since the various multipoles of the charge distribution determine the major features of the potential, the above equalities tell us that a scaling function representation gives the most faithful mapping between a continuous and discretized charge distribution for electrostatic problems.

## III. POISSON'S EQUATION IN A BASIS SET OF INTERPOLATING SCALING FUNCTIONS

As is well known, the following integral equation gives the potential for free boundary conditions:

$$V(\mathbf{r}) = \int d\mathbf{r}' \frac{1}{|\mathbf{r}-\mathbf{r}'|} \rho(\mathbf{r}'). \quad (7)$$

We are interested in the values of the potential on the same grid that was used for the charge density. Denoting the po-

tential on the grid point  $\mathbf{r}_{j_1, j_2, j_3} = (x_{j_1}, y_{j_2}, z_{j_3})$  by  $V_{j_1, j_2, j_3} = V(\mathbf{r}_{j_1, j_2, j_3})$ , we have

$$V_{j_1, j_2, j_3} = \sum_{i_1, i_2, i_3} \rho_{i_1, i_2, i_3} \int d\mathbf{r}' \frac{\phi_{i_1}(x') \phi_{i_2}(y') \phi_{i_3}(z')}{|\mathbf{r}_{j_1, j_2, j_3} - \mathbf{r}'|}. \quad (8)$$

The above integral defines the discrete kernel

$$K(i_1, j_1; i_2, j_2; i_3, j_3) = \int d\mathbf{r}' \phi_{i_1}(x') \phi_{i_2}(y') \phi_{i_3}(z') \frac{1}{|\mathbf{r}_{j_1, j_2, j_3} - \mathbf{r}'|}. \quad (9)$$

Since the problem is invariant under combined translations of both the source point  $(i_1, i_2, i_3)$  and the observation point  $(j_1, j_2, j_3)$ , the kernel depends only on the difference of the indices,

$$K(i_1, j_1; i_2, j_2; i_3, j_3) = K(i_1 - j_1, i_2 - j_2, i_3 - j_3), \quad (10)$$

and the potential  $V_{j_1, j_2, j_3}$  can be obtained from the charge density  $\rho_{i_1, i_2, i_3}$  by the following three-dimensional convolution:

$$V_{j_1, j_2, j_3} = \sum_{i_1, i_2, i_3} K(i_1 - j_1, i_2 - j_2, i_3 - j_3) \rho_{i_1, i_2, i_3}. \quad (11)$$

Once the kernel is available in Fourier space, this convolution can be evaluated with two FFTs at a cost of  $O(N \log N)$  operations where  $N = n_1 n_2 n_3$  is the number of three-dimensional grid points. Since all the quantities in the above equation are real, real-to-complex FFTs can be used to reduce the number of operations compared to the case where one would use ordinary complex-complex FFTs. Obtaining the kernel in Fourier space from the kernel  $K(j_1, j_2, j_3)$  in real space requires another FFT. The FFTs are performed using a modified version of the parallel FFT algorithm described in Ref. 11 that gives high performance on a wide range of computers.

It remains now to calculate the values of all the elements of the kernel  $K(k_1, k_2, k_3)$ . Solving a three-dimensional integral for each element would be much too costly and we use therefore a separable approximation of  $1/r$  in terms of Gaussians,<sup>12,13</sup>

$$\frac{1}{r} \approx \sum_k \omega_k e^{-p_k r^2}. \quad (12)$$

In this way all the complicated three-dimensional integrals become products of simple one-dimensional integrals. Using 89 Gaussian functions with the coefficients  $\omega_k$  and  $p_k$  suitably chosen, we can approximate  $1/r$  with an error less than  $10^{-8}$  in the interval  $[10^{-9}, 1]$ . If we are interested in a wider range, e.g., a variable  $R$  going from zero to  $L$ , we can use  $r = R/L$ :

$$\frac{L}{R} = \sum_k \omega_k e^{-(p_k/L^2)R^2}, \quad (13)$$

$$\frac{1}{R} = \frac{1}{L} \sum_k \omega_k e^{-p_k R^2}, \quad (14)$$

$$p_k = \frac{p_k}{L^2}. \quad (15)$$

With this approximation, we have that

$$K_{j_1, j_2, j_3} = \sum_{k=1}^{89} \omega_k K_{j_1}(p_k) K_{j_2}(p_k) K_{j_3}(p_k), \quad (16)$$

where

$$K_j(p_k) = \int \varphi_j(x) e^{-p_k x^2} dx \quad (17)$$

$$= \int \varphi_0(x) e^{-p_k(x-j)^2} dx. \quad (18)$$

So we only need to evaluate  $89 \times \max(\{n_1, n_2, n_3\})$  integrals of the type

$$K_j(p) = \int \varphi_0(x) e^{-p(x-j)^2} dx \quad (19)$$

for some value of  $p$  chosen between  $3 \times 10^{-5}$  and  $3 \times 10^{16}$ .

The accuracy in calculating the integrals can be further improved by using the refinement relation for interpolating scaling functions (3).

From (19), we can evaluate  $K_i(4p)$  as

$$K_i(4p) = \int \varphi(x) e^{-4p(x-i)^2} dx \quad (20)$$

$$= \frac{1}{2} \int \varphi(x/2) e^{-p(x-2i)^2} dx \quad (21)$$

$$= \frac{1}{2} \sum_j h_j \int \varphi_j(x) e^{-p(x-2i)^2} dx \quad (22)$$

$$= \frac{1}{2} \sum_j h_j K_{2i-j}(p). \quad (23)$$

The best accuracy in evaluating numerically the integral is attained for  $p < 1$ . For a fixed value of  $p$  given by Eq. (12), the relation (23) is iterated  $n = \lceil \log_4(p) \rceil$  times starting with  $p_0 = p/4^n$ . So the numerical calculation of the integrals  $K_i(p)$  is performed as follows: For each  $p$ , we compute the number  $n$  of required recursion levels and calculate the integral  $K_i(p_0)$ . The value of  $n$  is chosen such that  $p_0 \approx 1$  so we have a Gaussian function that is not too sharp. The evaluation of the interpolating scaling functions is fast on a uniform grid of points so we perform a simple summation over all the grid points. In Fig. 2, we show that 1024 points are enough to obtain machine precision. Note that the values of  $K_0(p)$  vary over many orders of magnitude as shown in Fig. 2.

#### IV. NUMERICAL RESULTS AND COMPARISON WITH OTHER METHODS

We have compared our method with the plane wave methods by Hockney<sup>5</sup> and Martyna and Tuckerman<sup>14</sup> as implemented in the CPMD electronic structure program.<sup>15</sup> As expected Hockney's method does not allow to attain high

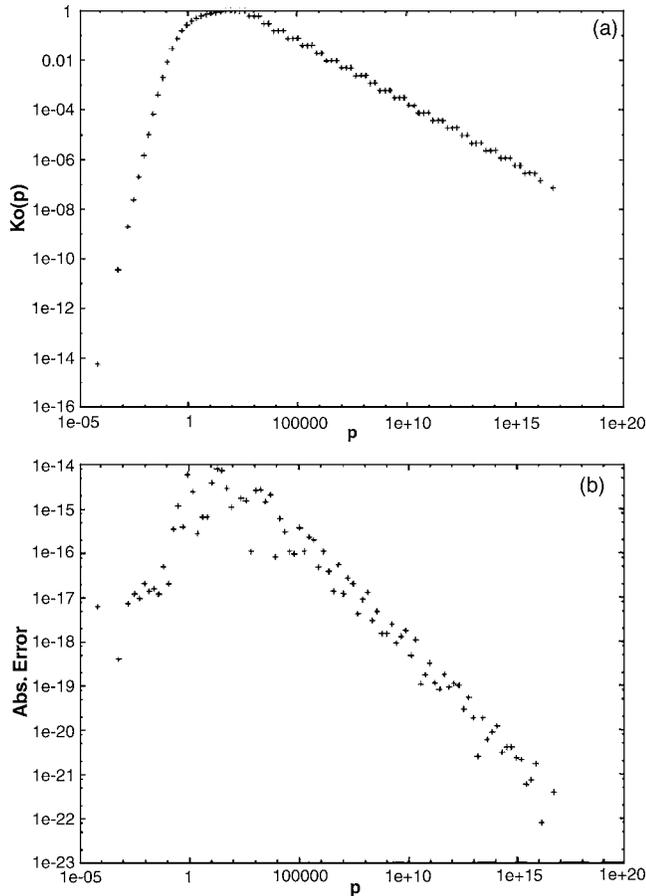


FIG. 2. Plots of the value of  $K_0(p)$  (a) and error of the integration defining its value (b) of for 1024 integration points for all the values of  $p$  used in the tensor decomposition of  $1/r$  in Gaussian functions.

accuracy. The method by Martyna and Tuckerman has a rapid exponential convergence rate which is characteristic for plane wave methods. Our new method has an algebraic convergence rate of  $h^m$  with respect to the grid spacing  $h$ . By choosing very high order interpolating scaling functions, we can get arbitrarily high convergence rates. Since convolutions are performed with FFT techniques, the numerical ef-

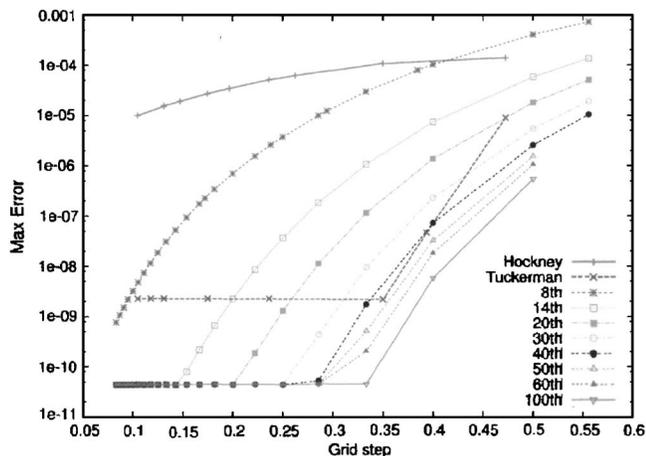


FIG. 3. Accuracy comparison between our method with interpolating scaling functions of different orders and the Hockney of Martyna-Tuckerman method as implemented in CPMD. The accuracy of our method is finally limited by the accuracy of the expansion of Eq. (12) with 89 terms.

TABLE I. The elapsed time in seconds required on a Cray XT3 (based on AMD Opteron processors) to solve Poisson's equation on a  $128^3$  grid as a function of the number of processors. Since Poisson's equation is typically solved many times, the time for setting up the kernel is not included. Including the setup time of the kernel increases the total timing by about 50%, since one additional FFT is needed.

1	2	4	8	16	32	64
0.92	0.55	0.27	0.16	0.11	0.08	0.09

fort does not increase as the order  $m$  is increased. The accuracy shown in Fig. 3 for the method of Martyna and Tuckerman is the accuracy in the central part of the cube that has  $1/8$  of the total volume of the computational cell. Outside this volume, errors blow up. So the main disadvantage of this method is that a very large computational volume is needed in order to obtain accurate results in a sufficiently large target volume. For this reason the less accurate Hockney method is generally preferred in the CPMD program.<sup>16</sup> There is, however, a modification of the method by Yarne *et al.*<sup>17</sup> (implemented in their PINY-MD code) that reduces the computational cost due to the large volume, but at the price of using different levels of resolutions for the long- and the short-range components.

A strictly localized charge distribution, i.e., a charge distribution that is exactly zero outside a finite volume, cannot be represented by a finite number of plane waves. This is an inherent contradiction in all the plane wave methods for the solution of Poisson's equation under free boundary conditions. For the test shown in Fig. 3, we used a Gaussian charge distribution whose potential can be calculated analytically. The Gaussian was embedded in a computational cell that was so large that the tails of the Gaussian were cut off at an amplitude of less than  $1 \times 10^{-16}$ . A Gaussian can well be represented by a relatively small number of plane waves, and so the above described problem is not important. For other localized charge distributions that are less smooth, a finite Fourier representation is worse and leads to a spilling of the charge density out of the original localization volume. This will lead to inaccuracies in the potential.

Table I shows the required CPU time for a  $128^3$  problem as a function of the number of processors on a Cray parallel computer. The parallel version is based on a parallel three-dimensional FFT.

A package for solving Poisson's equation according to the method described here can be downloaded from Ref. 18.

In conclusion, we have presented a method that allows to obtain the potential of a localized charge distribution under free boundary conditions with an  $O(N \log N)$  scaling in a mathematically clean way. Even though our method has the same scaling behavior as existing plane wave methods, it is not a plane wave method in the sense that neither the charge density nor the potential is ever represented by plane waves. Instead interpolating scaling functions are used for the representation of the charge density.

## ACKNOWLEDGMENTS

The authors acknowledge interesting discussions with Reinhold Schneider and Robert Harrison. This work was

supported by the European Commission within the Sixth Framework Programme through NEST-BigDFT (Contract No. BigDFT-511815) and by the Swiss National Science Foundation. Two of the authors (L.G. and T.D.) acknowledge financial support from the EU program IHP "Breaking Complexity" (Contract No. HPRN-CT 2002-00286). Research of one of the authors (G.B.) was partially supported by DOE Grant No. DE-FG02-03ER25583, DOE/ORNL Grant No. 4000038129, and DARPA/ARO Grant No. W911NF-04-1-0281. The timings were performed at the CSCS (Swiss Supercomputing Center) in Manno.

#### APPENDIX: PROOF OF EQ. (6)

In the present appendix, we are going to prove Eq. (6). Let  $\phi(x)$  be an interpolating scaling function of Delauriers-Dubuc, of the order  $m$ , and  $\rho_{i_1, i_2, i_3}$  be a three-dimensional array of constant coefficients. Let, further,

$$\rho(\mathbf{r}) = \sum_{i_1, i_2, i_3} \rho_{i_1, i_2, i_3} \phi(x - i_1) \phi(x - i_2) \phi(x - i_3). \quad (\text{A1})$$

Then,

$$\begin{aligned} \int d\mathbf{r} x^{l_1} y^{l_2} z^{l_3} \rho(\mathbf{r}) &= \int x^{l_1} y^{l_2} z^{l_3} \sum_{i_1, i_2, i_3} \rho_{i_1, i_2, i_3} \phi(x - i_1) \phi(x - i_2) \phi(x - i_3) d\mathbf{r} \\ &= \sum_{i_1, i_2, i_3} \rho_{i_1, i_2, i_3} \int x^{l_1} \phi(x - i_1) dx \int y^{l_2} \phi(y - i_2) dy \times \int z^{l_3} \phi(z - i_3) dz = \sum_{i_1, i_2, i_3} \rho_{i_1, i_2, i_3} i_1^{l_1} i_2^{l_2} i_3^{l_3}. \end{aligned}$$

$$\begin{aligned} \sum_{i_1, i_2, i_3} i_1^{l_1} i_2^{l_2} i_3^{l_3} \rho_{i_1, i_2, i_3} \\ = \int d\mathbf{r} x^{l_1} y^{l_2} z^{l_3} \rho(\mathbf{r}) \quad \text{if } 0 \leq l_1, l_2, l_3 < m. \end{aligned} \quad (\text{A2})$$

This follows from the fact, proven in Ref. 19 that the first  $m$  moments of the scaling function obey the formula

$$M_l = \int \phi(x) x^l dx = \delta_l, \quad l = 0, \dots, m-1. \quad (\text{A3})$$

Shift the integration variable, we have

$$\begin{aligned} \int \phi(x - j) x^l dx &= \int \phi(t) (t + j)^l dt \\ &= \int \phi(t) \sum_{p=0}^l C_l^p t^p j^{l-p} dt = j^l \end{aligned}$$

Then, inserting (A1) into the right side of (A2), we get

<sup>1</sup>P. Ewald, Ann. Phys. **64**, 251 (1921).

<sup>2</sup>L. Greengard and V. Rokhlin, J. Comput. Phys. **73**, 325 (1987); H. Cheng, L. Greengard, and V. Rokhlin, *ibid.* **155**, 468 (1999).

<sup>3</sup>M. Strain, G. Scuseria, and M. Frisch, Science **271**, 51 (1996); P. Maslen, C. Ochsenfeld, C. White, M. Lee, and M. Head-Gordon, J. Phys. Chem. **102**, 2215 (1998); J. Perez-Jorda and W. Yang, J. Chem. Phys. **107**, 1218 (1997).

<sup>4</sup>S. Goedecker and O. Ivanov, Solid State Commun. **105**, 665 (1998).

<sup>5</sup>R. W. Hockney, Methods Comput. Phys. **9**, 135 (1970).

<sup>6</sup>L. Füstli-Molnar and P. Pulay, J. Chem. Phys. **116**, 7795 (2002).

<sup>7</sup>J. R. Chelikowsky, N. Troullier, and Y. Saad, Phys. Rev. Lett. **72**, 1240 (1994).

<sup>8</sup>J. E. Pask, B. M. Klein, C. Y. Fong, and P. A. Sterne, Phys. Rev. B **59**, 12352 (1999).

<sup>9</sup>I. Daubechies, *Ten Lectures on Wavelets* (SIAM, Philadelphia, PA, 1992); S. Goedecker, *Wavelets and Their Application for the Solution of Differential Equations* (Presses Polytechniques Universitaires et Romandes, Lausanne, Switzerland, 1998).

<sup>10</sup>G. Deslauriers and S. Dubuc, Constructive Approx. **5**, 49 (1989).

<sup>11</sup>S. Goedecker, Comput. Phys. Commun. **76**, 294 (1993).

<sup>12</sup>G. Beylkin and L. Monzon, Appl. Comput. Harmon. Anal. **19**, 17 (2005); G. Beylkin and M. J. Mohlenkamp, SIAM J. Sci. Comput. (USA) **26**, 2133 (2005); Proc. Natl. Acad. Sci. U.S.A. **99**, 10246 (2002).

<sup>13</sup>The same expansion was used to find a wavelet representation of the entire integral operator in R. Harrison, G. Fann, T. Yanai, Z. Gan, and G. Beylkin, J. Chem. Phys. **121**, 11587 (2004).

<sup>14</sup>G. J. Martyna and M. E. Tuckerman, J. Chem. Phys. **110**, 2810 (1999).

<sup>15</sup>J. Hutter, A. Alavi, T. Deutsch, M. Bernasconi, S. Goedecker, D. Marx, M. Tuckerman, and M. Parrinello, CPMD, Version 3.3, Max-Planck-Institut für Festkörperforschung and IBM Zürich Research Laboratory, 1995–1999.

<sup>16</sup>D. Marx and J. Hutter, in *Modern Methods and Algorithms of Quantum Chemistry*, edited by J. Grotendorst (John von Neumann Institute for Computing, Jülich, 2000).

<sup>17</sup>D. A. Yarne, M. E. Tuckerman, and G. J. Martyna, J. Chem. Phys. **115**, 3531 (2001).

<sup>18</sup>www.unibas.ch/comphys/comphys/SOFTWARE

<sup>19</sup>N. Saito and G. Beylkin, IEEE Trans. Signal Process. **41**, 3584 (1993).