JCTC Journal of Chemical Theory and Computation

Adaptive Finite Element Method for Solving the Exact Kohn−Sham Equation of Density Functional Theory

Eric J. Bylaska,*,† Michael Holst,*,‡ and John H. Weare*,§

William R. Wiley Environmental Molecular Sciences Laboratory, Pacific Northwest National Laboratory, P.O. Box 999, Richland, Washington 99352, and Department of Mathematics and Department of Chemistry and Biochemistry, University of California, San Diego, La Jolla, California 92093

Received August 26, 2008

Abstract: Results of the application of an adaptive piecewise linear finite element (FE) based solution using the FETK library of M. Holst to a density functional theory (DFT) approximation to the electronic structure of atoms and molecules are reported. The severe problem associated with the rapid variation of the electronic wave functions in the near singular regions of the atomic centers is treated by implementing completely unstructured simplex meshes that resolve these features around atomic nuclei. This concentrates the computational work in the regions in which the shortest length scales are necessary and provides for low resolution in regions for which there is no electron density. The accuracy of the solutions significantly improved when adaptive mesh refinement was applied, and it was found that the essential difficulties of the Kohn−Sham eigenvalues equation were the result of the singular behavior of the atomic potentials. Even though the matrix representations of the discrete Hamiltonian operator in the adaptive finite element basis are always sparse with a linear complexity in the number of discretization points, the overall memory and computational requirements for the solver implemented were found to be quite high. The number of mesh vertices per atom as a function of the atomic number Z and the required accuracy ε (in atomic units) was estimated to be $v(\varepsilon, Z) \approx 122.37(Z^{2.2346}/\varepsilon^{1.1173})$, and the number of floating point operations per minimization step for a system of N_A atoms was found to be $O(N_A{}^3 v(\varepsilon, Z))$ (e.g., with $Z = 26$, $\varepsilon = 0.0015$ au, and $N_A = 100$, the memory requirement and computational cost would be ∼0.2 terabytes and ∼25 petaflops). It was found that the high cost of the method could be reduced somewhat by using a geometric-based refinement strategy to fix the error near the singularities.

1. Introduction

Kohn−Sham density functional theory (DFT)[1,2] which can be used to predict the structures, properties, and reactivities for a wide variety of solid state and molecular systems has become a state of the art tool. In many cases it can achieve chemical accuracy at a smaller cost than traditional quantum chemistry methods. It is now routine at this level of theory to perform simulations containing hundreds of atoms, and on today's parallel supercomputers, simulations containing over a thousand atoms are feasible; making realistic descriptions of material surfaces and defects possible. While current implementations of DFT are very efficient and the results are adequate for many cases,[3,4] the much wider application of these approaches to the even more demanding systems encountered in complex technology problems are limited. The limitations are that they use basis sets[5,6] and/or pseudopotentials[7] that are highly engineered, they scale as $O(N_A{}^3)$

* E-mail: eric.bylaska@pnl.gov (E.J.B.); mholst@ucsd.edu (M.H.); jweare@ucsd.edu (J.H.W.).

† Pacific Nortwest National Laboratory.

‡ Department of Mathematics, University of California, San Diego.

§ Department of Chemistry and Biochemistry, University of California, San Diego.

10.1021/ct800350j CCC: $40.75
Published on Web 03/05/2009

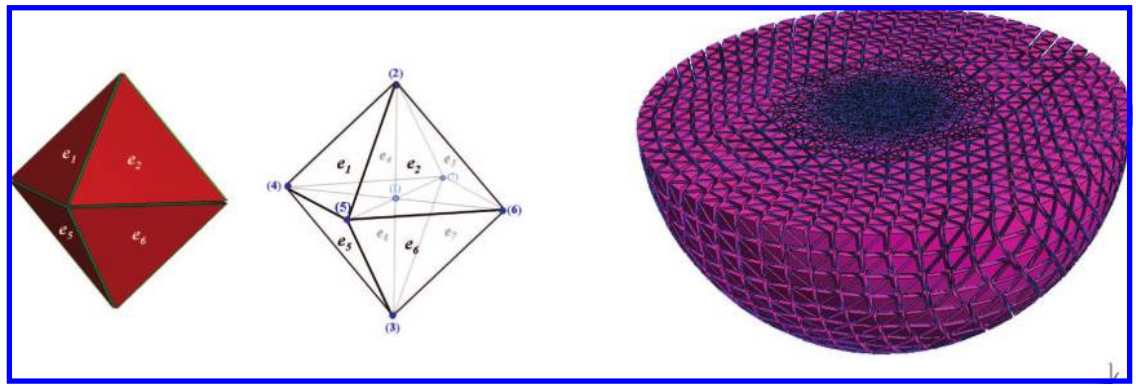

Figure 1. Examples of 3D finite element meshes. (left) Tetrahedral domain containing 8 elements and 7 vertex nodes. The elements are labeled e_l, and the nodes are labeled (m) in this figure. (right) Adaptive hemisphere domain containing 453 608 elements and 81 406 vertex nodes.

or $O(N_A^4)$ in the number of atoms, and the parallel scalings of existing solution methods are not good enough to exploit the performance of the next generation parallel computers. Hence, there is still a need to investigate other computational methods for solving DFT.

There have been a number of efforts to develop fast ab initio solvers based on real space solutions to the DFT equations.[8−27] Uniform finite difference gridding coupled with multilevel solvers has led to significant progress in the calculation of large systems[8−14] with large numbers of processors (~10 000 processors).[10] While these methods are often robust enough for predicting structural properties, they are not very efficient for describing multiple length scales and as a result do not have an accurate description near the atomic centers. In particular, when the interaction between the electron and the nucleus is described by the proper singular potential, $-(Ze/|\vec{r} - \vec{R}_I|)$, the singular behavior at $|\vec{r} - \vec{R}_I|$ can cause trouble with convergence. In fact, this kind of potential cannot be represented by uniform meshes methods. Adaptive finite element methods on the other hand, which can telescope down to the singularity, can in principle describe this kind of potential, and if used with a low order elements (i.e., piecewise finite elements) all the quantum mechanical operators can be represented by $O(N)$ sparse matrices, which can in principle limit the communications per processor to be $O(1)$.

Even though adaptive real space methods for DFT have shown some promise for describing singular electron−ion interactions, these methods have needed to use large number of elements for high Z atoms to describe the singular potential accurately. Earlier work by Bylaska et al.,[18] in which they developed a multilevel eigenvalue solver based on structured adapted mesh refinement and finite difference gridding worked well for simple systems such as H, and H_2^+. However, by $Z = 10$ (i.e., N_e), errors as large as 1 hartree were seen with this approach. These large errors led Kohn et al. to replace the atomic singular potentials with pseudopotentials and replace the finite difference solver with an adaptive finite element solver.[19] This new solver improved the accuracy somewhat, but at the time, it was too computationally intensive to be considered as a practical alternative to standard DFT solvers. Recently, Fattebert et al. have revisited these solvers and have shown them to be competitive with fast Fourier transform (FFT) solvers when very stiff pseudopotentials are used;[20] however, this work still had to rely on using pseudopotentials. Recent work by Batcho using spectral element methods[24] and Harrison et al. using a multiwavelet (high-order) basis,[21−23] has also been shown to be computationally competitive with standard DFT solvers and in some cases surpass them.

In this paper, we present an overview of our implementation of an unstructured adaptive finite element (FE) first principles solver and apply it to DFT equations which contain atomic singular potentials to estimate its overall memory and computational requirements. This solver is based on the FETK finite element framework of Holst.[28] The implementation is unique in that tetrahedral elements are used rather than parallelepipeds, and it also makes use of completely unstructured simplex meshes that have the advantage of being able to resolve the near singular features around atomic nuclei with minimal computational resources. This type solver has several potential advantages. It has compact support, it can be controlled by systematically increasing the number of the basis functions, it produces sparse matrices, it allows for the variable resolution in real space and can exactly represent potentials with "Z/r" singularities, and it does not require the use of a computationally intensive transform.

In section 2, a concise review of the FE method is given, and in section 3, the formulation of FE DFT equations is presented. In section 4, by using test problems which incorporate the critical issues of multiple length scales and the singular behavior of the potential, the overall memory and computational requirements per atom needed by the solver are estimated. The solver is then illustrated for several atoms and molecules including H, He, Li, Ne, H_2^+, and Li_2. Finally conclusions are given in section 5.

2. Background of the Finite Element (FE) Method

In the FE procedure,[29−34] the solution domain Ω (e.g., Figure 1) is divided up into connected polyhedral subregions or elements, $\{e_l\}_{l=1}^{L}$, where L is the number of elements. For each element e_l, a set of T_l nodes is chosen.

$$\{\vec{N}_t^{e_l} = (x_t^{e_l}, y_t^{e_l}, z_t^{e_l})\}_{t=1}^{T_l} \quad (1)$$

From these nodes, a global set of nodes is defined from the union of the element nodes,

$$\{\vec{N}_m\}_{m=1}^{M} = \cup_{l=1}^{L}\{\vec{N}_t^{e_l}\}_{t=1}^{T_l} \quad (2)$$

where M is the number of nodes in the finite element mesh. Nodes may be located at an element vertex, face, edge, or in its interior. A set of T_l basis functions are then defined for each element e_l. The basis functions are defined such that they are nonzero only inside the element, represented as simple low order polynomials, and have a value of 1 at its associated node, i.e.

$$\phi_t^{e_l}(\vec{x}) = \phi_t^{e_l}(x,y,z) = \sum_{n_1}\sum_{n_2}\sum_{n_3} a_{t,n_1,n_2,n_3} x^{n_1} y^{n_2} z^{n_3} \quad (3)$$

Using these basis functions, any piecewise polynomial function may be expanded as follows

$$u(\vec{x}) = \sum_{l=1}^{L}\sum_{t=1}^{T_l} \tilde{c}_{t,l}\phi_t^{e_l}(\vec{x}) \quad (4)$$

where $\tilde{c}_{t,\,l}$ are the expansion coefficients. This expansion is somewhat intricate given that neighboring elements share nodes with one another, which in turn results in certain expansion coefficients being equal to one another. For example, the expansion of a function using the finite element mesh shown in Figure 1 necessitates that the following coefficients be equal

$$\begin{aligned} c_1 = \tilde{c}_{11} = \tilde{c}_{12} = \tilde{c}_{13} = \tilde{c}_{14} = \tilde{c}_{15} = \tilde{c}_{16} = \tilde{c}_{17} = \tilde{c}_{18} \\ c_2 = \tilde{c}_{41} = \tilde{c}_{42} = \tilde{c}_{43} = \tilde{c}_{44} \\ c_3 = \tilde{c}_{45} = \tilde{c}_{46} = \tilde{c}_{47} = \tilde{c}_{48} \\ c_4 = \tilde{c}_{21} = \tilde{c}_{34} = \tilde{c}_{35} = \tilde{c}_{28} \\ c_5 = \tilde{c}_{22} = \tilde{c}_{31} = \tilde{c}_{36} = \tilde{c}_{25} \\ c_6 = \tilde{c}_{23} = \tilde{c}_{32} = \tilde{c}_{37} = \tilde{c}_{26} \\ c_7 = \tilde{c}_{24} = \tilde{c}_{34} = \tilde{c}_{38} = \tilde{c}_{27} \end{aligned} \quad (5)$$

To facilitate this mapping, a local to global index, $\tilde{m}(t, l)$, is defined. By using this index, the finite element expansion can then written as

$$u(\vec{x}) = \sum_{l=1}^{L}\sum_{t=1}^{T_l} c_{\tilde{m}(t,l)}\phi_t^{e_l}(\vec{x}) \quad (6)$$

Compared to eqs 5 and 6, this expansion is fairly uncomplicated. However, it can be simplified even further by introducing the following assembled finite element basis.

$$\eta_m(\vec{x}) = \sum_{l=1}^{L}\sum_{t=1}^{T_l} \phi_t^{e_l}(\vec{x})\delta_{m,\tilde{m}(t,l)} \quad (7)$$

With this assembled basis, the finite element expansion is

$$u(\vec{x}) = \sum_{m=1}^{M} c_m\eta_m(\vec{x}) \quad (8)$$

To facilitate the definition of the finite elements e_l and the corresponding basis functions as shown in eq 3, standard elements $\tilde{e}$ and their corresponding basis functions are introduced. This is done so that the basis functions and integrals for elements of different shapes are calculated readily, through a variable transformation, from the basis functions and integrals for a standard element. In this work, 3D tetrahedral elements with nodes at the vertices are used. The standard 3D tetrahedral element $\tilde{e}$ which covers the domain $[\tilde{x} = 0{:}1, \tilde{y} = 0{:}1 - \tilde{x}, \tilde{z} = 1 - \tilde{x} - \tilde{y}]$, is shown in Figure 2, and its standard local basis functions are

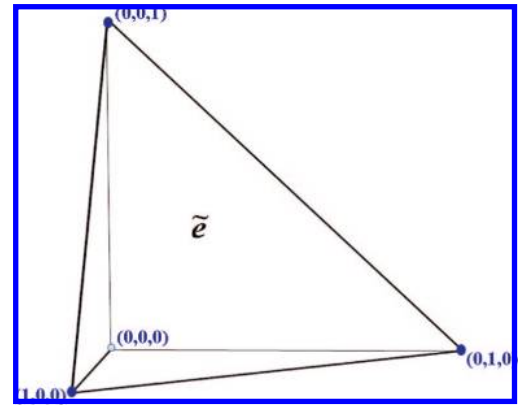

Figure 2. Standard 3D piecewise tetrahedral element.

$$\begin{aligned} \tilde{\phi}_1(\tilde{x},\tilde{y},\tilde{z}) &= 1 - \tilde{x} - \tilde{y} - \tilde{z} \\ \tilde{\phi}_2(\tilde{x},\tilde{y},\tilde{z}) &= \tilde{x} \\ \tilde{\phi}_3(\tilde{x},\tilde{y},\tilde{z}) &= \tilde{y} \\ \tilde{\phi}_4(\tilde{x},\tilde{y},\tilde{z}) &= \tilde{z} \end{aligned} \quad (9)$$

Integrals over the tetrahedral standard element,

$$I^{\tilde{e}}(\tilde{f}) = \int_{\tilde{e}} f(\vec{\tilde{x}})\,\mathrm{d}\vec{\tilde{x}} = \int_0^1 \mathrm{d}\tilde{z}\int_0^{1-\tilde{z}}\mathrm{d}\tilde{y}\int_0^{1-\tilde{z}-\tilde{y}}\mathrm{d}\tilde{x}\,\tilde{f}(\vec{\tilde{x}}) \quad (10)$$

can be computed for polynomial functions

$$\tilde{f}(\vec{\tilde{x}}) = \tilde{f}(\tilde{x},\tilde{y},\tilde{z}) = \tilde{x}^{n_1}\tilde{y}^{n_2}\tilde{z}^{n_3} \quad (11)$$

using the following analytic formula,

$$\begin{aligned} I^{\tilde{e}}(\tilde{x}^{n_1}\tilde{y}^{n_2}\tilde{z}^{n_3}) &= \int_0^1 \mathrm{d}\tilde{z}\int_0^{1-\tilde{z}}\mathrm{d}\tilde{y}\int_0^{1-\tilde{z}-\tilde{y}}\tilde{x}^{n_1}\tilde{y}^{n_2}\tilde{z}^{n_3} \\ &= \tilde{P}(n_1,0)\tilde{P}(n_2,n_1+1)\tilde{P}(n_3,n_1+n_2+2) \end{aligned} \quad (12)$$

where

$$\tilde{P}(a,b) = \sum_{k=0}^{b}\binom{b}{k}\frac{(-1)^k}{(a+k+1)} \quad (13)$$

Equations 12 and 13 are straightforward to compute. However, because the basis functions are only of a certain polynomial order $O(n)$, the integrals of eq 10 only need to be calculated to the same order in the finite element procedure. Therefore, the computation of the integrals of eq 10 can be further simplified by using a numerical method. In this work, the following formula is used

$$I^{\tilde{e}}(\tilde{f}) \approx \sum_{q=1}^{Q} w_q\tilde{f}(\vec{\tilde{x}}_q) \quad (14)$$

Table 1. 5-Point Tetrahedral Integration Points and Weights

formula type	points	weight
5-pt formula	(1/4, 1/4, 1/4)	−2/15
	(1/6, 1/6, 1/6)	3/40
	(1/2, 1/6, 1/6)	3/40
	(1/6, 1/2, 1/6)	3/40
	(1/6, 1/6, 1/2)	3/40

where $\{\vec{\tilde{x}}_q\}_{q=1}^{Q}$ and $\{w_q\}_{q=1}^{Q}$ are the integration point and weights respectively. Many different sets of integration points and weights can be constructed for use in eq 14; however, the computation will be more efficient for small Q. A 5-point formulation that can be used to integrate the standard 3D tetrahedral element to second-order is given in Table 1.

The affinte variable transformation is used to convert the basis function and integrals over an arbitrarily sized tetrahedral element e_l with vertices $\vec{x}_1^{e_l}$, $\vec{x}_2^{e_l}$, $\vec{x}_3^{e_l}$, and $\vec{x}_4^{e_l}$ in terms of a standard element $\tilde{e}$. This variable transformation is linear and invertible. It is defined by

$$\vec{x} = F^{e_l}\vec{\tilde{x}} + \vec{b}^{e_l} \tag{15}$$

or more explicitly by

$$\begin{aligned} x &= x_{e_l}(\tilde{x},\tilde{y},\tilde{z}) = F_{11}^{e_l}\tilde{x} + F_{12}^{e_l}\tilde{y} + F_{13}^{e_l}\tilde{z} + b_1^{e_l} \\ y &= y_{e_l}(\tilde{x},\tilde{y},\tilde{z}) = F_{21}^{e_l}\tilde{x} + F_{22}^{e_l}\tilde{y} + F_{23}^{e_l}\tilde{z} + b_2^{e_l} \\ z &= z_{e_l}(\tilde{x},\tilde{y},\tilde{z}) = F_{31}^{e_l}\tilde{x} + F_{32}^{e_l}\tilde{y} + F_{33}^{e_l}\tilde{z} + b_3^{e_l} \end{aligned} \tag{16}$$

where the matrix F^{e_l} is the Jacobean matrix and b^{e_l} is the location of the origin in the transformation.

$$F^{e_l} = \begin{bmatrix} (x_2^{e_l} - x_1^{e_l}) & (x_3^{e_l} - x_1^{e_l}) & (x_4^{e_l} - x_1^{e_l}) \\ (y_2^{e_l} - y_1^{e_l}) & (y_3^{e_l} - y_1^{e_l}) & (y_4^{e_l} - y_1^{e_l}) \\ (z_2^{e_l} - z_1^{e_l}) & (z_3^{e_l} - z_1^{e_l}) & (z_4^{e_l} - z_1^{e_l}) \end{bmatrix} \tag{17}$$

$$\vec{b}^{e_l} = \begin{bmatrix} x_1^{e_l} \\ y_1^{e_l} \\ z_1^{e_l} \end{bmatrix} \tag{18}$$

The inverse affine transformation is then

$$\vec{\tilde{x}} = (F^{e_l})^{-1}(\vec{x} - \vec{b}^{e_l})\vec{x} \tag{19}$$

or

$$\begin{aligned} \tilde{x} &= \tilde{x}_{e_l}(x,y,z) = (F^{e_l})_{11}{}^{-1}(x - b_1^{e_l}) + \\ &(F^{e_l})_{12}{}^{-1}(y - b_2^{e_l}) + (F^{e_l})_{13}{}^{-1}(z - b_3^{e_l}) \\ \tilde{y} &= \tilde{y}_{e_l}(x,y,z) = (F^{e_l})_{21}{}^{-1}(x - b_1^{e_l}) + \\ &(F^{e_l})_{22}{}^{-1}(y - b_2^{e_l}) + (F^{e_l})_{23}{}^{-1}(z - b_3^{e_l}) \\ \tilde{z} &= \tilde{z}_{e_l}(x,y,z) = (F^{e_l})_{31}{}^{-1}(x - b_1^{e_l}) + \\ &(F^{e_l})_{32}{}^{-1}(y - b_2^{e_l}) + (F^{e_l})_{33}{}^{-1}(z - b_3^{e_l}) \end{aligned} \tag{20}$$

where

$$\begin{aligned} (F^{e_l})_{11}{}^{-1} &= \frac{F_{22}^{e_l}F_{33}^{e_l} - F_{32}^{e_l}F_{23}^{e_l}}{|F^{e_l}|} \\ (F^{e_l})_{21}{}^{-1} &= \frac{F_{31}^{e_l}F_{23}^{e_l} - F_{21}^{e_l}F_{33}^{e_l}}{|F^{e_l}|} \\ (F^{e_l})_{31}{}^{-1} &= \frac{F_{21}^{e_l}F_{32}^{e_l} - F_{31}^{e_l}F_{22}^{e_l}}{|F^{e_l}|} \\ (F^{e_l})_{12}{}^{-1} &= \frac{F_{32}^{e_l}F_{13}^{e_l} - F_{12}^{e_l}F_{33}^{e_l}}{|F^{e_l}|} \\ (F^{e_l})_{22}{}^{-1} &= \frac{F_{11}^{e_l}F_{33}^{e_l} - F_{31}^{e_l}F_{13}^{e_l}}{|F^{e_l}|} \\ (F^{e_l})_{32}{}^{-1} &= \frac{F_{31}^{e_l}F_{12}^{e_l} - F_{11}^{e_l}F_{32}^{e_l}}{|F^{e_l}|} \\ (F^{e_l})_{13}{}^{-1} &= \frac{F_{12}^{e_l}F_{23}^{e_l} - F_{22}^{e_l}F_{13}^{e_l}}{|F^{e_l}|} \\ (F^{e_l})_{23}{}^{-1} &= \frac{F_{21}^{e_l}F_{13}^{e_l} - F_{11}^{e_l}F_{23}^{e_l}}{|F^{e_l}|} \\ (F^{e_l})_{33}{}^{-1} &= \frac{F_{11}^{e_l}F_{22}^{e_l} - F_{21}^{e_l}F_{12}^{e_l}}{|F^{e_l}|} \end{aligned} \tag{21}$$

Using this transformation, the global FE basis functions are written in terms of the standard local basis functions by

$$\phi_t^{e_l}(\vec{x}) = \begin{cases} \tilde{\phi}_t((F^{e_l})^{-1}\vec{x}) & \text{for } \vec{x} \in e_l \\ 0 & \text{otherwise} \end{cases} \tag{22}$$

and the gradients are written as

$$\frac{\mathrm{d}}{\mathrm{d}x^{(i)}}\phi_t^{e_l}(\vec{x}) = \begin{cases} \sum_{j=1}^{3}(F^{e_l})_{ji}{}^{-1}\frac{\mathrm{d}}{\mathrm{d}\tilde{x}^{(j)}}\tilde{\phi}_t((F^{e_l}\vec{x})) & \text{for } \vec{x} \in e_l \\ 0 & \text{otherwise} \end{cases} \tag{23}$$

With this transformation, the integral of a function f over an element is then

$$I^{e_l}(f) = \int_e f(\vec{x})\,\mathrm{d}\vec{x} \approx |F^{e_l}|\sum_{q=1}^{Q} w_q f(F^{e_l}\vec{\tilde{x}}_q) \tag{24}$$

The following element integrals are also needed for our adaptive multilevel finite element (FE) first principles solver.

$$G_r^{e_l}(f) = \int_{e_l}\phi_r^{e_l}(\vec{x})f(\vec{x})\,\mathrm{d}\vec{x} \approx |F^{e_l}|\sum_{q=1}^{Q}\tilde{\phi}_r(\vec{\tilde{x}}_q)f(F^{e_l}\vec{\tilde{x}}_q) \tag{25}$$

$$M_{rs}^{e_l} = \int_{e_l}\phi_r^{e_l}(\vec{x})\phi_s^{e_l}(\vec{x})\,\mathrm{d}\vec{x} \approx |F^{e_l}|\sum_{q=1}^{Q} w_q\tilde{\phi}_r(\vec{\tilde{x}}_q)\tilde{\phi}_s(\vec{\tilde{x}}_q) \tag{26}$$

$$K_{rs}^{e_l}(f) = \int_{e_l} \phi_r^{e_l}(\vec{x}) f(\vec{x}) \phi_s^{e_l}(\vec{x})\, \mathrm{d}\vec{x}$$

$$\approx |F^{e_l}| \sum_{q=1}^{Q} w_q \tilde{\phi}_r(\vec{\tilde{x}}_q) f(F^{e_l}\vec{\tilde{x}}_q) \tilde{\phi}_s(\vec{\tilde{x}}_q) \quad (27)$$

$$K_{rs}^{e_l}(u) = \int_{e_l} \phi_r^{e_l}(\vec{x}) \Big(\sum_{l'=1}^{L} \sum_{t=1}^{T_{l'}} c_{\tilde{m}(t,l')} \phi_t^{e_{l'}}(\vec{x}) \Big) \phi_s^{e_l}(\vec{x})\, \mathrm{d}\vec{x}$$

$$\approx |F^{e_l}| \sum_{q=1}^{Q} w_q \tilde{\phi}_r(\vec{\tilde{x}}_q) \Big(\sum_{t=1}^{T_l} c_{\tilde{m}(t,l)} \tilde{\phi}_t(\vec{\tilde{x}}_q) \Big) \tilde{\phi}_s(\vec{\tilde{x}}_q) \quad (28)$$

$$R_{rs}^{e_l}(f,\rho) = \int_{e_l} \phi_r^{e_l}(\vec{x}) f(\rho(\vec{x})) \phi_s^{e_l}(\vec{x})\, \mathrm{d}\vec{x}$$

$$\approx |F^{e_l}| \sum_{q=1}^{Q} w_q \tilde{\phi}_r(\vec{\tilde{x}}_q) f(\rho(F^{e_l}\vec{\tilde{x}}_q)) \tilde{\phi}_s(\vec{\tilde{x}}_q) \quad (29)$$

$$T_{rs}^{e_l} = \int_{e_l} \nabla\phi_r^{e_l}(\vec{x}) \cdot \nabla\phi_s^{e_l}(\vec{x})\, \mathrm{d}\vec{x}$$

$$\approx |F^{e_l}| \sum_{q=1}^{Q} w_q \Bigg[\sum_{i=1}^{3} \Bigg(\sum_{j=1}^{3} (F^{e_l})_{ji}^{-1} \frac{\mathrm{d}}{\mathrm{d}\tilde{x}^{(j)}} \tilde{\phi}_r(\vec{\tilde{x}}_q) \Bigg) \times \Bigg(\sum_{k=1}^{3} (F^{e_l})_{ki}^{-1} \frac{\mathrm{d}}{\mathrm{d}\tilde{x}^{(k)}} \tilde{\phi}_s(\vec{\tilde{x}}_q) \Bigg) \Bigg] \quad (30)$$

$$D_r^{e_l}(u_D) = -\int_{e_l} \phi_r^{e_l}(\vec{x}) \nabla^2 u_D(\vec{x})\, \mathrm{d}\vec{x}$$

$$= \int_{e_l} \nabla\phi_r^{e_l}(\vec{x}) \cdot \nabla u_D(\vec{x})\, \mathrm{d}\vec{x}$$

$$= \int_{e_l} \nabla\phi_r^{e_l}(\vec{x}) \cdot \nabla \Big(\sum_{l'=1}^{L} \sum_{t=1}^{T_{l'}} c_{\tilde{m}(t,l')} \phi_t^{e_{l'}}(\vec{x}) \Big)\, \mathrm{d}\vec{x}$$

$$= \sum_{t=1}^{T_l} c_{\tilde{m}(t,l)} T_{rt}^{e_l} \quad (31)$$

3. Formulation of FE DFT Equations

The electronic wave functions in density functional theory (DFT) in atomic units are given by the solutions to the following Kohn−Sham eigenvalue problem[1]

$$H\psi_i(\vec{x}) = \Big(-\frac{1}{2}\nabla^2 + V_{\mathrm{ext}} + V_{\mathrm{H}} + V_{\mathrm{xc}} \Big) \psi_i(\vec{x}) = \varepsilon_i \psi_i(\vec{x}) \quad (32)$$

$$\int_{\Omega} \psi_i(\vec{x}) \psi_j(\vec{x})\, \mathrm{d}\vec{x} = \delta_{ij} \quad (33)$$

where ε_i is an eigenvalue, and the wave functions $\{\psi_i\}$ satisfy the orthonormality constraints of a symmetric operator. In general, we require the lowest $N_e/2$ eigenvalues and wave functions for a spin-paired system, where N_e is the number of electrons in the system. The formulation for spin-unrestricted and fractionally occupied systems[2] are not presented here; however, generalizing the following equations to do so is quite straightforward. The external potential represents the external electrostatic field imposed on the system. For molecular systems V_{ext} represents the ion−electron interaction,

$$V_{\mathrm{ext}}(\vec{x}) = \sum_{I=1}^{N_A} \frac{-Z_I}{|\vec{x} - \vec{R}_I|} \quad (34)$$

where N_A is the number atoms, Z_I is the nuclear charge of atom I, and $\vec{R}_I$ is location of atom I. The Hartree potential, V_H, and the exchange-correlation potential, V_{xc}, are the effects of electron−electron interactions. Both of these potentials are functions of the electron density

$$\rho(\vec{x}) = \sum_{i=1}^{N_e/2} |\psi_i(\vec{x})|^2 \quad (35)$$

The exchange and correlation potential, V_{xc}, is a straightforward parametrized function of the electron density, e.g. Dirac exchange formula,[35]

$$V_{\mathrm{xc}}(\vec{x}) = -\Big(\frac{3}{\pi}\rho(\vec{x})\Big)^{1/3} = -\Bigg(\frac{3}{\pi}\sum_{i=1}^{N_e/2} |\psi_i(\vec{x})|^2\Bigg)^{1/3} \quad (36)$$

and the Hartree potential V_H is the solution to the Poisson equation

$$\nabla^2 V_{\mathrm{H}}(\vec{x}) = -4\pi\rho(\vec{x}) = -4\pi \sum_{i=1}^{N_e/2} |\psi_i(\vec{x})|^2 \quad (37)$$

Since both V_{xc} and V_H are functions of ρ, the Kohn−Sham eigenvalue problem must be solved self-consistently by an iterative algorithm. The standard approach for these type of problems is a Gummel-like iteration involving two computationally intensive kernels at each iteration:

(1) Calculation of the Hartree potential through the solution of the Poisson equation.

(2) Calculation of the eigenfunctions of the linearized generalized eigenvalue problem where the updated Hartree and exchange-correlation potentials are taken to be frozen.

The FE Poisson and FE DFT eigenvalue equations are generated by representing the Hartree potential and Kohn−Sham wave functions as a finite element expansion,

$$V_{\mathrm{H}}(\vec{x}) = \sum_{m=1}^{M} v_m \eta_m(\vec{x}) \quad (38)$$

$$\psi_i(\vec{x}) = \sum_{m=1}^{M} c_m^i \eta_m(\vec{x}) \quad (39)$$

and defining the boundary conditions for eqs 32 and 37. More extensive derivations of the weak formulation of the FE DFT and Poisson equations can be found in the work of Pask et al.[14] and Fattebert et al.[20] Free-space boundary conditions and periodic boundary conditions are the most common boundary conditions used for solving the Kohn−Sham eigenvalue equations. In this work, we chose to use free-space boundary conditions, i.e.

$$\begin{aligned} V_{\mathrm{H}}(|\vec{x}| \rightarrow \infty) &= 0 \\ \psi_i(|\vec{x}| \rightarrow \infty) &= 0 \end{aligned} \tag{40}$$

The problem with applying these boundary conditions is that the solution domain Ω does not go out to ∞. For eq 32, this is not a problem since the wave functions for most molecular systems decay exponentially and can readily be set to zero at $\partial\Omega$. However, for eq 37 with free-space boundary conditions, the potential decays $\propto 1/r$; hence, we must first calculate the boundary conditions on $\partial\Omega$. To do this, we use a high-order multipole expansion of the density to define the following far field expansion of the Hartree potential at the boundary,

$$V_{\mathrm{H}}(\vec{x} \in \partial\Omega) = \sum_{l=0}^{L_{\mathrm{MAX}}} \sum_{m=-l}^{l} N_{lm} M_{lm} \frac{T_{lm}(\hat{x})}{|\vec{x}|^{l+1}} \tag{41}$$

$$N_{lm} = \begin{cases} 1 & \text{for } m = 0 \\ 2\dfrac{(l-|m|)!}{(l+|m|)!} & \text{for } |m| > 0 \end{cases} \tag{42}$$

$$M_{lm} = \int_{\Omega} |\vec{x}'|^l \rho(\vec{x}') T_{lm}(\hat{x}')\, \mathrm{d}\vec{x}' \tag{43}$$

$$T_{lm}(\hat{x}) = \begin{cases} P_{l|m|}(\cos\theta) & \text{for } m = 0 \\ P_{l|m|}(\cos\theta)\cos|m|\varphi & \text{for } m > 0 \\ P_{l|m|}(\cos\theta)\sin|m|\varphi & \text{for } m < 0 \end{cases} \tag{44}$$

where $\hat{x} = (\cos\varphi \sin\theta, \sin\varphi \sin\theta, \cos\theta)$ and $P_{lm}(z)$ is an associated Legendre polynomial.[36] To include this boundary condition in the solution to Poisson equation the potential is broken up into two parts $V_{\mathrm{H}}(\vec{x}) = u(\vec{x}) + u_D(\vec{x})$, where the homogeneous part $u(\vec{x})$ is defined to be zero on the boundary, and the boundary part, $u_D(\vec{x})$, is the finite element expansion of eq 41. The solution to the homogeneous expansion coefficients, u_n, for the Hartree potential in eq 38) is then found by solving the following systems of linear equations,

$$A_{mn} u_n = f_n \tag{45}$$

where

$$A_{mn} = \sum_{l=1}^{L} \sum_{r=1}^{T_l} \sum_{s=1}^{T_l} \delta_{m,\tilde{m}(r,l)} \delta_{n,\tilde{m}(s,l)} (-T_{rs}^{e_l}) \tag{46}$$

and

$$f_n = \sum_{l=1}^{L} \sum_{r=1}^{T_l} \delta_{m,\tilde{m}(r,l)} \left(-4\pi G_r^{e_l} \left(\sum_{i=1}^{N_e/2} |\psi_i(\vec{x})|^2 \right) + D_r^{e_l}(u_D(\vec{x})) \right) \tag{47}$$

In this work, a standard preconditioned conjugate gradient solver was used to solve these equations.

Similarly, substituting eq 39 into eqs 32 and 33 produces the following generalized eigenvalue problem

$$H_{mn} c_n^i = \varepsilon_i S_{mn} c_n^i \tag{48}$$

and orthonormality conditions

$$\sum_{m=1}^{M} \sum_{n=1}^{M} c_m^i S_{mn} c_n^j = \delta_{ij} \tag{49}$$

where

$$S_{mn} = \int_{\Omega} \eta_m(\vec{x}) \eta_n(\vec{x})\, \mathrm{d}\vec{x} \tag{50}$$

$$H_{mn} = \int_{\Omega} \eta_m(\vec{x}) H \eta_n(\vec{x})\, \mathrm{d}\vec{x} \tag{51}$$

A formula for the matrix S_{mn} in terms of eq 26 can be readily be obtained by substituting eq 7 into eq 50.

$$S_{mn} = \int_{\Omega} \{ \sum_{l=1}^{L} \sum_{t=1}^{T} \phi_t^{e_l}(\vec{x}) \delta_{m,\tilde{m}(t,l)} \} \{ \sum_{k=1}^{L} \sum_{s=1}^{T} \phi_t^{e_l}(\vec{x}) \delta_{n,\tilde{m}(s,k)} \}\, \mathrm{d}\vec{x} \tag{52}$$

$$\begin{aligned} S_{mn} &= \int_{\Omega} \left\{ \sum_{l=1}^{L} \sum_{r=1}^{T_l} \phi_r^{e_l}(\vec{x}) \delta_{m,\tilde{m}(t,l)} \right\} \left\{ \sum_{k=1}^{L} \sum_{s=1}^{T_k} \phi_s^{e_k}(\vec{x}) \delta_{n,\tilde{m}(s,k)} \right\} \mathrm{d}\vec{x} \\ &= \sum_{l=1}^{L} \sum_{k=1}^{L} \sum_{r=1}^{T_l} \sum_{s=1}^{T} \delta_{m,\tilde{m}(r,l)} \delta_{n,\tilde{m}(s,k)} \int_{\Omega} \phi_r^{e_l}(\vec{x}) \phi_s^{e_k}(\vec{x})\, \mathrm{d}\vec{x} \\ &= \sum_{l=1}^{L} \sum_{k=1}^{L} \sum_{r=1}^{T_l} \sum_{s=1}^{T_k} \delta_{m,\tilde{m}(r,l)} \delta_{n,\tilde{m}(s,k)} \delta_{k,l} \int_{e_l} \phi_r^{e_l}(\vec{x}) \phi_s^{e_l}(\vec{x})\, \mathrm{d}\vec{x} \\ &= \sum_{l=1}^{L} \sum_{r=1}^{T_l} \sum_{s=1}^{T} \delta_{m,\tilde{m}(r,l)} \delta_{n,\tilde{m}(s,l)} \int_{e_l} \phi_r^{e_l}(\vec{x}) \phi_s^{e_l}(\vec{x})\, \mathrm{d}\vec{x} \\ &= \sum_{l=1}^{L} \sum_{r=1}^{T} \sum_{s=1}^{T_l} \delta_{m,\tilde{m}(r,l)} \delta_{n,\tilde{m}(s,l)} M_{rs}^{e_l} \end{aligned} \tag{53}$$

Similarly, a formula for the matrix H_{mn} in terms of eqs 30, 27, and 29 can also be obtained.

$$\begin{aligned} H_{mn} = \sum_{l=1}^{L} \sum_{r=1}^{T_l} \sum_{s=1}^{T_l} \delta_{m,\tilde{m}(r,l)} \delta_{n,\tilde{m}(s,l)} \Bigg(& \frac{1}{2} T_{rs}^{e_l} + K_{rs}^{e_l}(V_{\mathrm{ext}}) + \\ & K_{rs}^{e_l}(V_{\mathrm{H}}) + R_{rs}^{e_l}(V_{\mathrm{xc}}, \sum_{i=1}^{N_e/2} |\psi_i(\vec{x})|^2) \Bigg) \end{aligned} \tag{54}$$

Standard preconditioned eigenvalue solvers used in plane-wave DFT programs were used to solve the generalized eigenvalue equations.[20,37] However, as pointed out by Kohn et al., one potential problem in solving the Kohn−Sham equations with adaptivity is that the condition number of the discrete Kohn−Sham equations is dependent on the number of levels of refinement and that as many as two times the iterations will be needed for each new level of refinement.[19] In this work, we took a very simple approach to this problem. We first solved the equations at a coarse level of refinement and then projected it down for use as input at a finer level of refinement.

4. Adaptive FE DFT Solutions of Atoms and Molecules

The Kohn−Sham DFT equations contain several length scales because of the steepness of the atomic potentials. It is well-known that uniform FE meshes are not very efficient for these types of problems. Ideally, an FE mesh could be

adaptively refined only in the regions near the atom centers. However, in general, generating adaptive meshes of good quality is a difficult problem. Straightforward adaptive refinement procedures usually result in "nonconforming" meshes or meshes with hanging nodes. A globally "conforming" FE mesh is defined as a collection of elements which meet only at vertices and faces. While it is possible to develop a FE method based on nonconforming meshes, in general, FE meshes need to be conforming to ensure continuity of interpolated functions.[30,33] A basic algorithm to refine an existing conforming mesh is as follows. In the first step, the elements that have been selected for refinement are bisected. This step more then likely will produce a nonconforming mesh. The next step in the algorithm is then to mark for refinement the elements which contain hanging nodes. These steps proceed interactively until a conforming mesh is produced.[38,39] Many variants on this basic algorithm are possible. For example, the bisection could be along the longest edge or the newest vertex. In this work we used the conforming adaptive mesh refinement based on longest edge bisection. The exact algorithm used in our calculations is given in Scheme 1.

The adaptive FE solver was tested initially on the hydrogen-like atom. The Hamiltonian for this test problem has a deceptively simple form with only a single potential term.

$$H = -\frac{1}{2}\nabla^2 - \frac{Z}{|\vec{x}|} \tag{55}$$

The solutions to this eigenvalue problem are well-known and analytical solutions are available. However, the singular behavior at the origin can cause significant problems for numerical methods. In the case of the FE solver, a mesh vertex must be at the atom center (origin) in order for the Hamiltonian matrix of the FE solver not to contain a singularity in any of its elements. The lowest energy solution and energy are $\psi(\vec{x}) = (Z^{3/2}/\sqrt{\pi})\exp(-Z|\vec{x}|)$ and $\varepsilon = -Z^2/2$. Note that the severity of the singularity with increasing Z is reflected in the increasing localization of the solution.

Scheme 1: Conforming Adaptive FE Mesh Generation Algorithm.

1. Estimate the error $\gamma(e_l)$ for each element e_l in the FE mesh using the following formula

$$\gamma(e_l) = \left(\frac{1}{2}\{\xi(\vec{x}_2) + \xi(\vec{x}_3) + \xi(\vec{x}_4) - \xi(\vec{x}_1)\} - \xi(\vec{x}_c)\right)\frac{|F^{e_l}|}{6} \tag{56}$$

where $\vec{x}$, $\vec{x}_2$, $\vec{x}_3$, and $\vec{x}_4$ are the four vertices of the tetrahedral element e_l, $\vec{x}_c$ is its geometric center, and $\xi(\vec{x})$ is a user defined weight function having dimensions of density in atomic units.

2. Set refinement queues $Q_1 = Q_2 = \varnothing$.

3. Place elements with large errors ($\gamma(e_l) > \varepsilon$) in the refinement queue Q_1.

4. If $Q_1 = \varnothing$, then go to step 9.

5. If $Q_1 > \varnothing$, then proceed to step 6; otherwise, go to step 1.

6. Bisect the elements in Q_1 (removing from Q_1) using either $q-q$ tetrahedral bisection or $b-b$ bisection using the longest edge as shown in Figure 3 and place the nonconforming elements created in refinement queue Q_2.

7. Set $Q_1 \leftarrow Q_2$.

8. Go to step 5.

9. Done with refinement.

To define the adaptive FE mesh for this problem, local adaptivity was carried out starting from a uniform mesh using the geometric-based refinement strategy given in Scheme 1 with the following weight (atomic density) function

$$\xi(\vec{x}) = \frac{Z^3}{\pi}\exp(-2Z|\vec{x}|) \tag{57}$$

The initial uniform mesh used was generated by uniformly refining a seven element tetrahedral mesh four times using $q-q$ refinement (as shown in Figure 3) with the boundary vertices set to be at a radius of 10 au, resulting in mesh composed of 32 768 finite elements with 6017 vertices. For a given γ, the number of elements generated by refinement procedure was found to be nearly independent of Z. Furthermore, the number of vertices grows very rapidly for small tolerances, since at the lowest tolerance $\gamma = 10^{-4}$ there are approximately 11 000 vertices in the FE mesh, whereas at $\gamma = 10^{-7}$ there are approximately 220 000 vertices in the FE mesh. The number of vertices as a function of γ was found to be approximately,

$$v(\gamma) \approx 200\exp(-\log_{10}(\gamma)) \tag{58}$$

where v is the number of vertices.

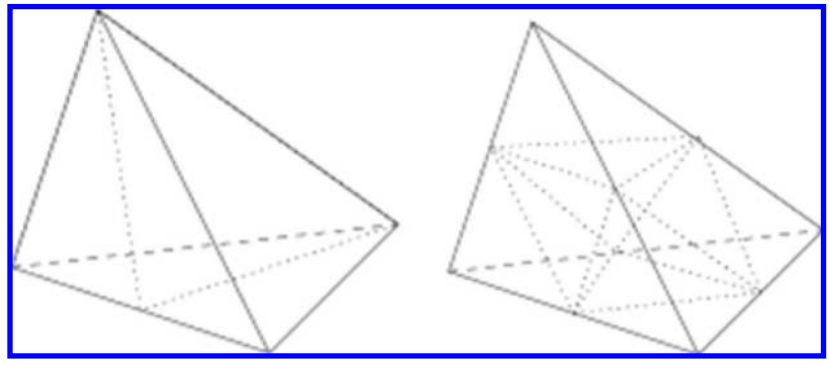

Figure 3. $b-b$ tetrahedral bisection (left) and $q-q$ tetrahedral bisection (right).

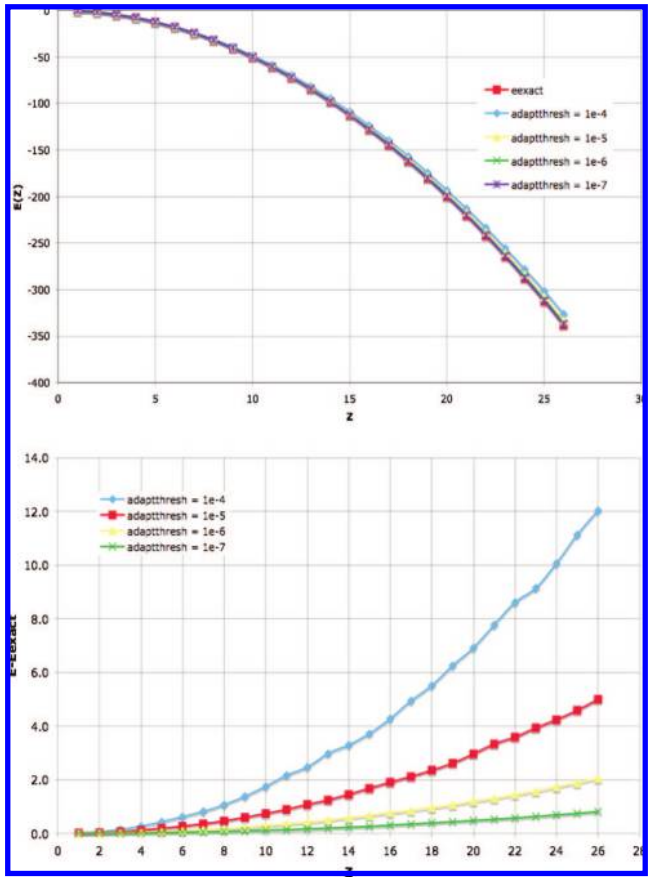

Figure 4. Eigenvalues and errors for the hydrogen-like atom as a function Z.

Not surprisingly, the placement of the elements was found to very dependent on Z. In particular, for higher Z the majority of elements were found to be very close to the atom center, while for lower Z the majority of elements were found to be located away from the atom center. For example, at a $\gamma = 10^{-7}$ refinement, the number of verticies within 1.0 au of the atom center was found to be 32 011 and 214 543 repectively for $Z = 1$ and $Z = 26$, while the number of elements outside this radius was found to be 162 103 and 6212 respectively. Furthermore, it was found that increasing the refinement threshold only increased the number of elements very near the atom centers, while the number of elements away from the atom center remained nearly constant.

In Figure 4, the lowest eigenvalues and errors of the hydrogen-like atoms ($Z = 1...26$) are shown at increasing levels of refinement. Not surprisingly, the accuracy of the solution improved significantly when adaptive refinement was applied. For a given refinement tolerance, the error grew quadratically as a function of Z. For $Z = 1$, the errors were found to be 0.0150 au ($\gamma = 10^{-4}$), 0.0069 au ($\gamma = 10^{-5}$), 0.0029 au ($\gamma = 10^{-6}$), and 0.0012 au ($\gamma = 10^{-7}$). When the singularity was strengthened, the errors were considerably larger. For $Z = 26$, the errors were found to be 12.0348 au ($\gamma = 10^{-4}$), 4.9934 au ($\gamma = 10^{-5}$), 2.0517 au ($\gamma = 10^{-6}$), and 0.08194 au ($\gamma = 10^{-7}$). Even though accurate solutions can be obtained with the current adaptive FE solver based on piecewise linear elements, extremely small adaptive tolerances (large FE meshes) will be required. On the basis of least-squares fitting, the error in terms of γ and Z was found to be approximately given by the following relation in atomic units.

$$\varepsilon(\gamma, Z) \approx 0.64423\exp(0.89503\log_{10}(\gamma))Z^2 \qquad (59)$$

Using eqs 58 and 59, one can estimate the number of vertices needed to obtain accuracies in the millihartree range.

$$\upsilon(\varepsilon, Z) \approx 122.37\frac{Z^{2.2346}}{\varepsilon^{1.1173}} \qquad (60)$$

For example, for $Z = 26$ and $\gamma = 10^{-14}$, the error and number of vertices needed are $\varepsilon \approx 0.0015$ au and $\upsilon \approx 240, 520, 857$. Hence, in order for the current adaptive FE solver to obtain chemical accuracies for molecules containing atoms with modest Z, the memory requirements are expected to be quite large (>10 Gb/atom). Given that the number of floating point operations per minimization step for a system of N_A atoms will be $O(N_A{}^3\upsilon(\varepsilon, Z))$ the overall memory requirement and computational cost of a simulation can be estimated. For example, to calculate 100 Fe atoms ($Z = 26$) at an accuracy of $\varepsilon = 0.0015$ au will require on the order of 2 terabytes at a cost of 25 petaflops per step.

The next test cases for the adaptive FE solver was the H, He, Li, and Ne atoms at the DFT level using the local density approximation (LDA) exchange-correlation functional.[40] Since the solutions to these equations are spherically symmetric, the accuracy of these FE DFT solutions can be checked by comparing them to solutions of the 1D-radial Kohn−Sham equation.

$$\left(-\frac{1}{2}\frac{d^2}{dr^2} + \frac{l(l+1)}{2r^2} - \frac{Z}{r} + 4\pi\int\frac{\rho(r')}{|r-r'|}r'^2\,dr' + V_{xc}(r)\right)\psi_{il}(r) = \varepsilon_{il}\psi_{il}(r) \qquad (61)$$

Table 2. LDA Energies and Errors in Atomic Units of H, He, Li, and Ne at Increasing Levels of Refinement

refinement	E_{LDA} (H)	error (H)	E_{LDA} (He)	error (He)	E_{LDA} (Li)	error (Li)	E_{LDA} (Ne)	error (Ne)
uniform	−0.438492	4.02E−02	−2.383364	4.51E−01	−5.624508	1.72E+00	−75.145028	5.31E+01
γ = 1e-2	−0.441826	3.68E−02	−2.602388	2.32E−01	−6.674924	6.69E−01	−117.391255	1.08E+01
γ = 1e-3	−0.456391	2.23E−02	−2.731748	1.03E−01	−7.031348	3.13E−01	−123.228792	5.00E+00
γ = 1e-4	−0.468855	9.82E−03	−2.789752	4.51E−02	−7.230246	1.14E−01	−126.047943	2.19E+00
γ = 1e-5	−0.474739	3.93E−03	−2.817929	1.69E−02	−7.294580	4.93E−02	−127.367719	8.66E−01
γ = 1e-6	−0.477234	1.44E−03	−2.830797	4.04E−03	−7.323754	2.01E−02	−127.917138	3.16E−01
γ = 1e-7	−0.477879	7.91E−04	−2.833847	9.89E−04	−7.333440	1.04E−02	*a*	*a*
γ = 1e-8	−0.478194	4.77E−04	*a*	*a*	*a*	*a*	*a*	*a*

[a] Required more than 2 GB of memory.

This 1D-radial equation was solved with a Hermann−Skilman telescoping grid and an Adams fifth-order predictor-correction method.[7] From solving this simplified equation, the exact LDA energies for H, He, Li, and Ne were found to be −0.47867, −2.83484, −7.34386, and −128.2335 au, respectively. For the adaptive FE solutions, the initial uniform FE mesh and the adaptive FE meshes were generated in the same way as the hydrogen-like atoms above, except that the weight functions were taken to be the all electron densities obtained from the solutions to the 1D-radial Kohn−Sham equation.

The LDA energies and errors for H, He, Li, and Ne are reported in Table 2. As expected, the accuracy of the solutions significantly improved when adaptive mesh refinement was applied. Just as for the non-self-consistent hydrogen-like problem, the error grew quadratically as a function of Z for a given refinement tolerance. The errors were also found to be of the same order as with the non-self-consistent hydrogen-like problem, confirming that the essential difficulties of the Kohn−Sham eigenvalues equation are the result of the singular behavior of the atomic potentials.

The next test case for the adaptive FE solver was for the simplest molecule, H_2^+. This problem is very similar to the hydrogen atom in that there is only one electron; however, unlike the hydrogen atom, there are now two centers with singularities located at $\vec{R}_1$ and $\vec{R}_2$. The Hamiltonian for this molecule is,

$$H = -\frac{1}{2}\nabla^2 - \frac{Z_1}{|\vec{x} - \vec{R}_1|} - \frac{Z_2}{|\vec{x} - \vec{R}_2|} \tag{62}$$

where $Z_1 = Z_2 = 1$. Having more than one center complicates the FE mesh generation considerably. To define the adaptive FE mesh for this problem, the geometric-based refinement strategy given in Scheme 1 was used with the following weight function,

$$\xi(\vec{x}) = \frac{Z_1^3}{\pi}\exp(-2Z_1|\vec{x} - \vec{R}_1|) + \frac{Z_2^3}{\pi}\exp(-2Z_2|\vec{x} - \vec{R}_2|) \tag{63}$$

The singularities at the ion centers were accommodated by modifying the initial uniform mesh (6017 vertices, $R = 10a_0$) by moving the vertex nearest to each ion center to lie on top of it. The adaptive solver produced solutions that were similar in accuracy to the hydrogen atom. In Figure 5, the binding energy curve for increasing levels of refinement is shown. The binding energy of H_2^+ at a distance $|\vec{R}_1 - \vec{R}_2|$ is defined as the total energy of molecule at this distance minus the energy of the molecule at infinite separation. Even though large errors are seen with the uniform mesh, the agreement with the analytic result with, even low levels of, adaptive refinement is remarkably good, producing smooth binding energy curves.

As a final test case for the adaptive FE solver, we chose to calculate the binding energy curve for Li_2. This seemingly simple molecule is difficult to calculate. The ground-state solution has three molecular orbitals ($1\sigma_g$, $1\sigma_u$, and $2\sigma_g$) shown in Figure 6. The bottom two molecular orbitals are very localized on the atoms. The top molecular orbital is

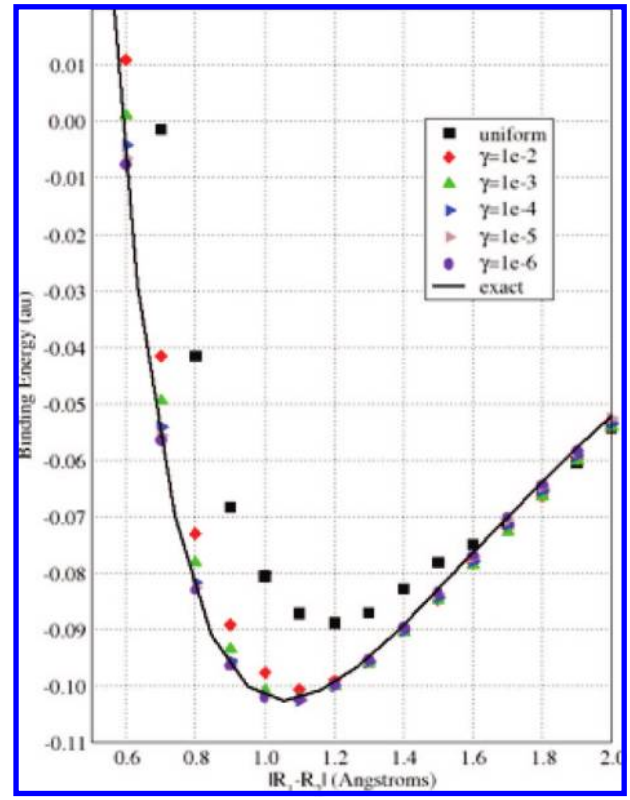

Figure 5. Binding energy curves for H_2^+ obtained with adaptive gridding defined by the geometric-based refinement strategy.

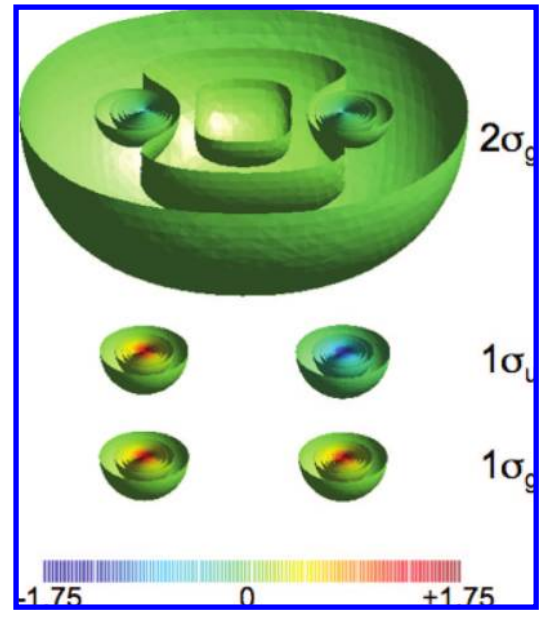

Figure 6. Ground-state molecular orbitals ($1\sigma_g$, $1\sigma_u$, and $2\sigma_g$) of LDA for Li_2 obtained with the FE DFT solver.

considerably more delocalized, but it also contains a localized part. To define the adaptive FE mesh for this problem, the geometric-based refinement strategy given in Scheme 1 was used with the following weight function,

$$\xi(\vec{x}) = \rho^{\mathrm{LDA}}_{\mathrm{Li\ atom}}(|\vec{x} - \vec{R}_1|) + \rho^{\mathrm{LDA}}_{\mathrm{Li\ atom}}(|\vec{x} - \vec{R}_2|)$$

where, $\rho^{\mathrm{LDA}}_{\mathrm{Li\ atom}}(r)$ was obtained by using a spline fit of the solution to eq 61 for the Li atom. The same initial mesh as

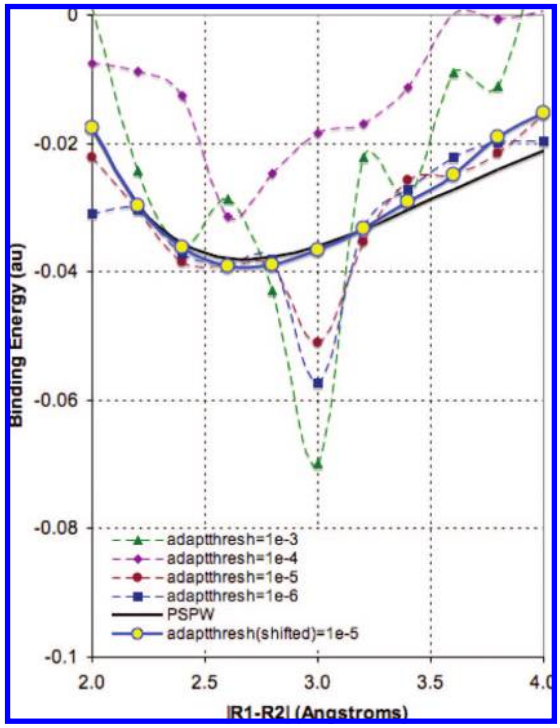

Figure 7. Binding energy curves for Li_2 obtained with adaptive gridding defined by the geometric-based refinement strategy. The "adaptthresh(shifted)=1e-5" curve was obtained with a $\gamma = 1e-5$ adaptive mesh where the initial mesh was modified by shifting procedure of eq 64. The PSPW curve, shown for comparison, was obtained with a NWChem pseudopotential planewave calculation.[4]

for the H_2^+ molecule was used, and vertices nearest to each ion center were moved to lie on top of them.

In Figure 7, the binding energy curve for increasing levels of refinement is shown. Even though a strategy very similar to what was used for the H_2^+ molecule was used, very large errors in the binding energy curves are seen with adaptive refinement at $\gamma = 1e-6$. In analyzing the solution, it was found that the majority of error was from the eigenvalues of the $1\sigma_g$ and $1\sigma_u$ molecular orbitals. Since these orbitals are very localized on the atoms, their eigenvalues are expected to be nearly constant as a function of $|\vec{R}_1 - \vec{R}_2|$. However, it was found that their eigenvalues fluctuated by as much as 0.1 au for the uniform mesh down to 0.01 au for the $\gamma = 1e-6$ mesh. While these errors are slightly smaller than the absolute errors seen for the Li atom in Table 2, they are still too large relative to the LDA binding energy of Li_2, which is roughly 0.04 au.

It was found that the errors in the binding energy for Li_2 could be reduced further by shifting a cloud of vertices near the ion center instead of just a single vertex nearest to each ion center ($\vec{v}_{\text{nearest}}$). To do this, for each ion each of the vertices in the mesh ($\vec{v}_i$) were moved by

$$\vec{v}_i = \vec{v}_i + (\vec{R}_I - \vec{v}_{\text{nearest}}) f(|\vec{v}_i - \vec{v}_{\text{nearest}}|) \qquad (64)$$

where $f(r)$ is the screening function

$$f(r) = 1 - \left[1 - \exp\left(-\left(\frac{r}{R}\right)^N\right)\right]^N \qquad (65)$$

and N and R are adjustable parameters, chosen to be 8 and 1.5 au, respectively, which define the atom center region. When this initial shifting procedure is used, it was found that an accurate binding energy curve was obtained by the $\gamma = 1e-5$ adaptive mesh (adaptthresh(shifted)=1e-5 curve in Figure 7). This result suggests that the placement of the mesh close to the atom centers is the main source of error, and an overlapping grid method can be used to reduce the errors (cancelation of errors) in structure and bond energies of the system. We note that carefully choosing mesh close to atom centers in order to reduce integration errors was also used by Batcho.[24] In this work, the mesh was generated by partitioning the volume around each singularity with a cube that was subdivided into six pyramids.

5. Conclusion

We have implemented an unstructured adaptive FE DFT program. The severe problem associated with the rapid variation of the electronic wave functions in the near singular regions of the atomic centers was treated by using unstructured simplex meshes that resolve these features around atomic nuclei. This approach uses a minimal amount of computational resources by concentrating the computational work in the regions in which the shortest length scales are necessary and provides for low resolution in regions for which there is no electron density. The matrix representations of the discrete Hamiltonian operator in the adaptive finite element basis are always sparse due to the local support nature of finite element basis functions. As a result, application of the Hamiltonian operator is $O(N)$ in the number of discretization points.

The overall memory and computational requirements for the solver implemented were found to be quite high. By using the solution to the hydrogen-like atom, the overall memory and computational requirements per atom needed by the solver were estimated. The number of mesh vertices per atom as a function of the atomic number Z and the required accuracy ε was estimated to be $v(\varepsilon, Z) \approx 122.37(Z^{2.2346}/\varepsilon^{1.1173})$. These meshing requirements were also found to hold for the full DFT solutions. The errors in the LDA energies of H, He, Li, and Ne were found to be of the same order as the hydrogen-like atom, which confirmed that the essential difficulty of solving the Kohn−Sham eigenvalue equation is the result of the singular behavior of the atomic potentials. This estimate can be used determine the overall memory requirement and computational cost of a simulation, since the number of floating point operations per DFT minimization step for a system of N_A atoms will be $O(N_A^3 v(\varepsilon, Z))$ (e.g., $Z = 26$, $\varepsilon = 0.0015$ au, and $N_A = 100$, the memory requirement and computational cost would be ~2 terabytes and ~25 petaflops).

Despite the high cost of the method, it was found that strategies for fixing the error near the atomic potential singularities, such as a geometric-based refinement strategy can be used to reduce the errors in structure and bond energies of the system. In this work, to define the adaptive FE mesh for a problem, local adaptivity was carried out by starting from an uniform mesh and adapting using a conforming adaptation procedure where the error was

determined by using a weight function composed of the sum of the atomic densities for the problem. For the simple H_2^+ molecule this strategy was found to work very well. However, for the Li_2 molecule very large errors in the binding energy curves were seen even when the geometric-based adaptive refinement procedure was used. It was found that the errors in the binding energy for Li_2 could be reduced further by shifting a cloud of vertices near the ion center instead of just a single vertex nearest to each ion center ($\vec{v}_{\text{nearest}}$) in the initial uniform mesh. When this initial shifting procedure was used in combination with the geometric-based adaptation procedure, it was found that an accurate binding energy curve could be obtained. These results showed that the placement of the mesh close to the atom centers is the main source of error in the method, and it suggests that an overlapping grid method could be used to reduce the errors ("cancellation of errors") in structure and bond energies of the molecule.

At present, our adaptive FE DFT solver uses piecewise linear elements which are $O(h^2)$ accurate. It was shown at this low order of accuracy that very large FE meshes will be needed to obtain the millihartree or better accuracy desired for molecules and materials with large Z atoms. Unless very large machines are used, the memory requirements (and computational cost) is unlikely to be competitive with more standard solution methods. Others have shown that memory and computational requirements of FE DFT can be substantially reduced by using higher-order FE elements[19,20] or spectral element methods.[21−24] However, to date these methods have relied on using parallelpiped elements along with a special treatement of the singularity by pseudopotentials[19,20] or with a special integration technique.[22,24] Our current work differs from these prior works in that we use tetrahedral elements rather than parallelpiped elements. This allows for considerably more flexibility in the grid generations as opposed to parallelpiped elements. Future work will focus on higher order FE DFT solvers which make use of tetrahedral elements.

Acknowledgment. This research was supported by the DOE ASCR Multiscale Mathematics program and the DOE BES Geosciences program. The Pacific Northwest National Laboratory (PNNL) is operated by Battelle Memorial Institute. Some of the calculations were performed on the MPP2 computing system at the Molecular Science Computing Facility in the William R. Wiley Environmental Molecular Sciences Laboratory (EMSL) at PNNL. EMSL is a national scientific user facility sponsored by the Department of Energy's Office of Biological and Environmental Research and located at Pacific Northwest National Laboratory. We also wish to thank the Scientific Computing Staff, Office of Energy Research, and the U.S. Department of Energy for a grant of computer time at the National Energy Research Scientific Computing Center (Berkeley, CA).

References

(1) Kohn, W.; Sham, L. J. Self-consistent equations including exchange and correlation effects. *Phys. Rev.* **1965**, *140*, A1133–A1138.

(2) Parr, R. G.; Yang, W. G. *Density-Functional Theory of Atoms and Molecules*; Oxford University Press: New York, 1989.

(3) Apr, A.; Gao, F.; Kristić, P. S.; Wells, J. C.; Windus, T. L. NWChem for Material Science. *Comput. Mater. Sci.* **2003**, *28*, 209–221.

(4) Bylaska, E. J.; de Jong, W. A.; Govind, N.; Kowalski, K.; Straatsma, T. P.; Valiev, M.; Wang, D.; Apra, E.; Windus, T. L.; Hammond, J.; Nichols, P.; Hirata, S.; Hackler, M. T.; Zhao, Y.; Fan, P.-D.; Harrison, R. J.; Dupuis, M.; Smith, D. M. A.; Nieplocha, J.; Tipparaju, V.; Krishnan, M.; Wu, Q.; Van Voorhis, T.; Auer, A. A.; Nooijen, M.; Brown, E.; Cisneros, G.; Fann, G. I.; Fruchtl, H.; Garza, J.; Hirao, K.; Kendall, R.; Nichols, J. A.; Tsemekhman, K.; Wolinski, K.; Anchell, J.; Bernholdt, D.; Borowski, P.; Clark, T.; Clerc, D.; Dachsel, H.; Deegan, M.; Dyall, K.; Elwood, D.; Glendening, E.; Gutowski, M.; Hess, A.; Jaffe, J.; Johnson, B.; Ju, J.; Kobayashi, R.; Kutteh, R.; Lin, Z.; Littlefield, R.; Long, X.; Meng, B.; Nakajima, T.; Niu, S.; Pollack, L.; Rosing, M.; Sandrone, G.; Stave, M.; Taylor, H.; Thomas, G.; van Lenthe, J.; Wong, A.; Zhang, Z. *NWChem, A Computational Chemistry Package for Parallel Computers*;, Version 5.1, Pacific Northwest National Laboratory: Richland, WA, 2007.

(5) Dunning, T. H., Jr.; Peterson, K. A.; Woon, D. E. Gaussian Basis Sets for Use in Correlated Calculations. In *Encyclopedia of Computational Chemistry*; Schleyer, P.v.R., Ed.; John Wiley & Sons Ltd: New York, 1997.

(6) Dunlap, B. I.; Connolly, J. W. D.; Sabin, J. R. On some approximations in applications of X theory. *J. Chem. Phys.* **1979**, *71*, 3396–3402.

(7) Hamann, D. R. Generalized norm-conserving pseudopotentials. *Phys. Rev. B* **1989**, *40*, 2980–2987.

(8) Tsuchida, E.; Tsukada, M. Electronic-structure calculations based on the finite-element method. *Phys. Rev. B* **1995**, *52*, 5573–5578.

(9) Bernholc, J.; Briggs, E. I.; Sullivan, D. J.; Brabec, C. J.; Nardelli, M. B.; Rapcewicz, K.; Roland, C.; Wensell, M. Real-space multigrid methods for large-scale electronic structure problems. *Int. J. Quantum Chem.* **1997**, *65*, 531–543.

(10) Bernholc, J.; Hodak, M.; Lu, W. C. Recent developments and applications of the real-space multigrid method. *J. Phys.: Condensed Matter* **2008**, *20*, 294205.

(11) Chelikowsky, J. R.; Troullier, N.; Saad, Y. Finite-Difference-Pseudopotential Method: Electronic Structure Calculations without a Basis. *Phys. Rev. Lett.* **1994**, *72*, 1240–1243.

(12) Chelikowsky, J. R.; Troullier, N.; Wu, K.; Saad, Y. Higher-order finiet-difference pseudopotential method: An application to diatomic molecules. *Phys. Rev. B* **1994**, *50*, 11355–11364.

(13) Briggs, E. I.; Sullivan, D. J.; Bernholc, J. Large-scale electronic-structure calculations with multigrid acceleration. *Phys. Rev. B* **1995**, *52*, R5471–R5474.

(14) Pask, J. E.; Klein, B. M.; Sterne, P. A.; Fong, C. Y. Finite-element methods in electronic-structure theory. *Comput. Phys. Commun.* **2001**, *135*, 1–34.

(15) Cho, K.; Arias, T. A.; Joannopoulos, J. D.; Lam, P. K. Wavelets in Electronic Structure Calculations. *Phys. Rev. Lett.* **1993**, *71*, 1808–1811.

(16) Lippert, R. A.; Arias, T. A.; Edelman, A. Multiscale Computation with Interpolating Wavelets. *J. Comput. Phys.* **1998**, *140*, 278–310.

(17) Arias, T. A. Multiresolution analysis of electronic structure: semicardinal and wavelet bases. *Rev. Mod. Phys.* **1999**, *71*, 267–311.

(18) Bylaska, E. J.; Kohn, S. R.; Baden, S. B.; Edelman, A.; Kawai, R.; Ong, M. E. G.; Weare, J. H.; Scalable Parallel Numerical Methods and Software Tools for Material Design. In *Proceeding of the Seventh SIAM Conference on Parallel Processing for Scientific Computing*, San Francisco, CA, 1995.

(19) Kohn, S.; Weare, J.; Ong, E.; Baden, S. Parallel Adaptive Mesh Refinement for Electronic Structure Calculations. In *Eigth SIAM Conference on Parallel Processing for Scientific Computing*, Minneapolis, MN, 1997.

(20) Fattebert, J. L.; Hornung, R. D.; Wissink, A. M. Finite element approach for density functional theory. *J. Comput. Phys.* **2007**, *223*, 759–773.

(21) Harrison, R. J.; Fann, G. I.; Yanai, T. G.; Gan, Z.; Beylkin, G. Multiresolution quantum chemistry: Basic theory and initial applications. *J. Chem. Phys.* **2004**, *121*, 11587–11598.

(22) Yanai, T.; Fann, G. I.; Gan, Z. T.; Harrison, R. J.; Beylkin, G. Multiresolution quantum chemistry in multiwavelet bases: Hartree-Fock exchange. *J. Chem. Phys.* **2004**, *121*, 6680–6688.

(23) Yanai, T.; Fann, G. I.; Gan, Z. T.; Harrison, R. J.; Beylkin, G. Multiresolution quantum chemistry in multiwavelet bases: Analytic derivatives for Hartree-Fock and density functional theory. *J. Chem. Phys.* **2004**, *121*, 2866–2876.

(24) Batcho, P. F. Computational method for general multicenter electronic structure calculations. *Phys. Rev. E* **2000**, *61*, 7169–7183.

(25) Modine, N. A.; Zumback, G.; Kaxiras, E. Adaptive-coordinate real-space electronic-structure calculations for atoms, molecules and solids. *Phys. Rev. B* **1997**, *55*, 10289–10301.

(26) Torsti, T.; Lindberg, V.; Makkonen, I.; Ogando, E.; Rasanen, E.; Saarikoski, H.; Puska, M. J.; Nieminen, R. M. Real-space electronic-property calculations for nanoscale structures. In *Handbook of Theoretical and Computational Nanotechnology*; Rieth, M., Schommers, W., Eds.; Forschungszentrum Karlsruhe: Germany, 2006.

(27) Modine, N. A.; Zumbach, G.; Kaxiras, E. Adaptive coordinate real-space electronic structure calculations for atoms, molecules and solids. *Phys. Rev. B* **1997**, *55*, 10289–10301.

(28) Holst, M. Adaptive numerical treatment of elliptic systems on mainfolds. *Adv. Comput. Math.* **2001**, *15*, 139–191.

(29) Axelsson, O.; Barker, V. A. *Finite Element Solution of Boundary Value Problems: Theory and Computation*; SIAM: Philadelphia, 2001.

(30) Braess, D. *Finite Elements: Theory, Fast Solvers and Applications in Solid Mechanics*, 2nd ed.; Cambridge Univeristy Press: Cambridge 2005.

(31) Brenner, S. C.; Scott, L. R., *The Mathematical Theory of Finite Element Methods*, 2nd ed.; Springer-Verlag: New York, 2002.

(32) Strang, G. Piecewise Polynomials and the Finite Element Method. *Bull. Am. Math. Soc.* **1973**, *79*, 1128–1137.

(33) Norrie, D. H.; de Vries, G. *The Finite Element Method*; Academic Press: New York, 1973.

(34) Zienkiewicz, O. C.; Morgan, M. *Finite Elements and Approximation*, Dover Publications: New York, 1983.

(35) Dirac, P. A. M. Note on exchange phenomena in the Thomas atom. *Proc. Cambridge Philos. Soc.* **1930**, *26*, 376–385.

(36) Sansone, G. *Orthogonal Functions*, revised English ed.; Dover Publications: New York, 1991.

(37) Hasnip, P. J.; Pickard, C. J. Electronic energy minimisation with ultrasoft pseudpotentials. *Comput. Phys. Commun.* **2006**, *174*, 24–29.

(38) Bank, R. E.; Holst, M. A New Paradigm for Parallel Adaptive Meshing Algorithms. *Siam J. Sci. Comput.* **2000**, *22*, 1411–1443.

(39) Arnold, D. N.; Mukherjee, A.; Pouly, L. Locally Adapted Tetrahedral Meshes using Bisection. *Siam, J. Sci. Comput.,* **2000**, *22*, 431–448.

(40) Vosko, S. H.; Wilk, L.; Nusair, M. Accurate Spin-Dependent Electron Liquid Correlation Energies for Local Spin-Density Calculations - A Critical Analysis. *Can. J. Phys.* **1980**, *58*, 1200–1211.

CT800350J