

# Adaptive Finite Element Method for Solving the Exact Schrödinger Equation of Density Functional Theory: Is it Computable?

Eric J. Bylaska\*

*Environmental Molecular Sciences Laboratory,  
Pacific Northwest National Laboratory,  
P.O. Box 999, Richland, Washington 99352*

Mike Holst<sup>†</sup>

*Department of Mathematics, University of California,  
San Diego, La Jolla, California 92093*

John H. Weare<sup>‡</sup>

*Department of Chemistry and Biochemistry,  
University of California, San Diego, La Jolla, California 92093*

## Abstract

Results of the application of an adaptive finite element (FE) based solution using the FETK library of M. Holst to 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  $\epsilon$  (in atomic units) was estimated to be  $v(\epsilon, Z) \approx 122.37 * \frac{Z^{2.2346}}{\epsilon^{1.1173}}$ , and the number of floating point operations per minimization step for a system of  $N_A$  atoms was found to be  $O(N_A^3 * v(\epsilon, Z))$  (e.g.  $Z = 26$ ,  $\epsilon = 0.0015$  au, and  $N_A=100$ , the memory requirement and computational cost would be  $\sim 0.2$  terabytes and  $\sim 25$  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.

---

\*Electronic address: [eric.bylaska@pnl.gov](mailto:eric.bylaska@pnl.gov)

†Electronic address: [mholst@ucsd.edu](mailto:mholst@ucsd.edu)

‡Electronic address: [jweare@ucsd.edu](mailto:jweare@ucsd.edu)

## I. INTRODUCTION

Kohn-Sham Density Functional Theory (DFT) [32] has become a state of the art tool in the calculation of the properties of solid state and molecular systems, predicting structures, properties, and reactivity of a wide variety of systems. In many cases it can achieve chemical accuracy at a smaller cost than other ab initio techniques. 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 adequate for many cases [1, 2], there are limitations to the much wider application of these approaches to the even more demanding systems encountered in complex technology problems. The most important shortcoming is that the parallel scaling of existing solution methods are not sufficient to exploit the performance of the next generation parallel computers that are like to contain over 100K processors. Other limitations of standard implementations of DFT are that they use basis sets [3, 4] and/or pseudopotentials [5] that are highly engineered, scale as  $O(N_A^3 \dots N_A^4)$  in the number of atoms, and the time scales of many interesting processes are orders of magnitude larger than can be directly simulated. Hence, there is still a need to investigate other computational methods for solving DFT.

Most popular solvers for DFT today are based on either the plane-wave method or the Gaussian orbital method. Both these methods suffer from the use of basis functions that do not have compact support, resulting in dense matrix operators and computationally intensive transforms which are difficult to implement efficiently on massively parallel computers. The limitations of these standard DFT solvers motivated the development of various real space solutions to the DFT equations. There have been a number of efforts to develop fast ab initio solvers based on real space solutions to the DFT equations [6, 7, 8, 9, 10, 11, 12, 13, 14, 16, 17, 18, 19, 20, 21, 22, 23, 24]. Uniform finite difference gridding coupled with multilevel solvers has led to significant progress in the calculation of large systems [6, 7, 8, 9, 10, 11, 12] with large numbers of processors ( $\sim 10K$  processors) [8]. 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 accurate description near the atomic centers. In particular, when the interaction between the electron and the nucleus is described by

the proper singular potential,  $-\frac{Ze}{|\vec{r}-\vec{R}_I|}$ , 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, these methods have needed to use a large number of elements for high Z atoms to be described accurately. Earlier work by Bylaska *et al.* [16], 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. Ne) 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 [17]. 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, Fattentburtt *et al.* have revisited these solvers and have shown them to be competitive with FFT solvers when very stiff pseudopotentials are used [18], however this work still had to rely on using pseudopotentials. The recent work of Harrison *et al.* using a multi-wavelet (high-order) basis [19, 20, 21], has also shown to be computationally competitive with standard DFT solvers and in some cases surpass them. However, in order to make their method efficient they still had to rely on smoothing the atomic singular potentials.

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 [25]. 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 giving resolution of the near singular features around atomic nuclei using minimal computational resources. This type of solver is one of the more popular solvers for partial differential equations. This method has several potential advantages over other popular solvers. It has compact support,

it can be controlled systematically by increasing the number of the basis functions, 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 II, a concise review of the FE method is given, and in section III the formulation of FE DFT equations is presented. In Section IV, 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<sub>2</sub><sup>+</sup>, and Li<sub>2</sub>. Finally conclusions are given in section V.

## II. BACKGROUND OF THE FINITE ELEMENT (FE) METHOD

In the FE procedure [26, 27, 28, 29, 30, 31], the solution domain  $\Omega$  (e.g. Fig. 1) is divided into a number of closed regions 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.

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

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

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

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 non-zero 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 on the left of Fig. 1 necessitates the following coefficients being equal

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

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

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

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.

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

With this assembled basis, the finite element expansion is simply written as

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

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 basis functions and integrals for the elements of different shapes can be readily calculated, through a variable transformation, in terms of just basis functions and integrals for the standard element. In this work 3D tetrahedral elements with nodes at the vertices were 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 Fig. 2, and its standard local basis functions are

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

Integrals over the tetrahedral standard element,

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

can readily be computed for polynomial functions

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

using the following analytic formula,

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

where

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

Eqs. (12) and (13) are fairly straightforward to compute. However, the computation of integrals of Eq. 11 can be further simplified. Since 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. A numerical method for computing the integrals can do this. In this work the following formula was used

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

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 computational procedure will be more efficient for small  $Q$ . A 5-point formulation which can be used to integrate the standard 3D tetrahedral element to second order is given in TableI.

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  $\tilde{e}$  the affine variable transformation is used. This variable transformation is linear and invertible. It is defined by

$$\vec{x} = F^{e_l} \vec{\tilde{x}} \quad (15)$$

or more explicitly by

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

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

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

$$b^{e_l} = \begin{bmatrix} x_1^{e_l} \\ y_1^{e_l} \\ z_1^{e_l} \end{bmatrix} \quad (18)$$

The inverse affine transformation is then

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

or

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

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

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

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

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

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

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

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

$$\begin{aligned}
K_{rs}^{e_l}(f) &= \int_{e_l} \phi_r^{e_l}(\vec{x}) f(\vec{x}) \phi_s^{e_l}(\vec{x}) d\vec{x} \\
&\approx |F^{e_l}| \sum_{q=1}^Q w_q \tilde{\phi}_r(\vec{x}_q) f(F^{e_l} \vec{x}_q) \tilde{\phi}_s(\vec{x}_q)
\end{aligned} \tag{27}$$

$$\begin{aligned}
K_{rs}^{e_l}(u) &= \int_{e_l} \phi_r^{e_l}(\vec{x}) \left( \sum_{l=1}^L \right) f(\vec{x}) \phi_s^{e_l}(\vec{x}) d\vec{x} \\
&\approx |F^{e_l}| \sum_{q=1}^Q w_q \tilde{\phi}_r(\vec{x}_q) f(F^{e_l} \vec{x}_q) \tilde{\phi}_s(\vec{x}_q)
\end{aligned} \tag{28}$$

$$\begin{aligned}
R_{rs}^{e_l}(f, \rho) &= \int_{e_l} \phi_r^{e_l}(\vec{x}) f(\rho(\vec{x})) \phi_s^{e_l}(\vec{x}) d\vec{x} \\
&\approx |F^{e_l}| \sum_{q=1}^Q w_q \tilde{\phi}_r(\vec{x}_q) f\left(\rho\left(F^{e_l} \vec{x}_q\right)\right) \tilde{\phi}_s(\vec{x}_q)
\end{aligned} \tag{29}$$

$$\begin{aligned}
T_{rs}^{e_l} &= \int_{e_l} \nabla \phi_r^{e_l}(\vec{x}) \cdot \nabla \phi_s^{e_l}(\vec{x}) d\vec{x} \\
&\approx |F^{e_l}| \sum_{q=1}^Q w_q \left[ \sum_{i=1}^3 \left( \sum_{j=1}^3 (F^{e_l})_{ji}^{-1} \frac{d}{d\vec{x}(j)} \tilde{\phi}_r(\vec{x}_q) \right) \left( \sum_{k=1}^3 (F^{e_l})_{ki}^{-1} \frac{d}{d\vec{x}(k)} \tilde{\phi}_s(\vec{x}_q) \right) \right]
\end{aligned} \tag{30}$$

### III. FORMULATION OF FE DFT EQUATIONS

The electronic wavefunctions in Density Functional Theory (DFT) in atomic units are given by the solutions to the following Kohn-Sham eigenvalue problem [32]

$$H\psi_i(\vec{x}) = \left( -\frac{1}{2}\nabla^2 + V_{ext} + V_H + V_{xc} \right) \psi_i(\vec{x}) = \epsilon_i \psi_i(\vec{x}) \tag{31}$$

$$\int_{\Omega} \psi_i(\vec{x}) \psi_j(\vec{x}) d\vec{x} = \delta_{ij} \tag{32}$$

where  $\epsilon_i$  is an eigenvalue, and the wavefunctions  $\{\psi_i\}$  satisfy the orthonormality constraints of a symmetric operator. In general, we require the lowest  $N_e/2$  eigenvalues and wavefunctions, where  $N_e$  is the number of electrons in the system. The external potential represents the external electrostatic field imposed on the system. For molecular systems  $V_{ext}$  represents the ion-electron interaction,

$$V_{ext}(\vec{x}) = \sum_{I=1}^{N_A} \frac{-Z_I}{|\vec{x} - \vec{R}_I|} \tag{33}$$

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,  $V_{xc}$ , the effects of electron-electron interactions. Both these potentials are functions of the electron density

$$\rho(\vec{x}) = \sum_{i=1}^{N_e/2} |\psi_i(\vec{x})|^2 \tag{34}$$

The exchange and correlation potentials  $V_{xc}$ , is a straightforward parameterized function of the electron density, e.g. Dirac exchange formula [33],

$$V_{xc}(\vec{x}) = -\left(\frac{3}{\pi}\rho(\vec{x})\right)^{1/3} = -\left(\frac{3}{\pi}\sum_{i=1}^{N_e/2}|\psi_i(\vec{x})|^2\right) \quad (35)$$

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

$$\nabla^2 V_H(\vec{x}) = -4\pi\rho(\vec{x}) = -4\pi\sum_{i=1}^{N_e/2}|\psi_i(\vec{x})|^2 \quad (36)$$

Since both  $V_{xc}$  and  $V_H$  are functions of  $\rho$ , 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 wavefunctions as a finite element expansion,

$$V_H(\vec{x}) = \sum_{m=1}^M v_m \eta_m(\vec{x}) \quad (37)$$

$$\psi_i(\vec{x}) = \sum_{m=1}^M c_m^i \eta_m(\vec{x}) \quad (38)$$

and defining the boundary conditions for Eqs. (31) and (36). 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.

$$\begin{aligned} V_H(|\vec{x}| \rightarrow \infty) &= 0 \\ \psi_i(|\vec{x}| \rightarrow \infty) &= 0 \end{aligned} \quad (39)$$

The problem with applying these boundary conditions is that the solution domain  $\Omega$  does not go out to  $\infty$ . For Eq. (31) this is not a problem since the wavefunctions for most molecular systems decay exponentially and can readily be set to zero at  $\partial\Omega$ . However, for

Eq. (36) 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,

$$V_H(\vec{x} \in \partial\Omega) = \sum_{l=0}^{L_{MAX}} \sum_{m=-l}^l N_{lm} M_{lm} \frac{T_{lm}(\hat{x})}{|\vec{x}|^{l+1}} \quad (40)$$

$$N_{lm} = \begin{cases} 1 & \text{for } m = 0 \\ 2 \frac{(l-|m|)!}{(l+|m|)!} & \text{for } |m| > 0 \end{cases} \quad (41)$$

$$M_{lm} = \int_{\Omega} |\vec{x}'|^l \rho(\vec{x}') T_{lm}(\hat{x}') d\vec{x}' \quad (42)$$

$$T_{lm}(\hat{x}) = \begin{cases} P_{l|m|}(\cos\theta) & \text{for } m = 0 \\ P_{l|m|}(\cos\theta) \cos|m|\varphi & \text{for } m > 0 \\ P_{l|m|}(\cos\theta) \sin|m|\varphi & \text{for } m < 0 \end{cases} \quad (43)$$

where  $\hat{x} = (\cos\varphi \sin\theta, \sin\varphi \sin\theta, \cos\theta)$  and  $P_{lm}(z)$  is an associated Legendre Polynomial [34]. Using this boundary condition, the solution to expansion coefficients for the Hartree potential in Eq. (37) is then found by solving the following systems of linear equations,

$$A_{mn} v_n = f_n \quad (44)$$

where

$$A_{mn} = \sum_{l=1}^L \sum_{r=1}^{T_l} \sum_{s=1}^{T_l} \delta_{m,\tilde{m}(r,l)} \delta_{n,\tilde{m}(s,l)} (-T_{rs}^{e_l}) \quad (45)$$

and

$$f_n = -4\pi \sum_{l=1}^L \sum_{r=1}^{T_l} \delta_{m,\tilde{m}(r,l)} G_r^{e_l} \left( \sum_{i=1}^{N_e/2} |\psi_i(\vec{x})|^2 \right) \quad (46)$$

Similarly, substituting Eq.(38) into Eqs.(31) and (32) produces the following generalized eigenvalue problem

$$H_{mn} c_n^i = \epsilon_i S_{mn} c_n^i \quad (47)$$

and orthonormality conditions

$$\sum_{m=1}^M \sum_{n=1}^M c_m^i S_{mn} c_n^j = \delta_{ij} \quad (48)$$

where

$$S_{mn} = \int_{\Omega} \eta_m(\vec{x}) \eta_n(\vec{x}) d\vec{x} \quad (49)$$

$$H_{mn} = \int_{\Omega} \eta_m(\vec{x}) H \eta_n(\vec{x}) d\vec{x} \quad (50)$$

A formula for the matrix  $S_{mn}$  in terms of Eq. (26) can be readily be obtained by substituting Eq. (7) into Eq. (49).

$$S_{mn} = \int_{\Omega} \left\{ \sum_{l=1}^L \sum_{t=1}^{T_l} \phi_t^{e_l}(\vec{x}) \delta_{m, \tilde{m}(t,l)} \right\} \left\{ \sum_{k=1}^L \sum_{s=1}^{T_s} \phi_s^{e_l}(\vec{x}) \delta_{n, \tilde{m}(s,k)} \right\} d\vec{x} \quad (51)$$

$$\begin{aligned} S_{mn} &= \int_{\Omega} \left\{ \sum_{l=1}^L \sum_{r=1}^{T_l} \phi_r^{e_l}(\vec{x}) \delta_{m, \tilde{m}(t,l)} \right\} \left\{ \sum_{k=1}^L \sum_{s=1}^{T_k} \phi_s^{e_k}(\vec{x}) \delta_{n, \tilde{m}(s,k)} \right\} d\vec{x} \\ &= \sum_{l=1}^L \sum_{k=1}^L \sum_{r=1}^{T_l} \sum_{s=1}^{T_k} \delta_{m, \tilde{m}(r,l)} \delta_{n, \tilde{m}(s,k)} \int_{\Omega} \phi_r^{e_l}(\vec{x}) \phi_s^{e_k}(\vec{x}) d\vec{x} \\ &= \sum_{l=1}^L \sum_{k=1}^L \sum_{r=1}^{T_l} \sum_{s=1}^{T_k} \delta_{m, \tilde{m}(r,l)} \delta_{n, \tilde{m}(s,k)} \delta_{k,l} \int_{e_l} \phi_r^{e_l}(\vec{x}) \phi_s^{e_l}(\vec{x}) d\vec{x} \\ &= \sum_{l=1}^L \sum_{r=1}^{T_l} \sum_{s=1}^{T_l} \delta_{m, \tilde{m}(r,l)} \delta_{n, \tilde{m}(s,l)} \int_{e_l} \phi_r^{e_l}(\vec{x}) \phi_s^{e_l}(\vec{x}) d\vec{x} \\ &= \sum_{l=1}^L \sum_{r=1}^{T_l} \sum_{s=1}^{T_l} \delta_{m, \tilde{m}(r,l)} \delta_{n, \tilde{m}(s,l)} M_{rs}^{e_l} \end{aligned} \quad (52)$$

Similarly, a formula for the matrix  $H_{mn}$  in terms of Eqs. (30), (28), and (29) can also be obtained.

$$H_{mn} = \sum_{l=1}^L \sum_{r=1}^{T_l} \sum_{s=1}^{T_l} \delta_{m, \tilde{m}(r,l)} \delta_{n, \tilde{m}(s,l)} \left( \frac{1}{2} T_{rs}^{e_l} + K_{rs}^{e_l} (V_{ext}) + K_{rs}^{e_l} (V_H) + R_{rs}^{e_l} \left( V_{xc}, \sum_{i=1}^{N_e/2} |\psi_i(\vec{x})|^2 \right) \right) \quad (53)$$

#### IV. ADAPTIVE FE DFT SOLUTIONS OF ATOMS AND MOLECULES

The Kohn-Sham DFT equations contains 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, a FE mesh could 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 “non-conforming” 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 non-conforming meshes, in general, FE meshes need to be conforming to ensure continuity of interpolated functions [27, 30]. 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 non-conforming mesh. The next step in the algorithm is then to mark for refinement the elements which contain hanging nodes. These steps procede iteratively until a conforming mesh is produced [35, 36]. 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 Algorithm1.

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

$$H = -\frac{1}{2}\nabla^2 - \frac{Z}{|\vec{x}|} \quad (54)$$

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) inorder for the Hamiltonian matrix of the FE solver not to contain a sinularity in any of its elements. The lowest energy solution and energy are  $\psi(\vec{x}) = \frac{Z^{\frac{3}{2}}}{\sqrt{\pi}} \exp(-Z|\vec{x}|)$  and  $\epsilon = -\frac{Z^2}{2}$ . Note that the severity of the singularity with increasing  $Z$  is reflected in the increasing localization of the solution.

---

**Algorithm 1** Conforming Adaptive FE Mesh Generation

---

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

$$\gamma(e_l) = \left( \frac{1}{2} \{ \xi(\vec{x}_2) + \xi(\vec{x}_3) + \xi(\vec{x}_4) - \xi(\vec{x}_1) \} - \xi(\vec{x}_c) \right) * \frac{|F^{e_l}|}{6} \quad (55)$$

where  $\vec{x}$ ,  $\vec{x}_2$ ,  $\vec{x}_3$ ,  $\vec{x}_4$  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.

2. Set refinement queues  $Q_1 = Q_2 = \emptyset$
  3. Place elements with large errors ( $\gamma(e_l) > \epsilon$ ) in the refinement queue  $Q_1$ .
  4. If  $Q_1 = \emptyset$  then go to step 9.
  5. If  $Q_1 > \emptyset$  then procede 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 Fig. 3 and place the non-conforming 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 Algorithm1 with the following weight function

$$\xi(\vec{x}) = \frac{Z^3}{\pi} \exp(-2Z|\vec{x}|) \quad (56)$$

The initial uniform mesh used was generated by uniformly refining a 7 element tetrahedral mesh 4 times using q-q refinement (as shown in Fig. 3) with the boundary vertices set to be at a radius of  $10a_0$ , resulting in mesh composed of 32,768 finite elements with 6,017 vertices. For a given  $\gamma$ , the 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  $\gamma$  was found to be approximately,

$$v(\gamma) \approx 200 \exp(-\log_{10}(\gamma)) \quad (57)$$

where  $v$  is the number of vertices.

In Fig. 4, the lowest eigenvalues and errors of the hydrogeon-like atoms ( $Z = 1 \dots 26$ ) 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 ( $\gamma = 10^{-4}$ ), 0.0069 ( $\gamma = 10^{-5}$ ), 0.0029 ( $\gamma = 10^{-6}$ ), and 0.0012 ( $\gamma = 10^{-7}$ ). When the singularity was strengthened the errors were considerably larger. For  $Z = 26$ , the errors were found to be 12.0348 ( $\gamma = 10^{-4}$ ), 4.9934 ( $\gamma = 10^{-5}$ ), 2.0517 ( $\gamma = 10^{-6}$ ), and 0.08194 ( $\gamma = 10^{-7}$ ). 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. Based on least squares fitting, the error in terms of  $\gamma$  and  $Z$  was found to be approximately given by the following relation.

$$\epsilon(\gamma, Z) \approx 0.64423 \exp(0.89503 \log_{10}(\gamma)) Z^2 \quad (58)$$

Using Eqs. (57) and (58), one can estimate the number of vertices needed to obtain accuracies in the millihartree range.

$$v(\epsilon, Z) \approx 122.37 \frac{Z^{2.2346}}{\epsilon^{1.1173}} \quad (59)$$

For example, for  $Z = 26$  and  $\gamma = 10^{-14}$ , the error and number of vertices needed are  $\epsilon \approx 0.0015$  and  $v \approx 240,520,857$ . 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 per atom). Given that the number of floating point operations per minimization step for a system of  $N_A$  atoms will be  $O(N_A^3 * v(\epsilon, Z))$  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  $\epsilon = 0.0015$  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 [37]. 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.

$$\left( -\frac{1}{2} \frac{d^2}{dr^2} + \frac{l(l+1)}{2r^2} - \frac{Z}{r} + 4\pi \int \frac{\rho(r')}{|r-r'|} r'^2 dr' + V_{xc}(r) \right) \psi_{il}(r) = \varepsilon_{il} \psi_{il}(r) \quad (60)$$

This 1d-radial equation was solved with a Hermann-Skilman telescoping grid and an Adams 5<sup>th</sup> order predictor-correction method [5]. From solving this simplified equation, the exact LDA energies for H, He, Li, and Ne were found to be -0.47867 au, -2.83484 au, -7.34386 au, 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 Na are reported in TableII. 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,

$$H = -\frac{1}{2} \nabla^2 - \frac{Z_1}{|\vec{x} - \vec{R}_1|} - \frac{Z_2}{|\vec{x} - \vec{R}_2|} \quad (61)$$

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 the geometric based refinement strategy given in Algorithm1 was used with the following weight function,

$$\xi(\vec{x}) = \frac{Z_1^3}{\pi} \exp\left(-2Z_1|\vec{x} - \vec{R}_1|\right) + \frac{Z_2^3}{\pi} \exp\left(-2Z_2|\vec{x} - \vec{R}_2|\right) \quad (62)$$

The singularities at the ion centers were accomodated by modify the initial uniform mesh (6,017 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 Fig. 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 seamingly simple molecule is difficult to calculate. The ground state solution has three molecular orbitals ( $1\sigma_g$  and  $1\sigma_u$ , and  $2\sigma_g$ ) shown in Fig.6. The bottom two molecular orbitals are very localized on the atoms. The top molecular orbital is considerably more delocalized, but it also contains a localized part. To define the adaptive FE mesh for this problem the the geometric based refinement strategy given in Algorithm1 was used with the following weight function,

$$\xi(\vec{x}) = \rho_{Li\ atom}^{LDA}\left(|\vec{x} - \vec{R}_1|\right) + \rho_{Li\ atom}^{LDA}\left(|\vec{x} - \vec{R}_2|\right)$$

where,  $\rho_{Li\ atom}^{LDA}(r)$  was obtained by using a spline fit of the the solution to Eq.60 for the Li atom. The same initial mesh as for the  $H_2^+$  molecule was used, and verticies nearest to each ion center were moved to lie on top of them. In Fig. 7, the binding energy curve for increasing levels of refinement is shown.

In Fig. 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 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 eigevalues 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 TableII, it is still significant given that the LDA binding energy for Li<sub>2</sub> is roughly 0.04 au.

It was found that the errors in the binding energy for Li<sub>2</sub> 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 ( $\vec{v}_{nearest}$ ). To do this, for each ion each of the vertices in the mesh ( $\vec{v}_i$ ) were moved by

$$\vec{v}_i = \vec{v}_i + \left( \vec{R}_I - \vec{v}_{nearest} \right) * f(|\vec{v}_i - \vec{v}_{nearest}|) \quad (63)$$

where  $f(r)$  is the screening function

$$f(r) = 1 - \left[ 1 - \exp\left(-\left(\frac{r}{R}\right)^N\right) \right]^N \quad (64)$$

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 Fig. 7). This result suggests that an overlapping grid method can be used to reduce the errors ("cancellation of errors") in structure and bond energies of the system.

## V. 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  $\epsilon$  was estimated to be  $v(\epsilon, Z) \approx 122.37 * \frac{Z^{2.2346}}{1.1173}$ . 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 to 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  $O(N_A^3 * v(\epsilon, Z))$  (e.g.  $Z = 26$ ,  $\epsilon = 0.0015\text{au}$ , and  $N_A = 100$ , the memory requirement and computational cost would be  $\sim 2$  terabytes and  $\sim 25$  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 a 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  $\text{H}_2^+$  molecule this strategy was found to work very well. However, for the  $\text{Li}_2$  molecule very large errors in the binding energy curves were seen even when the geometric adaptive refinement procedure. It was found that the errors in the binding energy for  $\text{Li}_2$  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 ( $\vec{v}_{nearest}$ ) in the 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 piece-wise 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. However, it is

anticipated the memory requirements can be reduced by using higher-order FE elements.

### Acknowledgments

This research was supported by the DOE ASCR Multiscale Mathematics program and the DOE BES Geosciences program. The Pacific Northwest National Laboratory 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 operations are supported by the DOE's Office of Biological and Environmental Research. 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).

- 
- [1] E. Aprà, E. J. Bylaska, D. J. Dean, A. Fortunelli, F. Gao, P. S. Kristić, J. C. Wells, and T. L. Windus, "NWChem for Material Science," *Computational Materials Science*, **28**, 209-221 (2003).
- [2] E. J. Bylaska, W. A. de Jong, N. Govind, K. Kowalski, T. P. Straatsma, M. Valiev, D. Wang, E. Apra, T. L. Windus, J. Hammond, P. Nichols, S. Hirata, M. T. Hackler, Y. Zhao, P.-D. Fan, R. J. Harrison, M. Dupuis, D. M. A. Smith, J. Nieplocha, V. Tipparaju, M. Krishnan, Q. Wu, T. Van Voorhis, A. A. Auer, M. Nooijen, E. Brown, G. Cisneros, G. I. Fann, H. Fruchtl, J. Garza, K. Hirao, R. Kendall, J. A. Nichols, K. Tsemekhman, K. Wolinski, J. Anchell, D. Bernholdt, P. Borowski, T. Clark, D. Clerc, H. Dachsel, M. Deegan, K. Dyall, D. Elwood, E. Glendening, M. Gutowski, A. Hess, J. Jaffe, B. Johnson, J. Ju, R. Kobayashi, R. Kutteh, Z. Lin, R. Littlefield, X. Long, B. Meng, T. Nakajima, S. Niu, L. Pollack, M. Rosing, G. Sandrone, M. Stave, H. Taylor, G. Thomas, J. van Lenthe, A. Wong, and Z. Zhang, "NWChem, A Computational Chemistry Package for Parallel Computers, Version 5.1" (2007), Pacific Northwest National Laboratory, Richland, Washington 99352-0999, USA.
- [3] T.H. Dunning, Jr. K.A. Peterson, and D.E. Woon, "Gaussian Basis Sets for Use in Correlated Calculations," in *Encyclopedia of Computational Chemistry*, Ed. P.v.R. Schleyer, John Wiley

- & Sons Ltd (1997).
- [4] B.I. Dunlap, J.W.D. Connolly, and J.R. Sabin, "On some approximations in applications of X theory," *Journal of Chemical Physics*, **71**, 3396-3402 (1979).
  - [5] D.R. Hamann, "Generalized norm-conserving pseudopotentials," *Physical Review B*, **40**, 2980-2987 (1989).
  - [6] E. Tsuchida and M. Tsukada, "Electronic-structure calculations based on the finite-element method," *Physical Review B*, **52**, 5573-5578, (1995).
  - [7] J. Bernholc, E.I. Briggs, D.J. Sullivan, C.J. Brabec, M.B. Nardelli, K. Rapcewicz, C. Roland, and M. Wensell, "Real-space multigrid methods for large-scale electronic structure problems", *International Journal of Quantum Chemistry* **65**, 531-543 (1997).
  - [8] J. Bernholc, M. Hodak, and W.C. Lu, "Recent developments and applications of the real-space multigrid method," *Journal of Physics-Condensed Matter*, **20**, Art. No. 294205 (2008).
  - [9] J.R. Chelikowsky, N. Troullier, and Y. Saad, "Finite-Difference-Pseudopotential Method: Electronic Structure Calculations without a Basis," *Physical Review Letters*, **72**, 1240-1243, (1994).
  - [10] J.R. Chelikowsky, N. Troullier, K.Wu, and Y. Saad, "Higher-order finite-difference pseudopotential method: An application to diatomic molecules," *Physical Review B*, **50**, 11355-11364 (1994).
  - [11] E.L. Briggs, D.J. Sullivan, and J. Bernholc, "Large-scale electronic-structure calculations with multigrid acceleration," *Physical Review B*, **52**, R5471-R5474 (1995).
  - [12] J.E. Pask, B.M. Klein, P.A. Sterne, and C.Y. Fong, "Finite-element methods in electronic-structure theory," *Computer Physics Communications*, **135**, 1-34 (2001).
  - [13] K. Cho, T.A. Arias, J.D. Joannopoulos, and P.K. Lam, "Wavelets in Electronic Structure Calculations," *Physical Review Letters*, **71**, 1808-1811 (1993).
  - [14] R.A. Lippert, T.A. Arias, A. Edelman, "Multiscale Computation with Interpolating Wavelets," *Journal of Computational Physics*, **140**, 278-310 (1998).
  - [15] T.A. Arias, "Multiresolution analysis of electronic structure: semicardinal and wavelet bases," *Reviews of Modern Physics*, **71**, 267-311, (1999).
  - [16] E.J. Bylaska, S.R. Kohn, S.B. Baden, A. Edelman, R. Kawai, M.E.G. Ong, and J.H. Weare, "Scalable Parallel Numerical Methods and Software Tools for Material Design", *Proceeding of the Seventh SIAM Conference on Parallel Processing for Scientific Computing*, San Francisco, CA (1995).

- [17] S. Kohn, J. Weare, E. Ong, and S. Baden, “Parallel Adaptive Mesh Refinement for Electronic Structure Calculations”, Eighth SIAM Conference on Parallel Processing for Scientific Computing, Minneapolis, MN (1997).
- [18] J.L. Fattebert, R.D. Hornung, and A.M. Wissink, “Finite element approach for density functional theory”, *Journal of Computational Physics* **223**, 759-773 (2007).
- [19] R.J. Harrison, G.I. Fann, T.G. Yanai, Z. Gan, and G. Beylkin, “Multiresolution quantum chemistry: Basic theory and initial applications”, *Journal of Chemical Physics* **121**, 11587-11598 (2004).
- [20] T. Yanai, G.I. Fann, Z.T. Gan, R.J. Harrison, and G. Beylkin, “Multiresolution quantum chemistry in multiwavelet bases: Hartree-Fock exchange”, *Journal of Chemical Physics* **121**, 6680-6688 (2004).
- [21] T. Yanai, G.I. Fann, Z.T. Gan, R.J. Harrison, and G. Beylkin, “Multiresolution quantum chemistry in multiwavelet bases: Analytic derivatives for Hartree-Fock and density functional theory”, *Journal of Chemical Physics* **121**, 2866-2876 (2004).
- [22] N.A. Modine, G. Zumbach and E. Kaxiras, “Adaptive-coordinate real-space electronic-structure calculations for atoms, molecules and solids”, *Physical Review B*, **55**, 10289-10301 (1997).
- [23] T. Torsti, V. Lindberg, I. Makkonen, E. Ogando, E. Rasanen, H. Saarikoski, M.J. Puska, and R.M. Nieminen, “Real-space electronic-property calculations for nanoscale structures,” in *Handbook of Theoretical and Computational Nanotechnology*, Eds. by M. Rieth and W. Schommers, Forschungszentrum Karlsruhe, Germany (2006).
- [24] N.A. Modine, G. Zumbach, and E. Kaxiras, “Adaptive coordinate real-space electronic structure calculations for atoms, molecules and solids,” *Physical Review B*, **55**, 10289-10301 (1997).
- [25] M. Holst “Adaptive numerical treatment of elliptic systems on manifolds”, *Advances in Computational Mathematics*, **15**, 139-191 (2001).
- [26] O. Axelsson and V.A. Barker, *Finite Element Solution of Boundary Value Problems: Theory and Computation*, SIAM, Philadelphia 2001.
- [27] D. Braess, *Finite Elements: Theory, Fast Solvers and Applications in Solid Mechanics*, 2nd edition, Cambridge University Press, Cambridge 2005.
- [28] S.C. Brenner and L.R. Scott, *The Mathematical Theory of Finite Element Methods*, 2nd edition, Springer-Verlag, New York 2002.
- [29] G. Strang, *Piecewise Polynomials and the Finite Element Method*, Bulletin of the American

- Mathematical Society, **79**, 1128-1137 (1973).
- [30] D.H. Norrie and G. de Vries, *The Finite Element Method*, Academic Press, New York 1973.
- [31] O.C. Zienkiewicz, and K. Morgan, *Finite Elements and Approximation*, Dover Publications, New York 1983.
- [32] W. Kohn and L.J. Sham, “Self-consistent equations including exchange and correlation effects”, *Physical Review*, **140**, A1133-A1138 (1965).
- [33] P.A.M. Dirac, “Note on exchange phenomena in the Thomas atom,” *Proceedings of the Cambridge Philosophical Society*, **26**, 376-385 (1930).
- [34] G. Sansone, *Orthogonal Functions*, Revised English edition, Dover Publications, New York (1991).
- [35] R.E. Bank and M. Holst, “A New Paradigm for Parallel Adaptive Meshing Algorithms”, *Siam J. Sci. Comput.* **22**, 1411-1443 (2000).
- [36] D.N. Arnold, A. Mukherjee, and L. Pouly, “Locally Adapted Tetrahedral Meshes using Bisection”, *Siam, J. Sci. Comput.* **22**, 431-448 (2000).
- [37] S.H. Vosko, L. Wilk and M. Nusair, “Accurate Spin-Dependent Electron Liquid Correlation Energies for Local Spin-Density Calculations - A Critical Analysis,” *Canadian Journal of Physics*, **58**, 1200-1211, (1980).

# Tables

Formula Type	Points	Weight
5 pt formula	$(1/4, 1/4, 1/4)$	$-2/15$
	$(1/6, 1/6, 1/6)$	$3/40$
	$(1/2, 1/6, 1/6)$	$3/40$
	$(1/6, 1/2, 1/6)$	$3/40$
	$(1/6, 1/6, 1/2)$	$3/40$

Table I: 5-point Tetrahedral Integration points and weights.

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

Table II: LDA Energies and errors of H, He, Li, and Ne at increasing levels of refinement. (<sup>a</sup>required more than 2 GB of memory).

# Figures

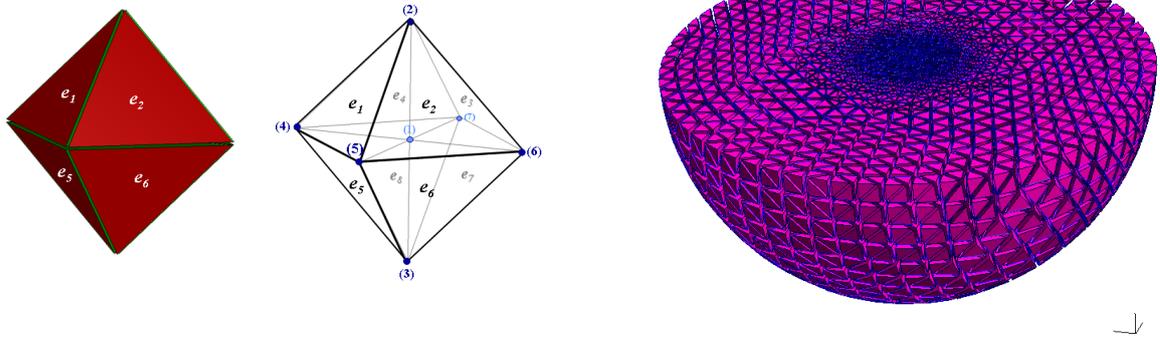


Figure 1: Examples of a 3D finite element meshes. Left: tetrahedral domain containing 8 elements and 7 vertex nodes. The elements are labeled  $e_l$  and the nodes are labeled  $(m)$  in this figure. Right: adaptive hemisphere domain containing 453,608 elements and 81,406 vertex nodes.

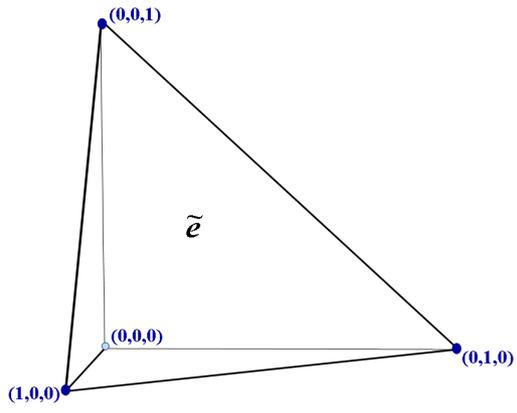


Figure 2: Standard 3D piece-wise tetrahedral element.

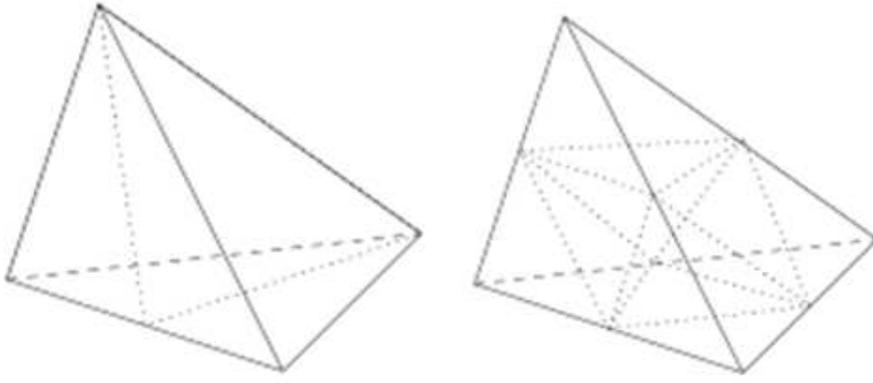


Figure 3: b-b tetrahedral bisection (left), and q-q tetrahedral bisection (right).

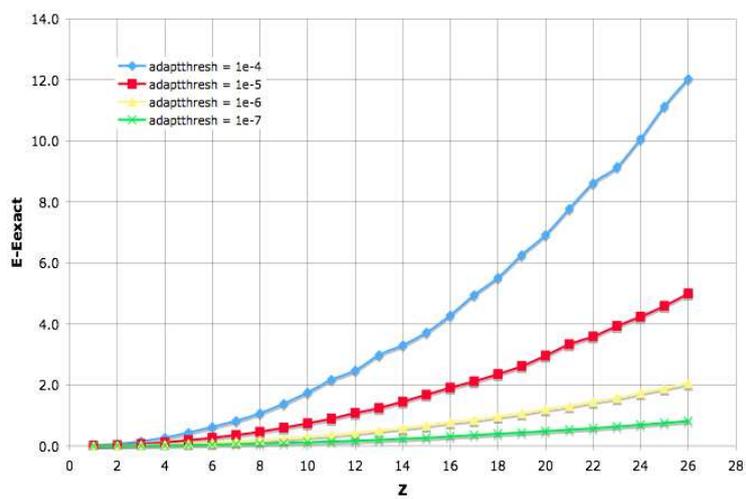
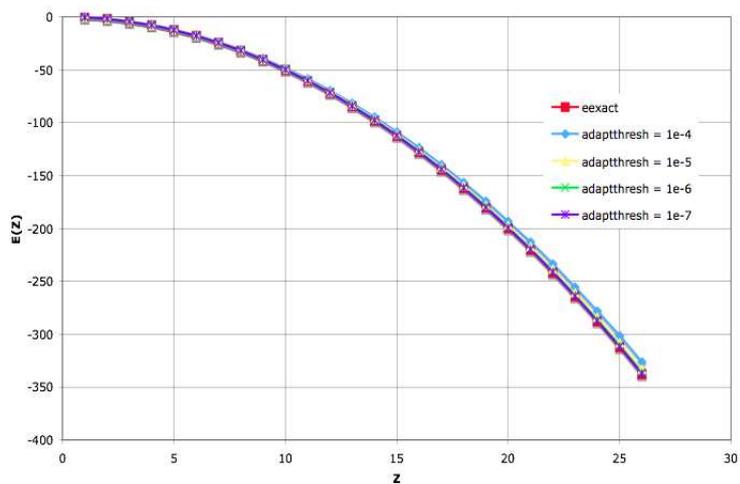


Figure 4: Eigenvalues and errors for the hydrogen-like atom as a function  $Z$ .

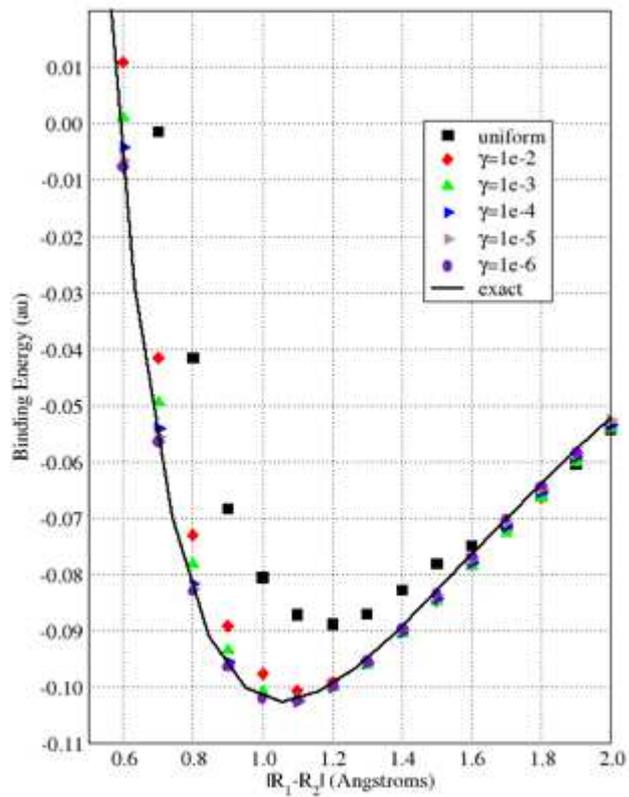


Figure 5: Binding energy curves for  $H_2^+$  obtained with adaptive gridding defined by the geometric based refinement strategy.

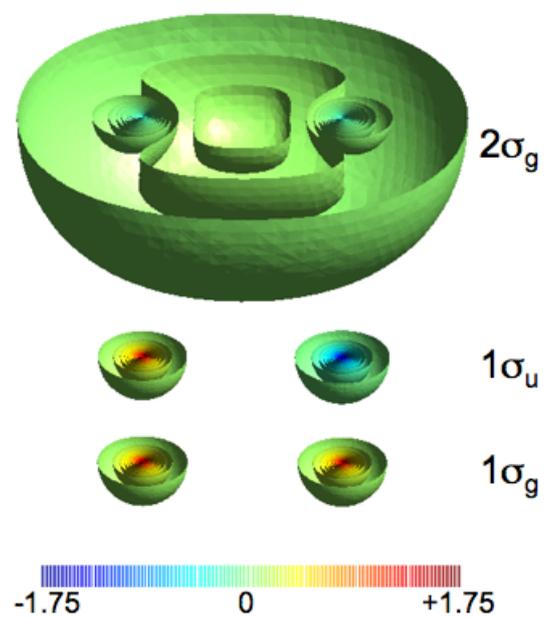


Figure 6: The ground state molecular orbitals ( $1\sigma_g$ ,  $1\sigma_u$ , and  $2\sigma_g$ ) of LDA for  $\text{Li}_2$  obtained with the FE DFT solver.

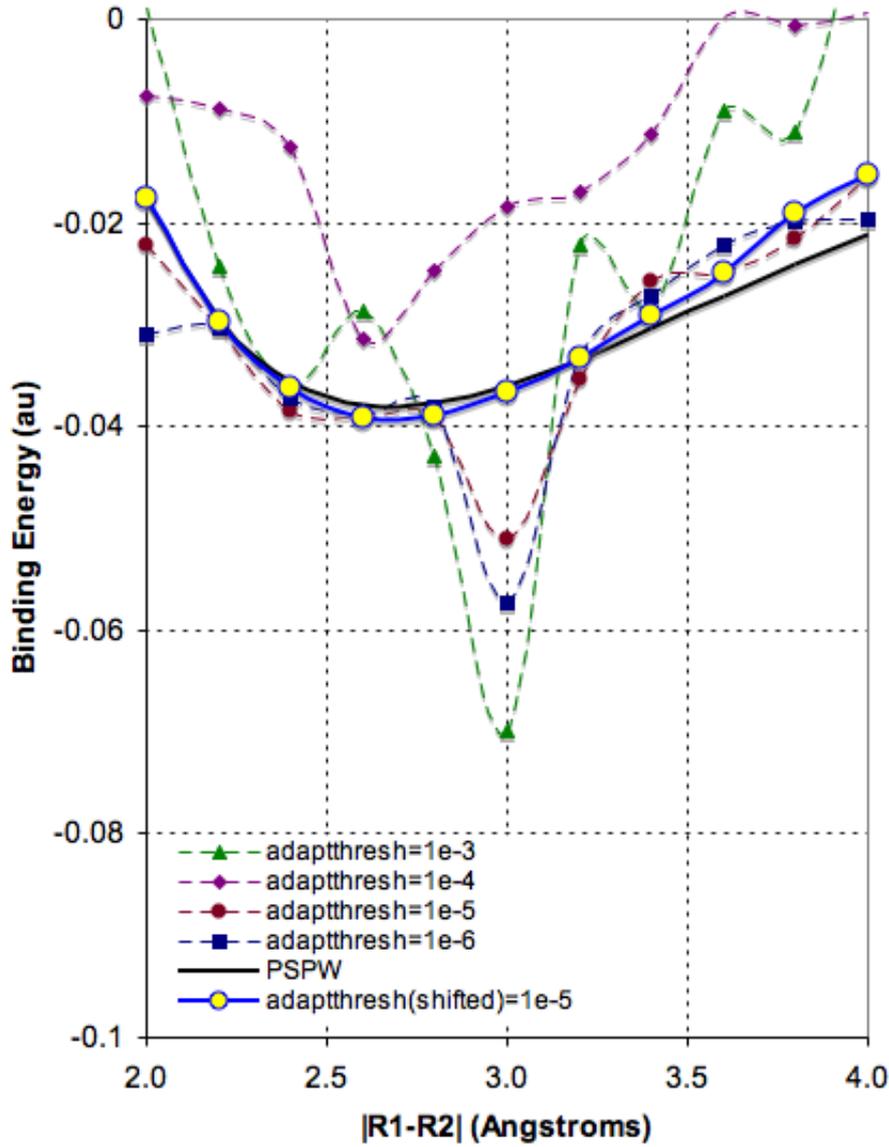


Figure 7: Binding energy curves for  $\text{Li}_2$  obtained with adaptive gridding defined by the geometric based refinement strategy. The  $\text{adaptthresh}(\text{shifted})=1e-5$  curve was obtained with a  $\gamma = 1e-5$  adaptive mesh where the initial mesh was modified by shifting procedure of Eq. 63. The PSPW curve, shown for comparison, was obtained a NWChem pseudopotential plane-wave calculation [2].