


2012

Reformulation of the Muffin-Tin Problem in Electronic Structure Calculations within the Feast Framework

Alan R. Levin

University of Massachusetts Amherst

Follow this and additional works at: <https://scholarworks.umass.edu/theses>

 Part of the [Electrical and Electronics Commons](#), [Electronic Devices and Semiconductor Manufacturing Commons](#), and the [Numerical Analysis and Computation Commons](#)

Levin, Alan R., "Reformulation of the Muffin-Tin Problem in Electronic Structure Calculations within the Feast Framework" (2012).
Masters Theses 1911 - February 2014. 923.

Retrieved from <https://scholarworks.umass.edu/theses/923>

This thesis is brought to you for free and open access by ScholarWorks@UMass Amherst. It has been accepted for inclusion in Masters Theses 1911 - February 2014 by an authorized administrator of ScholarWorks@UMass Amherst. For more information, please contact scholarworks@library.umass.edu.

**REFORMULATION OF THE MUFFIN-TIN PROBLEM IN
ELECTRONIC STRUCTURE CALCULATIONS WITHIN
THE FEAST FRAMEWORK**

A Thesis Presented

by

ALAN R. LEVIN

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL AND COMPUTER ENGINEERING

September 2012

Electrical and Computer Engineering

© Copyright by Alan R. Levin 2012

All Rights Reserved

**REFORMULATION OF THE MUFFIN-TIN PROBLEM IN
ELECTRONIC STRUCTURE CALCULATIONS WITHIN
THE FEAST FRAMEWORK**

A Thesis Presented

by

ALAN R. LEVIN

Approved as to style and content by:

Eric Polizzi, Chair

Neal Anderson, Member

Ashwin Ramasubramaniam, Member

Christopher Hollot, Department Chair
Electrical and Computer Engineering

ACKNOWLEDGEMENTS

Although countless people have helped me reach where I am today, there are a few people I would like to explicitly offer my sincerest thanks to for making this thesis possible:

-Dr. Eric Polizzi who, for 3 years, has been a truly amazing advisor for both my Honor's and Master's theses; he first exposed me to semiconductors and nanoelectronics (the fields I would come to love); his support and guidance has been endless with my thesis and my other interests, such as teaching, which he has gone above and beyond to encourage; his passion and skill in both research and teaching is truly remarkable and has been a constant inspiration during my time at UMass,

-Dr. Neal Anderson for serving on my committee, preparing me for my research with amazing courses in semiconductor physics and quantum mechanics, and always being available to answer questions on a variety of subjects,

-Dr. Ashwin Ramasubramaniam for serving on my committee and offering a unique perspective on my research,

-Dr. Dennis Goeckel for his invaluable advice about graduate school,

-Kimberly Donoughe for agreeing to make several figures for this thesis, despite the short notice I would give her,

-My roommate, Styck, who was willing to adventure through graduate school with me,

-Finally, my parents, Alan and Deb, grandmother, Joan, and the rest of my family for their constant love and support in all my endeavours.

This thesis is based upon work supported by the National Science Foundation under Grant No ECCS 0846457.

ABSTRACT

REFORMULATION OF THE MUFFIN-TIN PROBLEM IN ELECTRONIC STRUCTURE CALCULATIONS WITHIN THE FEAST FRAMEWORK

SEPTEMBER 2012

ALAN R. LEVIN

B.S., UNIVERSITY OF MASSACHUSETTS AMHERST

M.S.E.C.E., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor Eric Polizzi

This thesis describes an accurate and scalable computational method designed to perform nanoelectronic structure calculations. Built around the FEAST [1] framework, this method directly addresses the nonlinear eigenvalue problem. The new approach allows us to bypass traditional approximation techniques typically used for first-principle calculations. As a result, this method is able to take advantage of standard [2] muffin-tin type domain decomposition techniques without being hindered by their perceived limitations. In addition to increased accuracy, this method also has the potential to take advantage of parallel processing for increased scalability.

The Introduction presents the motivation behind the proposed method and gives an overview of what will be presented for this thesis. Chapter 1 explains how electronic structure calculations are currently performed, including an overview of Density Functional Theory and the advantages and disadvantages of various numerical techniques. Chapter 2 describes, in detail, the method proposed for this thesis, including

mathematical justification, a matrix-level example, and a description of implementing the FEAST algorithm. Chapter 3 presents and discusses results from numerical experiments for Hydrogen and various Hydrogen molecules, Methane, Ethane, and Benzene. Chapter 4 concludes with a summary of the presented work and its impact in the field.

TABLE OF CONTENTS

	Page
ACKNOWLEDGEMENTS	iv
ABSTRACT	v
LIST OF TABLES	ix
LIST OF FIGURES	x
INTRODUCTION	1
CHAPTER	
1. ELECTRONIC STRUCTURE CALCULATIONS	3
1.1 Physical Modeling	4
1.1.1 Density Functional Theory	4
1.1.2 Ionic Potential	6
1.2 Numerical Modeling	6
1.2.1 Plane Wave Expansion Methods	6
1.2.2 Localized Orbital Methods	7
1.2.3 Atomic Decomposition (Muffin-tin) Methods	8
2. METHODOLOGY	9
2.1 Mesh and Real-Space Techniques	10
2.2 FEAST Eigenvalue Solver	10
2.3 Mathematical Methodology	13
2.4 Discrete Matrix-level Description	15
2.4.1 General Approach	16
2.4.2 Application	17
2.5 Implementation of FEAST-RCI	18

2.6	Parallelization	20
2.7	Hybrid Basis	20
3.	NUMERICAL EXAMPLES	23
3.1	Hydrogen	23
3.2	Dihydrogen Cation	25
3.3	Trihydrogen Dication	26
3.4	Methane	27
3.5	Ethane	27
3.6	Benzene	28
3.7	Discussion	29
	CONCLUSION	35
	BIBLIOGRAPHY	36

LIST OF TABLES

Table	Page
1.1	Previously undefined variables for (1.2)-(1.5). 5
2.1	Calculated energy levels for H_2^+ using quadratic (P_2) FEM and spectral decomposition hybrid basis sets of 9 and 598. 22
3.1	Previously undefined variables for (3.1). 24
3.2	Matrix sizes for Hydrogen atom using full and muffin-tin meshes. 24
3.3	Calculated and exact 1st energy level for Hydrogen. 24
3.4	Matrix sizes for Dihydrogen Cation molecule using full and muffin-tin meshes. 25
3.5	Calculated and exact 1st energy level for Dihydrogen Cation. 25
3.6	Matrix sizes for Trihydrogen Dication molecule using full and muffin-tin meshes. 26
3.7	Calculated and estimated exact 1st energy level for Trihydrogen Dication. 26
3.8	Matrix sizes for Methane molecule using full and muffin-tin meshes. 27
3.9	Matrix sizes for Ethane molecule using full and muffin-tin meshes. 28
3.10	Matrix sizes for Benzene molecule using full and muffin-tin meshes. 28
3.11	Calculated and numerical estimate for 1st energy level for Benzene. 29

LIST OF FIGURES

Figure	Page
1.1	Flowchart of various discretization methods available within the DFT framework (which will be discussed in Section 1.2) [24]. Pictured above: plane wave expansion (PWE), empirical pseudopotential (EPM), atomic pseudopotential (APM), plane wave (PW), localized orbitals (LO), linear combination of atomic orbitals (LCAO), atomic decomposition (ADM), augmented plane waves (APW), linear APW (LAPW), muffin-tin orbital (MTO), and linear muffin-tin orbital (LMTO). 5
1.2	Representation of exact (dotted blue) and pseudo- (solid red) potentials and resulting wave function [24]. 7
2.1	A 2D real-space muffin-tin domain, with a fine atomic mesh (left) and a coarse interstitial mesh (right). Created using Triangle [29]. 9
2.2	A 3D real-space muffin-tin domain decomposition, with the surface (top-left) and cut-out (bottom-left) of the fine atomic mesh and a coarse interstitial mesh with embedded atomic meshes (right). 11
2.3	Energy levels for H_2^+ using P_2 FEM (blue), $M=9$ Hybrid (red), and $M=598$ Hybrid (Green). 21
3.1	A 3D mesh containing an H atom, surrounded by a coarse interstitial mesh. 24
3.2	A 3D mesh containing an H_2^+ molecule, surrounded by a coarse interstitial mesh. 25
3.3	A 3D mesh containing an H_3^{++} molecule, surrounded by a coarse interstitial mesh. 26
3.4	A top-down view embedded within a coarse mesh (left) and a side view (right) of a CH_4 molecule. 27
3.5	A top-down view embedded within a coarse mesh (left) and a side view (right) of a C_2H_6 molecule. 28

3.6	A 3D mesh containing an C_6H_6 molecule, surrounded by a coarse interstitial mesh.	29
3.7	Number of nodes for full (blue), atomic (green), and interstitial (red) meshes versus number of atoms per molecule.	31
3.8	Actual (blue) and ideal (red) relative speed-up times for factorization vs. number of atoms (threads). Here, 1 corresponds to 2.28 seconds.	32
3.9	Actual (blue) and ideal (red) relative speed-up time for solve vs. number of atoms (threads). Here, 1 corresponds to 0.61 seconds.	33
3.10	Actual relative speed-up time for interstitial solve step (blue) and interstitial nodes per number of threads (red) vs. number of atoms (threads). Here, 1 corresponds to 0.15 seconds.	34

INTRODUCTION

Having fully entered into the nanoregime of electronic design and manufacturing, numerical modeling techniques that can assist with the understanding and development of these devices is more important than ever. In a nanoelectronic device, structures are typically on the same order of magnitude as electron wavelengths, and active regions are often smaller than coherence lengths [3],[4]. This signifies an end to the “duct tape pseudo-quantum models” and a need for full-scale, first-principle quantum models [5],[6],[7]. The practice of modeling is nothing new to the electronics industry. For some time, VLSI tools have been standard for designing microelectronic devices [7]. Traditionally, however, quantum models have been mostly used to help with general scientific understanding of nanodevices, usually after devices are already being mass-produced [4]. This is no longer a sustainable model, now that even the most basic prototyping requires expensive cleanrooms and machinery to accommodate the significantly smaller sizes and tighter tolerances of nanodevices [4]. Along with this increase in basic costs comes the increase in complexity. Quantum effects greatly increase optimization challenges [7]. Additionally, these quantum effects can often have unforeseen consequences, greatly affecting the potential feasibility of an electronic device [7]. Because of the increased cost and complexity of modern nanoelectronic devices, accurate and scalable quantum numerical modeling must become an innate part of device design.

The problem of performing electronic structure calculations is particularly important. Electronic structure calculations reveal vital information about the physical and electronic properties of a system, such as energy spectrum, density of states, and

band structure. Many methods are currently widely used to perform these calculations. Most are based on Hartree-Fock or Density Functional methods, which produce a framework to ease the solution of the many-body Schrödinger equations. The resulting equation, however, is still too challenging to solve directly for large-scale systems. Because of this, a large array of numerical methods (such as plane wave expansion, linearized augmented plane waves, linearized muffin-tin orbitals, and linear combination of atomic orbitals [15]) have been developed. As will be discussed in Chapter 1, each of these methods relies heavily on approximations as a result of either a) linearization techniques used to handle the resulting nonlinear eigenvalue problem, or b) the pseudopotential models used to simplify the computational domain.

The method proposed in this thesis is based on embedded self-energy (Green's function) techniques on a real-space muffin-tin decomposition. Although these techniques have been used before [2],[16], this method differs in that it does not attempt to approximate the resulting eigenvalue problems. Within the FEAST [1] eigenvalue algorithm framework, this method is able to directly solve the resulting nonlinear eigenvalue problem. The two main advantages of this are:

- 1) Accuracy:** A higher accuracy can be achieved by eliminating the traditional approximation steps and by using the full ionic potential, and
- 2) Scalability:** Because of the the muffin-tin decomposition and the FEAST framework, atoms can be solved for individually and in parallel.

This thesis is broken into four chapters. Chapter 1 explains the current methods used for electronic structure calculations and discusses their advantages and disadvantages. Chapter 2 gives a detailed explanation of the mathematical methodology, FEAST implementation, and meshing techniques. Chapter 3 presents the results of numerical experiments for H, H₂⁺, H₃⁺⁺, CH₄, C₂H₆, and C₆H₆. Chapter 4 concludes with a summary and a discussion on the impact of this work.

CHAPTER 1

ELECTRONIC STRUCTURE CALCULATIONS

A perfect, error-free electronic structure model would require a full-scale, many-body, quantum mechanical description of the system. Although this can be done with minor effort for a one-particle system, the situation becomes increasingly harder as particles are added. As can easily be shown [11], the Schrödinger equation for a two-particle system (within the Born-Oppenheimer Approximation) is:

$$\left[-\frac{\hbar}{2} \left(\frac{1}{m_1} \nabla_1^2 + \frac{1}{m_2} \nabla_2^2 + V_{ion}(r_1, r_2) \right) \right] \Psi(r_1, r_2) = E\Psi(r_1, r_2). \quad (1.1)$$

Although the Hamiltonian for this equation is still linear, because of the interaction between particles, it is incredibly challenging (and with additional particles, quickly becomes impossible) to solve because of the rapid increase in system matrix size [10],[11]. Therefore, first-principle electronic structure calculations are instead performed using Hartree-Fock (HF) or Density Functional Theory (DFT) methods. Kohn-Sham DFT (KS-DFT) is particularly well regarded as it is less computationally demanding and allows, in practice, for exact calculation of ground state density and system energy [14]. Within the KS-DFT framework, a number of approximation schemes, such as plane wave expansion, localized orbitals, and atomic decomposition methods, can be used to ease the resulting equation. In this chapter, the physical and numerical models most commonly used for electronic structure calculations are described.

1.1 Physical Modeling

1.1.1 Density Functional Theory

DFT is based on the principle that any system property within the many-body problem can be described by a functional of the ground state density $n_0(\mathbf{r})$. KS-DFT provides a method for creating approximations of $n_0(\mathbf{r})$ for large-scale systems [25]. The set of linear equations to be solved within the KS-DFT framework is [20]:

$$\rho(r) = 2 \sum_{i=1}^{N_E} |\psi_i(r)|^2, \quad (1.2)$$

$$V_{eff}[\rho(r)] = V_{ion}(r) + V_h[\rho(r)] + V_{xc}[\rho(r)], \quad (1.3)$$

$$-\nabla^2 V_h(r) = \frac{\rho(r)}{\epsilon}, \quad (1.4)$$

$$V_{xc}[n](r) = \frac{\delta}{\delta n(r)} E_{xc}[n], \quad (1.5)$$

where relevant variables are defined in Table 1.1. The solution of this system leads to the eigenvalue Schrödinger-type equation:

$$\left(-\frac{\hbar^2}{2m} \nabla^2 - qV_{eff}[\rho(r)] \right) \psi_i(r) = E_i \psi_i(r), \quad (1.6)$$

or its equivalent discrete (matrix representation) form:

$$(\mathbf{H} + \mathbf{U}_{\text{eff}})\psi_i = E_i \mathbf{S}_i \psi_i, \quad (1.7)$$

whose solution yields the \mathbf{S}_i (mass matrix) orthonormal eigenstates ψ_i , which reproduces the density of the original many-body system.

Although the resulting set of equations is indeed linear, it is often the most time-consuming computational step because a) the discretization techniques needed for acceptable accuracy lead to large system matrices, and b) the number of eigenpairs

V_{eff}	Kohn-Sham potential
$\rho(r)$	Electron density
N_E	Number of electronic states
V_{ion}	Core potential
V_h	Hartree potential
V_{xc}	Exchange correlation potential
E_{xc}	Exchange correlation energy

Table 1.1. Previously undefined variables for (1.2)-(1.5).

needed to compute the electron density is proportional to the number of atoms in the system [10]. Because of this, numerical techniques are needed to further assist with the solution of the Schrödinger equation. Table 1.1 presents a flowchart of the various discretization methods (which will be discussed in Section 1.2) that are used within the DFT framework.

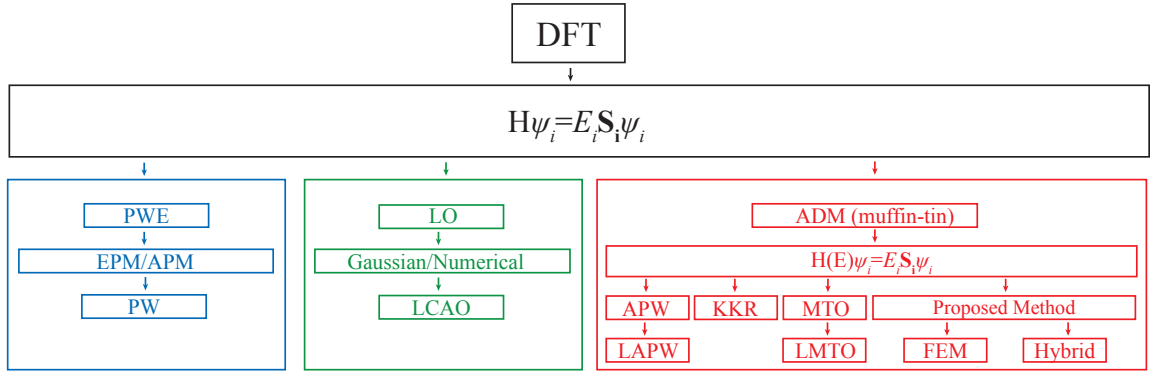


Figure 1.1. Flowchart of various discretization methods available within the DFT framework (which will be discussed in Section 1.2) [24]. Pictured above: plane wave expansion (PWE), empirical pseudopotential (EPM), atomic pseudopotential (APM), plane wave (PW), localized orbitals (LO), linear combination of atomic orbitals (LCAO), atomic decomposition (ADM), augmented plane waves (APW), linear APW (LAPW), muffin-tin orbital (MTO), and linear muffin-tin orbital (LMTO).

1.1.2 Ionic Potential

In order to solve (1.1), a model for the ionic potential V_{ion} is needed. The potential of an atom can be broken into two regions: the core electrons (those that reside very close to the nucleus and have little effect on atomic interaction) and valence electrons (those that are responsible for molecular bonding and transport phenomena) [21]. As can be seen in Figure 1.2, by eliminating the potential of the core electrons, calculations can be simplified without dramatically changing the wave function. This line of reasoning gives rise to the two most common methods: empirical (EPM) and ab-initio (APM) pseudopotential [20]. With the EPM, the pseudopotential is fitted to experimental data. While this can produce highly accurate results, it requires empirical data for every structure that is to be simulated. The APM is based on an all-electron calculation in which the wave function of the core atoms is absorbed into the wave function of the valence electrons, and the resulting eigenpairs are matched at the designated cut-off region r_c [22]. Although this method can be used to create accurate results, the linear combination of the core and valence electrons requires special treatment to mitigate error, which becomes increasingly time consuming as the system size increases [20],[22]. In order to bypass the inaccuracies and challenges incurred by pseudopotential methods, a true all-electron model can be used. Although this method is occasionally used already [20], the rapid potential variation of the core electrons can cause significant numerical issues. As will be shown in Chapter 2, the advantages of an all-electron potential can be benefited from without succumbing to the typical disadvantages.

1.2 Numerical Modeling

1.2.1 Plane Wave Expansion Methods

Plane wave expansion (PWE) methods are used for solving the ground states and electron density within the DFT framework [26]. They are particularly useful

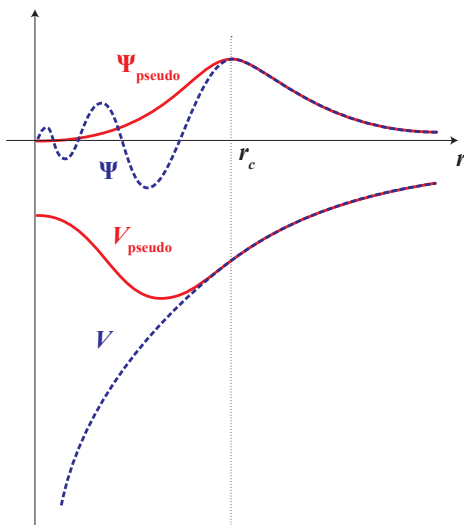


Figure 1.2. Representation of exact (dotted blue) and pseudo- (solid red) potentials and resulting wave function [24].

for periodic crystals, in which the calculations are greatly simplified because of Fast Fourier Transform (FFT) techniques. Because plane waves are eigenfunctions of the Schrödinger equation with constant potential, they can be used as a basis function for the nearly-free-electron approximation. Near the core, a constant potential is not an accurate representation; instead, it is described by a linear combination (expansion) of constant potentials. Pseudopotentials (EPM/APM) are often used to limit the expansion that needs to be performed.

1.2.2 Localized Orbital Methods

Localized orbital (LO) methods expand the wave function into a linear combination of energy-independent orbitals, each representing an atom within the crystal [27]. Basis functions are either a) gaussian or b) numerical. The most widely used LO method is the linear combination of atomic orbitals (LCAO) method. The LCAO method assumes that the number of molecular orbitals is equal to the number of atomic orbitals. So, for a system with n atomic orbitals, there would exist n molecular orbitals of the form:

$$\phi_i = \sum_{r=1}^n c_{ir} \chi_r, \quad \forall i \in [1, n], \quad (1.8)$$

where ϕ_i is the i^{th} molecular orbital, χ_r is the r^{th} atomic orbital, and c_{ir} is a weighting function. LCAO methods are widely used in quantum chemistry, although the methodology for choosing appropriate orbitals for accuracy and efficiency can be problematic.

1.2.3 Atomic Decomposition (Muffin-tin) Methods

Atomic decomposition (muffin-tin) methods, such as augmented plane wave (APW) and muffin-tin orbitals (MTO), separate the relatively constant interstitial regions from the rapidly varying atomic regions [28]. Since the 1930's, muffin-tin domain decomposition techniques [2] have been used, which, today, allow for parallel processing of the system. Embedded energy techniques [16] have also long been used to reduce the size of the system matrices. Unfortunately, these numerical techniques give rise to a reduced, but nonlinear, eigenvalue equation:

$$\hat{\mathbf{H}}(E)\psi = E\hat{\mathbf{S}}\psi. \quad (1.9)$$

In order to solve the resulting nonlinear eigenvalue problem, a second layer of approximation techniques, such as linear APW (LAPW) and linear MTO (LMTO) are generally used. In this thesis, an alternate approach is proposed using the FEAST [1] framework to bypass these traditionally unavoidable approximation techniques and solve directly for (1.9).

CHAPTER 2

METHODOLOGY

This proposed method solves the electronic structure problem of (1.9) without the previously necessary numerical approximations of other methods described in Section 1.2. This method uses a real-space muffin-tin domain decomposition to separate atoms from the interstitial region, as seen in Figure 2.1. Embedded self-energy (Green’s function) techniques are then used to ease the solution of the Schrödinger equation. The FEAST [1] eigenvalue solver is used to compute the equations that naturally arise. Not only does this method solve the electronic structure *exactly* within KS-DFT, it also offers a number of performance advantages due to its inherent ease of parallelization. In this chapter, specific meshing techniques, the FEAST algorithm, mathematical and matrix-level justifications are discussed.

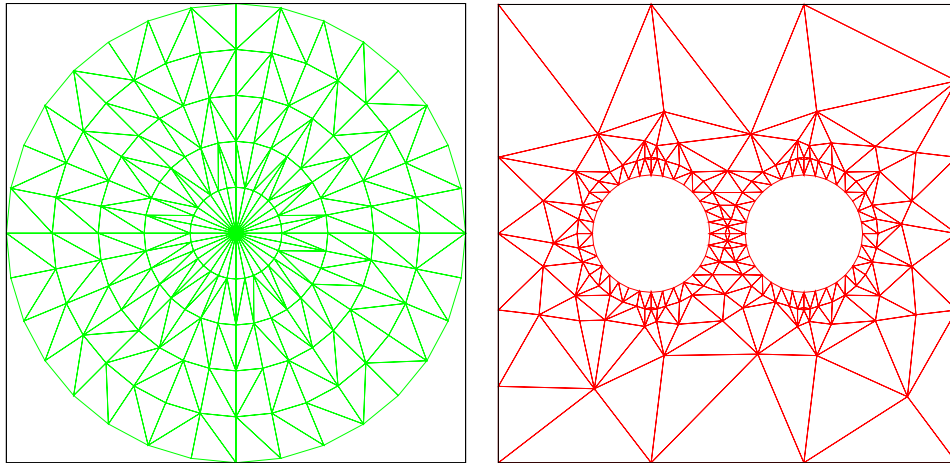


Figure 2.1. A 2D real-space muffin-tin domain, with a fine atomic mesh (left) and a coarse interstitial mesh (right). Created using Triangle [29].

2.1 Mesh and Real-Space Techniques

As will be shown, this method is not mesh dependent; however, a real-space mesh is helpful both for demonstration and for verification. Real-space techniques have a number of advantages over PW or LCAO approaches, including locality (which results in sparse system matrices) and the ability to incorporate adaptive local mesh refinements [20]. FEM is particularly advantageous because it preserves the hermitian properties of the Hamiltonian matrix for closed systems and produces natural Neumann boundary conditions that simplify the derivation of the self-energy functions [18].

TetGen [17], a tetrahedral mesh generator and 3D Delaunay triangulator, was used to create the FEM mesh, as seen in Figure 2.2. Fine atomic meshes were created by alternating layers of icositetrahedrons and rhombicuboctahedrons. The coarse interstitial region was then created to fill in the space between the atomic meshes. With this muffin-tin domain decomposition technique, it is possible to position the most points inside the atomic meshes, where the largest variations occur. By choosing higher-order basis functions, further refinement is possible. By choosing different basis functions for the atomic and interstitial meshes (to be discussed in Section 2.7), it is possible to increase accuracy around the atoms without greatly increasing the calculation domain.

2.2 FEAST Eigenvalue Solver

One of the major challenges for modeling any physical system is dealing with the nonlinear eigenvalue equations that arise. Modern linear solvers are robust and feature advanced error estimators, but this is not the case with nonlinear solvers [19]. Additionally, as simulated system-size and required accuracy both increase, scalability becomes increasingly important. The traditional method of choice is to

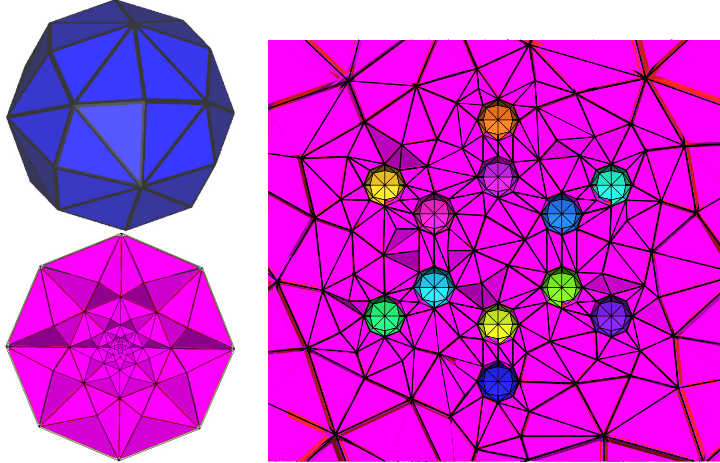


Figure 2.2. A 3D real-space muffin-tin domain decomposition, with the surface (top-left) and cut-out (bottom-left) of the fine atomic mesh and a coarse interstitial mesh with embedded atomic meshes (right).

linearize nonlinear eigenvalue problems; however, this further increases the system size and can lack the necessary accuracy for modern applications [19].

The FEAST [1] eigenvalue solver takes a difference approach, inspired from the density matrix representation and contour integration in quantum mechanics. Unlike traditional eigenvalue solvers, FEAST is based on performing a contour integration over the complex energy space, which, in the case of the embedding approach, replaces the nonlinear eigenvalue problem with a set of well-defined linear systems. With FEAST, the difficulty of solving an eigenvalue problem has been replaced with solving a linear system with multiple right-hand sides [18]. Before applying the FEAST algorithm for solving (1.9) (described in Section 2.3), a brief summary [15] of the computing steps for solving (1.7) is outlined below.

Starting with a set of M_0 linearly independent random vectors $\mathbf{Y}_{N \times M_0} = \{y_1, y_2, \dots, y_{M_0}\}$, where M_0 is chosen to be greater than the number of eigenvalues M in the search interval (i.e. M_0 represents the over-estimation of M which is not known *a priori*), a new set of vectors $\mathbf{Q}_{N \times M_0} = \{q_1, q_2, \dots, q_{M_0}\}$ is obtained as follows:

$$\mathbf{Q}_{N \times M_0} = -\frac{1}{2\pi i} \int_C dZ \mathbf{G}(Z) \mathbf{Y}_{N \times M_0}, \quad (2.1)$$

where C represents a complex contour from E_{min} to E_{max} . In practice, the vectors \mathbf{Q} in (2.1) can be computed using a high-order numerical integration where only very few linear systems $\mathbf{G}(Z)\mathbf{Y}$ need to be solved along a complex contour C [1] i.e.

$$(Z\mathbf{S} - \mathbf{H})\mathbf{Q}^{(Z)} = \mathbf{Y}, \quad (2.2)$$

where $\mathbf{Q}^{(Z)}$ denotes the set of responses at a given pivot energy Z for a given set of excitations \mathbf{Y} in the entire computational domain Ω .

Subsequently, by computing

$$\mathbf{H}_{\mathbf{Q}_{M_0 \times M_0}} = \mathbf{Q}^\dagger \mathbf{H} \mathbf{Q} \quad (2.3)$$

and

$$\mathbf{S}_{\mathbf{Q}_{M_0 \times M_0}} = \mathbf{Q}^\dagger \mathbf{S} \mathbf{Q}, \quad (2.4)$$

a projected reduced dense eigenvalue problem of size M_0 can be formed:

$$\mathbf{H}_{\mathbf{Q}} \Phi = \epsilon \mathbf{S}_{\mathbf{Q}} \Phi. \quad (2.5)$$

This reduced problem can be solved using standard eigenvalue routines for dense systems to obtain all the eigenpairs (ϵ_m, Φ_m) . By setting $E_m = \epsilon_m$ and computing $\Psi_{N \times M_0} = \mathbf{Q}_{N \times M_0} \Phi_{M_0 \times M_0}$, it follows that if E_m lies inside the contour, it is an eigenvalue solution and its eigenvector is Ψ_m (the m^{th} column of Ψ). The eigenvectors Ψ are also naturally \mathbf{S} -orthonormal if the eigenvectors of the reduced problem are $\mathbf{S}_{\mathbf{Q}}$ -orthonormal. Accuracy can be systematically improved using a new set of initial guess vectors $\mathbf{Y} = \mathbf{S}\Psi$ iteratively up until convergence (usually obtained in 2-3 iterations).

2.3 Mathematical Methodology

Mathematical justification for this technique is based heavily on work by [16], which presents a method for embedding with Neumann boundary conditions (the boundary conditions used for this method). If we consider the interstitial region Ω_0 and atomic regions $\Omega_j(j > 0)$, it can be shown that (1.9) can be solved based solely on calculations performed in Ω_0 , given the proper boundary conditions Γ_j for each Ω_j interface. We are then left to solve:

$$\mathbf{H}_0\Psi(x) = E\mathbf{S}\Psi(x), \quad x \in \Omega_0 \quad (2.6)$$

where \mathbf{H}_0 is the Hamiltonian of Ω_0 . From Green's theorem, we are able to relate $\Psi(x)$ to the appropriate boundary conditions Γ_j :

$$\frac{1}{2} \frac{\partial \Psi(x)}{\partial \eta_j} = \int_{\Gamma_j} dx' \Sigma_j(E, x, x') \Psi(x'), \quad x \in \Gamma_j (\forall j) \quad (2.7)$$

where $\hbar = m = 1$, η_j is the external normal at Γ_j , and Σ_j is the self-energy function, which can be found by the Green's function G_j :

$$(E - H_j)G_j(E, x, x') = \delta(x - x'), \quad x, x' \in \Omega_j (\forall j) \quad (2.8)$$

where \mathbf{H}_j is the Hamiltonian of Ω_j and G_j can be constructed with arbitrary boundary conditions at Γ_j . For this method, homogeneous Neumann boundary conditions (G_j with zero derivative on Γ_j) were used, resulting in the simplified expression for the self-energy Σ_j :

$$\Sigma_j(E, x, x') = G_j^{-1}(E, x, x'), \quad x, x' \in \Gamma_j (\forall j) \quad (2.9)$$

After discretization of 2.6 using the condition set in (2.7), the general nonlinear eigenvalue problem to be solved in Ω_0 takes the form:

$$(\mathbf{H}_0 - \bigcup_j \Sigma_j(E))\Psi_0 = E\mathbf{S}_0\Psi_0, \quad (2.10)$$

where \mathbf{S}_0 is the mass matrix in Ω_0 and \bigcup denotes a summation of matrices that have been reordered or reduced.

Up to this point, this method is completely independent of the basis. For a real-space FEM discretization, further simplifications can be made. Because $\Sigma_j(E)$ is non-zero for only a small number of matrix elements coupling all the unknown n_j on Γ_j , (2.8) and (2.9) can be used to produce:

$$\Sigma_j(E) = [\mathbf{G}_j]_{\Gamma_j} = ([\mathbf{I}_{n_j} 0 \dots 0](E\mathbf{S}_j - \mathbf{H}_j)^{-1}[\mathbf{I}_{n_j} 0 \dots 0]^T)^{-1}, \quad (2.11)$$

where \mathbf{S}_j is the mass matrix in Ω_j and the matrix Σ_j of size n_j contains the self-energy elements.

The resulting energy-dependent nonlinear eigenvalue problem (2.10) cannot be handled by traditional linear eigenvalue solvers. Although this problem is not impossible [30],[31], it remains practically very challenging. Within FEAST, however, (2.10) can be solved in the following manner:

Starting with a set of excitations $Y(x)$ in the continuum domain, the set of responses $Q^{(Z)}$ can be obtained by solving (2.6) only in Ω_0 :

$$(Z - H_0)Q^{(Z)}(x) = Y(x), \quad x \in \Omega_0 \quad (2.12)$$

where the boundary condition for $Q^{(Z)}$ on Γ_j should formally satisfy (2.7) augmented by a source term $-F_j^{(Z)}(x)$ (added to the right-hand side), which accounts for the effects of the excitations $Y(x)$ within the atomic regions Ω_j . For instance, using Neumann boundary conditions for G_j , the self-energies Σ_j are defined in (2.9) and it can be shown that:

$$F_j^{(Z)}(x) = \int_{\Gamma_j} dx' G_j^{-1}(Z, x, x') \int_{\Omega_j} dx'' G_j(Z, x', x'') Y(x''), \quad x \in \Gamma_j (\forall j). \quad (2.13)$$

Once $Q^{(Z)}$ is known in Ω_0 (and, therefore, all the Γ_j interfaces), the solutions in the Ω_j domains can be independently retrieved by solving the linear systems:

$$(Z - H_j)Q^{(Z)}(x) = Y(x), \quad x \in \Omega_j (\forall j), \quad (2.14)$$

with Dirichlet boundary conditions.

After discretization of (2.12), (2.7), and (2.13), solving (2.2) across the entire domain Ω can be replaced *exactly* by solving the linear system:

$$(Z\mathbf{S}_0 - \mathbf{H}_0 + \bigcup_j \Sigma_j(Z))\mathbf{Q}_0^{(Z)} = \mathbf{Y}_0 + \bigcup_j \mathbf{F}_j^{(Z)}, \quad (2.15)$$

where

$$\mathbf{F}_j^{(Z)} = \Sigma_j(Z)[\mathbf{G}_j(Z)\mathbf{Y}_j]_{\Gamma_j} \quad (2.16)$$

for the unknown components of the solutions $\mathbf{Q}_0^{(Z)}$ in Ω_0 and with a series of independent sub-problems (2.14) to obtain unknown components of the solutions $\mathbf{Q}_j^{(Z)}$ in the atom-centered regions Ω_j . Subsequently, the subspace \mathbf{Q} of (2.1) is obtained by integration of the set of solutions $\mathbf{Q}_0^{(Z)}$ and all $\mathbf{Q}_j^{(Z)}$ over the complex contour C . In practice, it is possible to construct \mathbf{H}_Q and \mathbf{S}_Q of (2.3) and (2.4), respectively, directly from the projection of \mathbf{H}_0 and \mathbf{S}_0 for Ω_0 and \mathbf{H}_j and \mathbf{S}_j for all Ω_j . Because (2.15) is solved only for a specific set of complex pivot energies Z , the nonlinearity of (2.10) in Ω_0 is then naturally removed.

2.4 Discrete Matrix-level Description

For clarity, the method outlined in Section 2.3 shall also be described in a discrete matrix-level form. Presented in this way, the benefits of FEM, as well as specific optimizations (such as non-refactorization methods) are more easily understood.

2.4.1 General Approach

Here, the goal is to solve for the discrete form of (2.14):

$$\mathbf{A}_j \begin{pmatrix} x_1 \\ x \end{pmatrix} = \begin{pmatrix} y_1 \\ y \end{pmatrix}. \quad (2.17)$$

The atomic system matrix \mathbf{A}_j is defined as:

$$\mathbf{A}_j = [\mathbf{Z}\mathbf{S}_j - \mathbf{H}_j] = \left(\begin{array}{c|cc} A_1 & A_{sa} & 0 \\ \hline A_{as} & & \\ 0 & & A_2 \end{array} \right), \quad (2.18)$$

where A_{as} is the transpose of A_{sa} . Here, the subscripts a and s represent atomic and surface interactions, respectively. The submatrix A_1 is of size $n_s \times N$ and the submatrix A_{as} is of size $n_s \times \hat{n}_s$.

The discrete Green's function \mathbf{G}_j of (2.8) can then be defined as:

$$\mathbf{G}_j = \mathbf{A}_j^{-1} = \left(\begin{array}{c|cc} G_{ss} & G_{sa} & \\ \hline G_{as} & G_{aa} & \end{array} \right). \quad (2.19)$$

Having defined the discrete matrix-level parameters, the system can, in general, be solved in the following manner:

1) Factorize: Using a SPIKE [38] factorization, the matrix \mathbf{A}_j can be written as:

$$\mathbf{A}_j = \left(\begin{array}{c|cc} \mathbf{I} & 0 & \\ \hline 0 & A_2 & \end{array} \right) \left(\begin{array}{c|cc} A_1 & A_{sa} & 0 \\ \hline W & \mathbf{I} & \end{array} \right), \quad (2.20)$$

where an LU factorization has been performed on the submatrix A_2 and W is obtained by solving:

$$A_2W = \begin{pmatrix} A_{as} \\ 0 \end{pmatrix}. \quad (2.21)$$

2) Solve: With the above factorization, it can be seen that solving $[Z\mathbf{S}_j - H_j]x = y$ is the same as solving:

$$\left(\begin{array}{c|cc} A_1 & A_{sa} & 0 \\ \hline W & \mathbf{I} & \end{array} \right) \begin{pmatrix} x_1 \\ x \end{pmatrix} = \begin{pmatrix} y_1 \\ A_2^{-1}y \end{pmatrix}. \quad (2.22)$$

We can now solve the following reduced system:

$$\left(\begin{array}{c|c} A_1 & A_{sa} \\ \hline W_T & I_{n_j} \end{array} \right) \begin{pmatrix} x_1 \\ x_T \end{pmatrix} = \begin{pmatrix} y_1 \\ (A_2^{-1}y)_T \end{pmatrix}, \quad (2.23)$$

where $W_T = [I_{n_j}, 0 \dots 0]W$, $x_T = [I_{n_j}, 0 \dots 0]x$, and $(A_2^{-1}y)_T = [I_{n_j}, 0 \dots 0](A_2^{-1}y)$.

3) Retrieve: Finally, x can be retrieved by solving:

$$x = A_2^{-1}y - Wx_1. \quad (2.24)$$

2.4.2 Application

The reordered self-energy Σ_j , which was defined in (2.11), can be repeated here, using the new notation, as:

$$\Sigma_j = G_{ss}^{-1} = ([\mathbf{I}_{n_j} 0 \dots 0] \mathbf{A}^{-1} [\mathbf{I}_{n_j} 0 \dots 0]^T)^{-1}. \quad (2.25)$$

In this particular case, $y_1 = \mathbf{I}$ and $y = 0$ in (2.17). Because of these conditions, $x_1 = G_{ss}^{-1} = \Sigma_j$. By solving (2.23) for x_1 with this new right-hand-side, it can be shown that:

$$\Sigma_j = (A_1 - A_{sa}W_T)^{-1}. \quad (2.26)$$

The interstitial system $\mathbf{A}_0 = [Z\mathbf{S}_0 - \mathbf{H}_0]$ is then updated (as seen in (2.15)) to include the self-energy Σ_j :

$$\hat{\mathbf{A}}_0 = \mathbf{A}_0 + \bigcup_j \Sigma_j. \quad (2.27)$$

To obtain F_j in (2.16), (2.17) is solved using $y_1 = 0$ (since the source term at the interface is already included in (2.15)) and $y = \hat{Y}_j$. One can then show that:

$$\mathbf{F}_j = -\mathbf{W}^\dagger \hat{\mathbf{Y}}_j, \quad (2.28)$$

where W^\dagger is the transpose of the entire SPIKE matrix W .

Having defined all of the above terms, \mathbf{Q}_0 can now be obtained by solving the linear system (2.15). Finally, \mathbf{Q}_j can be found by solving the linear system:

$$\left(\begin{array}{c|c} \mathbf{I} & 0 \\ \hline 0 & A_2 \end{array} \right) \begin{pmatrix} \mathbf{Q}_0 \\ \mathbf{Q}_j \end{pmatrix} = \begin{pmatrix} \mathbf{Q}_0 \\ y_j - \bigcup A_{sa} \hat{\mathbf{Q}}_0 \end{pmatrix}, \quad (2.29)$$

which implies that:

$$A_2 \mathbf{Q}_j = [y_j - \bigcup A_{sa} \hat{\mathbf{Q}}_0]. \quad (2.30)$$

2.5 Implementation of FEAST-RCI

The FEAST reverse communication (FEAST-RCI) [37] is used to solve (1.9). Algorithm 1 shows the pseudocode for implementing FEAST-RCI. The *ijob* parameter is initialized to -1. Once FEAST is called, *ijob* is used to identify the FEAST operation to be performed. When *ijob* equals 0, the code will exit FEAST. Within FEAST, there are several layers used to solve the given system for a given Z_e . Case(10) first solves for the self-energies Σ_j by solving (2.11) for all j . The interstitial Hamiltonian H_0 is then updated to include Σ_j , as seen in (2.15). The interstitial system is then

factorized. Case(11) first builds F_j using (2.16) and then solves the interstitial system (2.15). The solution for the atomic systems (the discrete form of (2.14)) are then solved using Dirichlet boundary conditions. The matrix-vector multiplication needed in (2.3) and (2.4) is performed in case(30) and case(40).

Algorithm 1 Pseudocode for implementing FEAST reverse communication interface (FEAST-RCI).

```

ijob = -1 ▷ Initialization
while ijob ≠ 0 do
  DFEAST_SRCI
  select case (ijob)
  case (10) ▷ Factorization
    for j Atoms do
       $\Sigma_j(E) = ([\mathbf{I}_{n_j} \ 0 \dots 0](E\mathbf{S}_j - \mathbf{H}_j)^{-1}[\mathbf{I}_{n_j} \ 0 \dots 0]^T)^{-1}$ 
    end for
    Update and factorize interstitial system  $(Z\mathbf{S}_0 - \mathbf{H}_0 + \sum_j \Sigma_j(Z))$ 

  case (11) ▷ Solve
    for j Atoms do
       $\mathbf{F}_j^{(Z)} = \Sigma_j(Z)\mathbf{G}_j(Z)\mathbf{Y}_j$ 
    end for
    Update right-hand side  $\mathbf{Y}_0 + \sum_j \mathbf{F}_j^{(Z)}$ 
    Solve  $(Z\mathbf{S}_0 - \mathbf{H}_0 + \sum_j \Sigma_j(Z))\mathbf{Q}_0^{(Z)} = \mathbf{Y}_0 + \sum_j \mathbf{F}_j^{(Z)}$  for  $\mathbf{Q}_0^{(Z)}$ 
    for j Atoms do
      Solve  $(Z\mathbf{S} - \mathbf{H}_j)\mathbf{Q}_j^{(Z)}(x) = \mathbf{Y}(x)$  for  $\mathbf{Q}_j^{(Z)}$ 
    end for

  case (30) ▷ Matrix-vector multiplication
    Perform  $\mathbf{H}\mathbf{Q}$  multiplication

  case (40) ▷ Matrix-vector multiplication
    Perform  $\mathbf{S}\mathbf{Q}$  multiplication

  end select

end while

```

2.6 Parallelization

As has been mentioned, this method naturally lends itself to parallelization. There are three main levels of parallelization that this method can take advantage of:

Level 1: The search interval within FEAST can be separated into many small, independent intervals, allowing them to be searched in parallel.

Level 2: For each search interval, there are N_e independent linear equations to be solved, which can be solved within FEAST in parallel.

Level 3: Because of the muffin-tin domain decomposition, it is possible to solve for the self-energies of the atomic regions Σ_j independently and in parallel.

Levels 1 and 2 of parallelism will be handled in FEAST and are not considered as part of this thesis, although they are important to consider for ultimate scalability. Level 3 has been implemented using OpenMP [36], allowing each atom to be solved on an individual core. Section 3.7 gives an in-depth analysis of the effects of parallelization on numerical experiments as well as the limitations of such an implementation. Due to restrictions of available hardware, fully-parallel level 3 numerical examples are limited to 8 atoms, allowing each atom to be solved on its own thread.

2.7 Hybrid Basis

As has been mentioned, the proposed method is independent of mesh type or basis. This provides for a very flexible system in which a specific basis set can be chosen for a particular problem. One example of this is using a spectral decomposition of energy levels instead of the fine atomic mesh. In this particular example, the Green's function $G(Z)$ is defined as:

$$G(Z) = \sum_{i=1}^M \frac{|X_i\rangle \langle X_i|}{Z - E_i} = \sum_{i=1}^M \frac{X_i X_i^\dagger}{Z - E_i}, \quad (2.31)$$

where M is the size of the basis and X_i^\dagger is the transpose of X_i .

As can be seen from Table 2.1, an increase in basis size increases accuracy. It should also be noted that the valence electrons (until they enter the interstitial region) are significantly more accurate than the core electrons, which is to be expected based on how this basis was built. Figure 2.3 shows how accuracy changes as a function of energy level. This particular hybrid basis could be used as a substitute for pseudopotential methods, or as a preconditioner for the exact method being proposed.

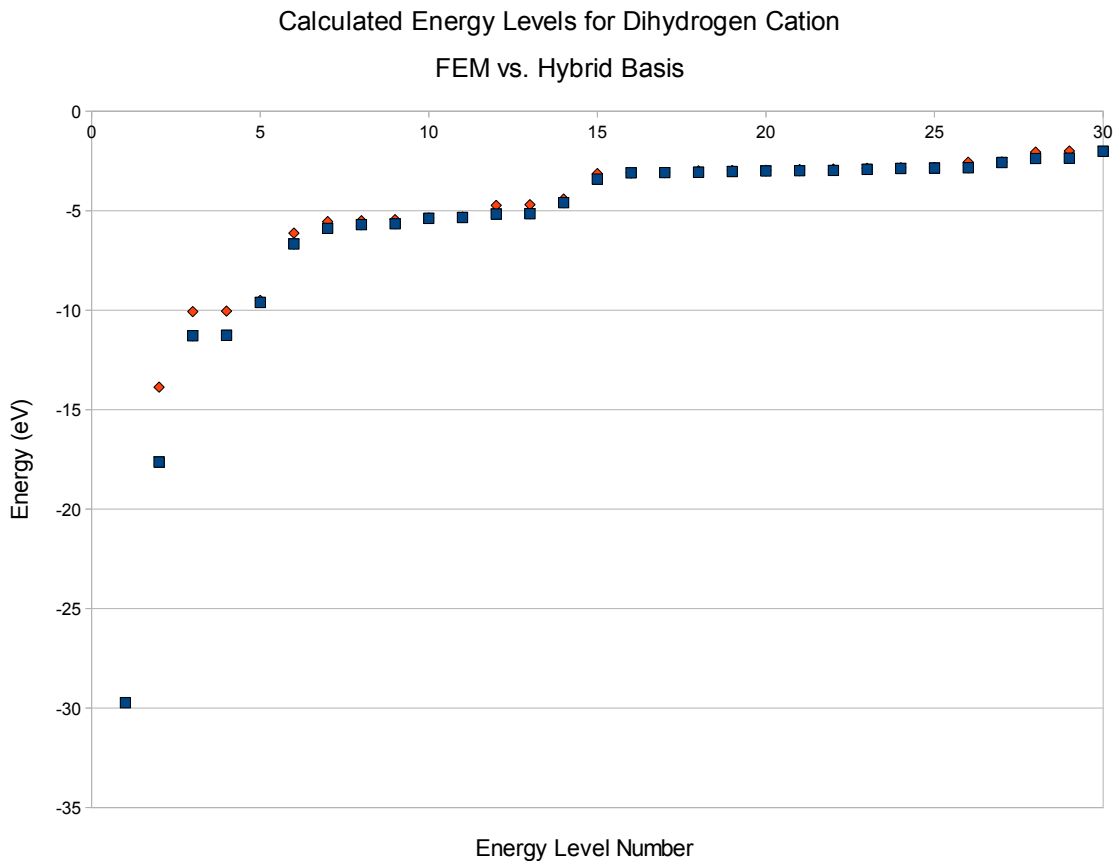


Figure 2.3. Energy levels for H_2^+ using P_2 FEM (blue), $M=9$ Hybrid (red), and $M=598$ Hybrid (Green).

FEM (P_2) (eV)	Hybrid (M=9) (eV)	Percent Error	Hybrid (M=598) (eV)	Percent Error
-29.7292	-	100.00%	-29.7230	0.02%
-17.6324	-13.8671	21.35%	-17.6241	0.05%
-11.2823	-10.0701	10.74%	-11.2820	0.00%
-11.2620	-10.0449	10.81%	-11.2617	0.00%
-9.6141	-9.5083	1.10%	-9.6126	0.02%
-6.6746	-6.1267	8.21%	-6.6725	0.03%
-5.8870	-5.5483	5.75%	-5.8868	0.00%
-5.6989	-5.5061	3.38%	-5.6989	0.00%
-5.6606	-5.4507	3.71%	-5.6605	0.00%
-5.3876	-5.3672	0.38%	-5.3876	0.00%
-5.3395	-5.3173	0.42%	-5.3395	0.00%
-5.1699	-4.7342	8.43%	-5.1698	0.00%
-5.1551	-4.6971	8.88%	-5.1550	0.00%
-4.5956	-4.4154	3.92%	-4.5951	0.01%
-3.4126	-3.1336	8.18%	-3.4117	0.03%
-3.0993	-3.0888	0.34%	-3.0993	0.00%
-3.0876	-3.0805	0.23%	-3.0876	0.00%
-3.0705	-2.9963	2.41%	-3.0704	0.00%
-3.0308	-2.9780	1.74%	-3.0308	0.00%
-3.0023	-2.9690	1.11%	-3.0022	0.00%
-2.9913	-2.9306	2.03%	-2.9913	0.00%
-2.9764	-2.9113	2.19%	-2.9764	0.00%
-2.9226	-2.8665	1.92%	-2.9226	0.00%
-2.8771	-2.8407	1.27%	-2.8771	0.00%
-2.8574	-2.8279	1.03%	-2.8574	0.00%
-2.8411	-2.5597	9.90%	-2.8411	0.00%
-2.5817	-2.5505	1.21%	-2.5817	0.00%
-2.3776	-2.0545	13.59%	-2.3775	0.00%
-2.3624	-2.0014	15.28%	-2.3623	0.00%
-2.0167	-	100.00%	-2.0164	0.02%

Table 2.1. Calculated energy levels for H_2^+ using quadratic (P_2) FEM and spectral decomposition hybrid basis sets of 9 and 598.

CHAPTER 3

NUMERICAL EXAMPLES

In this chapter, the numerical results for H, H₂⁺, H₃⁺⁺, CH₄, C₂H₆, and C₆H₆ are presented. Results for full and muffin-tin meshes are compared to analytic (when available) or numerical solutions for verification purposes. The chapter concludes with a discussion on accuracy and scalability based on the presented results. For all examples, a 16Å×16Å×16Å box was used for the interstitial region. The atomic region had a radius of 0.35Å.

3.1 Hydrogen

Figure 3.1 shows the mesh of the Hydrogen atom used in the first numerical example. The total matrix size for the full mesh can be calculated by:

$$N_{Full} = (N_{at} - N_s)At + N_{it}, \quad (3.1)$$

where relevant variables are defined in Table 3.1. It should be noted the total system size does not change based on the mesh type. Table 3.2 compares the matrix size (number of nodes) for full and muffin-tin meshes using quadratic (P₂) and cubic (P₃) refinement. Table 3.3 compares the calculated results to the analytic solution [33].

N_{Full}	Total number of nodes for full mesh
N_{at}	Number of nodes in atomic mesh
N_s	Number of nodes at interface
At	Number of atoms
N_{it}	Number of nodes in interstitial mesh

Table 3.1. Previously undefined variables for (3.1).

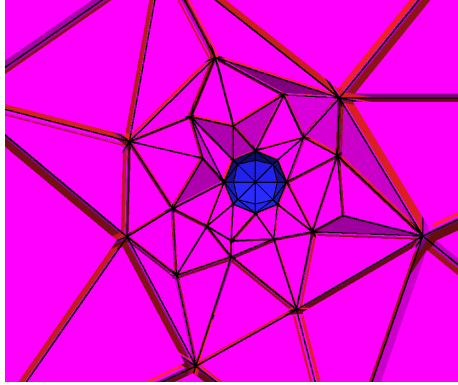


Figure 3.1. A 3D mesh containing an H atom, surrounded by a coarse interstitial mesh.

	Full Mesh	Muffin-tin Mesh			
	Matrix Size	Atomic	Interstitial	Interface	Total
P_2	4235	2841	1492	98	4235
P_3	13976	9457	4737	218	13975

Table 3.2. Matrix sizes for Hydrogen atom using full and muffin-tin meshes.

	Energy(ev)	Percent Error
P_2	-13.2414758254649	2.68%
P_3	-13.56387279724017	0.31%
Exact [33]	-13.6056925328194	-

Table 3.3. Calculated and exact 1st energy level for Hydrogen.

3.2 Dihydrogen Cation

Figure 3.2 shows the mesh of the Dihydrogen Cation used in the second numerical example. The atoms were separated by a distance of 2au (approximately 1.06Å). Table 3.4 compares the matrix size (number of nodes) for full and muffin-tin meshes using quadratic (P_2) and cubic (P_3) refinement. As with H, the total system size does not change between based on the mesh type. Table 3.5 compares the calculated results to the analytic solution [34].

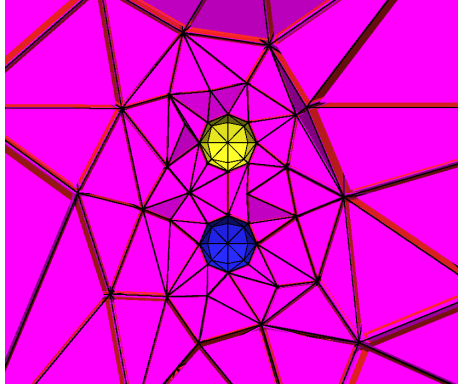


Figure 3.2. A 3D mesh containing an H_2^+ molecule, surrounded by a coarse interstitial mesh.

	Full Mesh	Muffin-tin Mesh			
	Matrix Size	Atomic	Interstitial	Interface	Total
P_2	7378	2841	1892	98	7378
P_3	24528	9457	6050	218	24528

Table 3.4. Matrix sizes for Dihydrogen Cation molecule using full and muffin-tin meshes.

	Energy(ev)	Percent Error
P_2	-29.7291582444973	0.92%
P_3	-29.9795763567402	0.08%
Exact [34]	-30.00420407	-

Table 3.5. Calculated and exact 1st energy level for Dihydrogen Cation.

3.3 Trihydrogen Dication

Figure 3.3 shows the mesh of the Trihydrogen Dication molecule used in the third numerical example. The atoms were separated by a distance of 1.68au (approximately 0.89Å). It should be noted that although H_3^{++} is not stable enough to exist [35], it is still useful for verification purposes. Table 3.6 compares the matrix size (number of nodes) for full and muffin-tin meshes using quadratic (P_2) and cubic (P_3) refinement. As with H and H_2^+ the total system size does not change between based on the mesh type. Table 3.7 compares the calculated results to the analytic solution [34].

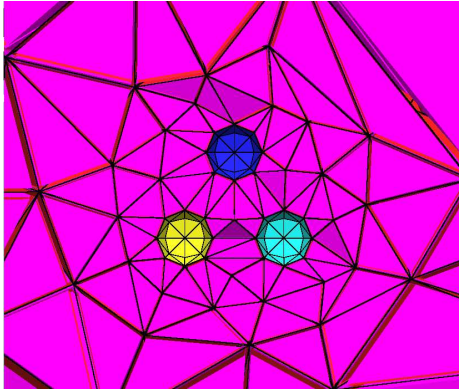


Figure 3.3. A 3D mesh containing an H_3^{++} molecule, surrounded by a coarse interstitial mesh.

	Full Mesh	Muffin-tin Mesh			
	Matrix Size	Atomic	Interstitial	Interface	Total
P_2	10430	2841	2201	98	10430
P_3	34791	9457	7074	218	34791

Table 3.6. Matrix sizes for Trihydrogen Dication molecule using full and muffin-tin meshes.

	Energy(ev)	Percent Error
P_2	-51.6733672948024	0.56%
P_3	-51.9378291242234	0.05%
Estimated Exact [34]	-51.966	-

Table 3.7. Calculated and estimated exact 1st energy level for Trihydrogen Dication.

3.4 Methane

Figure 3.4 shows the mesh of the Methane molecule used in the fourth numerical example. Table 3.8 compares the matrix size (number of nodes) for full and muffin-tin meshes using quadratic (P_2) and cubic (P_3) refinement. As with the Hydrogen molecules, the total system size does not change between based on the mesh type.

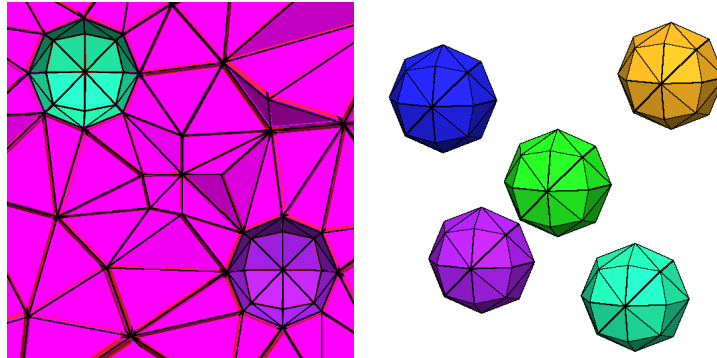


Figure 3.4. A top-down view embedded within a coarse mesh (left) and a side view (right) of a CH_4 molecule.

	Full Mesh	Muffin-tin Mesh			
	Matrix Size	Atomic	Interstitial	Interface	Total
P_2	16950	2841	3235	98	16950
P_3	56681	9457	16705	218	56681

Table 3.8. Matrix sizes for Methane molecule using full and muffin-tin meshes.

3.5 Ethane

Figure 3.5 shows the mesh of the Ethane molecule used in the fifth numerical example. Table 3.9 compares the matrix size (number of nodes) for full and muffin-tin meshes using quadratic (P_2) and cubic (P_3) refinement. As with the other examples, the total system size does not change between based on the mesh type.

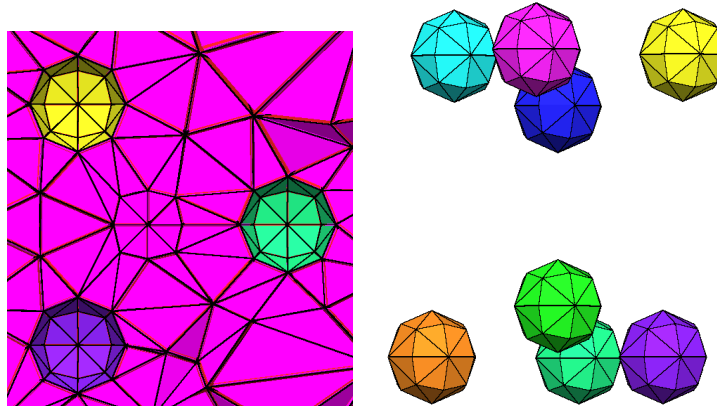


Figure 3.5. A top-down view embedded within a coarse mesh (left) and a side view (right) of a C_2H_6 molecule.

	Full Mesh	Muffin-tin Mesh			
	Matrix Size	Atomic	Interstitial	Interface	Total
P_2	27085	2841	5141	98	27085
P_3	90617	9457	16705	218	90617

Table 3.9. Matrix sizes for Ethane molecule using full and muffin-tin meshes.

3.6 Benzene

Figure 3.6 shows the mesh of the Benzene molecule used in the final numerical example. Table 3.10 compares the matrix size (number of nodes) for full and muffin-tin meshes using quadratic (P_2) and cubic (P_3) refinement. As with the other examples, the total system size does not change between based on the mesh type. An analytic solution for the electronic structure is not available for comparison; instead, Table 3.11 compares the calculated result to other published numerical solutions [42],[43].

	Full Mesh	Muffin-tin Mesh			
	Matrix Size	Atomic	Interstitial	Interface	Total
P_2	39881	2841	6965	98	39881
P_3	133579	9457	22711	218	133579

Table 3.10. Matrix sizes for Benzene molecule using full and muffin-tin meshes.

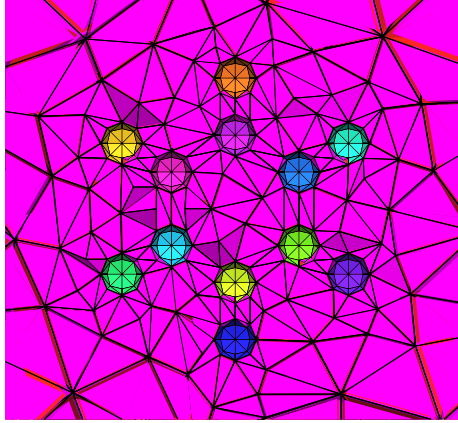


Figure 3.6. A 3D mesh containing an C_6H_6 molecule, surrounded by a coarse interstitial mesh.

	Energy(ev)	Percent Difference
P_3	-6261.14	-
HF (STO-3G)[42]	-6201.24	0.96%
DFT-LSDA (STO-3G)[42]	-6206.31	0.88%
P_3 ELMER [43]	-6262.57	0.02%
FHI-aims [43]	-6263.83	0.04%

Table 3.11. Calculated and numerical estimate for 1st energy level for Benzene.

3.7 Discussion

A few conclusions can be drawn from the results in Sections 3.1 - 3.5. First, it is important to note that accuracy increases as refinement increases; in fact, the accuracy can be arbitrarily increased (within the limitations of DFT) by increasing refinement. Additionally, results are the same for the full and muffin-tin meshes.

Second, although results are the same for both mesh types, scalability is not. Figure 3.7 shows a plot of the number of nodes for full, atomic, and interstitial meshes versus number of atoms per molecules. For the full mesh, as the number of atoms increases, the entire system scales linearly; however, for the muffin-tin mesh, only the interstitial region scales linearly and the atomic mesh remains constant. Because the atomic mesh contains the most points, great scalability is possible.

Figures 3.8 and 3.9 show the relative speed-up times for the factorization and solve steps, respectively. For these timing simulations, the number of threads was set equal to the number of atoms. Because of this, we would expect both the factorization and solve speed-ups to remain equal to 1. As can be seen, this is not quite the case. The factorization and solve steps are both affected by hardware limitations. Because of the shared resources between threads, it's impossible to achieve complete parallelization with the current system. Overall factorization time is particularly affected, as expected, by the interstitial factorization time, which can be seen in Figure 3.10. We note that, for very small systems, factorization time actually decreases. This is caused by a decrease in the number of nodes per the number of threads for those particular systems. Although the nodes/threads continues to decrease, factorization time begins increasing for larger systems. This is due to limitations within PARDISO [41], which is being used to factorize the system.

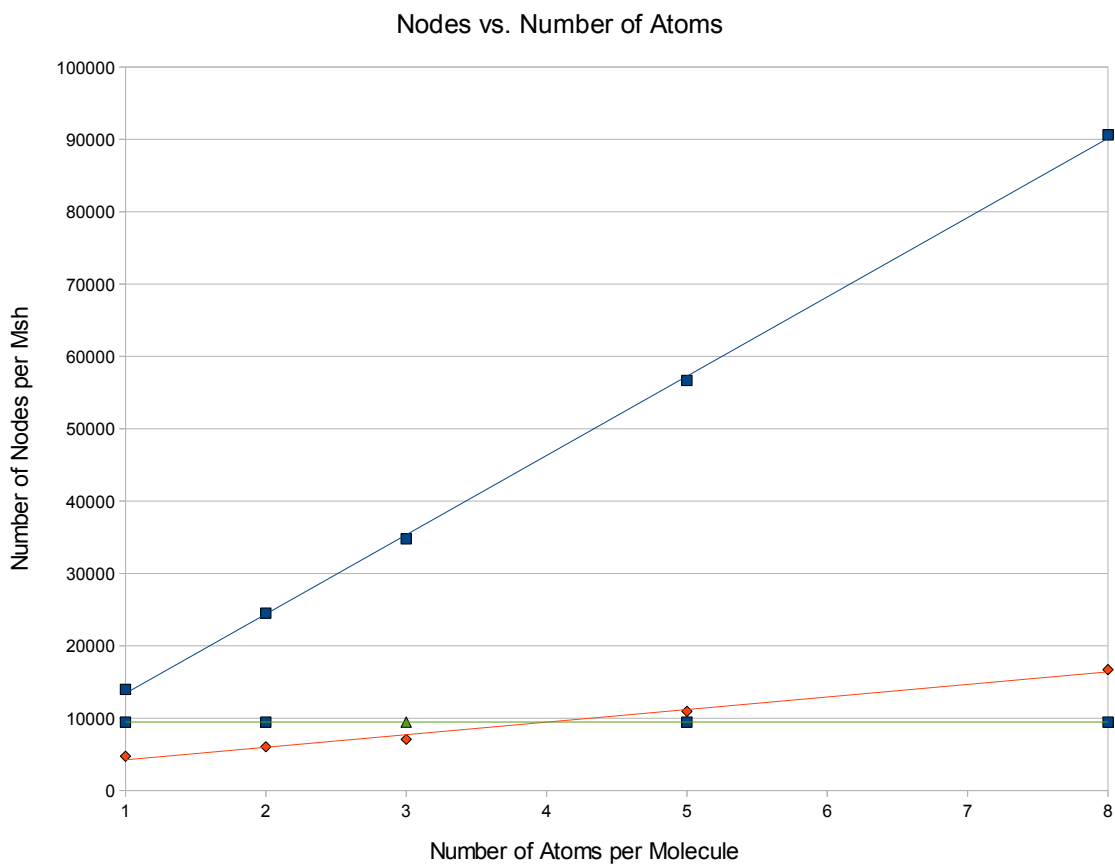


Figure 3.7. Number of nodes for full (blue), atomic (green), and interstitial (red) meshes versus number of atoms per molecule.

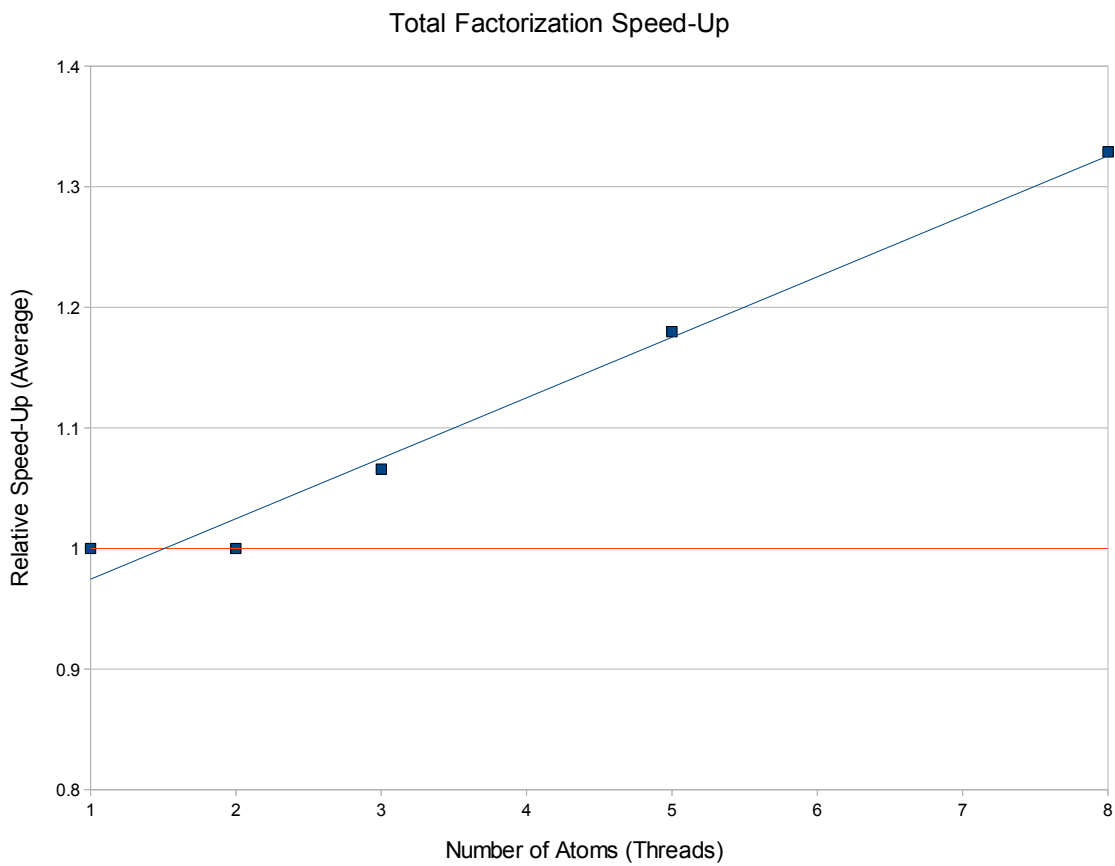


Figure 3.8. Actual (blue) and ideal (red) relative speed-up times for factorization vs. number of atoms (threads). Here, 1 corresponds to 2.28 seconds.

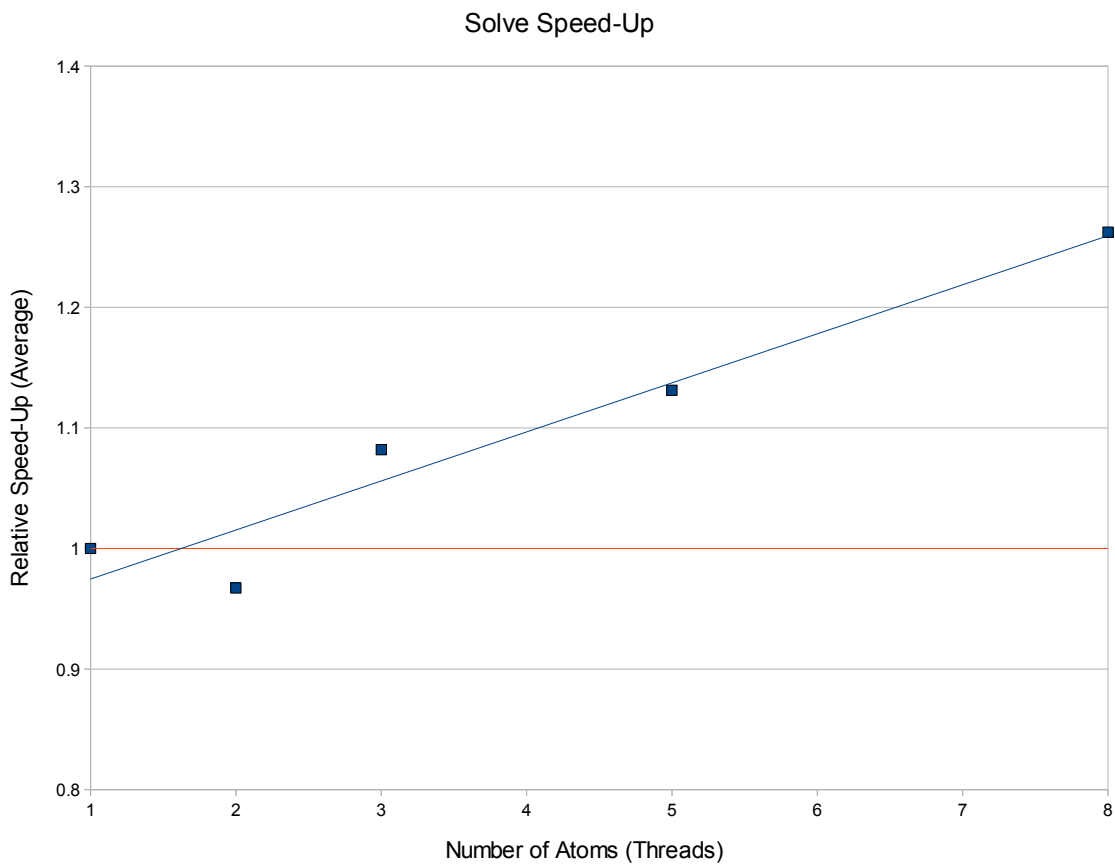


Figure 3.9. Actual (blue) and ideal (red) relative speed-up time for solve vs. number of atoms (threads). Here, 1 corresponds to 0.61 seconds.

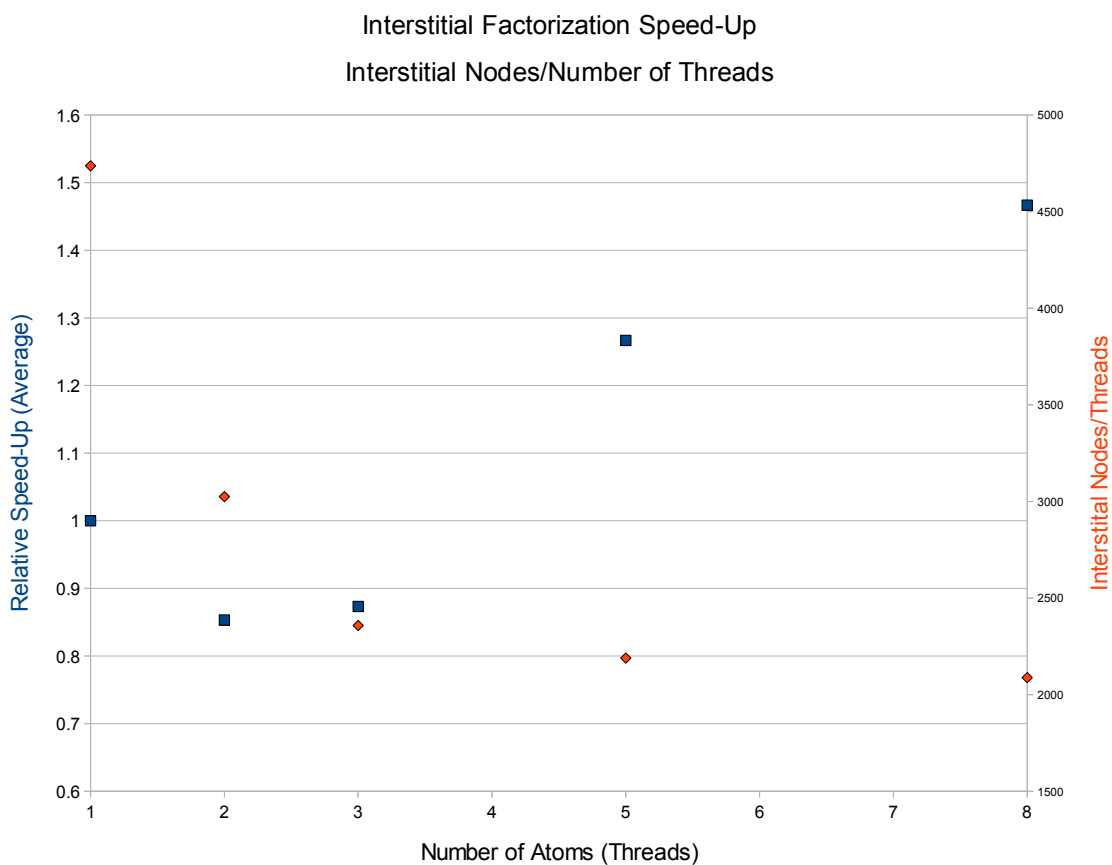


Figure 3.10. Actual relative speed-up time for interstitial solve step (blue) and interstitial nodes per number of threads (red) vs. number of atoms (threads). Here, 1 corresponds to 0.15 seconds.

CONCLUSION

In this thesis, an accurate and scalable method for performing nanoelectronic structure calculations has been presented. Built within the FEAST [1] framework, this method is based on muffin-tin methods originally presented by Slater in 1937 [2] and on the resulting embedded self-energies (Green’s functions). The most common traditional numerical techniques, such as plane wave expansion, linearized augmented plane waves, linearized muffin-tin orbitals, and linear combination of atomic orbitals, were briefly presented. Although these methods have been widely used with varying success, they all suffer from the added layer of approximation to linearize the resulting eigenvalue problem, or they rely on pseudopotential methods to ease the potential around the cores. These approximation techniques make these methods intrinsically less accurate than the method proposed in this thesis.

Although this method is mesh-independent, FEM was heavily focused on, as it provides a number of benefits, such as preserving the hermitian properties of the Hamiltonian, as described in Chapter 2. A rigorous mathematical explanation along with a matrix-level example provided justification for the proposed method, in which it was shown how Green’s functions were used in the form of self-energies to take advantage of the muffin-tin method within FEAST. Implementation was demonstrated through pseudocode of the FEAST-RCI and a description of parallelization techniques.

Finally, numerical experiments were performed to demonstrate accuracy and scalability. Meshes can be arbitrarily refined to increase accuracy (within the limits of DFT). Although scalability was not ideal, most inefficiencies were not method-specific (such as hardware and PARDISO [41] limitations) and could be improved in ways not related to this thesis.

This method allows for several new opportunities. It is now possible to perform *numerically exact* all-electron structure calculations for large-scale systems with modest hardware configurations. Hybrid basis functions, such as the one described in Section 2.7 can allow for more cost-effective approximate models, or can be used as preconditioners to speed up the exact approach solved using iterative solvers. Finally, because this method is built within the NESSIE nanoelectronics environment [39], the muffin-tin FEAST framework presented here is expected to speed-up the NESSIE time dependent and time-independent full self-consistent electronic structure calculations of molecules and solids.

BIBLIOGRAPHY

- [1] E. Polizzi, “Density-matrix-based algorithm for solving eigenvalue problems,” *Physics Review B*, vol. 79, 2009.
- [2] J. C. Slater, “Wave functions in a periodic potential,” *Physics Review*, 1937.
- [3] D. K. Ferry, et al., “Modeling of nanoelectronic and quantum devices,” NSTI-Nanotech, Boston, MA, 2004, vol. 3.
- [4] J. H. Luscombe, “Current issues in nanoelectronic modelling,” *Nanotechnology*, 1993.
- [5] B. P. Haley, et al., “Advancing nanoelectronic device modeling through petascale computing and deployment on nanoHUB,” SciDAC, San Diego, CA, 2009.
- [6] R. K. Cavin, et al., “A long-term view of research targets in nanoelectronics,” *Journal of Nanoparticle Research*, 2005.
- [7] M. Macucci, et al., “Status and perspectives of nanoscale device modelling,” *Nanotechnology*, pp. 136-142, 2001.
- [8] E. Polizzi and S. Datta, “Multidimensional nanoscale device modeling: the finite element method applied to the non-equilibrium Green’s function formalism,” *IEEE-Nano 2003*, San Francisco, CA, 2003, pp. 821-825.
- [9] G. Galli and M. Parrinello, “Large scale electronic structure calculations,” *Physical Review Letters*, vol. 69, no. 24, pp. 3547-3550, Dec. 1992.
- [10] E. Polizzi, “Bypassing traditional approximation techniques in electronic structure calculations: a fundamental and high-performance all-electron numerical framework,” unpublished.
- [11] D. Bohm, “Fluctuations, correlations, and eigenfunctions,” in *Quantum Theory*, New York: Dover Publications, 1989, pp. 208-209.
- [12] D. Bohm, “Theory of scattering,” in *Quantum Theory*, New York: Dover Publications, 1989, pp. 533-537.
- [13] A. L. Fetter and J. D. Walecka, “Fermi systems,” in *Quantum Theory of Many-Particle Systems*, New York: Dover Publications, 2003, pp. 121.

- [14] R. G. Parr, W. Yang, "The Kohn-Sham method: basic principles," in *Density-Functional Theory of Atoms and Molecules*, New York: Oxford University Press, 1989, pp. 144-145.
- [15] A. R. Levin, D. Zhang, E. Polizzi, "FEAST fundamental framework for electronic structure calculations: reformation of the muffin-tin problem," submitted.
- [16] J. Inglesfield, "A method of embedding," *Journal of Physics C: Solid State Physics*, vol. 14, 1981, pp. 3795-3806.
- [17] H. Si. (2001, Jan. 19). *TetGen* [Online]. Available: <http://wias-berlin.de/software/tetgen/>
- [18] A. Levin and E. Polizzi, "Embedded self-energy technique for solving arbitrary nanoelectronic systems using FEAST," in *IEEE Nano*, Portland, OR, 2011, pp. 1155-1158.
- [19] V. Mehrmann and H. Voss, "Nonlinear eigenvalue problems: a challenge for modern eigenvalue methods," *Mitteilungen der Gesellschaft für Angewandte Mathematik und Mechanik*, vol. 27, 2005 pp. 121-151.
- [20] D. Zhang, "First-principle electronic structure calculations within real-space mesh framework: applications to atoms, molecules and nanostructures," PhD dissertation, ECE Dept., Umass Amherst, Amherst, Ma, 2011.
- [21] W. E. Pickett, "Pseudopotential methods in condensed matter applications," *Computer Physics Reports*, vol. 9, no. 3, pp. 115-197, Apr., 1989.
- [22] M. T. Yin and M. L. Cohen, "Theory of ab initio pseudopotential calculations," *Physical Review B*, vol. 25, no. 12, pp. 7403-7412, June, 1982.
- [23] J. Kohenoff, *Electronic structure calculations for solids and molecules: theory and computational methods*, Cambridge University Press (2006).
- [24] Figure courtesy of Kimberly Donoughe.
- [25] R. M. Martin, "Density functional theory: foundations," in *Electronic Structure: Basic Theory and Practical Methods*, New York: Cambridge University Press, 2005, pp. 119.
- [26] R. M. Martin, "Plane waves and grids: basics," in *Electronic Structure: Basic Theory and Practical Methods*, New York: Cambridge University Press, 2005, pp. 236-250.
- [27] R. M. Martin, "Localized orbitals: tight-binding," in *Electronic Structure: Basic Theory and Practical Methods*, New York: Cambridge University Press, 2005, pp. 272-293.

- [28] R. M. Martin, "Augmented functions: APW, KKR, MTO," in *Electronic Structure: Basic Theory and Practical Methods*, New York: Cambridge University Press, 2005, pp. 313-341.
- [29] J. R. Shewchuk, "Triangle: Engineering a 2D quality mesh generator and Delaunay triangulator, in *Applied Computational Geometry: Towards Geometric Engineering*, Berlin: Springer-Verlag, 1996, pp. 203-222.
- [30] B. N. Harmon and D. D. Koelling, "Technique for rapid solution of the apw secular equation," *Journal of Physics C: Solid-State Physics*, vol. 7, no. 11, pp. 201, June, 1974.
- [31] E. Sjöstedt and L. Nordström, "Efficient solution of the nonlinear augmented plane wave secular equation, *Journal of Physics: Condensed Matter*, vol. 14, no. 47, pp. 12485-12494, Nov. 2002.
- [32] M. Naumov, et al., "A tearing-based hybrid parallel sparse linear system solver," *Journal of Computational and Applied Mathematics*, pp. 3025-3037, 2010.
- [33] "Hydrogen atom," *Wikipedia* [Online]. Available: http://en.wikipedia.org/wiki/Hydrogen_atom
- [34] J. Ackermann, B. Erdmann, and R. Roltzsch, "A self-adaptive multilevel finite element method for the stationary Schrödinger equation in three space dimensions," *Journal of Chemical Physics*, vol. 101, no. 9, pp. 7643-7650, Nov., 1994.
- [35] H. Medel-Cobaxin, A. Alijah, and A.V. Turbiner, "About non-existence of the molecular ion H_3^{++} ," *arXiv* [Online]. Available: <http://arxiv.org/abs/0805.1796>
- [36] OpenMP [Online.] Available: <http://openmp.org>
- [37] E. Polizzi, "A High-Performance Numerical Library for Solving Eigenvalue Problems: FEAST Solver v2.0 User's Guide" *arXiv*[Online]. Available: <http://arxiv.org/abs/1203.4031>
- [38] E. Polizzi and A. Sameh, "A parallel hybrid banded system solver: the SPIKE algorithm," *Parallel Computing*, vol. 32, pp. 177-194, 2006.
- [39] E. Polizzi and A. Sameh, "Numerical parallel algorithms for large-scale nanoelectronics simulations using NESSIE," *Journal of Computational Electronics*, vol. 3, pp. 363-266, 2005.
- [40] OpenMPI [Online.] Available: <http://www.open-mpi.org>
- [41] PARDISO [Online.] Available: <http://www.pardiso-project.org>
- [42] "Energies for C6H6 (Benzene)," *Computational Chemistry Comparison and Benchmark DataBase* [Online.] Available: <http://cccbdb.nist.gov/>

- [43] L. Lehtovaara, V. Havu, and M. Puska, “All-electron density functional theory and time dependent density functional theory with high-order finite elements,” *The Journal of Chemical Physics*, vol. 131, pp. 054103, 2009.