

2013

# A Non-Linear Eigensolver-Based Alternative to Traditional Self-Consistent Electronic Structure Calculation Methods

Brendan E. Gavin

*University of Massachusetts Amherst*

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

 Part of the [Atomic, Molecular and Optical Physics Commons](#), [Engineering Physics Commons](#), [Other Electrical and Computer Engineering Commons](#), and the [Quantum Physics Commons](#)

---

Gavin, Brendan E., "A Non-Linear Eigensolver-Based Alternative to Traditional Self-Consistent Electronic Structure Calculation Methods" (2013). *Masters Theses 1911 - February 2014*. 1123.

Retrieved from <https://scholarworks.umass.edu/theses/1123>

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](mailto:scholarworks@library.umass.edu).

**A NON-LINEAR EIGENSOLVER-BASED ALTERNATIVE  
TO TRADITIONAL SELF-CONSISTENT ELECTRONIC  
STRUCTURE CALCULATION METHODS**

A Thesis Proposal Presented

by

BRENDAN GAVIN

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 2013

Electrical and Computer Engineering

**A NON-LINEAR EIGENSOLVER-BASED ALTERNATIVE  
TO TRADITIONAL SELF-CONSISTENT ELECTRONIC  
STRUCTURE CALCULATION METHODS**

A Thesis Proposal Presented by  
**BRENDAN GAVIN**

Approved as to style and content by:

---

Eric Polizzi, Chair

---

Neal Anderson, Member

---

Dimitrios Maroudas, Member

---

Ashwin Ramasubramaniam, Member

---

Robert Jackson, Department Chair  
Electrical and Computer Engineering

## ACKNOWLEDGMENTS

Thanks to the faculty of UMass, who so frequently have taught me things that I didn't know that I didn't know.

## ABSTRACT

# A NON-LINEAR EIGENSOLVER-BASED ALTERNATIVE TO TRADITIONAL SELF-CONSISTENT ELECTRONIC STRUCTURE CALCULATION METHODS

SEPTEMBER 2013

BRENDAN GAVIN

B.Sc., UNIVERSITY OF MASSACHUSETTS, AMHERST

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

Directed by: Professor Eric Polizzi

This thesis presents a means of enhancing the iterative calculation techniques used in electronic structure calculations, particularly Kohn-Sham DFT. Based on the subspace iteration method of the FEAST eigenvalue solving algorithm, this non-linear FEAST algorithm (NLFEAST) improves the convergence rate of traditional iterative methods and dramatically improves their robustness. A description of the algorithm is given, along with the results of numerical experiments that demonstrate its effectiveness and offer insight into the factors that determine how well it performs.

# TABLE OF CONTENTS

	Page
ACKNOWLEDGMENTS .....	iii
ABSTRACT .....	iv
LIST OF TABLES .....	vi
LIST OF FIGURES .....	vii
CHAPTER	
1. THE MANY-BODY SCHRÖDINGER EQUATION AND THE SINGLE PARTICLE APPROXIMATION .....	1
1.1 The Many-Body Schrödinger Equation .....	1
1.2 The Single Particle Approximation .....	2
2. SOLVING THE KOHN-SHAM EQUATIONS .....	4
3. NLFEAST .....	7
3.1 Linear FEAST .....	7
3.2 Nonlinear FEAST .....	9
3.3 Numerical Results .....	10
3.3.1 Performance Comparison .....	11
3.3.2 Convergence Rate and $\mathbf{Q}$ subspace size .....	13
3.3.3 Convergence Rate and Contour Integration Accuracy .....	14
3.3.4 The Initial Guess .....	14
4. FUTURE WORK .....	22
BIBLIOGRAPHY .....	24

## LIST OF TABLES

Table	Page
3.1 Total energy results for NLFEST and NWChem for various molecules. ....	16

## LIST OF FIGURES

Figure	Page
3.1 The FEAST algorithm .....	16
3.2 The full NLFEAST algorithm .....	17
3.3 Results of numerical experiments comparing the performance of NLFEAST to that of DIIS for the Silane $\text{SiH}_4$ , Benzene $\text{C}_6\text{H}_6$ and caffeine $\text{C}_8\text{H}_{10}\text{N}_4\text{O}_2$ molecules. ....	18
3.4 Numerical experiments demonstrating the role of the number of retained $\mathbf{Q}^{(k)}$ subspaces for constructing the search subspace used by NLFEAST. ....	19
3.5 Experiments demonstrating the effect on convergence of the number of gauss points in the contour integration in NLFEAST. ....	20
3.6 Results of numerical experiments where the initial guess for the electron density was set to zero, $n = 0$ . ....	21



## CHAPTER 1

# THE MANY-BODY SCHRÖDINGER EQUATION AND THE SINGLE PARTICLE APPROXIMATION

### 1.1 The Many-Body Schrödinger Equation

The prediction of many of the electrical and chemical properties of any system of matter can, at least in principle, be performed by solving the many body Schrödinger equation with the Born-Oppenheimer approximation:

$$\hat{H}\psi_i(x_1, x_2, x_3 \dots x_{n_e}) = \lambda_i \psi_i(x_1, x_2, x_3 \dots x_{n_e}) \quad (1.1)$$

$$\hat{H} = \sum_{i=1}^{n_e} \frac{-\hbar^2}{2m} \frac{d^2}{dx_i^2} + V_{nuclear}(x_i) + \sum_{\langle i,j \rangle} \frac{q^2}{4\pi\epsilon_0} \frac{1}{\|x_i - x_j\|} \quad (1.2)$$

Perturbative approximations can offer some insights into some systems, but there is no known analytical means of actually solving the general many body Schrödinger equation. If the equation is discretized using some basis set with the appropriate boundary conditions it becomes a generalized eigenvalue problem, which can be solved straight-forwardly using well-established diagonalization algorithms. The discretized solution to the problem, however, cannot generally be stored in a computer due to the size of the solution space; given some set of basis function  $f_i(x)$ ,  $1 \leq i \leq n$ , the solution is approximated as:

$$\psi(x_1, x_2, x_3, \dots) \approx \sum_{i,j,k,\dots=1}^n \psi_{ijk\dots} f_i(x_1) f_j(x_2) f_k(x_3) \dots \quad (1.3)$$

where the ellipses indicate that an arbitrary number of indices may be used, the precise number of which depends on the number of degrees of freedom in the system.

The number of coefficients  $\psi_{ijk\dots}$  needed clearly scales exponentially with the size of the system. Although this can be mitigated somewhat by taking advantage of symmetries, such as the electron exchange (anti)symmetry, the storage requirements for the solution of the many body Schrödinger equation with a reasonable amount of accuracy still far exceed what is feasible on a classical computer for almost any system of interest.

## 1.2 The Single Particle Approximation

A method of addressing this problem that has met with a good amount of success is the single particle approximation. There are two primary methods that can be said to use a single particle approximation: Hartree-Fock and Density Functional Theory (DFT). This thesis will be dealing with the latter.

The idea behind DFT is that certain important properties of a many body quantum mechanical system, such as the ground state total energy and electron density, can be found by solving a Schrödinger equation for a related system of fictitious, non-interacting particles [2][3]. The fact that the particles are non-interacting means that all of the terms in the system Hamiltonian are entirely separable, and so only the single particle Hamiltonian needs to be diagonalized:

$$\hat{H}[n]\psi_i(x) = \lambda_i\psi_i(x) \quad n = \sum_{i=1}^{n_e} 2|\psi_i(x)|^2 \quad (1.4)$$

$$\hat{H}[n] = \frac{-\hbar^2}{2m} \frac{d^2}{dx^2} + V_{nuclear} + V_{hartree}[n] + V_{xc}[n] \quad (1.5)$$

where  $n$  is the electron density and  $n_e$  is the number of electrons in the system in question. These are the Kohn-Sham equations. The Hamiltonian here shares with the original many body Hamiltonian the nuclear potential  $V_{nuclear}$ , and the electrostatic energy from the electron-electron interactions is captured by the "Hartree" potential

$V_{Hartree}$ . The exchange-correlation  $V_{xc}$  potential accounts for the non-classical electron correlation and exchange effects.

The primary difference between these equations and the many body Schrödinger equation is the fact that the Kohn-Sham equations are nonlinear in the electron density; both  $V_{Hartree}$  and  $V_{xc}$  are functions of the electron density. These equations, taken together with the diagonalization procedure followed by the formation of a new electron density, form a fixed point map  $f(n)$  [1] that takes as input one electron density and produces another as an output; the solution to the equations is the density that, used as an input for the Kohn-Sham equations, yields an eigenvalue problem whose density evaluates to the same as the one used as an input:  $f(n^*) = n^*$ .

$$\begin{aligned} f(n) = & \text{generate } \hat{H}[n] \\ & \Downarrow \\ & \text{solve } \hat{H}[n]\psi(x) = \lambda\psi(x) \\ & \Downarrow \\ & \text{form } n_{new} = \sum_k 2|\psi_k(x)|^2 \end{aligned}$$

## CHAPTER 2

### SOLVING THE KOHN-SHAM EQUATIONS

The nonlinear nature of the Kohn-Sham equations makes traction with them significantly more problematic than with the linear Schrödinger equation, but the fact that their solution is storable in a computer makes this a worthwhile endeavor.

Finding the fixed point of a fixed point mapping like that of the Kohn-Sham equations is often difficult or impossible to accomplish analytically, and so the problem is first discretized and then solved using an iterative procedure; one begins with some sort of guess at the solution, and repeatedly updates that guess through some procedure until convergence is reached. "Convergence" is typically defined as the point when the difference between subsequent iterations of the variable that is being iterated over (i.e. the electron density, in this case) is smaller than some arbitrary constraint:  $\|n_{i+1} - n_i\| < \epsilon$ . Convergence may also be defined with respect to some function of the variable being iterated over; in DFT, convergence is often defined with respect to the total energy of the system:  $|E(n_{i+1}) - E(n_i)| < \epsilon$

The simplest kind of iterative procedure that one can use to solve a fixed point mapping is to simply iterate using the map until the solution converges:

Given an initial guess for  $n$ :

1. Calculate  $n_{i+1} = f(n_i)$
2. If  $\|n_{i+1} - n_i\| < \epsilon$ , stop; else, set  $n_i = n_{i+1}$  and GOTO step 1

This procedure rarely results in numerical convergence; it typically diverges rapidly.

A slightly more advanced procedure, called "simple mixing", generates an updated guess at the solution by forming a linear combination of the current guess and the output of the fixed point mapping:

Given an initial guess for  $n$ :

1. Calculate  $n_{i+1} = \alpha f(n_i) + (1 - \alpha)n_i$ ,  $0 \leq \alpha \leq 1$
2. If  $\|n_{i+1} - n_i\| < \epsilon$ , stop; else, set  $n_i = n_{i+1}$  and GOTO step 1

With a good initial guess and a sufficiently small  $\alpha$ , this procedure will often converge; the quality of the initial guess is very important, however, and the best choice for  $\alpha$  usually has to be determined by trial and error on a case-by-case basis. Convergence tends to take a very long time, needing hundreds or thousands of iterations.

Advanced techniques that offer good rates of convergence are Newton-type techniques that make use of knowledge of the Jacobian of the fixed point mapping in order to find an intelligent search direction for updating the guess at the solution.

The most common technique used in quantum chemistry for solving the Kohn-Sham equations is the Direct Inversion of Iterative Subspaces (DIIS) [7], which can be shown to be equivalent to a Newton-type technique [8]. With DIIS, one begins with several initial guesses (generated, for example, from a single guess followed by several iterations of simple mixing) and their residuals,  $R(n_i) = f(n_i) - n_i$ , and then generates a new guess by forming the linear combination of previous guesses that would result in minimizing the residual of that linear combination, were it the case that the fixed point mapping  $f(n)$  were linear, plus a correction term composed of the linear combination of the residuals [8].

**DIIS:**

Start with  $\{n_1, n_2, \dots, n_i\}$ ,  $\{r_1, r_2, \dots, r_i\}$ ,  $r_k = f(n_k) - n_k$

1. Find  $c_k$  to minimize  $\|\sum c_k r_k\|$  such that  $\sum c_k = 1$

2. Form  $n_{i+1} = \sum c_k n_k + c_k r_k$
3. Calculate  $r_{i+1} = f(n_{i+1}) - n_{i+1}$ , discard  $n_1, r_1$
4. Stop if  $\|r_{i+1}\| < \epsilon$ ; else, goto step 1

The operational assumption here is that, if the initial guesses are good enough, then they can be thought of as lying close together in or near the neighborhood of the fixed point, and that the treatment of the fixed point mapping as linear is therefore a reasonable approximation.

In practice, DIIS even performs fairly well even for initial guesses wherein the assumption of linearity is not such a good one. Even so, the algorithm is still quite sensitive to the choice of initial guess, and many iterations may be required before convergence is reached.

## CHAPTER 3

### NLFEAST

The fact that DIIS converges slowly or unreliably, depending on the initial guess and the system under study, motivates the exploration of additional techniques for finding the fixed point electron density of the Kohn-Sham equations. The following sections will discuss the reasoning behind, and products of, the research project that is the subject of this thesis, which is aimed at using an eigensolver algorithm called FEAST as the basis for solving the Kohn-Sham equations.

#### 3.1 Linear FEAST

FEAST is an algorithm for solving large sparse or small dense linear generalized eigenvalue problems,  $\mathbf{A}x = \lambda\mathbf{S}x$ , for the eigenvectors whose eigenvalues fall within some user-prescribed interval  $[\lambda_{min}, \lambda_{max}]$  [5]. The algorithm takes as its input a guess at the eigenvectors of interest and improves that guess dramatically through multiplication by an approximation of the spectral projector for the eigenvectors of interest (exact multiplication would immediately produce the exact subspace in which the desired solution lies):

$$\mathbf{X}_{i+1} = \rho\mathbf{S}\mathbf{X}_i \tag{3.1}$$

where  $\mathbf{X}_i$  is the matrix whose column vectors are the current guesses at the eigenvalues of interest and  $\rho = \sum_k x_k x_k^T$  is the spectral projector (and sometimes called the "density matrix") that spans the eigenvalues that are being sought. Multiplication

by  $\mathbf{S}$  is necessary because, for the generalized eigenvalue problem, the solutions  $x_k$  are  $\mathbf{S}$ -orthogonal.

A reduced eigenvalue problem is then produced by using the improved guess as a subspace in which to solve the original, large size eigenvalue problem:

$$\mathbf{Q} = \mathbf{X}_{i+1} \quad (3.2)$$

$$\mathbf{A}x = \lambda\mathbf{B}x \rightarrow \mathbf{Q}^T\mathbf{A}\mathbf{Q}x = \lambda\mathbf{Q}^T\mathbf{B}\mathbf{Q}x \quad (3.3)$$

The solutions to this problem, projected back into the full problem space, serve as the new guess for a solution to the full size problem. This entire procedure is repeated until some measure of convergence is met, that measure typically being the convergence of the trace of the eigenvalues.

In practice, the approximation of the multiplication by the spectral projector is accomplished by using an integral of the Green's function of the eigenvalue problem. The integral is performed in a closed contour in the complex plane around the interval where the eigenvalues of interest are expected to be found.

$$\rho\mathbf{S}\mathbf{X} = \oint_C (\mathbf{H} - z\mathbf{S})^{-1}\mathbf{S}\mathbf{X} dz \approx \sum_i \omega_i (\mathbf{H} - z_i\mathbf{S})^{-1}\mathbf{S}\mathbf{X} \quad (3.4)$$

This integral itself is exact, and the approximation of the spectral projection occurs when it is evaluated using a quadrature rule. It is worth emphasizing that although the spectral projection is approximated, the eigenvectors produced by the algorithm are not approximate; it is only their projection that is approximated, and repeated application of this procedure can be shown to result in convergence to the correct vectors [6].

Because a quadrature rule is used in performing the spectral projection, the computational effort of forming the subspace thus amounts to solving several linear systems,



$$(\mathbf{H} - z_i \mathbf{S})y = \omega_i \mathbf{S} \mathbf{X} \quad (3.5)$$

the number of which is proportional to the number of points used in the quadrature. It can be shown that this algorithm will converge to the correct solution, and that the it will converge more quickly if the contour integration is performed more accurately (i.e. if there are more points used in the quadrature) [6].

In practice, one provides an initial guess that forms a subspace of larger dimension than the number of eigenvectors that are expected to be found; the larger the subspace used for solving the reduced eigenvalue problem, the faster the rate of convergence is. The full linear FEAST algorithm is given in Fig 3.1.

### 3.2 Nonlinear FEAST

Nonlinear FEAST (NLFEAST), the subject of this research, is the application of the FEAST algorithm to the problem of solving the discretized nonlinear Kohn-Sham equations:

$$\mathbf{H}[n]\psi_i = \lambda_i \mathbf{S}\psi_i \quad n = \sum_{i=1}^{n_e} 2|\psi_i|^2 \quad (3.6)$$

$$\mathbf{H}[n] = \mathbf{T} + \mathbf{V}_{nuclear} + \mathbf{V}_{hartree}[n] + \mathbf{V}_{xc}[n] \quad (3.7)$$

where  $\mathbf{T}$  is the discretized kinetic energy operator and  $\mathbf{S}$  is the overlap matrix resulting from discretization using a non-orthogonal basis set, such as finite elements.

In its most essential form, the NLFEAST algorithm is the same as the FEAST algorithm, except that the projected eigenvalue problem is nonlinear in the eigenvectors being sought. Thus, at each FEAST iteration, the projected problem is solved by iterating in the density  $n = \sum 2|\psi_i|^2$  using one of the typical algorithms for solving nonlinear problems, such as DIIS.

$$\mathbf{H}[n]\psi_i = \lambda_i \mathbf{S}\psi_i \rightarrow \mathbf{Q}^T \mathbf{H}[n] \mathbf{Q}\psi_i = \lambda_i \mathbf{Q}^T \mathbf{S} \mathbf{Q}\psi_i \quad (3.8)$$

This basic form of the algorithm gives poor performance; numerical experiments indicate little or no convergence for even the simplest problems. We find that the algorithm's performance improves dramatically if, rather than replacing the projection subspace at each FEAST step, the subspace for the next step is generated by appending the columns of the new subspace to the old one:

$$\mathbf{Q}_{i+1} = [\mathbf{Q}_i \ \mathbf{Q}_{new}] \quad (3.9)$$

Convergence then becomes not only swift, but faster than that of DIIS used outside the context of the FEAST algorithm.

The fact that the subspace  $\mathbf{Q}$  is now composed of the subspaces from various previous iterations, none of which are necessarily orthogonal to each other, means that the projected overlap matrix  $\mathbf{Q}^T \mathbf{B} \mathbf{Q}$  is not necessarily symmetric positive definite. Because the symmetric positive definiteness of the overlap matrix is necessary in order for there to be a solution to the generalized eigenvalue problem, an additional step is required wherein the singular value decomposition of the subspace  $\mathbf{Q}$  is performed and only the left singular vectors are retained.

$$\mathbf{Q} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T \rightarrow \mathbf{Q} = \mathbf{U} \quad (3.10)$$

With these modifications, the full NLFEAST algorithm is given in fig 3.2

### 3.3 Numerical Results

Numerical experiments for this research consisted of finding the ground state electron density and total energy for various molecules using the NLFEAST and the DIIS algorithms, while keeping track of the convergence of each algorithm. Convergence

was measured as a relative error of either the trace of the eigenvalues being sought or the total energy, where "relative error" is defined as a normalized absolute difference between two iterations of the same quantity (e.g. for total energy, the relative error would be  $|E(n_{i+1}) - E(n_i)|/|E(n_{i+1})|$  at iteration  $i + 1$ ).

These algorithms were implemented in an in-house all-electron simulation framework that uses a real-space cubic finite element discretization, some details of which are described in Ref [4]. The DIIS technique was used both for performing the traditional iterative approach as well as for solving the NLFEAST nonlinear reduced system in Step-3c of Figure 3.2. Both DIIS procedures make use of a subspace composed of five successive generations of electron densities, but the maximum number of DIIS iterations for solving the reduced system in NLFEAST has been fixed to three (i.e. the non-linear reduced system is then solved only approximately). For most of the examples, increasing the number of the inner iterations further has had no effect on the overall NLFEAST convergence rate.

### 3.3.1 Performance Comparison

In all the experiments conducted thus far, including various molecules from  $H_2$  to  $C_{60}$ , the NLFEAST algorithm has outperformed SCF-DIIS both in terms of convergence rate and execution time. Three representative examples are shown in Figure 3.3 for the Silane, Benzene and Caffeine molecules. The relative error on the total energy at each iteration,  $|E(n_{i+1}) - E(n_i)|/|E(n_{i+1})|$ , is used as the measure of convergence for both approaches, as this is the most directly comparable measure of convergence for both DIIS-SCF and NLFEAST. The meaning of the outer-iterations is different for both algorithm since, for NLFEAST, it directly represents the number of contour integrations and non-linear reduced systems.

Because the implementation of DIIS in this research uses the FEAST algorithm as its eigenvalue solver, it is possible to have a direct comparison between the operations

counts for both DIIS-SCF and NLFEAST. In both cases the contour integration is the most time consuming single numerical operation. In the three examples provided in Figure 3.3, for example, SCF-DIIS requires  $5\times$  to  $7\times$  more contour integrations than NLFEAST for obtaining the same level of accuracy in the total energy (i.e.  $\sim 10^{-7}$ ). On the other hand, NLFEAST requires the solution of  $\sim 3\times$  as many Poisson equations as does SCF-DIIS. However, since the Poisson system matrix only needs to be factorized once at the beginning of the calculation, the remaining Poisson solve operations become negligible in comparison to both factorizing and solving the complex linear systems that arise from the contour integration. The rest of the numerical operations involved in NLFEAST, such as solving the reduced eigenvalue problem or performing the SVD in Step-3b of Figure 3.2, bring no significant computational overhead provided that the ratio  $N/M_0$  stays relatively large for a given search interval.

Table 3.1 summarizes the final total energies obtained by using NLFEAST for molecules ranging in size from  $H_2$  to  $C_{60}$ , and provides the results obtained by using the NWChem software package [9] for the same molecule geometries. Although these two pieces of software use different discretizations and convergence criteria, their results are in good agreement with each other. Because of the differences between the two codes, however, it was not possible to make a direct comparison between them for the purpose of comparing the rate of convergence.

For the simulations reported in Table 3.1, the following parameters were used for the NLFEAST algorithm: (i) a  $\mathbf{Q}$  search subspace that keeps increasing indefinitely until convergence, (ii) 16 nodes for the Gauss quadrature along the complex contour, (iii) a conventional initial guess as the starting point for the electron density (e.g. the all-electron result for the single atoms). The role that each of these parameters plays in the functioning of the algorithm will be discussed in greater detail in the following sections.

### 3.3.2 Convergence Rate and $\mathbf{Q}$ subspace size

As mentioned in Section 3.2, using only a single subspace  $\mathbf{Q}$  (in step-3a of Figure 3.2) returned by the contour integration procedure of FEAST provides an extremely poor convergence rate; a successful implementation of the NLFEAST algorithm thus requires expanding the subspace size beyond this. It may be extended by retaining only a finite number of the most recent subspaces, or it may be extended indefinitely until convergence. For the numerical experiments presented in Figure 3.3 and Table 3.1, the rate of convergence is expected to reach a maximum because all generated subspaces have been retained. Figure 3.4 shows the results of several numerical experiments that were performed for our selected three molecules using different sizes for the search subspace  $\mathbf{Q}$ .

These results show clearly that the rate of convergence of NLFEAST is directly related to the number of subspaces that are retained. They also show that it is, in practice, not necessary to retain every subspace that gets generated; retention of only four of the most recent subspaces provides a rate of convergence that is nearly as good as that obtained by retaining every subspace.

Worthy of particular note are the simulation results for Benzene and Caffeine where only a single subspace was retained, that being the subspace that was most recently generated. Here the algorithm does not appear to converge at all. Although this is not necessarily typical, and the solution may eventually converge using a more efficient approach for solving the non-linear reduced problem (e.g. using a larger number of inner iterations for the FEAST-DIIS problem which has been fixed to three in these simulations), it highlights the importance of extending the search subspace size for the success of this algorithm.

### 3.3.3 Convergence Rate and Contour Integration Accuracy

At each iteration of NLFEAST, the approximate subspace solution is improved through multiplication by the density matrix of the most current Hamiltonian. This step, Step 2 in Figure 3.2, is accomplished by performing a numerical contour integration of the Green's function multiplied by the approximate solution. In practice, a Gauss quadrature can be efficiently used here, which involves summing the solutions of several separate linear systems.

For all the results presented so far, 16 Gauss contour points were used to perform the quadrature. Figure 3.5 shows the results for simulations wherein the number of contour points is varied between 4 and 48. Although increasing the number of Gauss points can help to improve the convergence rate, the effect here is not quite as dramatic as the effect of increasing the subspace size. The relationship that exists here between the contour integration accuracy and the convergence rate is similar to the case in the linear FEAST algorithm, where both a larger subspace size and an improved contour integration accuracy also improve convergence rate [6].

### 3.3.4 The Initial Guess

Like other schemes for solving nonlinear eigenvector equations, NLFEAST requires an initial guess for the electron density  $n$ . Clearly, a good initial guess can provide faster convergence, but for NLFEAST the quality of the initial guess is not important as far as achieving convergence is concerned. Unlike other means of performing SCF iterations, this algorithm is capable of achieving convergence even when given an extremely poor initial guess, including the extreme case of no initial guess at all (i.e.  $n = 0$ ). Figure 3.6 shows the results of a several numerical experiments wherein the initial guess for the electron density was set to zero. Each experiment eventually resulted in convergence.

The algorithm's performance when given such a poor initial guess is partly related to the size of the system; more electrons typically means that a larger number of iterations is required before convergence is reached. Performance also appears to depend on the particular molecule under consideration. In Figure 3.6, CH<sub>4</sub> requires a larger number of iterations to reach a high level of convergence, despite being the second smallest system (in terms of number of electrons). It is likely that, for some molecules, the algorithm converges towards a local energy minimum before ultimately finding its way to the global minimum, which results in its progress being more delayed than it would be with a molecule where such a detour does not occur.

These results are given only as a demonstration of the robustness of the algorithm; in practice, one would always use some reasonable heuristic to derive a more sensible initial guess.

---



---

### Basic FEAST Algorithm

---

**Input:** interval  $[E_{\min}, E_{\max}]$ , and an (over)-estimation  $M_0$  of the number of eigenpairs

**1- Initialization**

Select  $M_0 > M$  random vectors  $\mathbf{Y} \in \mathbb{R}^{N \times M_0}$

**2- Contour Integration**

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

**3- Rayleigh-Ritz**

Form  $\mathbf{H}_{\mathbf{Q}_{M_0 \times M_0}} = \mathbf{Q}^T \mathbf{H} \mathbf{Q}$  and  $\mathbf{S}_{\mathbf{Q}_{M_0 \times M_0}} = \mathbf{Q}^T \mathbf{S} \mathbf{Q}$

Solve the reduced eigenvalue problem  $\mathbf{H}_{\mathbf{Q}} \Phi = \epsilon \mathbf{S}_{\mathbf{Q}} \Phi$

**4- Subspace Iteration**

(a) Set  $E_m = \epsilon_m$  and  $\mathbf{X}_{N \times M_0} = \mathbf{Q}_{N \times M_0} \Phi_{M_0 \times M_0}$

(b) Check convergence (i.e.  $\text{Trace}\{E_m\}$ )

(c) If needed go back to step **2** with  $\mathbf{Y} = \mathbf{S} \mathbf{X}$

**Output:** All the  $M < M_0$  eigenpairs  $(\{E_m, \mathbf{x}_m\})$ .

---



---

**Figure 3.1.** The FEAST algorithm

	#electrons	#iterations	$E_{\text{tot}}$ (eV)	NWChem
H <sub>2</sub>	2	5	-30.962	-30.959
CH <sub>4</sub>	10	5	-1091.45	-1091.69
H <sub>2</sub> O	10	7	-2065.24	-2065.48
CO	14	6	-3060.12	-3060.45
SiH <sub>4</sub>	18	9	-7907.42	-7909.78
Na <sub>2</sub>	22	8	-8785.57	-8786.61
C <sub>6</sub> H <sub>6</sub>	42	6	-6262.31	-6263.65
C <sub>8</sub> H <sub>10</sub> N <sub>4</sub> O <sub>2</sub>	102	9	-18364.5	-18360.3
C <sub>60</sub>	360	8	-61610.5	-61660.1

**Table 3.1.** Total energy results for NLFEAST and NWChem for various molecules.



---



---

### Non-linear eigenvector FEAST algorithm

---

**Input:** interval  $[E_{\min}, E_{\max}]$  including  $M$  lowest occupied states and an initial search subspace of size  $M_0 > M$

#### 1- Initialization

a-Define an initial guess for the electron density  $\mathbf{n}$

b-Construct Hamiltonian  $\mathbf{H}[\mathbf{n}]$

c-Select  $M_0 > M$  random vectors  $\mathbf{Y} \in \mathbb{R}^{N \times M_0}$

#### 2- Contour Integration at FEAST iteration $k$

Compute  $\mathbf{Q}_{N \times M_0}^{(k)} = -\frac{1}{2\pi i} \int_{\mathcal{C}} dZ (Z\mathbf{S} - \mathbf{H}[\mathbf{n}])^{-1} \mathbf{Y}_{N \times M_0}$ ,

#### 3- Non-linear Rayleigh-Ritz

a-Construct the subspace  $\hat{\mathbf{Q}} = \{\mathbf{Q}^{(k)}, \mathbf{Q}^{(k-1)}, \mathbf{Q}^{(k-2)}, \dots\}$

b-Compute  $\mathbf{Q} := \mathbf{U}$  using SVD i.e.  $\hat{\mathbf{Q}} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$

c- SCF Procedure

- construct Hamiltonian  $\mathbf{H}[\mathbf{n}]$   
(requires also solving the Poisson equation)
- form  $\mathbf{H}_{\mathbf{Q}_{M_0 \times M_0}} = \mathbf{Q}^T \mathbf{H}[\mathbf{n}] \mathbf{Q}$  and  $\mathbf{S}_{\mathbf{Q}_{M_0 \times M_0}} = \mathbf{Q}^T \mathbf{S} \mathbf{Q}$
- solve reduced eigenvalue problem  $\mathbf{H}_{\mathbf{Q}} \mathbf{\Phi} = \epsilon \mathbf{S}_{\mathbf{Q}} \mathbf{\Phi}$
- compute electron density  $\mathbf{n} = 2 \sum_{m=1}^M |\mathbf{Q} \mathbf{\Phi}_m|^2$
- Check convergence of  $\mathbf{n}$ , go back to step-c if needed with a new generated input density

#### 4- Subspace Iteration

Set  $E_m = \epsilon_m$  and compute  $\mathbf{X}_{N \times M_0} = \mathbf{Q}_{N \times M_0} \mathbf{\Phi}_{M_0 \times M_0}$

Check convergence (i.e. Trace  $\sum_m E_m \in [E_{\min}, E_{\max}]$ ).

If needed go back to step 2 with  $\mathbf{Y} = \mathbf{S}\mathbf{X}$

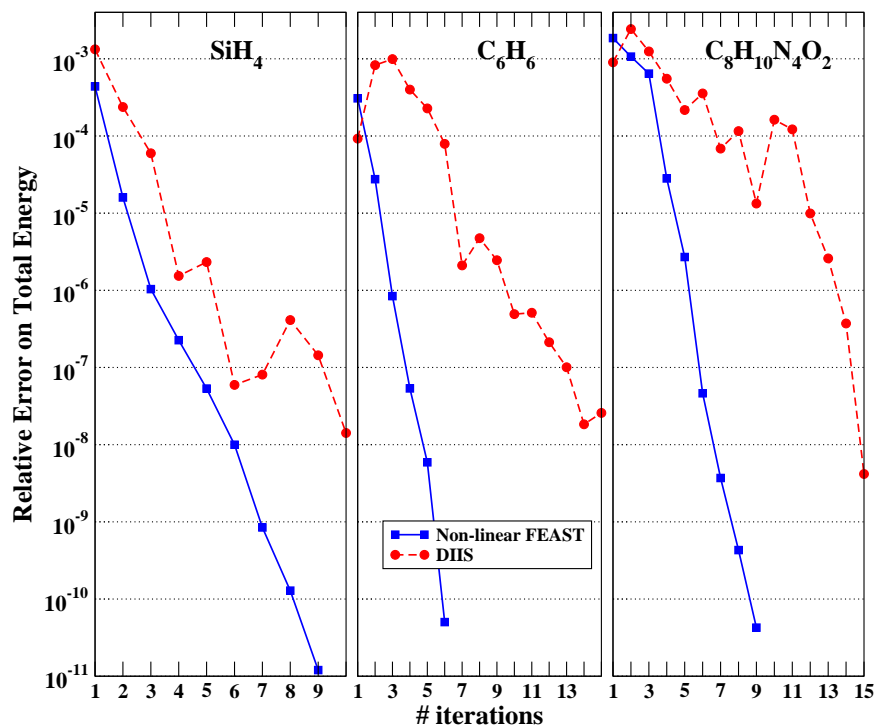
**Output:** All the  $M < M_0$  eigenpairs  $(\{E_m, \mathbf{x}_m\})$ .

---

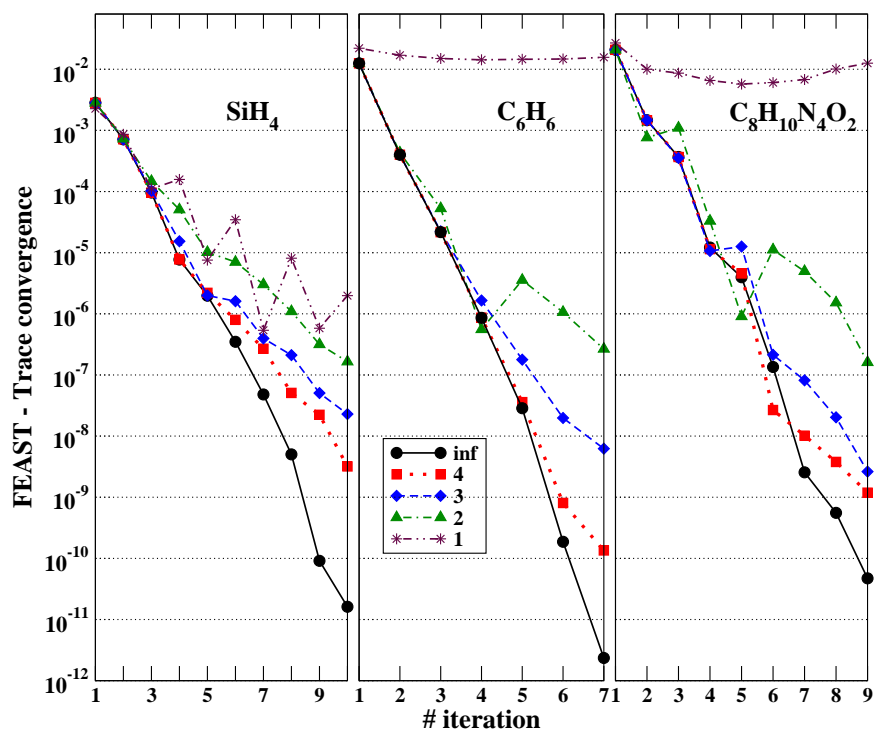


---

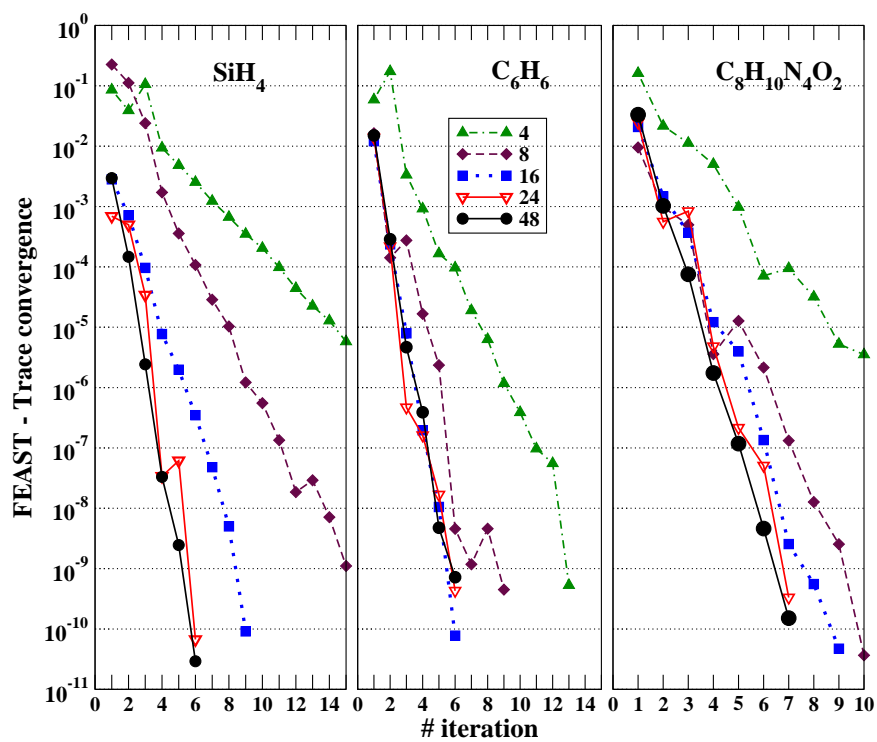
**Figure 3.2.** The full NLFEAST algorithm



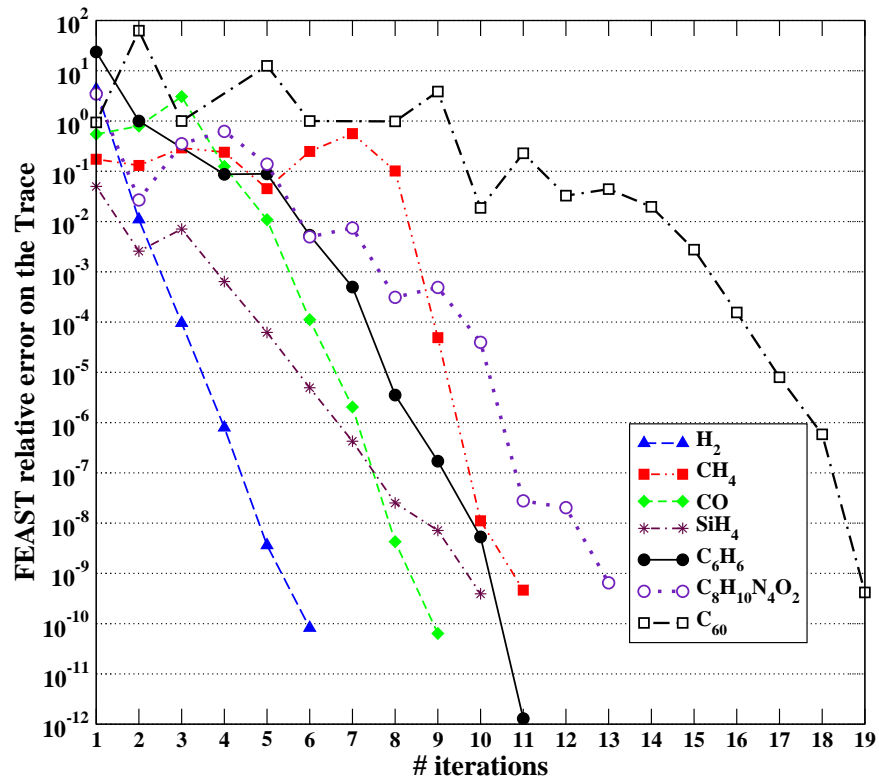
**Figure 3.3.** Results of numerical experiments comparing the performance of NLFEAST to that of DIIS for the Silane  $\text{SiH}_4$ , Benzene  $\text{C}_6\text{H}_6$  and caffeine  $\text{C}_8\text{H}_{10}\text{N}_4\text{O}_2$  molecules.



**Figure 3.4.** Numerical experiments demonstrating the role of the number of retained  $\mathbf{Q}^{(k)}$  subspaces for constructing the search subspace used by NLFEAST.



**Figure 3.5.** Experiments demonstrating the effect on convergence of the number of gauss points in the contour integration in NLFEAST.



**Figure 3.6.** Results of numerical experiments where the initial guess for the electron density was set to zero,  $n = 0$ .

## CHAPTER 4

### FUTURE WORK

The subspace iteration procedure of the FEAST algorithm can, with some small modifications, serve as an effective means of enhancing the traditional methods of solving the Kohn-Sham equations of Density Functional Theory. Numerical experiments performed on a variety of molecular geometries show that the resulting NLFEAST algorithm has a faster convergence rate than normal iterative procedures alone and that it gives a dramatic improvement in robustness.

Still left to be done is to address the fundamental bottleneck preventing the NLFEAST algorithm from being used effectively for systems of extremely large size: the reduced eigenvalue problem. Because the algorithm requires expanding the subspace size by an amount proportional to the number of electrons, systems with a large number of electrons will produce a prohibitively large and dense reduced eigenvalue problem to solve. Addressing this effectively will require taking advantage of the structure of the projected Hamiltonian and overlap matrices in order to find some means of parallelizing the solution of the reduced system, possibly with another layer of the FEAST algorithm. This may be a somewhat considerable undertaking, equal to this project in scope and difficulty, if indeed it succeeds.

As things stand now, the NLFEAST algorithm is able to effectively address problems of moderate size, and the primary barrier for its adoption in other electronic structure codes and projects is its implementation. It is proposed that, for the final version of this thesis, a reverse communication interface be developed that would

allow other projects to easily incorporate the NLFEAST algorithm while minimizing the number of implementational details that they need to concern themselves with.

## BIBLIOGRAPHY

- [1] Jerome, J., Sievert, P. , Ye, L. , Kim, I., and Freeman, A. . Convergence of density functional iterative procedures with a newton-raphson algorithm. *J. Comput. Elec. 6* (2007), 349–352.
- [2] Hohenberg, P., and Kohn, W. Inhomogeneous electron gas. *Phys. Rev. 136* (Nov 1964), B864–B871.
- [3] Kohn, W., and Sham, L. J. Self-consistent equations including exchange and correlation effects. *Phys. Rev. 140* (Nov 1965), A1133–A1138.
- [4] Levin, A. , Zhang, D. , and Polizzi, E. . Feast fundamental framework for electronic structure calculations: Reformulation and solution of the muffin-tin problem. *Comput. Phys. Comm. 183* (2012), 2370–2375.
- [5] Polizzi, E. Density-matrix-based algorithm for solving eigenvalue problems. *Phys. Rev. B 79* (2009), 115112.
- [6] Polizzi, E. , and Tang, P. . Subspace iteration with approximate spectral projection. <http://arxiv.org/abs/1302.0432>, 2013.
- [7] Pulay, P. Convergence acceleration of iterative sequences. the case of scf iteration. *Chem. Phys. Let. 2* (1980), 393–398.
- [8] Rohwedder, T. , and Schneider, R. . An analysis for the diis acceleration method used in quantum chemistry calculations. *J. Math. Chem. 49* (2011), 1889–1914.
- [9] Valieva, M. et al. Nwchem: A comprehensive and scalable open-source solution for large scale molecular simulations. *Comput. Phys. Comm. 181* (2010), 1477–1489.