

# Computational Material Science and Engineering

Eric Polizzi\*      Yousef Saad †

August 5, 2019

## Abstract

Computational methods in Materials science have made huge strides in recent years and parallel computing methodologies have played a major role in enabling such a progress. The goal of this chapter is to discuss the current state of the art in computational materials science as it stands today, illustrating advances in the development of parallel algorithms and the impact such algorithms have had in the area. The paper is intended to be accessible to a diverse scientific computing audience. The focus of the paper will be the Density Functional Theory methodology and the solution of the eigenvalue problems that are encountered in solving the resulting equations.

---

\*Dept. of Electrical and Computer Engineering, Department of Mathematics and Statistics, University of Massachusetts at Amherst. Work supported by NSF awards CCF-1510010 and SI2-SSE-1739423.

†Dept. of Computer Science and Engineering, University of Minnesota, Minneapolis, MN 55455. Work supported by NSF award CCF-1505970.

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Quantum descriptions of matter</b>	<b>3</b>
<b>3</b>	<b>Density Functional Theory and the Kohn-Sham equation</b>	<b>5</b>
3.1	The Kohn Sham equation . . . . .	6
3.2	Pseudopotentials . . . . .	7
3.3	Discretization . . . . .	7
3.3.1	Planewaves . . . . .	7
3.3.2	LCAO . . . . .	8
3.3.3	Real-space Methods . . . . .	8
3.4	Comparison of discretization approaches . . . . .	11
3.5	Computing the electron density . . . . .	11
<b>4</b>	<b>Solution of the eigenvalue problem</b>	<b>12</b>
4.1	Traditional methods: Subspace iteration, Lanczos and Davidson . . . . .	12
4.2	Nonlinear Chebyshev filtered subspace iteration . . . . .	14
4.3	EVSL: Filtering and spectrum slicing . . . . .	15
4.4	FEAST: Rational filtering and spectrum slicing . . . . .	18
<b>5</b>	<b>Conclusion</b>	<b>21</b>

## 1 Introduction

Among the many jobs running at any given time of a high-performance computing facility today, it is likely that those related to quantum mechanical calculations will figure prominently. The numerical simulations that arise from the modeling of matter are very demanding both in terms of memory and computational power. These simulations combine ideas and techniques from a variety of disciplines including physics, chemistry, applied mathematics, numerical linear algebra, and computer science.

Determining matter’s electronic structure can be a major challenge: The number of particles is large [a macroscopic amount contains  $\approx 10^{23}$  electrons and nuclei] and the physical problem is intrinsically complex.

The most significant change in computational methods used in materials in the past two decades has undoubtedly been the systematic use of parallel processing. This revolution in methodology has taken some time to unravel and then mature. For example, it was not clear in the early 1990s whether massively parallel computing could be achieved with vector processors or if a message passing interface would be best. There were phases in which programming models and languages took different directions. As architectures changed over the years, the software and techniques have been in constant flux. At the same time algorithms have also evolved considerably, in part to cope with the new computing environments and the enormous power afforded by new hardware.

Most of the gains in speed combine advances from 3 areas: simplifications or improvements from physical models, effective numerical algorithms, and powerful hardware+software tools.

In terms of physical models, the biggest advances in nanotechnology were made in the sixties with the emergence of Density Functional Theory (DFT) which made it possible to approximate the initial problem by one which involves unknowns that are functions of only one space variables instead of  $N$  space variables, for  $N$ -particle systems in the original Schrödinger equation. Thus instead of dealing with functions in  $\mathbb{R}^{3N}$  we only need to handle functions in  $\mathbb{R}^3$ . DFT provides (in principle) an exact method for calculating the ground state energy and electron density of a system of interacting electrons using exchange-correlation density functionals, and a set of single electron wavefunctions solution of an eigenvalue equation.

The number of atoms contained in nanostructures of technological interests usually range from few hundreds to many thousands posing a unique challenge for DFT electronic structure modeling and computation. Many modeling advances were made in designing various discretization techniques to accommodate atomistic systems with high level of accuracy. In addition, since both system size and number of needed eigenpairs to compute the electron density depend linearly on the number of atoms, progress in electronic structure calculations are tied together with advances in eigenvalue algorithm and their scalability on parallel architectures.

The goal of this paper is not to provide another exhaustive review of the state of the art in materials but rather to discuss the impact that parallel processing has had on the design of algorithms. From physics to algorithms, we will begin with a review of the basics, and then discuss the recent advances made in electronic structure calculations using appropriate discretization schemes and new parallel algorithms that can fully capitalize on modern HPC computing platforms.

## 2 Quantum descriptions of matter

Consider  $N$  nucleons of charge  $Z_n$  at positions  $\{\mathbf{R}_n\}$  for  $n = 1, \dots, N$  and  $M$  electrons at positions  $\{\mathbf{r}_i\}$  in space, for  $i = 1, \dots, M$ . The non-relativistic, time-independent Schrödinger equation that describes the physical state of the system can be written as:

$$\mathcal{H} \Psi = E \Psi \quad (1)$$

where the many-body wavefunction  $\Psi$  is of the form

$$\Psi \equiv \Psi(\mathbf{R}_1, \mathbf{R}_2, \mathbf{R}_3, \dots; \mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \dots) \quad (2)$$

and  $E$  is the total electronic energy. The Hamiltonian  $\mathcal{H}$  in its simplest form can be written as

$$\begin{aligned} \mathcal{H}(\mathbf{R}_1, \mathbf{R}_2, \mathbf{R}_3, \dots; \mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \dots) = & \sum_{n=1}^N \frac{-\hbar^2 \nabla_n^2}{2M_n} + \frac{1}{2} \sum_{\substack{n, n'=1, \\ n \neq n'}}^N \frac{Z_n Z_{n'} e^2}{|\mathbf{R}_n - \mathbf{R}_{n'}|} \\ & + \sum_{i=1}^M \frac{-\hbar^2 \nabla_i^2}{2m} - \sum_{n=1}^N \sum_{i=1}^M \frac{Z_n e^2}{|\mathbf{R}_n - \mathbf{r}_i|} + \frac{1}{2} \sum_{\substack{i, j=1 \\ i \neq j}}^M \frac{e^2}{|\mathbf{r}_i - \mathbf{r}_j|} \quad (3) \end{aligned}$$

Here,  $M_n$  is the mass of the nucleon,  $\hbar$  is Planck's constant divided by  $2\pi$ ,  $m$  is the mass of the electron, and  $e$  is the charge of the electron.

The above Hamiltonian includes the kinetic energies for each nucleon (first sum in  $\mathcal{H}$ ), and each electron (3rd sum), the inter-nuclei repulsion energies (2nd sum), the nuclei-electronic (Coulomb)

attraction energies (4th sum), and the electron-electron repulsion energies (5th sum). Each Laplacian  $\nabla_n^2$  involves differentiation with respect to the coordinates of the  $n^{\text{th}}$  nucleon. Similarly the term  $\nabla_i^2$  involves differentiation with respect to the coordinates of the  $i^{\text{th}}$  electron.

In principle, the electronic structure of any system is completely determined by (1) by finding the wavefunction  $\Psi$  that minimizes the energy  $\langle \Psi | \mathcal{H} | \Psi \rangle$  over all normalized wavefunctions  $\Psi$ . The function  $\Psi$  has a probabilistic interpretation: for the minimizing wavefunction  $\Psi$ ,

$$|\Psi(\mathbf{R}_1, \dots, \mathbf{R}_N; \mathbf{r}_1, \dots, \mathbf{r}_M)|^2 d^3\mathbf{R}_1 \cdots d^3\mathbf{R}_N d^3\mathbf{r}_1 \cdots d^3\mathbf{r}_M$$

represents the probability of finding nucleon 1 in volume  $|\mathbf{R}_1 + d^3\mathbf{R}_1|$ , nucleon 2 in volume  $|\mathbf{R}_2 + d^3\mathbf{R}_2|$ , etc. However, solving (1) is not practically feasible for systems that include more than just a few atoms.

The main computational difficulty stems from the nature of the wavefunction which depends on all coordinates of all particles (nuclei and electrons) simultaneously. To give an illustration of this, imagine we have 10 Atoms each with 14 electrons [e.g., Silicon]. This represents a total of  $15 \times 10 = 150$  particles. The wavefunction in its form without spin is  $\Psi(R_1, \dots, R_{14}, r_1, \dots, r_{140})$  and it must be discretized. A simple scheme would be some finite difference method. If we use 100 points for each of the 150 coordinates we would get a huge number of unknowns:

$$\# \text{ Unknowns} = \underbrace{100}_{\text{part.1}} \times \underbrace{100}_{\text{part.2}} \times \cdots \times \underbrace{100}_{\text{part.150}} = 100^{150}$$

The original Schrödinger equation (1) can be viewed as an eigenvalue problem: we need compute the smallest eigenvalue and associated eigenvector of the Hamiltonian. It can also be viewed from the point of view of optimization since finding the smallest eigenpair is known to be equivalent to finding the wavefunction  $\Psi$  that minimizes the Rayleigh quotient:

$$E = \langle \Psi | \mathcal{H} | \Psi \rangle \equiv \frac{\int \Psi^* \mathcal{H} \Psi d^3\mathbf{R}_1 d^3\mathbf{R}_2 d^3\mathbf{R}_3 \cdots d^3\mathbf{r}_1 d^3\mathbf{r}_2 d^3\mathbf{r}_3 \cdots}{\int \Psi^* \Psi d^3\mathbf{R}_1 d^3\mathbf{R}_2 d^3\mathbf{R}_3 \cdots d^3\mathbf{r}_1 d^3\mathbf{r}_2 d^3\mathbf{r}_3 \cdots} \quad (4)$$

The symbols *bra* (for  $\langle |$ ) and *ket* (for  $| \rangle$ ) are common in chemistry and physics. When applying the Hamiltonian to a state function  $\Psi$  the result is another state function:  $\Phi = |\mathcal{H}|\Psi \rangle$ . The inner product of this function with another function  $\Theta$  is  $\langle \Theta | \Phi \rangle$  which is a scalar.

The first, and basic, approximation made to reduce complexity is the *Born-Oppenheimer* or adiabatic approximation. This approximates separates the nuclear and electronic degrees of freedom: exploiting the fact that the nuclei have a much bigger mass than the electrons, it can be assumed that the electrons will respond “instantaneously” to the nuclear coordinates. This allows one to treat the nuclear coordinates as classical parameters. For most condensed matter systems, this assumption is highly accurate [79, 29]. Under this approximation the first term in (3) vanishes and the second becomes a constant, so we end up with the simplified Hamiltonian:

$$\mathcal{H}(\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \dots) = \sum_{i=1}^M \frac{-\hbar^2 \nabla_i^2}{2m} - \sum_{n=1}^N \sum_{i=1}^M \frac{Z_n e^2}{|\mathbf{R}_n - \mathbf{r}_i|} + \frac{1}{2} \sum_{\substack{i,j=1 \\ i \neq j}}^M \frac{e^2}{|\mathbf{r}_i - \mathbf{r}_j|}. \quad (5)$$

This simplified Hamiltonian is often taken as a practical replacement of the original problem.

Its eigenfunctions determine the *states*. There are infinitely many states, labeled 1, 2,  $\dots$  by increasing eigenvalue. Each eigenvalue represents an ‘energy’ level of the state. The state with

lowest energy (smallest eigenvalue) is the *ground state*. It determines stable structures, mechanical deformations, phase transitions, and phonons. States above the ground state are known as ‘excited states’. They are used to study many body effects, quasi-particles, electronic band gaps, optical properties, etc.

A direct numerical treatment of the Schrödinger equation using the simplified many-body Hamiltonian (5), leads to a deceptively simple linear eigenvalue problem which is still intractable because of its exponential growing dimension with the number of electrons. This limitation has historically motivated the need for lower levels of sophistication in the description of the electronic structure using a single electron picture approximation where the size of the Hamiltonian operator scales linearly with the number of electrons. It is within the single-electron picture that first-principle electronic structure calculations are usually performed [49] using either (post) Hartree-Fock type methods widely used in quantum chemistry, or as an alternative to wave function based methods, the Density Functional Theory (DFT) associated with the Kohn-Sham equations [31, 36]. The next Section will be reviewing the electronic structure modeling process using DFT.

### 3 Density Functional Theory and the Kohn-Sham equation

A breakthrough in the solution of the Schrödinger equation came with the discovery of Density Functional Theory. In a series of papers, Hohenberg, Kohn, and Sham established a theory in which the many body wave function was replaced by one-electron orbitals [48, 31, 36]. The basic idea is that the state of the system will now be expressed in terms of the charge density  $\rho$ , which is a distribution of probability, i.e.,  $\rho(\mathbf{r}_1)d^3\mathbf{r}_1$  represents - in a probabilistic sense - the number of electrons (all electrons) in the infinitesimal volume  $d^3\mathbf{r}_1$ . It is easy to calculate the charge density from a given wavefunction. The fundamental theorem which these authors were able to state is that this mapping is one-to-one, i.e., given the charge density it should be possible to obtain the ground state wavefunction. In essence there is a certain Hamiltonian – as defined by a certain potential (that depends on  $\rho$ ) whose minimum energy is reached for the ground state  $\Psi$ . Kohn and Sham wrote this Hamiltonian as

$$H_{KS} = \frac{\hbar^2}{2m} \nabla^2 + V_N(\rho) + V_H(\rho) + V_{xc}(\rho) \quad (6)$$

where  $V_N(\rho)$  is the external potential  $V_H(\rho)$  is the Hartree potential and  $V_{xc}(\rho)$  is the exchange correlation potential. Note the dependence on the charge density  $\rho$  which is itself implicitly defined from the set of occupied eigenstates  $\phi_i, i = 1, \dots, N$  of (6) by:

$$\rho(\mathbf{r}) = 2 \sum_{j=1}^{occup} |\phi_j(\mathbf{r})|^2, \quad (7)$$

where  $N$  is the number occupied states (i.e. number of electrons) and the factor 2 accounts for the electron spin.

### 3.1 The Kohn Sham equation

We can now write the *Kohn-Sham equation* [36] for the electronic structure of matter as

$$\left( \frac{-\hbar^2 \nabla^2}{2m} + V_N(\mathbf{r}) + V_H(\mathbf{r}) + V_{xc}[\rho(\mathbf{r})] \right) \phi_i(\mathbf{r}) = E_i \phi_i(\mathbf{r}) \quad (8)$$

As stated above the charge density is defined in terms of the orbitals  $\phi_i$  given by (7).

Given a charge density  $\rho$  the Hartree Potential  $V_H$  is the solution of Poisson equation:

$$\nabla^2 V_H = -4\pi\rho(r) \quad (9)$$

The exchange & correlation potential  $V_{xc}$  is unknown in theory but it is approximated by a potential in different ways, the simplest of which is the Local Density Approximation (LDA).

Therefore, this equation is usually solved “self-consistently” in the sense that if a given  $\rho^{in}$ , as obtained from a set of occupied states  $\phi_i(r)$ ,  $i = 1, \dots, N$  is utilized to compute new occupied states from (6), and a new charge density  $\rho^{out}$  is then computed according to (7) then  $\rho$  and  $\rho^{out}$  should be the same. The SCF procedure takes some initial approximate charge to estimate the exchange-correlation potential and this charge is used to determine the Hartree potential from (9). These approximate potentials are inserted in the Kohn-Sham equation and the total charge density determined as in (7). The “output” charge density is used to construct new exchange-correlation and Hartree potentials. The process is repeated until the input and output charge densities (or potentials) are close enough. This process is illustrated in Figure 1.

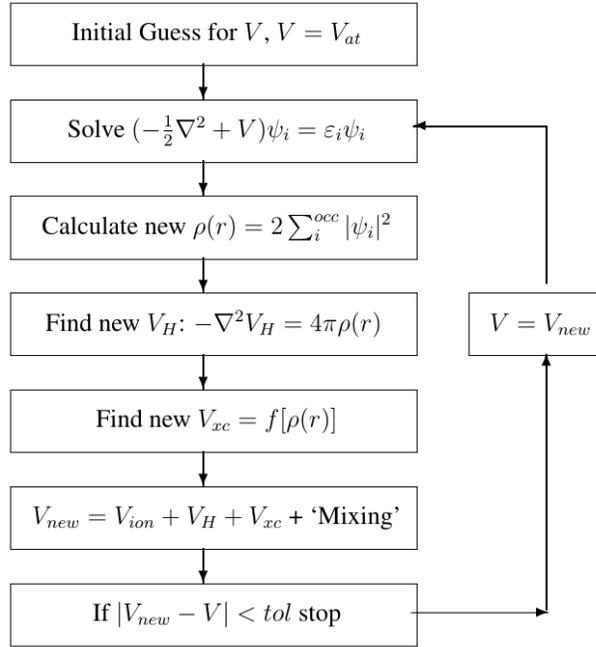


Figure 1: The Self-Consistent Field Iteration.

DFT has been widely used in computational material science and quantum chemistry over the past few decades, since it provides (in principle) an exact method for calculating the ground

state density and energy of a system of interacting electrons using a non-linear single electron equation associated with exchange-correlation (XC) functionals. In practice, the reliability of DFT depends on the numerical approximations used for the XC terms that range from the simplest local density approximation (LDA) to more advanced schemes which are still the subject of active research efforts [14, 65, 70]. Solutions of the DFT/Kohn-Sham problem are routinely used in the calculations of many ground state properties including: total energy and ionization potential; and, via perturbation: crystal-atomic structure, ionic forces, vibrational frequencies and phonon bandstructure.

## 3.2 Pseudopotentials

When discretizing the KS equation, we run into a major difficulty which arises from the different scales of the lengths involved. The inner (core) electrons are highly localized and tightly bound compared to the outer (valence electrons). Another major advance in the solid-state physics field was the advent of pseudopotential techniques which remove the core states from the problem and replacing the all electron potential by one that replicates only the chemically active, valence electron states[16]. This is possible because the physical properties of solids depend much more on the valence electrons than on the core electrons. The whole art is then to construct pseudopotentials that reproduce the valence state properties such as the eigenvalue spectrum and the charge density outside the ion core.

## 3.3 Discretization

One can identify three main discretization techniques that have been widely used over the past four decades by both the quantum chemistry and the solid-state physics communities [49]: (i) the plane wave expansion scheme, (ii) the linear combination of atomic orbitals (LCAO) (along with the dominant use of Gaussian local basis sets), and (iii) the real-space mesh techniques (also loosely called “numerical grids”) based on finite difference method (FDM), finite element method (FEM), spectral element or wavelets methods. Each of these approaches have pros and cons.

### 3.3.1 Planewaves

Planewave bases have been very popular in materials science and solid-state physics for performing bandstructure calculations. For example, in the context of pseudopotentials methods, planewave bases can be quite effective in representing the orbitals for crystalline periodic matter, requiring a small number of planewaves. This leads to a compact representation of the Schrödinger operator. The resulting matrix is dense in Fourier (plane wave) space, but it is not formed explicitly. Instead, matrix-vector product operations are performed with the help of fast Fourier transforms. This plane wave approach is akin to spectral techniques used in solving certain types of partial differential equations [24]. The plane wave basis used is of the form:

$$\psi_{\mathbf{k}}(\mathbf{r}) = \sum_{\mathbf{G}} \alpha(\mathbf{k}, \mathbf{G}) \exp(i(\mathbf{k} + \mathbf{G}) \cdot \mathbf{r}) \quad (10)$$

where  $\mathbf{k}$  is the wave vector,  $\mathbf{G}$  is a reciprocal lattice vector and  $\alpha(\mathbf{k}, \mathbf{G})$  represent the coefficients of the basis. Thus, each planewave is labeled by a wavevector which is a triplet of 3 integers,

i.e.,  $\mathbf{k} = (\mathbf{k}_1, \mathbf{k}_2, \mathbf{k}_3)$ . The vector parameter  $\mathbf{G}$  translates the periodicity of the wavefunction with respect to a lattice which attempts to describe a crystalline structure of the atoms.

### 3.3.2 LCAO

An appealing approach uses a basis set of orbitals localized around the atoms. This is the approach for example taken in the SIESTA code [68] where, with each atom  $a$ , is associated a basis set of functions which combine radial functions around  $a$  with spherical harmonics:

$$\phi_{lmn}^a(\mathbf{r}) = \phi_{ln}^a(\mathbf{r}_a)Y_{lm}(\hat{\mathbf{r}}_a)$$

where  $\mathbf{r}_a = \mathbf{r} - \mathbf{R}_a$ .

In contrast to plane wave methods, LCAO techniques cannot be universally and systematically improved towards convergence. On the positive side, LCAO benefits from a large collection of local basis sets that has been refined along the years by the quantum chemistry community to obtain high-level of accuracy in simulations. Atomic orbital basis also yields much smaller matrices and requires less memory than plane wave methods. The sparsity of the matrices depends on how many neighboring atoms are accounted for in the linear combination.

A popular basis employed with pseudopotentials is that of Gaussian orbitals[13, 32, 17, 33]. Gaussian bases have the advantage of yielding analytical matrix elements provided the potentials are also expanded in Gaussians. However, the implementation of a Gaussian basis is not as straightforward as with plane waves. For example, numerous indices must be employed to label the state, the atomic site, and the Gaussian orbitals used.

### 3.3.3 Real-space Methods

When applied to electronic structure calculations, real-space mesh techniques exhibit the following significant advantages: (i) they avoid deriving global basis sets for a specific problem by employing universal mathematical approximations at local regions in the physical space; (ii) they can easily handle the treatment of various boundary conditions such as Dirichlet, Neumann, or mixed (such as self-energy functions useful in transport problems [59]); (iii) they produce very sparse matrices and are cast as linear scaling electronic structure discretization methods; (iv) they allow to solve the Poisson equation for electrostatics using the same numerical grid; (v) they can benefit from the recent advances made in mathematical modeling techniques and numerical algorithm design including multigrids, domain decomposition, or direct and Krylov-subspace iterative techniques. All of these properties motivated the development of real-space mesh software packages for electronic structure calculations such as Octopus [8, 5], MIKA [3], PARSEC [37, 6], or NESSIE [4].

**Finite Differences** An appealing discretization alternative is to avoid traditional explicit bases altogether and work instead in real space, by discretizing the space variable. This can be achieved with Finite Difference Methods (FDM), see, e.g., [37, 22, 53, 41, 72, 30, 27, 9]. FDM is the simplest real space method which utilizes finite difference discretization on a cubic grid. One of the most popular schemes is to use regular grids with high-order discretizations [25] for the Laplacian which represents the kinetic energy operator. Such high order schemes significantly improve convergence of the eigenvalue problem when compared with standard, low order, finite

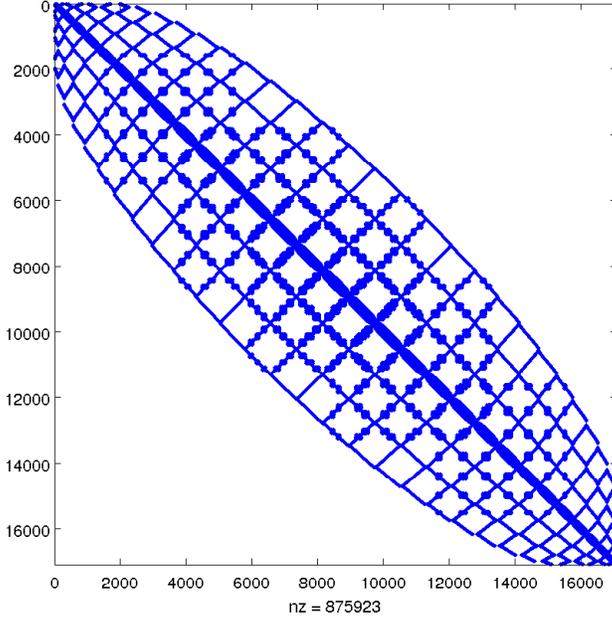


Figure 2: Matrix resulting from a 12-th order ( $M = 6$ ) FD discretization of the Kohn-Sham equation. The matrix is obtained from a Parsec simulation of a small Silicon cluster passivated by Hydrogen atoms (Si10H16). A spherical domain is used which explains the curved diagonals

difference methods. With a uniform grid where the points are described in a finite domain by  $(x_i, y_j, z_k)$ ,  $\frac{\partial^2 \psi}{\partial x^2}$  at  $(x_i, y_j, z_k)$  is approximated by

$$\frac{\partial^2 \psi}{\partial x^2} = \sum_{n=-M}^M C_n \psi(x_i + nh, y_j, z_k) + O(h^{2M}), \quad (11)$$

where  $h$  is the grid spacing. Thus using a total of  $2M + 1$  points in each direction yields an error of order  $O(h^{2M})$ . Algorithms are available to compute the coefficients  $C_n$  for arbitrary order in  $h$  [25].

With the kinetic energy operator expanded as in (11), one can set up a one-electron Schrödinger equation over a grid. One may assume a uniform grid, but this is not a necessary requirement. Once the Kohn Sham equation is discretized using high order finite differences, we obtain a standard matrix eigenvalue problem of the form:

$$A\psi = \lambda\psi \quad (12)$$

in which  $A$  is a real symmetric and sparse matrix. Note that the discretization (11) for the kinetic energy term, will lead to  $2M$  nonzero entries for each of the 3 directions, plus the diagonal entry, so we end up with a total of  $6M + 1$  nonzero entries, to which we need to add the nonzero entries that come from the other terms of the Hamiltonian. The Hartree and Exchange correlation terms usually lead to a diagonal matrix, while the external potential is non-local and leads to a sort of low-rank matrix centered around each atom. An example of such a matrix is shown in Figure 2.

A grid based on points uniformly spaced in a three dimensional cube is typically used. Many points in the cube are far from any atoms in the system and the wave function on these points may

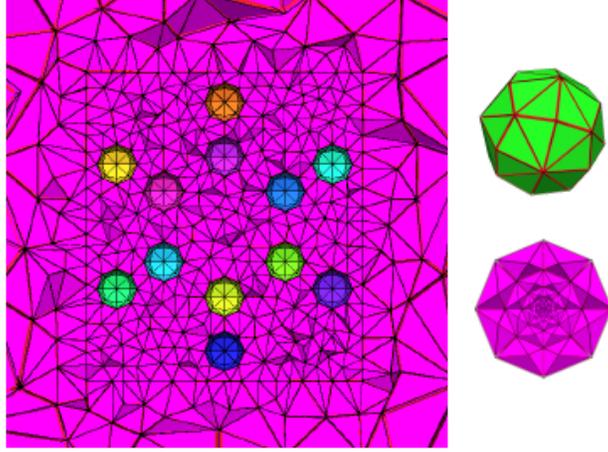


Figure 3: The figures represent a 2D cross-section of 3D finite element mesh using a coarser interstitial mesh (left) connecting all of the atoms of a Benzene molecule, and a much finer mesh (right) for the atom-centered regions suitable to capture the highly localized core states around the nuclei.

be replaced by zero. Special data structures may be used to discard these points and keep only those having a nonzero value for the wave function. The size of the Hamiltonian matrix is usually reduced by a factor of two to three with this strategy, which is quite important considering the large number of eigenvectors which must be saved. Further, since the Laplacian can be represented by a simple stencil, and since all local potentials sum up to a simple diagonal matrix, the Hamiltonian need not be stored explicitly as a sparse matrix. Handling the ionic pseudopotential is complex as it consists of a local and a non-local term. In the discrete form, the non-local term becomes a sum over all atoms,  $a$ , and quantum numbers,  $(l, m)$  of rank-one updates:

$$V_{ion} = \sum_a V_{loc,a} + \sum_{a,l,m} c_{a,l,m} \mathbf{U}_{a,l,m} \mathbf{U}_{a,l,m}^T \quad (13)$$

where  $\mathbf{U}_{a,l,m}$  are sparse vectors which are only non-zero in a localized region around each atom,  $c_{a,l,m}$  are normalization coefficients.

**Finite Elements** One of the main advantage of finite element method (FEM) is its flexibility to be used with non-uniform meshes and include local refinement by adding more nodes in various regions of interests. In electronic structure calculations, local refinement is important to capture the strong variations of potential and electron density in the vicinity of the atom center regions. Consequently, FEM has been employed in some electronic structure codes [42, 43] as a way to bypass the pseudopotential approach and consider the full core potential. These calculations are called all-electron calculations since both core and valence electrons are included.

As illustrated in Figure 3 with the example of a Benzene molecule, the 3D finite-element mesh can be built in two steps: (i) a 3D atom-centered mesh which is highly refined around the nucleus to capture the core states, and (ii) a much coarser 3D interstitial mesh that connects all the atom-centered regions. For the atom-centered mesh, successive layers of polyhedra as proposed in [42], along with cubic finite element, do provide high-level of accuracy for solving single atom systems.

Not only, the distance between layers can be systematically refined while approaching the nucleus, the outer-layer is consistently providing the same (relatively small) number of connectivity nodes that will be used by the coarser interstitial mesh at the surface with the atoms. This approach, used in the NESSIE code [4], is ideally suited for domain-decomposition techniques and parallel computing [34].

### 3.4 Comparison of discretization approaches

Real-space approaches have a number of advantages and have become popular in recent years, see [74, 19, 20, 18, 12, 27, 80, 54, 35, 41, 11, 24, 23]. It is worth mentioning that the Gordon Prize in 2011 was awarded to a team that relied on finite difference discretization [28] a testimony of the capability of this approach. One of the attractions of space approaches relative to planewaves is that they bypass many of the difficulties involved with non-periodic systems. Although the resulting matrices are usually (much) larger than with plane waves, they are sparse and the methods are easy to implement on parallel computers. However, even on sequential machines, real space methods can be faster than methods based on traditional approaches.

Comparing finite difference with finite element discretization methods one can say that finite elements are good at reducing the total number of variables involved but they may be more difficult to implement.

### 3.5 Computing the electron density

Within the SCF-DFT procedure, solving the linear and symmetric eigenvalue problem at each given iteration step, becomes a very challenging part of the calculations.

The most challenging aspect of electronic structure calculations is the high computational cost of calculating the electron density (7) at each step of the DFT/Kohn-Sham self-consistent iterations (see Figure 1). The electron density is traditionally calculated using all the wave functions (eigenvectors) solution of the Kohn-Sham eigenvalue problem over all occupied energy states. In order to characterize complex systems and nanostructures of current technological interests, many thousands of eigenpairs may indeed be needed. Indeed, all valence electrons (and core electrons if applicable) need to be included in the calculation.

An alternative approach to the wave function formalism consists of performing a contour integration of the Green's function matrix  $G(z) = (zB - A)^{-1}$  over the complex energy space [76, 71]. We note that  $A$  is the Hamiltonian matrix  $A$ , and  $B$  represents the basis function overlap matrix (i.e. or mass matrix) which is obtained after discretization ( $S = I$  using FDM). At zero temperature, the resulting expression for the electron density in real-space is:

$$\rho(\mathbf{r}) = -\frac{1}{\pi i} \int_{\mathcal{C}} \text{diag}(G(z)) dz = 2 \sum_{j=1}^{\text{occup}} |\phi_j|^2, \quad (14)$$

where the clockwise complex contour  $\mathcal{C}$  includes all the occupied eigenvalues. The contour integration technique represents a priori an attractive alternative approach to the traditional eigenvalue problem since the number of Green's function to be calculated (typically of order  $\sim O(10)$  using Gaussian quadrature) is independent of the size of the system. In addition, only the diagonal elements of the Green's function needs to be computed (independently) along the integration points.

This problem has motivated the development of new algorithms that are able to directly obtain the diagonal elements of the inverse of any sparse matrices. For 1D physical structures such as long nanowires which give rise to banded matrices after discretization, it is possible to perform efficient  $O(N)$  calculations for obtaining the diagonal elements of the Green’s function [10, 77, 47]. For arbitrary 3D systems (i.e. beyond nanowire structures), however, the numerical complexity of a direct solver such as PEXSI is  $O(N^2)$  [7].

The Green’s function-based alternative to the wave function formalism for computing electron density, gives rise to difficulties in algorithmic complexity, parallel scalability and accuracy. In that regard, it is difficult to bypass the wave function formalism, and progress in large-scale electronic structure calculations can be tied together with advances in numerical algorithms for addressing the eigenvalue problem that will be discussed in the next Section.

## 4 Solution of the eigenvalue problem

One significant characteristic of the eigenvalue problem that arises from the Kohn-Sham equation is that the number of required eigenvectors is proportional to the atoms in the system, and can grow up to thousands, possibly many more depending on the compound being studied. This means that we will have to store an eigenbasis consisting of a large number of vectors. In addition, the vectors of this basis need to be orthogonal. In fact, the biggest part of the cost of existing eigenvalue codes is related to orthogonalization.

In this Section, we will briefly review various diagonalization methods ranging from Lanczos and Davidson to polynomial and rational filtering, and introduce the notion of ‘slicing’. One of the main motivations of filtering is to allow ‘slices’ of the spectrum to be computed independently of one another and orthogonalization between eigenvectors in different slices is no longer necessary.

### 4.1 Traditional methods: Subspace iteration, Lanczos and Davidson

Large computations based on DFT approaches started in the 1970s after the breakthrough results of Kohn, Hohenberg, and Sham. The use of planewave bases dominated the arena of electronic structure from that period onward - starting with the trend-setting Car and Parrinello [15] article which was the catalyst in the development of computational codes using planewaves and pseudopotentials. Most computations in the mid-80s to the 1990s, and still today, rely on planewave bases. Since the matrices involved were dense and memory was expensive, this was a major limiting factor at the beginning. However, it was soon realized that it was not necessary to store the dense matrix if a code that accesses the matrix only to perform matrix-vector products (‘matvecs’ thereafter) is employed [50], see also [51]. This is achieved by working in Fourier space and using FFT to go back and forth from real to Fourier space to perform the operations needed for the matvec. An early code based on subspace iteration for eigenvalue problems and called Ritzit, initially written by Rutishauser in Algol [60], became a de facto standard.

The Lanczos algorithm [38] discovered in 1950 re-emerged in the early 1980s in the linear algebra community as a contender to subspace iteration due mainly to its superior effectiveness when computing a small number of eigenvalues at one end of the spectrum. In exact arithmetic, the Lanczos algorithm generates an orthonormal basis  $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_m$ , of the Krylov subspace

$\text{Span}\{\mathbf{v}, \mathcal{A}\mathbf{v}, \mathcal{A}^2\mathbf{v}, \dots, \mathcal{A}^{m-1}\mathbf{v}\}$  via an inexpensive 3-term recurrence of the form :

$$\beta_{j+1}\mathbf{v}_{j+1} = \mathcal{A}\mathbf{v}_j - \alpha_j\mathbf{v}_j - \beta_j\mathbf{v}_{j-1}.$$

In the above sequence,  $\alpha_j = \mathbf{v}_j^H \mathcal{A}\mathbf{v}_j$  and  $\beta_{j+1} = \|\mathcal{A}\mathbf{v}_j - \alpha_j\mathbf{v}_j - \beta_j\mathbf{v}_{j-1}\|_2$ . So the  $j^{\text{th}}$  step of the algorithm starts by computing  $\alpha_j$  and then proceeds to form the vector  $\hat{\mathbf{v}}_{j+1} = \mathcal{A}\mathbf{v}_j - \alpha_j\mathbf{v}_j - \beta_j\mathbf{v}_{j-1}$  and then  $v_{j+1} = \hat{\mathbf{v}}_{j+1}/\beta_{j+1}$ . Note that for  $j = 1$ , the formula for  $\hat{\mathbf{v}}_2$  changes to  $\hat{\mathbf{v}}_2 = \mathcal{A}\mathbf{v}_2 - \alpha_2\mathbf{v}_2$ .

Suppose that  $m$  steps of the recurrence are carried out and consider the tridiagonal matrix,

$$\mathcal{T}_m = \begin{pmatrix} \alpha_1 & \beta_2 & & & \\ \beta_2 & \alpha_2 & \beta_3 & & \\ & \ddots & \ddots & \ddots & \\ & & & \beta_m & \alpha_m \end{pmatrix}$$

Further, denote by  $\mathcal{V}_m$  the  $n \times m$  matrix  $\mathcal{V}_m = [\mathbf{v}_1, \dots, \mathbf{v}_m]$  and by  $\mathbf{e}_m$  the  $m^{\text{th}}$  column of if the  $m \times m$  identity matrix. After  $m$  steps of the algorithm, the following relation holds:

$$\mathcal{A}\mathcal{V}_m = \mathcal{V}_m\mathcal{T}_m + \beta_{m+1}\mathbf{v}_{m+1}\mathbf{e}_m^T.$$

It is observed, and can be theoretically shown, that some of the eigenvalues of the tridiagonal matrix  $\mathcal{H}_m$  will start approximating corresponding eigenvalues of  $\mathcal{A}$  when  $m$  becomes large enough. An eigenvalue  $\tilde{\lambda}$  of  $\mathcal{H}_m$  is called a Ritz value, and if  $\mathbf{y}$  is an associated eigenvector, then the vector  $\mathcal{V}_m\mathbf{y}$  is, by definition, the Ritz vector, i.e., the approximate eigenvector of  $\mathcal{A}$  associated with  $\tilde{\lambda}$ . If  $m$  is large enough, the process may yield good approximations to the desired eigenvalues  $\lambda_1, \dots, \lambda_s$  of  $\mathcal{H}$ , corresponding to the occupied states, i.e., all occupied eigenstates.

In practice, orthogonality of the Lanczos vectors, which is guaranteed in theory, is lost and this phenomenon takes place as soon as one of the eigenvectors starts to converge [55, 56]. Orthogonality can be reinstated in a number of ways, see [66, 40, 67, 39, 75].

The Davidson [52] method, is a sort of preconditioned version of the Lanczos algorithm, in which the preconditioner is the diagonal of  $\mathcal{A}$ . We refer to the generalized Davidson algorithm as a Davidson approach in which the preconditioner is not restricted to being a diagonal matrix (A detailed description can be found in [62].)

The Davidson algorithm differs from the Lanczos method in the way in which it defines new vectors to add to the projection subspace. Instead of adding just  $\mathcal{A}\mathbf{v}_j$ , it preconditions a given residual vector  $\mathbf{r}_i = (\mathcal{A} - \mu_i\mathcal{I})\mathbf{u}_i$  and adds it to the subspace (after orthogonalizing it against current basis vectors). The algorithm consists of an ‘‘eigenvalue loop’’ which computes the desired eigenvalues one by one (or a few at a time), and a ‘‘basis’’ loop which gradually computes the subspace on which to perform the projection. Consider the eigenvalue loop which computes the  $i^{\text{th}}$  eigenvalue and eigenvector of  $\mathcal{A}$ . If  $\mathcal{M}$  is the current preconditioner, and  $\mathcal{V} = [\mathbf{v}_1, \dots, \mathbf{v}_k]$  is the current basis the main steps of the main loop are as follows:

1. Compute the  $i^{\text{th}}$  eigenpair  $(\mu_k, \mathbf{y}_k)$  of  $\mathcal{C}_k = \mathcal{V}_k^T \mathcal{A} \mathcal{V}_k$ .
2. Compute the residual vector  $\mathbf{r}_k = (\mathcal{A} - \mu_k\mathcal{I})\mathcal{V}_k\mathbf{y}_k$
3. Precondition  $\mathbf{r}_k$ , i.e., compute  $\mathbf{t}_k = \mathcal{M}^{-1}\mathbf{r}_k$

4. Orthonormalize  $\mathbf{t}_k$  against  $\mathbf{v}_1, \dots, \mathbf{v}_k$  and call  $\mathbf{v}_{k+1}$  the resulting vector, so  $\mathcal{V}_{k+1} = [\mathcal{V}_k, \mathbf{v}_{k+1}]$ .
5. Compute last column-row of  $\mathcal{C}_{k+1} = \mathcal{V}_{k+1}^T \mathcal{A} \mathcal{V}_{k+1}$

The original Davidson approach used the diagonal of the matrix as a preconditioner but this works only for limited cases. For planewave bases, it is possible to construct fairly effective preconditioners by exploiting the lower order bases. By this we mean that if  $\mathcal{H}_k$  is the matrix representation obtained by using  $k$  planewaves, we can construct a good approximation to  $\mathcal{H}_k$  from  $\mathcal{H}_m$  with  $m \ll k$ , by completing it with a diagonal matrix representing the larger (undesirable) modes. Note that these matrices are not explicitly computed as they are dense. This possibility of building lower dimensional approximations to the Hamiltonian which can be used to precondition the original matrix constitutes an advantage of planewave-based methods.

## 4.2 Nonlinear Chebyshev filtered subspace iteration

A big disadvantage of the Lanczos and Davidson iterations is that they do not allow to exploit previous bases that have been calculated from earlier SCF iterations. A look at Figure 1 indicates that what matters for convergence is how well the procedure is approximating the basis of the subspace corresponding to the  $n$  occupied states. At the next SCF iteration, the Lanczos algorithm starts with one vector only. This means that we cannot fully take advantage of the basis that has been computed previously. In contrast, the subspace iteration algorithm is ideal in this context. All we need to do at the next SCF iteration is update the Hamiltonian – and use whatever subspace we had from the previous SCF iteration. This constitutes a major attraction of subspace iteration. Another attraction is clearly its added parallelism.

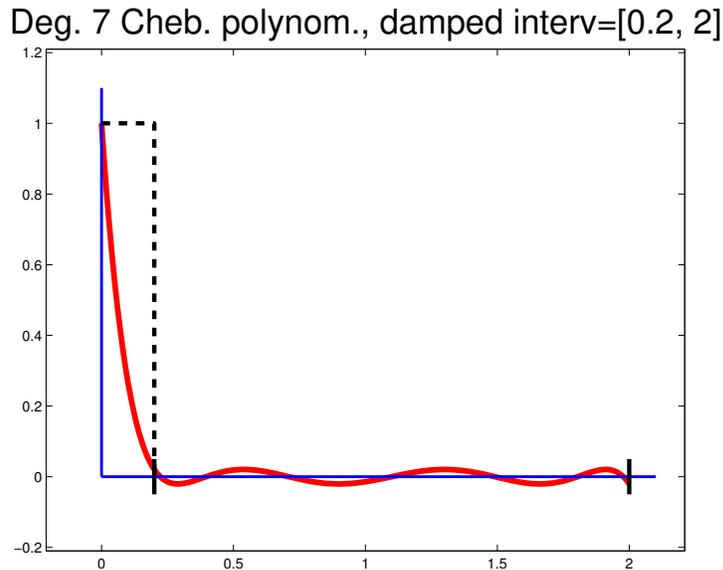


Figure 4: Degree 8 Chebyshev filter

The main ingredient of a subspace iteration procedure is the Chebyshev filtering. Given a basis  $[v_1, \dots, v_m]$ , each vector is 'filtered' as  $\hat{v}_i = P_k(A)v_i$ , where  $p_k$  is a low degree polynomial whose goal is to enhance the wanted components of these vectors in the desired eigenvectors of  $A$ . The

most common filters used are shifted and scaled Chebyshev polynomials. If  $[a, b]$  is the interval containing unwanted eigenvalues, those that must be dampened, then we use the polynomial

$$p_k(t) = \frac{C_k(l(t))}{C_k(l(g))}; \quad \text{with} \quad l(t) = \frac{2t - b - a}{b - a}$$

where  $C_k$  is the Chebyshev polynomial of degree  $k$  of the first kind and  $g$  is some approximation of the eigenvalue that is farthest from the center  $(a + b)/2$  of the interval – which is used for scaling. One such polynomial of degree 7 is shown in Figure 4. The 3-term recurrence of Chebyshev polynomial is exploited to compute  $p_k(A)v$ . If  $B = l(A)$ , then  $C_{k+1}(t) = 2tC_k(t) - C_{k-1}(t) \rightarrow w_{k+1} = 2Bw_k - w_{k-1}$ . Algorithm 4.1 provides an illustration of Chebyshev filtering.

---

**Algorithm 4.1:**  $[Y] = \text{Chebyshev\_filter}(X, m, a, b, g)$

---

```

1  $e = (b - a)/2$ ;  $c = (a + b)/2$ ;  $\sigma = e/(c - g)$ ;  $\tau = 2/\sigma$ ;
2  $Y = (\mathcal{A} * X - c * X) * (\sigma/e)$ ;
3 for  $i = 2$  to  $m$  do
4    $\sigma_{new} = 1/(\tau - \sigma)$ ;
5    $Y_t = (\mathcal{A} * Y - c * Y) * (2 * \sigma_{new}/e) - (\sigma * \sigma_{new}) * X$ ;
6    $X = Y$ ;  $Y = Y_t$ ;  $\sigma = \sigma_{new}$ ;
```

---

What was discussed above is what might be termed a standard SCF approach in which a filtered subspace iteration is used to compute the eigenvalues at each SCF iteration. The subspace iteration can also be used in a *nonlinear* way. In the nonlinear Subspace iteration, the filtering step is not used to compute eigenvectors accurately. Instead the basis is filtered and the Hamiltonian is updated immediately using these vectors. In essence the process amounts to removing one loop from the algorithm in that the SCF and the diagonalization loops merged. The new SCF iteration is illustrated in Figure 5. Experiments reported in [78] reported that this procedure can yield a factor of 10 speed-up over the more traditional one in which the inner eigenvalue loop is kept.

### 4.3 EVSL: Filtering and spectrum slicing

We mentioned earlier that a big part of the cost of computing a large number of eigenvectors is to maintain orthogonality between these vectors. The number of vectors to orthogonalize is typically of the order of the number of states which is itself proportional to the number of particles, and so the cost increases quadratically with the number of particles. This was observed early on and a number of articles sought inexpensive alternatives. One of the main ideas proposed was one based on *filtering*, i.e., transforming the Hamiltonian so as to enhance or magnify the desired part of the spectrum by a polynomial of rational transformation to enable a projection method like subspace iteration, to extract the desired eigenvalues easily. An early contribution along these lines is the article by Zunger [73] which discusses a scheme whereby the Hamiltonian  $\mathcal{H}$  is replaced by  $B = (\mathcal{H} - \sigma I)^2$ . Extracting the smallest eigenpairs of  $B$  will yield the eigenvectors associated with the eigenvalues closest to the shift  $\sigma$ . A similarly simple technique is one that is based on shift-and-invert [56] which uses a rational filter.

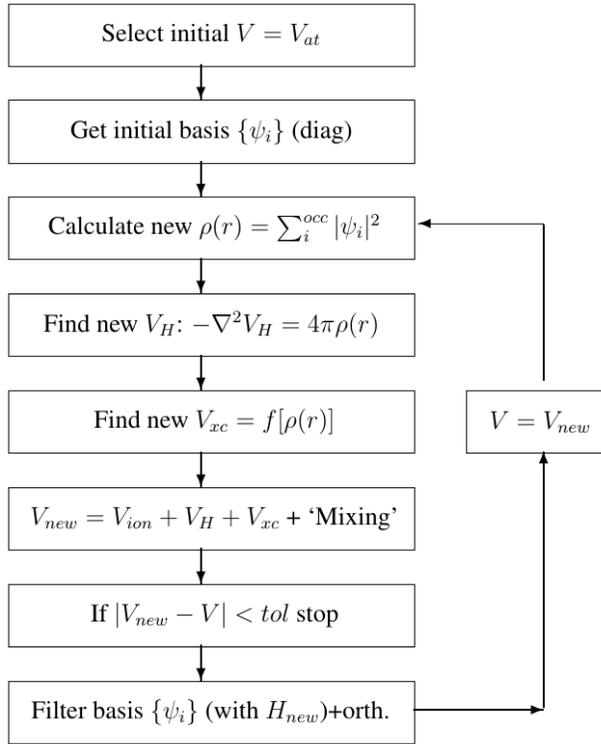


Figure 5: The Self-Consistent Field Iteration with a nonlinear subspace iteration approach

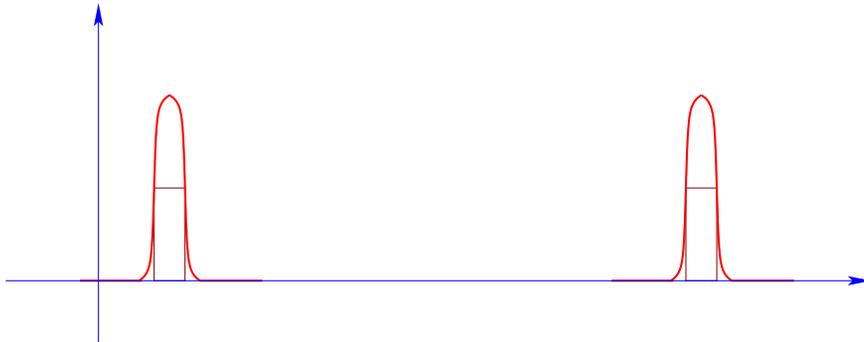


Figure 6: Two filters to compute two slices of the spectrum that are far apart. Note that eigenvectors associated with two distinct slices need not be orthogonalized against each other.

The essence of a filtering technique is to replace the original matrix  $\mathcal{A}$  by  $B = \phi(\mathcal{A})$ , where the filter  $\phi$  is either a polynomial or rational function. The main advantage of filtering is that it allows to compute different parts of the spectrum independently. A *spectrum slicing* method refers to a technique that computes the desired spectrum by sub-intervals or ‘slices’. The recently developed package named EVSL (for Eigenvalues Slicing Library) relies entirely on this strategy [1, 44, 45]. Figure 6 illustrates the main motivation for this strategy, namely that eigenvectors associated with slices that are far apart need not be orthogonalized against each other.

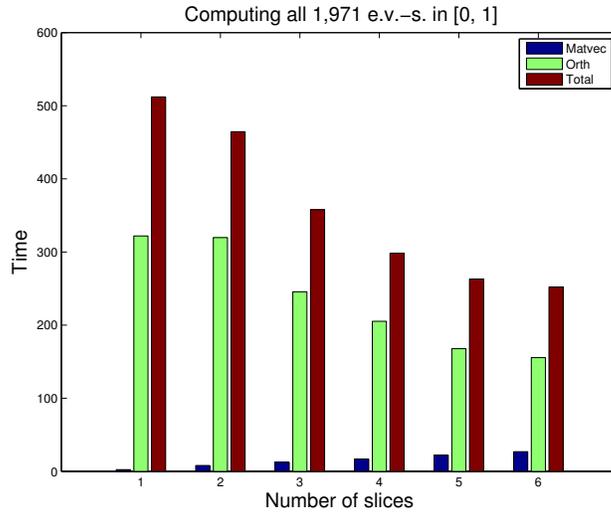


Figure 7: Cost of calculating all eigenvalues of a Laplacian matrix in the interval  $[0, 1]$  by a polynomial filtered non-restart Lanczos method. There are 1,971 eigenvalues in the interval and they are computed by slicing the spectrum into 1, 2,  $\dots$ , 6 sub-intervals.

The gain in computational cost that comes from avoiding limiting or orthogonalization can be significant both in terms of computational time and in terms of memory. For example, Figure 7 illustrates a calculation with EVSL in which all eigenvalues in the interval  $[0, 1]$  of a Laplacian discretized on a  $49 \times 49 \times 49$  centered finite difference grid. A spectrum slicing strategy is exploited and the total cost is shown as the number of intervals varies from 1 to 6. Note that in EVSL the degree of the polynomial filter is computed automatically. One can observe that orthogonalization costs are drastically reduced along with costs related to the projection process. At the same time the cost of matvecs increases but it remains insignificant relative to the rest. This calculation is performed without fully taking advantage of parallelism. If a fully parallel computation were to be implemented, each of the total times would have been divided by the number of intervals used.

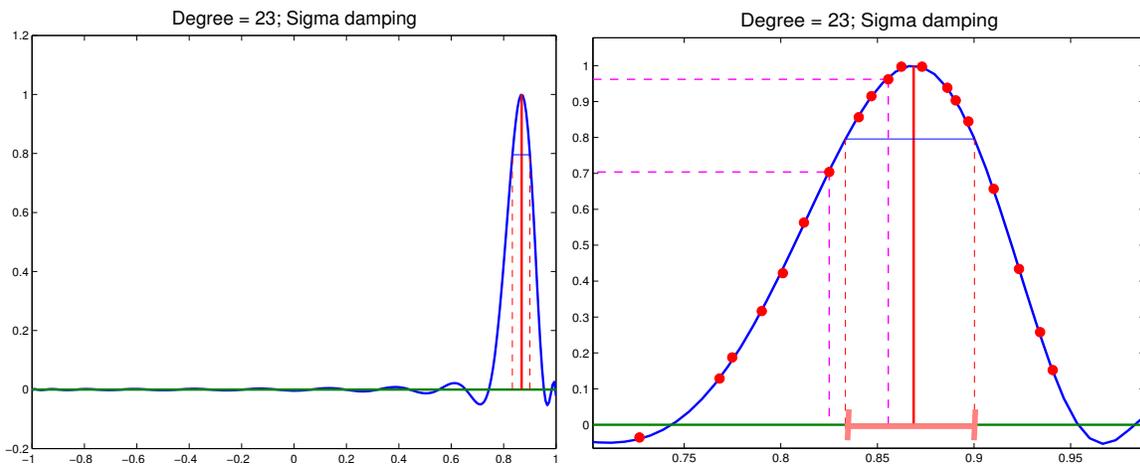


Figure 8: A filter polynomial of degree 23 (left) and a zoom of the same polynomial near the interval of interest (right).

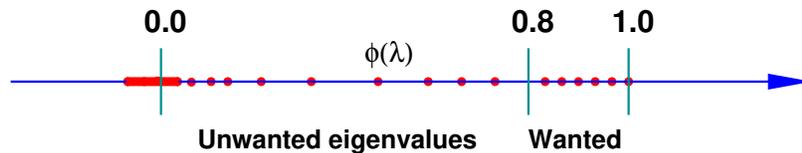


Figure 9: Eigenvalues of the filtered matrix with the filter of Figure 8

To illustrate how polynomial filtering is combined with a non-restarted version of the Lanczos algorithm we show in Figure 8 a polynomial filter of the type used in EVSL. In the figure, an eigenvalue  $\lambda_i$  located inside the interval of desired eigenvalues is transformed to  $\phi(\lambda_i)$ . The filter is designed so that any eigenvalue  $\lambda_i$  located inside the interval of desired eigenvalues is transformed into an eigenvalue  $\phi(\lambda_i)$  that is larger than or equal to a certain value (called the ‘bar’) which is  $\beta = 0.8$  in the figure. This makes it easy to distinguish between wanted eigenvalues ( $\phi(\lambda_i) \geq \beta$ ) and unwanted ones ( $\phi(\lambda_i) < \beta$ ). Figure 9 shows the filtered eigenvalues for the same problem. As is highlighted in the figure, all wanted eigenvalues of the original problem are now eigenvalues that are not smaller than  $\beta = 0.8$  for the filtered matrix. It is therefore possible to devise a strategy whereby these eigenvalues are all computed from a Lanczos algorithm with full reorthogonalization and no restarts. If the degree of the polynomial is well selected and the sub-interval contains a reasonable number of eigenvalues this strategy works quite well in practice.

EVSL solves large sparse real symmetric standard and generalized eigenvalue problems. In order to enable a spectrum slicing strategy, the methods in EVSL rely on a quick calculation of the spectral density of a given matrix, or a matrix pair. Once this is done the driver will then cut the interval into slices so that each slice will have approximately the same number of eigenvalues. What distinguishes EVSL from other currently available packages is that EVSL relies entirely on filtering techniques. While much effort has been devoted to develop effective polynomial filtering the package also implements rational filters. The projection methods developed in the package are the Lanczos methods without restart, or with thick restart, as well as the subspace iteration method. Various interfaces are available for various scenarios, including matrix-free modes, whereby the user can supply his/her own functions to perform matrix-vector operations or to solve sparse linear systems. A fully parallel version is currently being developed.

#### 4.4 FEAST: Rational filtering and spectrum slicing

Equation (14) indicates that the contour integration technique does not provide a natural route for obtaining the individual occupied wave functions but rather the summation of their amplitudes square. The FEAST algorithm was originally proposed to reconcile both wave function and Green’s function formalism and provide an efficient and scalable new approach for solving the eigenvalue problem [58]. FEAST can be applied for solving both standard and generalized form of the Hermitian or non-Hermitian problem, and it belongs to the family of contour integration eigensolvers along with the Sakurai and Sugiura (SS) method [63, 64]. In contrast to the Krylov-based SS method, FEAST is a subspace iteration method that uses the Rayleigh-Ritz projection and an approximate spectral projector as a filter [57]. Given a Hermitian generalized eigenvalue problem  $AX = BX\Lambda$  of size  $n$ , the algorithm in Figure 10 outlines the main steps of a generic Rayleigh-Ritz subspace iteration procedure for computing  $m$  eigenpairs. At convergence, the algorithm yields the  $B$ -orthonormal eigensubspace  $Y_m \equiv X_m = \{x_1, x_2, \dots, x_m\}_{n \times m}$  and associated

0. Start: Select random subspace  $Y_{m_0} \equiv \{y_1, y_2, \dots, y_{m_0}\}_{n \times m_0}$  ( $n \gg m_0 \geq m$ )
1. Repeat until convergence
2. Compute  $Q_{m_0} = \rho(B^{-1}A)Y_{m_0}$
3. Orthogonalize  $Q_{m_0}$
4. Compute  $A_Q = Q_{m_0}^H A Q_{m_0}$  and  $B_Q = Q_{m_0}^H B Q_{m_0}$
5. Solve  $A_Q W = B_Q W \Lambda_Q$  with  $W^H B_Q W = I_{m_0 \times m_0}$
6. Compute  $Y_{m_0} = Q_{m_0} W$
7. Check convergence of  $Y_{m_0}$  and  $\Lambda_{Q_{m_0}}$  for the  $m$  wanted eigenvalues
8. End

Figure 10: Subspace iteration method with Rayleigh-Ritz projection

eigenvalues  $\Lambda_{Q_m} \equiv \Lambda_m$ . Taking  $\rho(B^{-1}A) = B^{-1}A$ , yields the bare-bone subspace iteration (generalization of the power method) which converges towards the  $m$  dominant eigenvectors with the linear rate  $|\lambda_{m_0+1}/\lambda_i|_{i=1, \dots, m}$  [56, 61]. This standard approach is never used in practice. Instead, it is combined with filtering using the function  $\rho$  which aims at improving the convergence rate (i.e.  $|\rho(\lambda_{m_0+1})/\rho(\lambda_i)|_{i=1, \dots, m}$ ) by increasing the gap between wanted and unwanted eigenvalues. An ideal filter for the interior eigenvalue problem which maps all  $m$  wanted eigenvalues to one and all unwanted ones to zero, can be derived from the Cauchy (or Dunford) integral formula:

$$\rho(\lambda) = \frac{1}{2\pi i} \oint_{\mathcal{C}} dz (z - \lambda)^{-1}, \quad (15)$$

where the wanted eigenvalues are located inside a complex contour  $\mathcal{C}$ . The filter then becomes a spectral projector, with  $\rho(B^{-1}A) = X_m X_m^H B$ , for the eigenvector subspace  $X_m$  (i.e.  $\rho(B^{-1}A)X_m = X_m$ ) and can be written as:

$$\rho(B^{-1}A) = \frac{1}{2\pi i} \oint_{\mathcal{C}} dz (zB - A)^{-1} B. \quad (16)$$

FEAST uses a numerical quadrature to approximately compute the action of this filter onto a set of  $m_0$  vectors along the subspace iterations. The resulting rational function  $\rho_a$  that approximates the filter (15) is given by:

$$\rho_a(z) = \sum_{j=1}^{n_e} \frac{\omega_j}{z_j - z}, \quad (17)$$

where  $\{z_j, \omega_j\}_{1 \leq j \leq n_e}$  are the nodes and related weights of the quadrature. We obtain for the subspace  $Q_{m_0}$  in step 2 of the algorithm in Figure 10:

$$Q_{m_0} = \rho_a(B^{-1}A)Y_{m_0} = \sum_{j=1}^{n_e} \omega_j (z_j B - A)^{-1} B Y_{m_0} \equiv X \rho_a(\Lambda) X^H B Y_{m_0}. \quad (18)$$

In practice,  $Q_{m_0}$  can be computed by solving a small number of (independent) shifted linear systems over a complex contour.

$$Q_{m_0} = \sum_{j=1}^{n_e} \omega_j Q_{m_0}^{(j)}, \quad \text{with } Q_{m_0}^{(j)} \text{ solution of } (z_j B - A) Q_{m_0}^{(j)} = B Y_{m_0} \quad (19)$$

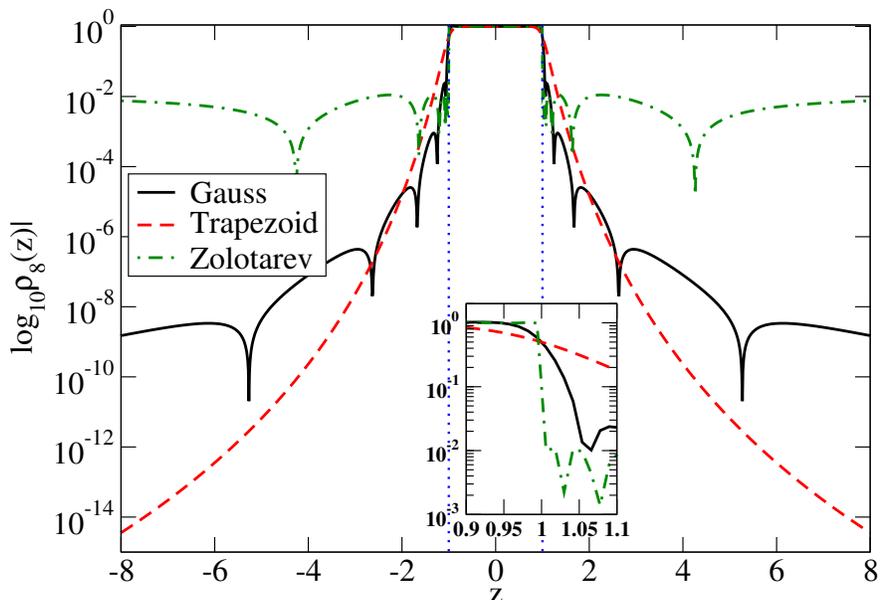


Figure 11: Variations of the rational functions  $\rho_8(\lambda)$  ( $n_e = 8$  contour points) associated with Gauss, Trapezoidal and Zolotarev quadrature rules. While Trapezoidal presents a more regular decay than Gauss, the latter produces smaller values for the rational function just outside the edges of the search interval  $|z| > 1$ . From the caption, we note that Zolotarev presents a dramatic drop in the rational function at  $z = 1$  (i.e., fastest possible decay), but this value quickly saturates.

As shown in Figure 11, a relatively small number of quadrature nodes (using Gauss, Trapezoidal or Zolotarev [26] rules) on a circular contour suffices to produce a rapid decay of the function  $\rho_a$  from  $\approx 1$  within the search contour to  $\approx 0$  outside. In comparison with more standard polynomial filtering [69, 61], the rational filter (17) can lead to a very fast convergence of the subspace iteration procedure. In addition, all the  $m$  desired eigenvalues are expected to converge at the same rate (since  $\rho_a(\lambda_i) \simeq 1$  if  $\lambda_i$  is located within the search interval). The convergence rate of FEAST does not only depend upon the decay properties of the rational function  $\rho_a$ , but also on the size of the search subspace  $m_0$  which must not be chosen smaller than the number of eigenvalues inside the search contour (i.e.  $m_0 \geq m$ ). Users of the FEAST eigensolver[2] are then responsible for specifying an interval to search for the eigenvalues and a subspace size  $m_0$  that overestimate the number of the wanted eigenvalues. Once these conditions are satisfied, FEAST offers the following set of appealing features:

- (i) high robustness with well-defined convergence rate  $|\rho_a(\lambda_{m_0+1})/\rho_a(\lambda_i)|_{i=1,\dots,m}$ ;
- (ii) all multiplicities naturally captured;
- (iii) no explicit orthogonalization procedure on long vectors required in practice (i.e., step-3 in Figure 10 is unnecessary as long as  $B_Q$  is positive definite). We note in (18) that  $Q_{m_0}$  is naturally spanned by the eigenvector subspace;
- (iv) reusable subspace capable to generate suitable initial guess when solving a series of eigenvalue problems such the ones that appear in DFT-SCF iterations;
- (v) can exploit natural parallelism at three different levels: search intervals can be treated sep-

arately (no overlap) while maintaining orthogonality - linear systems can be solved independently across the quadrature nodes of the complex contour - each complex linear system with  $m_0$  multiple right-hand-sides can be solved in parallel. Consequently, within a parallel environment, the algorithm complexity depends on solving a single linear system using a direct or an iterative method.

Using FEAST, the total number of processes  $n_{pp}$  can be distributed over three levels of parallelism: (i) eigenvalue level parallelism using  $i$  filters (i.e.  $i$  slices); (ii) block level parallelism where all the  $k$  linear systems are solved independently; (iii) domain level parallelism which handles the system matrices and the multiple right-hand-sides using the remaining  $p$  processes available since  $n_{pp} = i \times k \times p$ . Achieving a good balancing with a suitable distribution of the parallel resources among all slices, would require that the number of eigenvalues in each slice be about the same. Obviously, it can be quite challenging for a user to perform a customized slicing by first guessing the distribution of the eigenvalue spectrum. Recent work on stochastic estimates can be helpful in this regard [46, 21]. One possible estimate on the eigenvalue count in an interval consists of approximating the trace of the spectral projector by exploiting the rational function expansion (19) i.e.

$$\text{tr}(\tilde{P}) \approx \frac{n}{n_v} \sum_{j=1}^k \omega_j \sum_{i=1}^{n_v} v_i^T (\sigma_j B - A)^{-1} B v_i. \quad (20)$$

The cost of this estimation can remain relatively small since the linear systems can be solved with low accuracy and with a very small number of right-hand-sides  $n_v$ . Furthermore, if the factorizations can already be computed at each complex shift  $\sigma_j$ , they can be reused in the subsequent subspace iteration.

## 5 Conclusion

Atom-by-atom large-scale first-principle calculations have become critical for supplementing the experimental investigations and obtaining detailed electronic structure properties and reliable characterization of emerging nanostructures. First-principle calculations most often rely on a succession of modeling trade-offs between accuracy and performances, which can be broadly divided into four major steps: (i) physical, (ii) mathematical, (iii) discretization, and (iv) computing. These modeling steps contain different layers of numerical approximations which are most often tightly tied together. In order to improve on current software implementation by fully capitalizing on modern HPC computing platforms, it is essential to revisit not one, but all the various stages of the electronic structure modeling process which have been summarized in this chapter.

Solutions of the DFT/Kohn-Sham problem are routinely used in the calculations of many ground state properties of small molecular systems or crystal unit-cells containing a handful of atoms. In order to characterize large-scale complex systems and nanostructures of current technological interest, the SCF-DFT procedure would require repeated computations of many tens of thousands of eigenvectors, for eigenvalue systems that can have sizes in the tens of millions. In this case, a divide-and-conquer approach that can compute wanted eigenpairs by parts, becomes mandatory, since windows or slices of the spectrum can be computed independently of one another and orthogonalization between eigenvectors in different slices is no longer necessary. All these is-

sues have originally motivated the development of the EVSL and FEAST approaches that were discussed here.

## References

- [1] *Evsl*. <http://www.cs.umn.edu/~saad/software/EVSL>.
- [2] *Feast*. <http://www.feast-solver.org>.
- [3] *Mika*. <http://www.csc.fi/physics/mika>.
- [4] *Nessie*. <http://www.nessie-code.org>.
- [5] *Octopus*. <http://www.tddft.org/programs/octopus/>.
- [6] *Parsec*. <http://www.ices.utexas.edu/parsec/index.html>.
- [7] *Pexsi*. [https://math.berkeley.edu/~linlin/pexsi/download/doc\\_v0.6.0/](https://math.berkeley.edu/~linlin/pexsi/download/doc_v0.6.0/).
- [8] X. ANDRADE, J. ALBERDI-RODRIGUEZ, D. A. STRUBBE, M. J. T. OLIVEIRA, F. NOGUEIRA, A. CASTRO, J. MUGUERZA, A. ARRUABARRENA, S. G. LOUIE, A. ASPURU-GUZIK, A. RUBIO, AND M. A. L. MARQUES, *Time-dependent density-functional theory in massively parallel computer architectures: the octopus project*, Journal of Physics: Condensed Matter, 24 (2012), p. 233202.
- [9] X. ANDRADE, D. A. STRUBBE, U. D. GIOVANNINI, A. H. LARSEN, M. J. T. OLIVEIRA, J. ALBERDI-RODRIGUEZ, A. VARAS, I. THEOPHILOU, N. HELBIG, M. VERSTRAETE, L. STELLA, F. NOGUEIRA, A. ASPURU-GUZIK, A. CASTRO, M. A. L. MARQUES, AND A. RUBIO, *Real-space grids and the octopus code as tools for the development of new simulation approaches for electronic systems*, Physical Chemistry Chemical Physics, 17 (2015), pp. 31371–31396.
- [10] S. BARONI AND P. GIANNOZZI, *Towards very large-scale electronic-structure calculations*, Europhysics Letters (EPL), 17 (1992), pp. 547–552.
- [11] T. L. BECK, *Real-space mesh techniques in density functional theory*, Rev. Mod. Phys., 74 (2000), pp. 1041–1080.
- [12] E. L. BRIGGS, D. J. SULLIVAN, AND J. BERNHOLC, *Large-scale electronic-structure calculations with multigrid acceleration*, Phys. Rev. B, 52 (1995), pp. R5471–R5474.
- [13] A. BRILEY, M. R. PEDERSON, K. A. JACKSON, D. C. PATTON, AND D. V. POREZAG, *Vibrational frequencies and intensities of small molecules: All-electron, pseudopotential, and mixed-potential methodologies*, Phys. Rev. B, 58 (1997), pp. 1786–1793.
- [14] K. BURKE, *Perspective on density functional theory.*, The Journal of chemical physics, 136 15 (2012), p. 150901.

- [15] R. CAR AND M. PARRINELLO, *Unified approach for molecular dynamics and density functional theory*, Phys. Rev. Lett., 55 (1985), pp. 2471–2474.
- [16] J. R. CHELIKOWSKY AND M. L. COHEN, *Pseudopotentials for semiconductors*, in Handbook of Semiconductors, T. S. Moss and P. T. Landsberg, eds., Elsevier, Amsterdam, 2nd edition, 1992.
- [17] J. R. CHELIKOWSKY AND S. G. LOUIE, *First-principles linear combination of atomic orbitals method for the cohesive and structural properties of solids: Application to diamond*, Phys. Rev. B, 29 (1984), pp. 3470–3481.
- [18] J. R. CHELIKOWSKY, N. TROULLIER, X. JING, D. DEAN, N. BIGGELI, K. WU, AND Y. SAAD, *Algorithms for the structural properties of clusters*, Comp. Phys. Comm., 85 (1995), pp. 325–335.
- [19] J. R. CHELIKOWSKY, N. TROULLIER, AND Y. SAAD, *The finite-difference-pseudopotential method: Electronic structure calculations without a basis*, Phys. Rev. Lett., 72 (1994), pp. 1240–1243.
- [20] J. R. CHELIKOWSKY, N. TROULLIER, K. WU, AND Y. SAAD, *Higher order finite difference pseudopotential method: An application to diatomic molecules*, Phys. Rev. B, 50 (1994), pp. 11355–11364.
- [21] E. DI NAPOLI, E. POLIZZI, AND Y. SAAD, *Efficient estimation of eigenvalue counts in an interval*, Numerical Linear Algebra with Applications, 23 (2016), pp. 674–692.
- [22] J.-L. FATTEBERT AND J. BERNHOLC, *Towards grid-based  $O(N)$  density-functional theory methods: Optimized nonorthogonal orbitals and multigrid acceleration*, Phys. Rev. B, 62 (2000), pp. 1713–1722.
- [23] ———, *Towards grid-based  $O(N)$  density-functional theory methods: Optimized nonorthogonal orbitals and multigrid acceleration*, Phys. Rev. B, 62 (2000), pp. 1713–1722.
- [24] C. Y. FONG, *Topics in Computational Materials Science*, World Scientific, 1998.
- [25] B. FORNBERG AND D. M. SLOAN, *A review of pseudospectral methods for solving partial differential equations*, Acta Numer., 94 (1994), pp. 203–268.
- [26] S. GÜTTEL, E. POLIZZI, P. TANG, AND G. VIAUD, *Zolotarev quadrature rules and load balancing for the feast eigensolver*, SIAM Journal on Scientific Computing, 37 (2015), pp. A2100–A2122.
- [27] F. GYGI AND G. GALLI, *Real-space adaptive-coordinate electronic-structure calculations*, Phys. Rev. B, 52 (1995), pp. R2229–R2232.
- [28] Y. HASEGAWA, J. IWATA, M. TSUJI, D. TAKAHASHI, A. OSHIYAMA, K. MINAMI, T. BOKU, F. SHOJI, A. UNO, M. KUROKAWA, H. INOUE, I. MIYOSHI, AND M. YOKOKAWA, *First-principles calculations of electron states of a silicon nanowire with 100,000 atoms on the k computer*, in Proceedings of 2011 International Conference for High

Performance Computing, Networking, Storage and Analysis, SC '11, New York, NY, USA, 2011, ACM, pp. 1:1–1:11.

- [29] A. HAUG, *Theoretical Solid State Physics*, Pergamon Press, 1972.
- [30] M. HEIKANEN, T. TORSTI, M. J. PUSKA, AND R. M. NIEMINEN, *Multigrid method for electronic structure calculations*, Phys. Rev. B, 63 (2001), pp. 245106–245113.
- [31] P. HOHENBERG AND W. KOHN, *Inhomogeneous electron gas*, Phys. Rev., 136 (1964), pp. B864–B871.
- [32] K. A. JACKSON, M. R. PEDERSON, D. V. POREZAG, Z. HAJNAL, AND T. FRAUNHEIM, *Density-functional-based predictions of Raman and IR spectra for small Si clusters*, Phys. Rev. B, 55 (1997), pp. 2549–2555.
- [33] R. W. JANSEN AND O. F. SANKEY, *Ab initio linear combination of pseudo-atomic-orbital scheme for the electronic properties of semiconductors: Results for ten materials*, Phys. Rev. B, 36 (1987), pp. 6520–6531.
- [34] J. KESTYN, V. KALANTZIS, E. POLIZZI, AND Y. SAAD, *Pfeast: A high performance sparse eigenvalue solver using distributed-memory linear solvers*, in Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC '16, Piscataway, NJ, USA, 2016, IEEE Press, pp. 16:1–16:12.
- [35] Y. H. KIM, I. H. LEE, AND R. M. MARTIN, *Object-oriented construction of a multigrid electronic structure code with Fortran*, Comp. Phys. Comm., 131 (2000), pp. 10–25.
- [36] W. KOHN AND L. J. SHAM, *Self-consistent equations including exchange and correlation effects*, Phys. Rev., 140 (1965), pp. A1133–A1138.
- [37] L. KRONIK, A. MAKMAL, M. L. TIAGO, M. M. G. ALEMANY, M. JAIN, X. HUANG, Y. SAAD, AND J. R. CHELIKOWSKY, *PARSEC the pseudopotential algorithm for real-space electronic structure calculations: recent advances and novel applications to nano-structure*, Phys. Stat. Sol. (B), 243 (2006), p. 1063–1079.
- [38] C. LANCZOS, *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*, J. Research Nat. Bur. Standards, 45 (1950), pp. 255–282.
- [39] R. M. LARSEN, *PROPACK: A software package for the symmetric eigenvalue problem and singular value problems on Lanczos and Lanczos bidiagonalization with partial reorthogonalization, SCCM, Stanford University*  
URL: <http://sun.stanford.edu/~rmunk/PROPACK/>.
- [40] —, *Efficient Algorithms for Helioseismic Inversion*, PhD thesis, Dept. Computer Science, University of Aarhus, DK-8000 Aarhus C, Denmark, October 1998.
- [41] I. H. LEE, Y. H. KIM, AND R. M. MARTIN, *One-way multigrid method in electronic-structure calculations*, Phys. Rev. B, 61 (2000), p. 4397.

- [42] 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, 131 (2009), p. 054103.
- [43] A. R. LEVIN, D. ZHANG, AND E. POLIZZI, *Feast fundamental framework for electronic structure calculations: Reformulation and solution of the muffin-tin problem*, Computer Physics Communications, 183 (2012), pp. 2370 – 2375.
- [44] R. LI, Y. XI, L. ERLANDSON, AND Y. SAAD, *The EigenValues Slicing Library (EVSL): Algorithms, implementation, and software*, Tech. Rep. ys-2018-02, Dept. Computer Science and Engineering, University of Minnesota, Minneapolis, MN, 2018. Submitted. ArXiv: <https://arxiv.org/abs/1802.05215>.
- [45] R. LI, Y. XI, E. VECHARYNSKI, C. YANG, AND Y. SAAD, *A Thick-Restart Lanczos algorithm with polynomial filtering for Hermitian eigenvalue problems*, SIAM Journal on Scientific Computing, 38 (2016), pp. A2512–A2534.
- [46] L. LIN, Y. SAAD, AND C. YANG, *Approximating spectral densities of large matrices*, SIAM Review, 58 (2016), pp. 34–65.
- [47] L. LIN, C. YANG, J. C. MEZA, J. LU, L. YING, AND E. WEINAN, *Selinv - an algorithm for selected inversion of a sparse symmetric matrix*, ACM Trans. Math. Softw., 37 (2011), pp. 40:1–40:19.
- [48] S. LUNDQVIST AND N. H. MARCH, eds., *Theory of the Inhomogeneous Electron Gas*, Plenum, 1983.
- [49] R. MARTIN, R. MARTIN, AND C. U. PRESS, *Electronic Structure: Basic Theory and Practical Methods*, Cambridge University Press, 2004.
- [50] J. L. MARTINS AND M. COHEN, *Diagonalization of large matrices in pseudopotential band-structure calculations: Dual-space formalism*, Phys. Rev. B, 37 (1988), pp. 6134–6138.
- [51] J. L. MARTINS, N. TROULLIER, AND S.-H. WEI, *Pseudopotential plane-wave calculations for ZnS*, Phys. Rev. B, 43 (1991), pp. 2213–2217.
- [52] R. B. MORGAN AND D. S. SCOTT, *Generalizations of Davidson’s method for computing eigenvalues of sparse symmetric matrices*, SIAM J. Sci. Comput., 7 (1986), pp. 817–825.
- [53] T. ONO AND K. HIROSE, *Timesaving double-grid method for real-space electronic structure calculations*, Phys. Rev. Lett., 82 (1999), pp. 5016–5019.
- [54] —, *Timesaving double-grid method for real-space electronic structure calculations*, Phys. Rev. Lett., 82 (1999), pp. 5016–5019.
- [55] C. C. PAIGE, *The Computation of Eigenvalues and Eigenvectors of Very Large Sparse Matrices*, PhD thesis, University of London, 1971.
- [56] B. N. PARLETT, *The Symmetric Eigenvalue Problem*, no. 20 in Classics in Applied Mathematics, SIAM, Philadelphia, 1998.

- [57] P. PETER TANG AND E. POLIZZI, *Feast as a subspace iteration eigensolver accelerated by approximate spectral projection*, SIAM Journal on Matrix Analysis and Applications, 35 (2014), pp. 354–390.
- [58] E. POLIZZI, *A density matrix-based algorithm for solving eigenvalue problems*, phys. rev. B, 79 (2009).
- [59] E. POLIZZI AND S. DATTA, *Multidimensional nanoscale device modeling: the finite element method applied to the non-equilibrium green's function formalism*, in 2003 Third IEEE Conference on Nanotechnology, 2003. IEEE-NANO 2003., vol. 1, Aug 2003, pp. 40–43 vol.2.
- [60] H. RUTISHAUSER, *Simultaneous iteration for symmetric matrices*, in Handbook for automatic computations (linear algebra), J. Wilkinson and C. Reinsch, eds., New York, 1971, Springer Verlag, pp. 202–211.
- [61] Y. SAAD, *Numerical Methods for Large Eigenvalue Problems*, John Wiley, New York, 1992.
- [62] Y. SAAD, A. STATHOPOULOS, J. R. CHELIKOWSKY, K. WU, AND S. OGUT, *Solution of large eigenvalue problems in electronic structure calculations*, BIT, 36 (1996), pp. 563–578.
- [63] T. SAKURAI AND H. SUGIURA, *A projection method for generalized eigenvalue problems using numerical integration*, Journal of Computational and Applied Mathematics, 159 (2003), pp. 119 – 128. Japan-China Joint Seminar on Numerical Mathematics; In Search for the Frontier of Computational and Applied Mathematics toward the 21st Century.
- [64] T. SAKURAI AND H. TADANO, *Cirr: a rayleigh-ritz type method with contour integral for generalized eigenvalue problems*, Hokkaido Mathematical Journal, 36 (2007), p. 745757.
- [65] L. J. SHAM, *Theoretical and computational development some efforts beyond the local density approximation*, International Journal of Quantum Chemistry, Vol. 56, 4 , pp 345-350, (2004).
- [66] H. D. SIMON, *Analysis of the symmetric Lanczos algorithm with reorthogonalization methods*, Linear Algebra Appl., 61 (1984), pp. 101–132.
- [67] ———, *The Lanczos algorithm with partial reorthogonalization*, Math. Comp., 42 (1984), pp. 115–142.
- [68] J. M. SOLER, E. ARTACHO, J. D. GALE, A. GARCIA, J. JUNQUERA, P. ORDEJÓN, AND D. SÁNCHEZ-PORTAL, *The SIESTA method for ab-initio order-N materials simulation*, J. Phys.: Condens. Matter, 14 (2002), pp. 2745–2779.
- [69] D. C. SORENSEN, *Implicit application of polynomial filters in a k-step Arnoldi method*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 357–385.
- [70] E. M. STOUDENMIRE, L. O. WAGNER, S. R. WHITE, AND K. BURKE, *One-dimensional continuum electronic structure with the density matrix renormalization group and its implications for density functional theory*, Phys. Rev. Lett. 109, I5, 056402, (2012).

- [71] J. TAYLOR, H. GUO, AND J. WANG, *Ab initio modeling of quantum transport properties of molecular electronic devices*, Phys. Rev. B, 63 (2001), p. 245407.
- [72] T. TORSTI, M. HEISKANEN, M. PUSKA, AND R. NIEMINEN, *MIKA: A multigrid-based program package for electronic structure calculations*, Int. J. Quantum Chem., 91 (2003), pp. 171–176.
- [73] L.-W. WANG AND A. ZUNGER, *Electronic structure pseudopotential calculations of large ( 1000 atoms) si quantum dots*, J. Phys. Chem., 98 (1994), pp. 2158–2165.
- [74] S. R. WHITE, J. W. WILKINS, AND M. P. TETER, *Finite-element method for electronic structure*, Phys. Rev. B, 39 (1989), pp. 5819–5833.
- [75] K. WU AND H. SIMON, *A parallel Lanczos method for symmetric generalized eigenvalue problems*, Tech. Rep. 41284, Lawrence Berkeley National Laboratory, 1997. Available on line at <http://www.nerisc.gov/research/SIMON/planso.html>.
- [76] R. ZELLER, J. DEUTZ, AND P. DEDERICHS, *Application of complex energy integration to selfconsistent electronic structure calculations*, Solid State Communications, 44 (1982), pp. 993 – 997.
- [77] D. ZHANG AND E. POLIZZI, *Linear scaling techniques for first-principle calculations of large nanowire devices*, 2008 NSTI Nanotechnology Conference and Trade Show. Technical Proceedings, Vol. 1 pp12-15, (2008).
- [78] Y. ZHOU, Y. SAAD, M. L. TIAGO, AND J. R. CHELIKOWSKY, *Parallel self-consistent-field calculations via Chebyshev-filtered subspace acceleration*, Phy. rev. E, 74 (2006), p. 066704.
- [79] J. M. ZIMAN, *Electrons and Phonons*, Oxford University Press, 1960.
- [80] G. ZUMBACH, N. A. MODINE, AND E. KAXIRAS, *Adaptive coordinate, real-space electronic structure calculations on parallel computers*, Solid State Commun., 99 (1996), pp. 57–61.