A Spectrum Slicing Method for the Kohn-Sham Problem

Grady Schofield[a], James R. Chelikowsky[a,b], Yousef Saad[c]

[a] *Center for Computational Materials, Institute for Computational Engineering and Sciences, University of Texas, Austin, TX 78712*
[b] *Departments of Physics and Chemical Engineering, University of Texas, Austin, TX 78712*
[c] *Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN 55455*

Abstract

Solving the Kohn-Sham equation, which arises in density functional theory, is a standard procedure to determine the electronic structure of atoms, molecules, and condensed matter systems. The solution of this nonlinear eigenproblem is used to predict the spatial and energetic distribution of electronic states. However, obtaining a solution for large systems is computationally intensive because the problem scales super-linearly with the number of atoms. Here we demonstrate a divide and conquer method that partitions the necessary eigenvalue spectrum into slices and computes each partial spectrum on an independent group of processors in parallel. We focus on the elements of the spectrum slicing method that are essential to its correctness and robustness such as the choice of filter polynomial, the stopping criterion for a vector iteration, and the detection of duplicate eigenpairs computed in adjacent spectral slices. Some of the more prominent aspects of developing an optimized implementation are discussed.

Keywords: Spectrum slicing, Hermitian eigenproblem, Kohn-Sham equation, sparse parallel eigensolver, polynomial filtering

1. Introduction

Treating the electronic structure problem in density functional theory (DFT) requires calculating a large number of eigenvalues and eigenvectors (eigenpairs) for the Kohn-Sham equation [10]. The number of eigenpairs corresponds to one for each pair of valence electrons in the system of interest (unless an explicit spin description is required). Here we consider only the valence states [4] and do not consider magnetic systems, *i.e.*, each eigenpair is interpreted as representing two distinct spin states, although the extension to magnetic states is straightforward. The Kohn-Sham equation is written in atomic units ($\hbar = e = m = 1$) as follows:

$$\begin{aligned} \left(-\frac{1}{2}\nabla^2 + V_{ion} + V_H[\rho] + V_{xc}[\rho]\right)\psi_i &= \epsilon_i \psi_i \\ \rho &= \sum_{i=1}^{N_{occ}} f_i |\psi_i|^2 \end{aligned} \tag{1}$$

where V_{ion} is the external potential, often taken to be an ion core pseudopotential [4], V_H is the Hartree potential or electrostatic potential, V_{xc} is the effective exchange-correlation potential, ρ is the charge density of the valence electrons, and $\{(\psi_i, \epsilon_i)\}_{i=1}^{N_{occ}}$ are the eigenpairs for the occupied states. f_i is given by a Fermi-Dirac distribution, usually replaced by the Heaviside step function, which is the limit of zero temperature, *i.e.*, f_i is unity for an occupied state and zero for an empty state such that the system is neutral. Since ρ is computed from the set of eigenpairs but also appears in the left hand side of (1), obtaining the correct

Email addresses: grady@ices.utexas.edu (Grady Schofield), jrc@ices.utexas.edu (James R. Chelikowsky), saad@cs.umn.edu (Yousef Saad)

electronic structure requires an iterative procedure that finds a fixed point of (1) in terms of ρ. At the fixed point, the potentials computed from ρ, V_H and V_{xc}, generate the same ρ through (1). So the solution creates a "self-consistent field" [10].

As the number of electrons (or atoms) grows, the computational complexity of the Kohn-Sham problem scales super-linearly. In particular, enforcing an orthogonality condition on the approximation subspace for the eigenvectors results in cubic scaling. Since the prefactor associated with cubically scaling operations is small compared to that of quadratic operations, for a sequence of increasingly larger systems all of moderate size, the runtime exhibits predominantly quadratic growth. We note that some methods exist that attempt linear scaling solutions of the Kohn-Sham equation [7, 6], but these methods are usually quite specialized. Here we develop a method that partitions the occupied states into slices. The eigenproblem on each slice is computed independently of the others. Our implementation of this method parallelizes the eigensolver with respect to these independent slices and provides a scalable approach to the Kohn-Sham problem.

2. Filter Operators on a Spectral Slice

Let H be the discretized Hamiltonian for the operator on the left hand side of (1). Because H is Hermitian, its eigenvectors, $\{u_i\}_{i=1}^n$, form an orthonormal basis for $\mathbb{R}^n$. For $v \in \mathbb{R}^n$, we have

$$Hv = H\sum_{i=1}^{n} c_i u_i = \sum_{i=1}^{n} \lambda_i c_i u_i.$$

Since $\{\lambda_i\}_{i=1}^n \in \mathbb{R}$, we may pass from a polynomial $\tilde{p} : \mathbb{C}^{n\times n} \to \mathbb{C}^{n\times n}$ to the corresponding polynomial, $p : \mathbb{R} \to \mathbb{R}$, as follows

$$\tilde{p}(H)v = \tilde{p}(H)\sum_{i=1}^{n} c_i u_i = \sum_{i=1}^{n} c_i \tilde{p}(H)u_i = \sum_{i=1}^{n} c_i p(\lambda_i)u_i.$$

A judicious choice of $p(x)$ leads to an operator, $\tilde{p}(H)$, that enhances eigencomponents from only a segment of the spectrum. Therefore, the eigenpairs in that segment will lie in the span of a subspace generated by repeated applications of this operator.

Our algorithm must compute several eigenpairs associated with the smallest eigenvalues of (1). In practice, the largest necessary eigenvalue, $\lambda_{N_{occ}}$, is not known, and inevitably some number of unnecessary eigenpairs with $\lambda_i > \lambda_{N_{occ}}$ will be computed. To simplify the discussion, we will refer to $\lambda_{N_{occ}}$ as if it were known beforehand.

A polynomial resembling the step function

$$\phi(x) = \begin{cases} 1 & : \quad x \le \lambda_{N_{occ}} \\ 0 & : \quad x > \lambda_{N_{occ}} \end{cases}$$

would yield an ideal operator for the task. Unfortunately the discontinuity at $\lambda_{N_{occ}}$ causes any approximation of ϕ to exhibit rapid oscillations around $\lambda_{N_{occ}}$. The desired property of $p(x)$ is to decay as rapidly as possible beyond $\lambda_{N_{occ}}$. Even a sharp cut off function with arbitrarily high regularity will result in approximations that exhibit rapid oscillations around $\lambda_{N_{occ}}$. For this reason, we take the simplest approach of approximating ϕ and smoothing the result.

To this end, we employ the Chebyshev-Jackson approximation (2) [8, 13].

$$f(x) \approx \sum_{i=0}^{k} \gamma_i g_i^k T_i(x) \tag{2}$$

$$\gamma_i = \begin{cases} \dfrac{1}{\pi}\displaystyle\int_{-1}^{1} \frac{1}{\sqrt{1-x^2}} f(x)dx & : \quad i = 0 \\ \dfrac{2}{\pi}\displaystyle\int_{-1}^{1} \frac{1}{\sqrt{1-x^2}} f(x)T_i(x)dx & : \quad i > 0 \end{cases} \tag{3}$$

This modification of the usual Chebyshev approximation attenuates higher order terms in the sum. Letting $\alpha_k = \frac{\pi}{k+2}$, we have for g_i^k from [9]:

$$g_i^k = \frac{\left(1 - \frac{i}{k+2}\right)\sin(\alpha_k)\cos(i\alpha_k) + \frac{1}{k+2}\cos(\alpha_k)\sin(i\alpha_k)}{\sin(\alpha_k)}. \tag{4}$$

For a given term in (2), g_i^k depends only on the degree of approximation. The location of the filter within

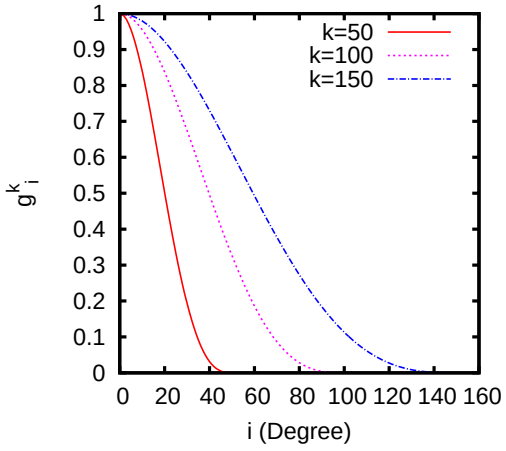


Figure 1: Attenuation coefficient g_i^k for k=50,100,150.

the spectrum and the exact filter function, ϕ, are not taken into account. g_i^k is plotted for a few values of k in Figure 1.

For the specific case where $f(x)$ is a step function on $[a, b]$ we have the following equations for γ_i

$$\gamma_i = \begin{cases} \frac{1}{\pi}\left(\arccos(a) - \arccos(b)\right) & : \quad i = 0 \\ \frac{2}{\pi}\left(\frac{\sin(i\arccos(a)) - \sin(i\arccos(b))}{i}\right) & : \quad i > 0 \end{cases}. \tag{5}$$

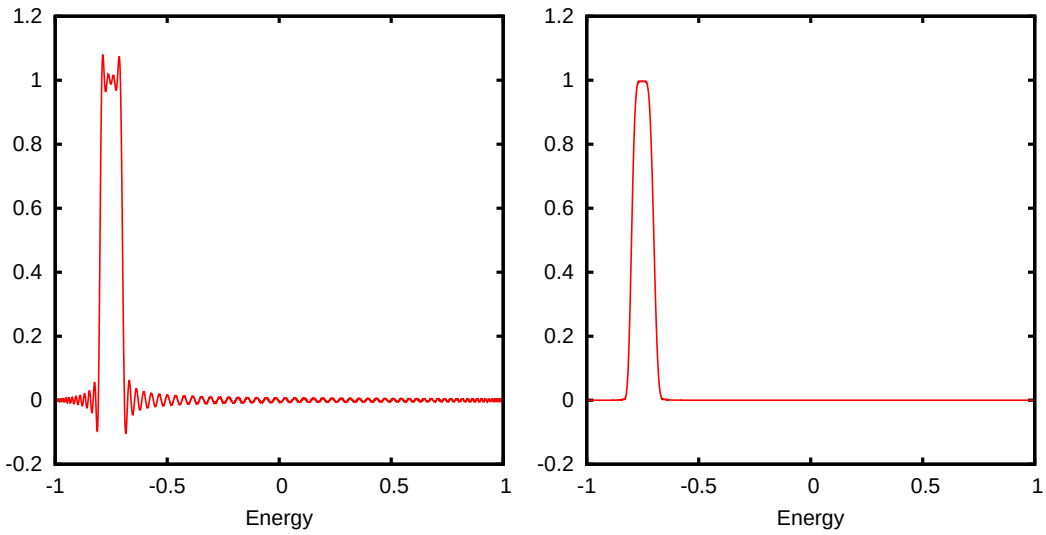
Calculating the coefficients for the approximation of a given degree is straightforward and $O(n)$ in the degree of approximation. To see the effect this attenuation factor has on the usual Chebyshev approximation, see Figures 2 and 3.

The degree of approximation needed to obtain a sufficiently sharp $p(x)$ increases with the number of slices packed into a fixed span of the spectrum. As more slices are needed in a calculation, the overlap between adjacent filter polynomials must decrease. The extent to which $p(x)$ faithfully represents the step function is controlled by a transition width, w, and a magnification factor, m.

For an interval, $[a, b]$, in the spectrum we require

$$\frac{p(a)}{p(a-w)} > m \text{ and } \frac{p(b)}{p(b+w)} > m. \tag{6}$$

This ensures that the eigencomponents in $[a, b]$ are magnified by a factor of m over the components outside of $[a - w, b + w]$. This criteria, while appearing less precise than an expression involving a norm, reflects the functional impact $p(x)$ will have when used in the eigensolver. Let N_f be the number of filters, which is identical to the number of slices. When $N_f > 1$, we would like to set the transition width to keep the



(a) Chebyshev approximation

(b) Chebyshev-Jackson approximation

Figure 2: Approximation of a step function equal to 1 on $(-0.8, -0.7)$. Both polynomials have the same degree.

overlap between adjacent filters proportional to the width of an individual interval. To achieve this, given a width, w, that works well for a single filter, a transition width w/N_f should be employed in the case of multiple filters. Therefore the transition width should be made inversely proportional to N_f. As shown in Figure 4, the degree of a filter that satisfies (6) is inversely proportional to the transition width, w/N_f. These two inverse relationships cause the degree of a filter to scale linearly in N_f.

As the number of slices increases, the average number of times the filter must be applied to form the slice subspace decreases linearly. This is because the average number is proportional to the total number of states in each interval. In a parallel implementation, where the eigenpairs for each slice are calculated independently, this would nominally hold constant the wall clock time spent in the filtering operation while the total computer time would increase. Increased filtering cost is incurred in exchange for a decrease in the cost of super-linear scaling procedures used in the eigensolver, making this method most advantageous as the number of atoms increases.

A related issue in employing (2) comes from a filter's location. The distribution of the Chebyshev polynomials' roots is denser at the endpoints of the spectrum, giving better accuracy in that region for lower order approximations. So obtaining a uniform degree of sharpness for all filters in a given calculation requires higher degree polynomials in the interior of the spectrum. Figure 5 shows the degree of a filter with a fixed width and sharpness criterion as the filter slides across the range $[-1, 1]$. The difference in filter order leads to a load balancing problem. To add a new vector to an approximation subspace, processors for interior slices spend more time filtering than those for exterior ones. There are two main approaches to this problem.

On one hand, a code could setup the slices heterogeneously, varying the width of each slice based on some model of runtime that uses a guess of where the new eigenvalues lie. Such a guess is easy to produce because the spectrum from the previous SCF iteration, shifted and scaled into the new Hamiltonian's bounds, is usually good enough. However, developing a model for runtime is more difficult. The number of vectors needed in the subspace is affected by nearby spectral gaps as well as the presence of nearly degenerate eigenvalues. The multiplicity of a degeneracy is also a factor, with higher multiplicity requiring more subspace vectors per converged eigenpair than lower multiplicity. Furthermore, reducing the width of an interior slice too much necessitates reducing the transition width of the associated filter as well as either of the adjacent filters if the density of eigenvalues at the shared boundary is high enough. Issues like this can

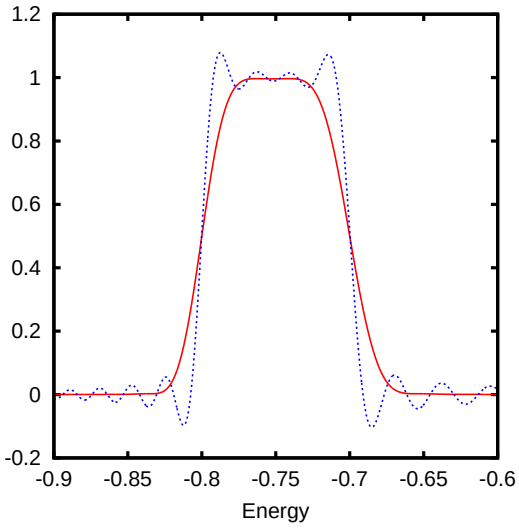


Figure 3: Close up of Chebyshev approximation (dashed) and smoothed Chebyshev-Jackson polynomial (solid) for a step function nonzero on $(-0.8, -0.7)$ over the interval $[-1, 1]$.

make a model dependent on a number of heuristics.

Another possibility is to take a simple approach to laying out the slices, perhaps giving them equal width, and address load balancing dynamically. One approach that we have implemented is for the left neighbor of a slice (being more exterior, it typically finishes first) to send the converged eigenpairs in the transition region to its right neighbor. At the end of each loop of Algorithm 2, the processors for a slice check to see if there is a message from the left neighbor. If so, then the processors incorporate these converged vectors into their approximation subspace by orthogonalization. This technique generally decreases the runtime of a calculation by 20%. Other dynamic load balancing techniques can be applied, and this is a route for further research of spectrum slicing methods.

3. Creation of the Approximation Subspace for a Spectral Slice

Given an operator $f(H) : \mathbb{R}^n \to \mathbb{R}^n$ that enhances eigencomponents from only an interval, $[a, b]$, of the total spectrum, we can easily develop a vector iteration that yields an approximation subspace for the eigenpairs in that interval. Algorithm 2 is such an iteration. Starting with a normalized initial vector v_0, we construct a subspace, V, from repeated application of $f(H)$, $\widetilde{v}_{i+1} = f(H)v_i$. Since $\widetilde{v}_{i+1}$ is to be used to augment the subspace, we are only interested in its component orthogonal to V. Let v_{i+1} denote that component so that $\widetilde{v}_{i+1} = v_{i+1} + v'_{i+1}$ where $v_{i+1} \perp V$ is the vector with which to augment V and $v'_{i+1} \in \text{span}(V)$ is of no interest. The vector v_{i+1} will be mapped under $f(H)$ in the next iteration.

With a procedure for building a subspace, we need a criterion to terminate the iteration. One option is to perform the Rayleigh-Ritz method, shown in Algorithm 1, on V and examine the approximate eigenpairs (Ritz pairs) and their residuals. However, the cost of periodically computing the approximate eigenpairs is high. The approximations always need to be computed using Algorithm 1 once convergence is reached, but using them to detect convergence can be even more costly because Algorithm 1 must be repeated when convergence is not detected.

While costly, such a test can be reliable as long as the eigenpairs of interest are at either end point of the spectrum. However, for an interior slice, the Rayleigh-Ritz procedure produces vectors whose Ritz value is a weighted average of eigenvalues on the left and right side of the slice. This averaging will routinely put a few of the Ritz values in $[a, b]$, though the associated residuals would be large. One is then faced with developing some criterion for throwing out Ritz pairs that appear to be in the slice. That task is made

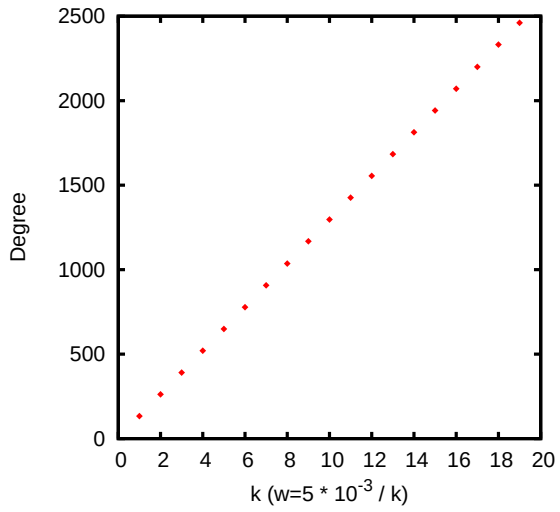


Figure 4: The linear relationship between the degree of a filter and decreasing transition width, w. Here, $m = 2$ and $w = 5 \times 10^{-3}/k$.

Algorithm 1: The Rayleigh-Ritz method

Input: Orthonormal basis, $V \in \mathbb{C}^{N \times m}$, and the Hamiltonian matrix, $H \in \mathbb{C}^{N \times N}$
Output: Approximate eigenpairs (λ_i, ψ_i) and their Ritz residuals, σ_i

$R = V^{\mathrm{H}} H V$; `// form Rayleigh quotient matrix,` $O(Nm^2)$
$Q^{\mathrm{H}} \Lambda Q = R$; `// diagonalize` R, $O(m^3)$
$\Psi = VQ$; `// form Ritz vectors,` $O(Nm^2)$
for $i = 1$ **to** m **do** `// compute each residual,` $O(Nm)$
  $\sigma_i = \|H\psi_i - \lambda_i \psi_i\|_2$
end

more difficult since nearly degenerate subspaces are typically resolved last among all eigenpairs in the slice; and until resolution happens, the Rayleigh-Ritz procedure reports the degenerate subspace as one or two Ritz vectors with large residuals. This makes examination of the subspace's Ritz data alone insufficient for developing a stopping criterion.

There is an alternative which requires computing only the Ritz value of v_{i+1}, the component of the most recently produced vector which is orthogonal to the current approximation subspace. When convergence is obtained, the eigenpairs in $[a, b]$ are contained in span(V). Thus, v_{i+1} has no components from that interval in the spectrum, *i.e.*, in the eigenvector expansion

$$v_{i+1} = \sum_{k=1}^{N} c_k \psi_k$$

and $c_k = 0$ for all k with $\lambda_k \in [a, b]$. Since v_{i+1} is normalized, the Ritz value

$$\mu_{i+1} = v_{i+1}^{\mathrm{H}} H v_{i+1} = \sum_{k=1,N} c_k^2 \lambda_k \tag{7}$$

is a convex combination of the eigenvalues outside of $[a, b]$. If the average of the eigenvalues in (7) is heavily weighted to the positive side, the Ritz value will leave this interval once convergence is obtained.

In the electronic structure problem, we are always interested in the smallest eigenvalues, those of lowest energy, because they correspond to the occupied states. The number of occupied states is far smaller than the

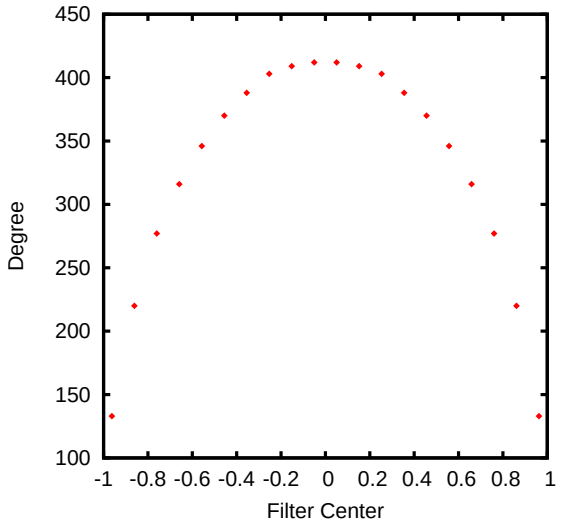


Figure 5: The location of the filter effects the degree of approximation necessary to achieve a given sharpness criterion. Here the filter spans a width of 0.025 on $[-1, 1]$ with a transition width of 0.005 and magnification factor equal to 2. Note that in the electronic structure problem, the solution involves only a small portion, for instance 2%, of the left end of this interval.

total number of eigenpairs in the spectrum of H. A typical example is the system used in the computational experiment to follow. The occupied states fill only 2% of the spectral radius, and they constitute only 0.3% of the total number of eigenvectors. Moreover, the calculation is done with a grid spacing close to the maximum that can be used on this system, 0.7 au. For a given problem, grid spacing determines the spectral radius and the total number of eigenvectors, both increasing with decreasing grid spacing. In a system with a transition metal, typically a smaller grid spacing would be required; and the principle on which the convergence criteria is based would still apply. Therefore, in the Kohn-Sham problem, (7) leaves $[a, b]$ quickly after convergence is reached, making it a useful measure for stopping the iteration.

The Chebyshev-Jackson polynomials allow this technique to be effective. Without smooth filters, convergence in the wanted eigenspace is followed by convergence in the peaks and troughs of the step function approximation's ripple, see Figure 3. Convergence in these unnecessary areas makes the Ritz value of v_{i+1} stagnate for many more iterations.

In Algorithm 2, the stopping criteria is given by the input *max_outside*. The iteration stops when the Ritz value is outside of $[a, b]$ for *max_outside* iterations. In the computational example we set this value to 10. While specifying a stopping criteria in this way is heuristic, the more precise application of Sylvester's theorem must be ruled out as it requires a complete LDLT factorization of a shifted Hamiltonian [12].

Figure 6 displays this Ritz value, computed in Line 2 of Algorithm 2, at each iteration. Note that the Ritz value moves outside of the upper boundary of the interval in latter iterations. This indicates convergence of eigenpairs in the slice. In the calculation, the number of eigenpairs with residuals below 10^{-10} is 63 and the total number of iterates is 128. So this criteria stops the iteration and requires roughly 2 filter operations per converged vector. Of the 63 converged vectors, only 41 reside within the slice. To some extent, excess is unavoidable; but its cost can be recouped somewhat by the optimization described at the end of Section 2.

Aside from the stopping criterion, computing the orthogonal component in line 1 of Algorithm 2 can be problematic. As this iteration nears convergence, the eigenpairs in $[a, b]$ will be contained in span(V) and the result of the operator $f(H)$ will be nearly linearly dependent with V. Modified Gram-Schmidt and QR factorization will both fail to produce an orthogonal component in this situation. A Gram-Schmidt algorithm with reorthogonalization must be used here [5].

Dividing the spectrum into slices requires an estimate of the Fermi energy, or better, a knowledge of the highest occupied eigenstate. In our implementation, we use the Fermi level from the previous self-consistency

Algorithm 2: Slice subspace creation method

Input: max_outside, the stopping criterion; H, the Hamiltonian matrix; f, the filter polynomial over $[a, b]$

Output: At least all eigenpairs of H lying in $[a, b]$

$v_0 = \text{Random}(N)$;
$count = 0$;
$V = \emptyset$;
while $count < max\_outside$ **do**
  $\tilde{v}_{i+1} = f(H)v_i$;
1 Compute the normalized component $v_{i+1} \perp V$ of $\tilde{v}_{i+1}$;
  $V = V \cup \{v_{i+1}\}$;
2 **if** $v_{i+1}^H H v_{i+1} > b$ **then** $count = count + 1$;
  else $count = 0$;
end
Apply Algorithm 1 to V

iteration as an estimate, and place the last filter at some distance beyond that point to ensure a few extra states are included. This means that we cannot use the filtering algorithm for the first self-consistency step unless the user can provide a reliable guess at the Fermi level. Hence, for the first self-consistency iteration, we compute the spectrum using Algorithm 2, but with only one slice and with a low degree filter that gradually slopes upward such as that in Figure 9. This requires a much less accurate estimate of the Fermi level and can be modified adaptively [16]. It is also necessary to have a guess of the minimum and maximum eigenvalues of the spectrum, but this is easily obtained with a few iterations of the Lanczos algorithm [11].

Algorithm 2 computes the required states lying in $[a, b]$, but it also inevitably calculates converged states immediately outside of the boundary of $[a, b]$. In enumerating the final set of states from which the density and forces are calculated, eigenvalues that lie very close to either end point of an interval may be counted twice.

4. Identifying Duplicate Eigenpairs

Eigenpairs near interval boundaries present a problem where, in enumerating the complete set of occupied states, these eigenpairs may be counted twice. When an algorithm uses the Ritz value

$$\mu_i = \psi_i^{\mathrm{H}} H \psi_i$$

and residual

$$\sigma_i = \|H\psi_i - \mu_i \psi_i\|_2$$

to determine which Ritz pairs lie in an interval, there must be some slack in the condition owing to the fact that the approximated eigenpair may lie anywhere in the interval $(\mu_i - \sigma_i, \mu_i + \sigma_i)$. Therefore a Ritz pair with μ_i slightly outside of the interval may be approximating an eigenpair inside of the interval. This is the essence of the problem with duplicate eigenpairs. Algorithm 2 run on $[a, b]$ and $[b, c]$ may count eigenpairs near b twice. In Figure 7 eigenpairs in the shaded region are computed for both slices. Most of the eigenpairs may be eliminated by a simple test using the Ritz value and residual, but eigenpairs close enough to the boundary at -4.11 eV require a more robust test.

To avoid double counting, the principal angles between the subspaces of Ritz vectors near an interval boundary may be used. If Q_L and Q_R are the Ritz vectors near the boundary for the left and right slice then the singular value decomposition of $Q_L^{\mathrm{H}} Q_R = U \Sigma V^{\mathrm{H}}$ yields the cosine of the principal angles through

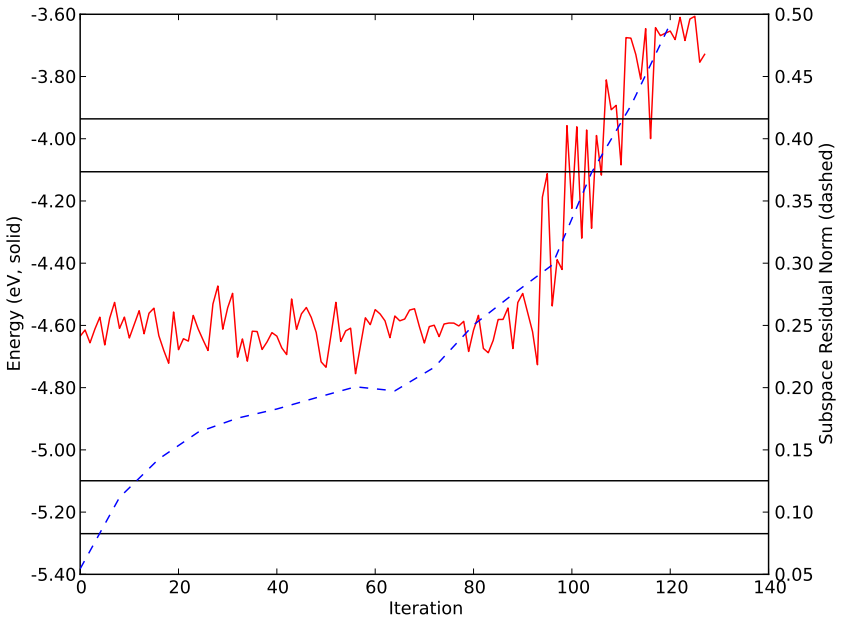


Figure 6: This data is for one of the slices from the calculation described in Section 5. The dashed line shows the residual, $\|HW - \Lambda W\|_F$, of the subspace produced by applying the Rayleigh-Ritz procedure to the approximation subspace, V, in Algorithm 2. The solid line shows the Ritz value, computed on Line 2 of Algorithm 2. The inner horizontal lines are the upper and lower bounds of the spectral slice, $[-5.06, -4.11]$. The outer horizontal lines, bounding $[-5.23, -3.89]$, are formed by expanding the slice bounds by the filter's transition width, 0.17. Algorithm 2 waits for the Ritz value to exceed the upper line at -3.89 for at least 10 iterations. We use a block size of 8 in this calculation. Hence, the limit of 10 can be exceeded by at most 7.

the singular values in Σ, [2],

$$\Sigma = \begin{bmatrix} \cos(\theta_1) & 0 & \cdots & 0 \\ 0 & \cos(\theta_2) & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \cos(\theta_n) \end{bmatrix}.$$

Assume the singular values in Σ are ordered and suppose k of the singular values are near 1. By

$$Q_L^{\mathrm{H}} Q_R = U \Sigma V^{\mathrm{H}} \Rightarrow (Q_L U)^{\mathrm{H}} (Q_R V) = \Sigma \approx \left[\begin{array}{c|c} I^{k\times k} & 0 \\ \hline 0 & 0 \end{array}\right]$$

we see that the first k columns of $Q_L U$ are a basis for the subspace of duplicate vectors in the left slice. By examining the first k columns of U for some of their largest elements, we can find the specific columns of Q_L that are duplicates. Let N_L be the number of vectors in Q_L. Define the vector $m \in \mathbb{R}^{N_L}$ so that

$$m_i = \max_{1 \le j \le k} |U_{i,j}|.$$

The largest k elements of m have indices corresponding to the duplicated vectors in Q_L. We cannot simply examine the columns of U for their maximum elements. In the event that the duplicate eigenvectors form a nearly degenerate subspace, such a test can find the maximum across two separate columns to be in the same row. For example, if Q_L contained five vectors, and three of them formed a degenerate subspace, then

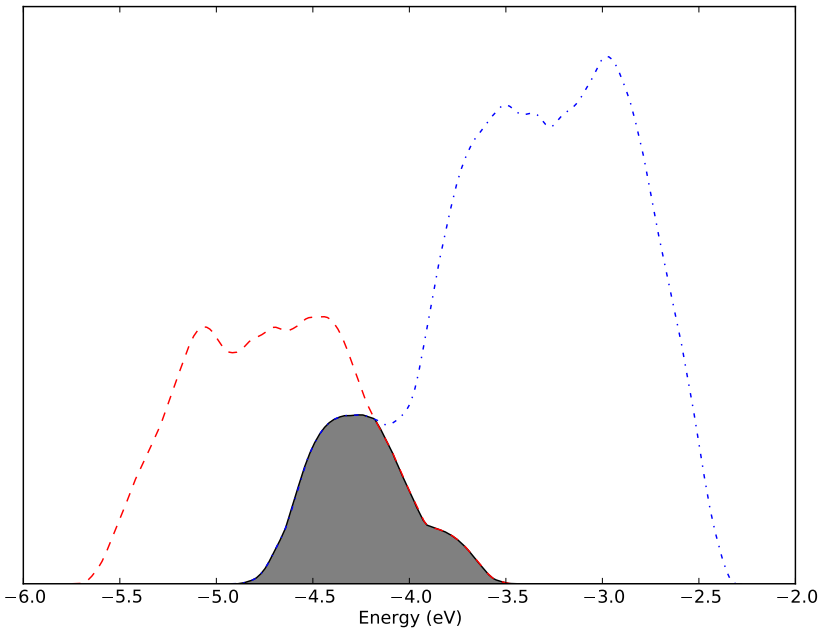


Figure 7: The density of states for two out of sixteen slices in the calculation of Section 5. Eigenpairs in the shaded region are computed for both slices using Algorithm 2. To avoid counting duplicate eigenpairs near the interval boundary, Algorithm 3 must be used.

a possibility for U, shown in (8), would indicate that the first, second and fourth vectors of Q_L are a basis for the degenerate space. Examining only the maximum absolute value of the elements, the maximum is found for column one and two in the first row, and for column three in the fourth row. Hence row two would be overlooked. Note that in practice, one can not simply examine the rows in the first k columns for nonzeros, as would work in this example. The matrix in (8) is an ideal case, and precision issues can add slight perturbations making any of the zero elements nonzero. Therefore a method like the one described above is necessary for identifying the relevant rows.

$$U = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 & 0 \\ \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ \frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{6}} & -\sqrt{\frac{2}{3}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{8}$$

With the indices for basis vectors of the duplicate subspace in Q_L, the code can remove duplicates in the left slice without explicitly forming any state vectors.

For a slice at $[a, b]$, discarding all converged eigenpairs for which $(\mu_i - \sigma_i, \mu_i + \sigma_i) \cap [a, b] = \emptyset$ and applying Algorithm 3 will result in the union across all slices of all remaining eigenpairs giving an accurate count of the occupied states.

One situation that will cause a problem for this algorithm is if Algorithm 2 does not find a complete basis for a degenerate subspace with an eigenvalue near the boundary. In the simplest case, Q_L could consist of two vectors that span a degenerate space, but Q_R could have only one basis vector. In this situation, the code would have only one singular value to work with, and would only identify one duplicate. Indeed, it would identify one of the vectors in Q_L as a duplicate and remove it. Without further reorthogonalization at some later stage, the basis for this degenerate space would not be orthogonal. We do not handle this in the code. We assume that Algorithm 2 will always find a complete basis for any degenerate or nearly degenerate

Algorithm 3: Remove duplicates in slices before final state enumeration

Input: Converged eigenpairs for two adjacent slices at $[a, b]$ and $[b, c]$; tolerance $\delta < 1$ with $\delta \approx 1$
Output: Indices for the set of duplicate eigenvectors to remove from the slice at $[a, b]$

if *slice over* $[a, b]$ **then**
  for $i = 1$ **to** N_L **do** `// loop over converged states`
    if $(\mu_i - \sigma_i, \mu_i + \sigma_i) \cap [b, c]$ **then** $Q_L = Q_L \cup \{\psi_i\}$
  end
end
if *slice over* $[b, c]$ **then**
  for $i = 1$ **to** N_R **do** `// loop over converged states`
    if $(\mu_i - \sigma_i, \mu_i + \sigma_i) \cap [a, b]$ **then** $Q_R = Q_R \cup \{\psi_i\}$
  end
end
$U\Sigma V^{\mathrm{H}} = Q_L^{\mathrm{H}} Q_R$; `// assume` $\Sigma_{i,i} \geq \Sigma_{i+1,i+1}$ `so Line 1 is` max `over contiguous elements`
$k = 0$;
if $\Sigma_{i,i} > \delta$ **then**
  $k = k + 1$;
end
for $i = 1$ **to** N_L **do**
1 $m_i = \max_{1 \leq j \leq k} |U_{i,j}|$;
end
Sort the sequence $\{\{m_i, i\}_{i=1}^{N_L}\}$ descending by the first coordinate;
The second coordinate of the first k elements of the sorted sequence gives the column index of a duplicate in Q_L;

space near the boundary. In practice, the Algorithm 2 will always find more states than necessary on either side of the boundary.

5. Computational Result

We have computed the occupied states of a passivated nanocrystal, $Si_{275}H_{172}$, using the spectrum slicing algorithm. The geometry of the nanocrystal is fixed by assuming a bulk crystal fragment. Unsaturated bonds are passivated by hydrogen-like atoms. This is a modestly sized nanostructure and is typical of a contemporary electronic structure problem [3]. 636 eigenpairs or occupied states are required for a solution. This calculation was done in the framework of the real-space code PARSEC [4]. A histogram of the states is shown in Figure 8. The result computed using the spectrum slicing solver with 16 slices agrees *exactly* with the standard PARSEC solver. For this problem, the number of slices used here is unnecessarily high. Our calculations for this system are meant as a "viability" test or a proof-in-principle for the algorithm as opposed to a fully optimized example of the algorithm.

The order of filter polynomials ranges from 290 for the leftmost filter to 1090 for the innermost filter. The order of these polynomials is much higher than that of those used in the standard PARSEC solver where the degree is usually less than 25. For any configuration, the cost of filtering in a spectrum slicing algorithm is higher than in a standard method. However, this algorithm reduces the quadratically and cubically scaling costs for other operations that dominate the runtime in larger systems. To demonstrate how the spectrum slicing method becomes advantageous for large subspaces, we compare the runtimes for a calculation using Algorithm 2 employing a single low-degree filter to those of Algorithm 2 run with a single higher degree window filter. This calculation is done for a fixed system where the number of wanted eigenpairs exceeds the Fermi level. This is an important setting for time-dependent density functional theory, where a large number of unoccupied Kohn-Sham orbitals are used to calculate excited state properties [14].

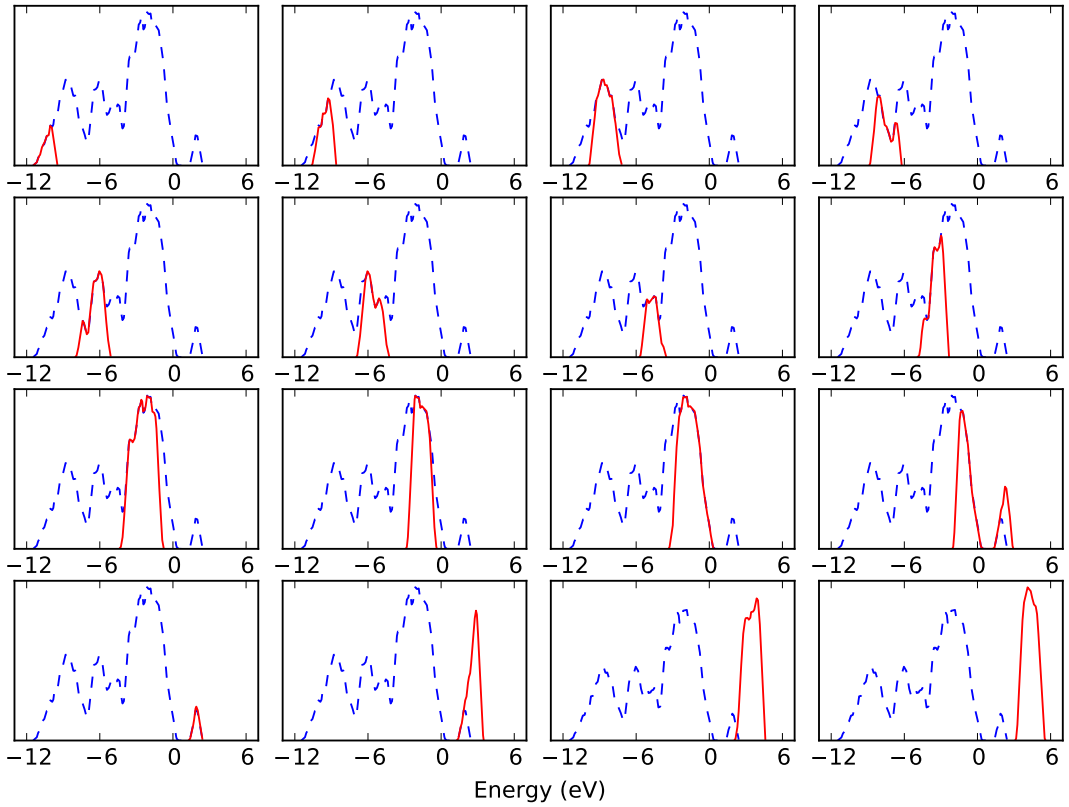


Figure 8: States for unrelaxed $Si_{275}H_{172}$ as computed by Algorithm 2 using 16 slices. The states computed for a slice are shown by the solid line. The dashed line is the union of all wanted states. Algorithm 2 inevitably produces converged eigenpairs for some states outside of the interval boundaries, hence the overlap between states of adjacent slices.

For the calculation done with low-degree filters, the entire set of occupied states is computed at once, similar to the algorithm used in [1, 15]. The filters used for the comparison are shown in Figure 9. By fitting the run times to a model of the computational complexity we can estimate at what number of computed unoccupied states the use of multiple filters becomes advantageous.

A model for the operation count of the loop in Algorithm 2 is

$$T(N_{occ}) = \alpha N_{occ}^2 + \beta D_f N_{occ} + \gamma, \tag{9}$$

where α, β, and γ are to be determined by data. The first term in equation 9 comes from the orthogonalization cost, and the second term comes from filtering cost. D_f is the degree of a filter. Because the degree increases as filters are added, we need it to be factored out of β.

In a model for systems where the number of atoms can increase, the first two terms would contain an additional $O(N_{occ})$ factor to account for the length of a vector. This factor is $O(N_{occ})$ because adding states corresponds to adding some species of atom and hence some amount of volume and its associated grid points. Hence orthogonalization is cubic in the number of occupied states when the number of atoms increases. Therefore, in this situation, the model would be

$$T(N_{occ}) = \alpha N_{occ}^3 + \beta D_f N_{occ}^2 + \gamma.$$

In the model, (9), we account only for the time spent on filtering and orthogonalization, the body of the loop in Algorithm 2. We do not consider time spent executing Algorithm 1, the Rayleigh-Ritz

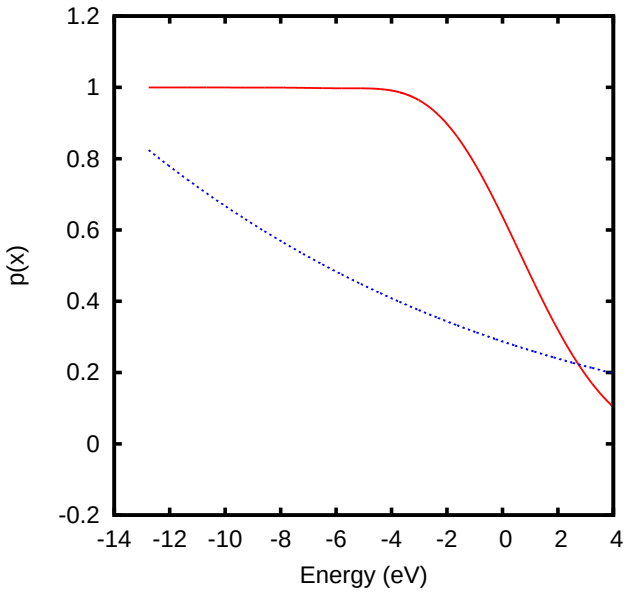


Figure 9: A low degree (22) filter (dashed) enhancing the occupied states in $[-12.8, 0]$ and a filter of degree 118 (solid) approximating a step function in the same interval. The polynomial extends to the end of the discrete Hamiltonian's spectrum at 655 eV.

procedure. This omission ultimately makes the estimate pessimistic for the spectrum slicing method because the Rayleigh-Ritz procedure has steps that scale super-linearly in the number of states. Consequently the spectrum slicing method reduces the cost of the Rayleigh-Ritz procedure. This model will not reflect that benefit, but the omission of the Rayleigh-Ritz cost improves clarity by allowing us to treat only the costs that are incurred on a per iteration basis.

In this model, adding N_f filters corresponds to decreasing N_{occ} by a factor of $1/N_f$ and increasing the degree of a filter polynomial by a factor of N_f. This increase is necessary for the filter to become sharper as discussed in section 2. Therefore with multiple filters we have

$$\begin{aligned} T(N_{occ}) =& \alpha \left(\frac{N_{occ}}{N_f} \right)^2 + \beta D_f N_f \frac{N_{occ}}{N_f} + \gamma \\ =& \alpha \left(\frac{N_{occ}}{N_f} \right)^2 + \beta D_f N_{occ} + \gamma. \end{aligned} \tag{10}$$

N_f is introduced in the numerator of the second term by the increased degree of the filter polynomial. However it is also appears in the denominator by a reduced number of filter applications caused by the reduced subspace size for the smaller slice.

The run times for each calculation are displayed in Figure 10. Fitting the data, which has been computed up to 636 states, and extrapolating we arrive at the plots for a single slice and a single low degree filter in Figure 11. The break-even point occurs under $3,000$ states. This is for only a modest sized nanostructure. The largest nanostructure studied by this group has been for a nanocrystal of over 10,000 atoms, requiring 20,000 eigenpairs. While many factors affect the exact crossover point such as the relative efficiency of the codes for matrix-vector operations and dense super-linear scaling procedures, this estimate is derived from real data and shows that spectrum slicing techniques should be applicable for even moderately large systems accessible on current computing hardware.

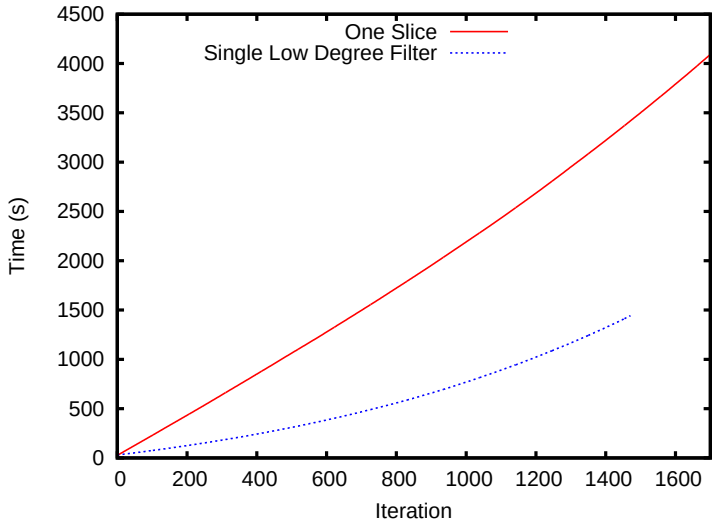


Figure 10: Run times for the first SCF iteration of $Si_{275}H_{172}$ using the filters displayed in Figure 9

6. Conclusion

In this paper we demonstrated a method by which the Kohn-Sham eigenproblem can be solved in a parallel divide and conquer fashion. The cost of orthogonalization and the Rayleigh-Ritz procedure, which scale cubically in the number of states, is reduced by the reduction in the size of the eigenspace for each spectral interval. For systems with tens of the thousands of states, an approach of this nature is the only viable course given the computational complexity of the problem. The independent nature of the calculations done on each spectral interval allow an additional layer of parallelization to be implemented where the calculations for each slice are done on independent groups of processors. For a typical system, the silicon nanocrystal, we estimate the spectrum slicing method to be preferred over the fast Chebyshev-Davidson [16] method currently used in PARSEC once the number of necessary states exceeds 3000.

Acknowledgments

We wish to acknowledge support from the National Science Foundation under grants No. DMR-0941645 and OCI-1047997 and from The Welch Foundation under grant No. F-1708. This research used resources at the National Energy Research Scientific Computing Center (NERSC) and the Oak Ridge Leadership Computing Facility (OLCF), located in the National Center for Computational Sciences at Oak Ridge National Laboratory. Both NERSC and OLCF are supported by the Office of Science of the U.S. Department of Energy under Contracts No. DE-AC02-05CH11231 and DE-AC05-00OR22725 respectively. The National Science Foundation provided computational resources through TeraGrid at the Texas Advanced Computing Center (TACC) under grant No. TG-DMR090026. The third author acknowledges support from the Minnesota Supercomputing Institute.

References

[1] C. Bekas, E. Kokiopoulou, and Y. Saad. Polynomial filtered Lanczos iterations with applications in density functional theory. *SIAM Journal on Matrix Analysis and Applications*, 30(1):397–418, 2008.

[2] Ake Bjorck and Gene H. Golub. Numerical methods for computing angles between linear subspaces. *Mathematics of Computation*, 27(123):579–594, 1973.

[3] J. R. Chelikowsky, T.-L. Chan, M. M. G. Alemany, and G. Dalpian. *Rep. Prog. Phys.*, 74:046501, 2011.

[4] J.R. Chelikowsky. The pseudopotential-density functional method applied to nanostructures. *Journal of Physics D: Applied Physics*, 33(8):R33–R50, 2000.

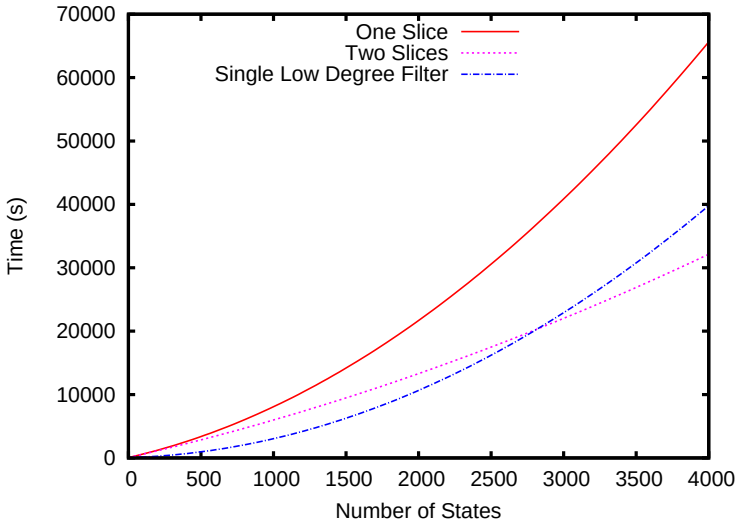


Figure 11: Extrapolation of the run times in Figure 10 where the average number of iterations per converged state has been taken into account and the model from (10) has been used to arrive at an estimate for the case of two slices. The performance of the spectrum slicing algorithm surpasses the low degree filtering method before 3000 states are reached.

[5] J. W. Daniel, W. B. Gragg, L. Kaufman, and G. W. Stewart. Reorthogonalization and stable algorithms for updating the Gram-Schmidt QR factorization. *Mathematics of Computation*, 30:772–795, 1976.

[6] C. Fonseca Guerra, J. G. Snijders, G. te Velde, and E. J. Baerends. *Theor Chem Acc*, 99:391, 1998.

[7] E. Hernandez and M. J. Gillan. *Phys. Rev. B*, 51:10157, 1995.

[8] Dunham Jackson. *The Theory of Approximation*. American Mathematical Society, New York, 1930.

[9] Laurent O. Jay, Hanchul Kim, Yousef Saad, and James R. Chelikowsky. Electronic structure calculations in plane-wave codes without diagonalization. *Computer Physics Communications*, 118:21–30, 1999.

[10] W. Kohn and L. J. Sham. Self-consistent equations including exchange and correlation effects. *Phys. Rev.*, 140:A1133, 1965.

[11] C. Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *Journal of Research of the National Bureau of Standards*, 45:255–282, 1950.

[12] B.N. Parlett. *The Symmetric Eigenvalue Problem*. Prentice Hall, Englewood Cliffs, 1980.

[13] Theodore J. Rivlin. *An Introduction to the Approximation of Functions*. Dover, New York, 1981.

[14] Dario Rocca, Ralph Gebauer, Yousef Saad, and Stefano Baroni. Turbo charging time-dependent density-functional theory with Lanczos chains. *J. Chem. Phys.*, 128:154105, 2008.

[15] Y. Zhou, Y. Saad, M.L. Tiago, and J.R. Chelikowsky. Self-consistent-field calculations using Chebyshev-filtered subspace iteration. *Journal of Computational Physics*, 219:172–184, 2006.

[16] Yunkai Zhou and Yousef Saad. A Chebyshev-Davidson algorithm for large symmetric eigenproblems. *SIAM Journal on Matrix Analysis and Applications*, 29(3):954–971, 2007.