Lawrence Berkeley National Laboratory
LBL Publications

Title
Thick-Restart Lanczos Method for Electronic Structure Calculations

Permalink
https://escholarship.org/uc/item/20r866kv

Authors
Wu, K
Canning, A
Simon, H D
et al.

Publication Date
1999-03-01

Copyright Information
This work is made available under the terms of a Creative Commons Attribution License, available at https://creativecommons.org/licenses/by/4.0/

eScholarship.org Powered by the California Digital Library
University of California

LBNL-42917

ERNEST ORLANDO LAWRENCE
BERKELEY NATIONAL LABORATORY

Thick-Restart Lanczos Method for Electronic Structure Calculations

K. Wu, A. Canning, H.D. Simon, and L.-W. Wang

Computing Sciences Directorate
National Energy Research Scientific Computing Division

March 1999

REFERENCE COPY
Does Not
Circulate

Bldg. 50 Library - Ref.
Lawrence Berkeley National Laboratory

Copy 1

LBNL-42917

DISCLAIMER

This document was prepared as an account of work sponsored by the United States Government. While this document is believed to contain correct information, neither the United States Government nor any agency thereof, nor the Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by its trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or the Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof or the Regents of the University of California.

LBNL-42917

Thick-Restart Lanczos Method for Electronic Structure Calculations

K. Wu, A. Canning, H.D. Simon, and L.-W. Wang

Computing Sciences Directorate
National Energy Research Scientific Computing Division
Ernest Orlando Lawrence Berkeley National Laboratory
University of California
Berkeley, California 94720

March 1999

This work was supported by the Director, Office of Science, Office of Laboratory Policy and Infrastructure Management, of the U.S. Department of Energy under Contract No. DE-AC03-76SF00098, and by the Office of Basic Energy Sciences, Division of Material Sciences, under Contract No. DE-AC36-83CH10093.

Thick-restart Lanczos method for electronic structure calculations

K. Wu A. Canning H. D. Simon
NERSC, Lawrence Berkeley National Laboratory, Berkeley, California 94720
L.-W. Wang
National Renewable Energy Laboratory, Golden, Colorado 80401

March 3, 1999

Abstract

This paper describes two recent innovations related to the classic Lanczos method for eigenvalue problems, namely the thick-restart technique and dynamic restarting schemes. Combining these two new techniques we are able to implement an efficient eigenvalue problem solver. This paper will demonstrate its effectiveness on one particular class of problems for which this method is well suited: linear eigenvalue problems generated from non-selfconsistent electronic structure calculations.

1 Introduction

The Lanczos method is a very simple and yet effective algorithm for finding extreme eigenvalues of large matrices. Since it only needs to access the matrix through matrix-vector multiplications, the user has the flexibility of choosing the most appropriate matrix-vector multiplication scheme to reduce computer memory usage and the computation time. There is never any need to explicitly store the full matrix which can be prohibitively large in many electronic structure calculations. There are two common ways of implementing the Lanczos method depending on whether or not the Lanczos vectors are stored. When the Lanczos vectors are not stored, they have to be recomputed when needed for re-orthogonalization or computing eigenvectors. This scheme is usually used without re-orthogonalization and only to compute eigenvalues. Since there is no re-orthogonalization, the Lanczos vectors will lose orthogonality after a number of steps and the Lanczos method may generate spurious solutions [3, 21]. Though spurious eigenvalues can be effectively identified, however, less Lanczos steps would be needed if the orthogonality is maintained. If the eigenvectors are also wanted, the Lanczos iterations are repeated after the eigenvalues are found. This is a significant amount of additional work. For the applications under consideration, both eigenvalues and eigenvectors are needed, therefore it is more appropriate to store the Lanczos vectors. When the Lanczos vectors are stored, the loss of orthogonality problem can be corrected by re-orthogonalization [11, 12, 16] and no spurious eigenvalues are generated. Because each Lanczos step generates one vector, a large amount of computer memory may be required to store all the Lanczos vectors. If the re-orthogonalization is necessary, the time needed to carry out a Lanczos step increases as more Lanczos vectors are generated. For these reasons, the Lanczos algorithm that stores the Lanczos vectors is usually restarted after a certain number of steps.

The restarted versions often use considerably more matrix-vector multiplications than the non-restarted version to compute the same eigenvalues. In recent years, newly developed restarting strategies have significantly reduced the number of matrix-vector multiplications used for other

restarted eigenvalue methods. The two most successful ones are the implicitly restarted Arnoldi method [9, 17] and the dynamic thick-restart Davidson method [18, 27]. Compared to the Arnoldi method and the Davidson method, the Lanczos method uses less arithmetic operations per step. Therefore we would like to apply these restarting strategies on the Lanczos method. The implicitly restarted Lanczos method has been studied elsewhere [2] and implemented in ARPACK [9]. Here we describe a thick-restart Lanczos method. Because the thick-restart procedure is only a slight modification of the Rayleigh-Ritz procedure, it is easier to implement than the implicitly restarted Lanczos method. More importantly because we have conducted detailed analysis of exactly how many Ritz pairs to save during restarting, our implementation of the thick-restart Lanczos method is considerably more effective than ARPACK on most of the eigenvalue problems tested [30].

Many electronic structure calculations result in a non-linear eigenvalue problem where the lowest eigenvectors, corresponding to the electronic states of the physical system, are required. This problem is normally solved by iterating a linearized form of the non-linear problem, to self-consistency. In these cases it is advantageous to extrapolate from previous steps to produce a good starting guess for the eigenvectors of the next step in the self-consistent iteration. For this reason iterative eigensolvers that can take advantage of a good starting guess such as the Davidson method [4] and the Conjugate Gradient (CG) method [19], are the most commonly used. Since the simple Lanczos method cannot take an arbitrary number of starting vectors, it is more appropriate for linear eigenvalue problems. The test problems chosen in this paper are calculations of quantum dot structures with empirical pseudopotentials [22, 31] resulting in linear eigenvalue problems.

The goal of this paper is to introduce two new innovations on the Lanczos method to the reader and show the effectiveness of the improved method through a number of examples. We will compare the new variations of the Lanczos method against the older variations and demonstrate that the new methods scale well as the number of required eigenvalues increases and as the matrix size increases. We will also discuss how the Lanczos method computes the eigenvectors associated with a degenerate eigenvalue and how to choose appropriate parameters in order to achieve the correct multiplicity.

Because the algorithm used in this paper is not yet widely known, we state the algorithm and the restarting strategy so that the reader can implement their own version of the program. The main body of the algorithm is described in Section 2. A number of restarting strategies are discussed in Section 3. After describing the new algorithm, we present comparisons against other versions of the Lanczos method and the scaling properties of the new methods in Section 4, and discuss the question of computing degenerate eigenvalues and the workspace requirement in Section 5. Some concluding remarks are given in Section 6.

2 The thick-restart Lanczos algorithm

The thick-restart Lanczos algorithm combines the Lanczos algorithm with the thick-restart technique to form a new restarted eigenvalue method. It is designed to solve symmetric or Hermitian eigenvalue problems of the form,

$$Ax = \lambda x,$$

where A is the matrix, λ is an eigenvalue of A and x is the corresponding eigenvector. The Lanczos eigenvalue method computes approximate values of λ and x which will also be denoted by λ and x. Typically as more Lanczos steps are performed, the approximate values become closer to the exact values. The effectiveness of the method can be measured by the time it needs to compute the solutions to a desired level of accuracy.

The Lanczos method for eigenvalue problems has two conceptually distinct parts, one to construct the Lanczos basis and the other to compute the approximate solutions using a projection method usually the Rayleigh-Ritz projection [12]. The approximate eigenvalues and eigenvectors computed using the Rayleigh-Ritz projection are commonly referred to as the Ritz values and the Ritz vectors [12] and the vectors of the Lanczos basis are also known as the Lanczos vectors. In the restarted Lanczos algorithm the two basic steps of constructing a basis and performing the projection are carried out as usual. However, after a specified number of Lanczos vectors are built, a linear combination of the basis vectors is selected to start the Lanczos algorithm again by using the same workspace to store the new basis vectors. The thick-restart Lanczos algorithm [29] is a particular version of the restarted Lanczos method. It differs from the simple restarted Lanczos method in that it can save an arbitrary portion of the current Lanczos basis. This flexibility can be effectively used to enhance the performance of the restarted Lanczos method as demonstrated by the implicitly restarted Lanczos method [2] which is mathematically equivalent to the thick-restart Lanczos method[29]. Compared to the implicitly restarted Lanczos method, the thick-restart Lanczos method is simpler in two ways. The thick-restart procedure is only a slight modification of the Rayleigh-Ritz procedure and therefore it is simpler than the implicit restart procedure. The implicitly restarted Lanczos method needs a post-processing step to compute the eigenvectors after the eigenvalues are computed. The thick-restart Lanczos method does not need this step [17, 29].

The thick-restart Lanczos method described next is suitable for floating-point arithmetic implementation. The main difference between this one and the one for exact arithmetic is that this one has a *re-orthogonalization* step. The re-orthogonalization scheme shown here includes a local re-orthogonalization and a global re-orthogonalization. It guarantees that the Lanczos vectors are orthogonal to machine precision (ϵ) and coefficients α_i and β_i are accurate to the order of $\epsilon\|A\|$. This ensures no spurious solutions are computed and it allows us to compute both eigenvalues and eigenvectors simultaneously.

Assuming there is enough computer memory to store $m+1$ Lanczos vectors, the thick-restart Lanczos algorithm progressively builds its basis vectors as follows.

ALGORITHM 1 **Initialization**

To start solving a new eigenvalue problem, take a starting vector, normalize it and store the result in q_1 $(k = 0)$.

When restarting, the quantities $\alpha_1, \ldots, \alpha_k$, $\beta_1, \ldots, \beta_k$, $q_1, \ldots, q_k$ *and* q_{k+1} *shall satisfy*

$$Aq_i = \alpha_i q_i + \beta_i q_{k+1}, \qquad i = 1, \ldots, k. \tag{1}$$

Iterate

For $i = k+1, \ldots, m$,

1. $q_{i+1} = Aq_i$,
2. $\alpha_i = q_i^T q_{i+1}$,
3. **orthogonalization**:
   If $i > k+1$,

$$q_{i+1} \leftarrow q_{i+1} - \alpha_i q_i - \beta_{i-1} q_{i-1}, \tag{2}$$

   else

$$q_{i+1} \leftarrow q_{i+1} - \alpha_i q_i - \sum_{j=1}^{k} \beta_j q_j. \tag{3}$$

4. **re-orthogonalization**:
   - *If* $i > k+1$, $\eta = \alpha_i^2 + \beta_{i-1}^2$, *else* $\eta = \alpha_i^2 + \sum_{j=1}^{k} \beta_j^2$.
   - *If* $q_{i+1}^T q_{i+1} \geq \eta$, *perform the local re-orthogonalization*

$$q_{i+1} \leftarrow q_{i+1} - q_i q_i^T q_{i+1} - q_{i-1} q_{i-1}^T q_{i+1}, \tag{4}$$

   else if $q_{i+1}^T q_{i+1} \geq \epsilon^2 \eta$, *perform the global re-orthogonalization*

$$q_{i+1} \leftarrow q_{i+1} - \sum_{j=1}^{i} q_j q_j^T q_{i+1}, \tag{5}$$

   else, replace q_{i+1} *with a random vector that is orthogonal to* $[q_1, \ldots, q_i]$.
   - *Before updating* q_{i+1} *using Equation 4 or 5, replace* α_i *by* $\alpha_i + q_i^T q_{i+1}$. *However, do not modify* α_i *if* q_{i+1} *is replaced by a random vector.*
5. **normalization**: $\beta_i = \|q_{i+1}\|$, $q_{i+1} \leftarrow q_{i+1}/\beta_i$.
   If q_{i+1} *is a random vector, set* β_i *to zero after* q_{i+1} *is normalized.*

The second part of the algorithm performs the Rayleigh-Ritz projection. As in the usual projection step of any eigenvalue method, it computes the Ritz values and the Ritz vectors. The main difference is that it also prepares the quantities involved in Equation 1 to allow the thick-restart Lanczos algorithm to restart with an arbitrary number of vectors. We will only give the basic procedure in this section and leave the discussion on exactly how may vectors to save, i.e., the restarting strategies, to the next section. For convenience of describing the restarting procedure, we define $Q_m \equiv [q_1, \ldots, q_m]$ and $T_m \equiv Q_m^T A Q_m$. No computation is required to generate T_m because it can be assembled from $\alpha_1, \ldots, \alpha_m$ and $\beta_1, \ldots, \beta_{m-1}$. Because the Lanczos vectors Q_m are orthogonal to machine precision in the proceeding algorithm, the matrix T_m is accurate as well. This in turn ensures the Ritz values and the Ritz vectors are accurate and Equation 1 is satisfied to machine precision.

$$T_m = \begin{pmatrix} \alpha_1 & & & \beta_1 & & & \\ & \ddots & & \vdots & & & \\ & & \alpha_k & \beta_k & & & \\ \beta_1 & \cdots & \beta_k & \alpha_{k+1} & \beta_{k+1} & & \\ & & & \beta_{k+1} & \alpha_{k+2} & \ddots & \\ & & & & \ddots & \ddots & \beta_{m-1} \\ & & & & & \beta_{m-1} & \alpha_m \end{pmatrix}. \tag{6}$$

ALGORITHM **2 Restarting scheme**

1. *Find all eigenvalues and eigenvectors of* T_m, $T_m Y = YD$, *where the columns of* Y *are eigenvectors and the diagonal elements of* D *are the eigenvalues. The Ritz values are* $d_{i,i}, i = 1, \ldots m$.
2. *Choose* k *Ritz values to be saved, denote the Ritz values as* $\lambda_1, \ldots, \lambda_k$ *and renumber the corresponding eigenvectors of* T_m *as* $y_1, \ldots, y_k$.
3. *Let* $Y_k \equiv [y_1, \ldots, y_k]$, $\hat{q}_{k+1} = q_{m+1}$ *and replace the first* k *columns of* Q_m *with* $Q_m Y_k$, *i.e.,* $\hat{Q}_k = Q_m Y_k$. *The corresponding* $\hat{\alpha}_i$ *and* $\hat{\beta}_i$ *are defined as:*

$$\hat{\alpha}_i = \lambda_i, \quad \hat{\beta}_i = \beta_m y_{m,i}, \quad i = 1, \ldots, k. \tag{7}$$

In the actual implementation, the quantities $\hat{Q}_{k+1}$, $\hat{\alpha}_i$ and $\hat{\beta}_i$, occupy the same storage as the corresponding quantities Q_{k+1}, α_i and β_i. We distinguish them here only to make clear what are new quantities to be used in the next Lanczos iteration and what are old quantities to be discarded. It is easy to verify that $\hat{Q}_{k+1}$, $\hat{\alpha}_i$ and $\hat{\beta}_i$ satisfy Equation 1 [29], which means that they can be used to restart Algorithm 1. When entering Algorithm 1 for the first time, it is hard to satisfy Equation 1 with $k > 0$. Thus the thick-restart Lanczos method is usually started initially with only one vector. It is easy to implement a block version of the above algorithm, in which case, a block of starting vectors can be used.

What makes the above algorithm different from the naive explicit restarted Lanczos method is that k is significantly larger than one. When k is set to one during the restarting phase, the thick-restart Lanczos algorithm reduces to a simple explicitly restarted Lanczos algorithm. The explicitly restarted Lanczos algorithm is usually effective in finding one extreme eigenvalue. On the other hand, saving a large number of vectors when restarting as in the thick-restart Davidson method and the implicitly restarted Arnoldi method have been shown to be effective in finding a few eigenvalues [2, 18, 27]. Methods that save a large portion of the existing basis also work well when the maximum basis size m is close to the number of eigenvalues computed. For this reason, the ability to restart with an arbitrary number of Ritz vectors is an important property of the new method.

So far we have described all implementation details of the new algorithm except step 2 of the restarting procedure and how to perform convergence tests. Typical convergence tests for symmetric eigenvalue problems use either residual norms or estimated errors in the eigenvalues. In the experiment reported later, we declare a Ritz pair converged if its residual norm is less than 10^{-6}, $\|r_i\| = |\hat{\beta}_i| < 10^{-6}$. The restarting strategies will be discussed in the next section.

3 Restarting strategies

Two of the crucial decisions to be made during the thick-restart Lanczos algorithm are which Ritz pairs to save and exactly how many. Based on the analyses of Morgan [10], saving the Ritz values near the wanted eigenvalues could enhance the convergence rate of the restarted methods. The saved Ritz vectors may not be accurate approximations to their corresponding eigenvectors, but they approximately deflate the spectrum, increase the separation between the wanted eigenvalue and the rest of the spectrum and increase the convergence rate of the restarted Lanczos method. Since we only use the Lanczos method to compute extreme eigenvalues, the Ritz pairs saved are those with the largest Ritz values and the smallest Ritz values. The remainder of this section describes our attempt to identify exactly how many Ritz pairs should be saved. There are other arguments that can be used to guide the design of restarting schemes. A comprehensive review can be found elsewhere [30], in this section we will only describe two restarting strategies based on approximate deflation.

The research work that is more closely related to this one is the dynamic thick-restart scheme used in the dynamic thick-restart Davidson method [18]. In this paper the decision of how many vectors to save is based on maximizing the effective gap ratio. Assuming the m Ritz pairs are in ascending order of the Ritz values, if we are to save Ritz pairs $1, \ldots, k_l$ and $k_r + 1, \ldots, m$, the effective gap ratio for computing the smallest eigenvalue is defined to be

$$\gamma = \frac{\lambda_{k_l} - \lambda_1}{\lambda_{k_r+1} - \lambda_1}.$$

When computing more than one eigenvalue, the gap ratio is initially computed with the outermost

Ritz value as the reference. After the outermost eigenvalue has reached convergence, the effective gap ratio γ is computed with the next eigenvalue as the reference. For example, if the smallest Ritz value has converged, the effective gap ratio is computed as $\gamma = (\lambda_{k_l} - \lambda_2)/(\lambda_{k_r} - \lambda_2)$. The reference Ritz value serves a similar role as the target in the Davidson method [5] and we shall also call it the target in this paper.

Typically, the computed Ritz values are never exactly identical even if the corresponding eigenvalues are identical. In these cases, γ is a monotonic function if either k_l or k_r is fixed. The effective gap ratio increases as the difference between k_l and k_r decreases. For this reason, the maximum γ is always achieved when $k_r = k_l + 1$. This is usually not a good choice since it requires one to perform Rayleigh-Ritz projection and compute $m-1$ Ritz pairs after every matrix-vector multiplication. In practice, saving $m-1$ Ritz vectors often yields smaller residual norm reduction per matrix-vector multiplication than saving $m/2$ Ritz pairs. To understand this, we notice that the definition of the effective gap ratio γ is only accurate if the Ritz values $\lambda_1, \ldots, \lambda_{k_l}$ are close to the k_l smallest eigenvalues and $\lambda_{k_r+1}, \ldots, \lambda_m$ are close to the $m - k_r$ largest eigenvalues. Since m is much smaller than the size of matrix A as k_l becomes close k_r, the above conditions are not satisfied and γ is significantly larger than the actual effective gap ratio.

To prevent an over-aggressive choice of k_l or k_r, researchers have previously chosen to enforce the condition of $k_r \geq k_l + 3$ [18]. After extensive testing, we found that the following restriction gives much better timing results for the restarted Lanczos method, $k_r \geq k_l + \min(m - n_{eig}, 2(m - n_c)/5)$, where m is the maximum basis size, n_{eig} is the number of eigenvalues to be computed, n_c is the number of desired eigenvalues that have converged already. In actual implementation, we only need to consider $k_r = k_l + \min(m - n_{eig}, 2(m - n_c)/5)$ when performing the search for the best γ. This leads to a simpler searching algorithm than in the previous implementation.

If the effective gap ratio γ is accurate, after each Lanczos step, the residual norm of the target eigenvalue should decrease by a factor that is proportion to $e^{-\gamma}$ [10]. Based on this, the above dynamic restarting scheme maximizes the expected residual norm reduction during each Lanczos step. An alternative approach is to consider maximizing the residual norm reduction for the entire restarted loop. If k Ritz pairs are saved, the Lanczos algorithm can proceed $m - k$ step before restarting. The residual norm is expected to decrease by a factor proportional to $e^{-(m-k)\gamma}$. To maximize the residual norm reduction of the next restarted loop, we need to maximize μ,

$$\mu \equiv (m - k)\gamma.$$

Since μ is not a monotonic function like γ, to find its maximum value, we need to compare all possible choices of k_l and k_r. Our tests show that $k_r \geq k_l + \min(m - n_{eig}, 2(m - n_c)/5)$ is also a reasonable restriction on the search range for this scheme.

It is possible to construct more dynamic restarting schemes based on either empirical observation or other heuristics. However, through our tests, we have found that the above two schemes work well for the eigenvalue problems from electronic structure calculations studied in this paper. More detailed studies of various dynamic restarting heuristics can be found elsewhere [30].

4 Timing results

In this section, we will use electronic structure calculations of semiconductor nanosystems to demonstrate the effectiveness of our new method. The systems contain 512 to 250,000 atoms, thus far beyond the range of *ab initio* calculations. To describe the electronic structures of such large systems, the empirical pseudopotential has been used. In this scheme [24], the total potential of the

Table 1: Test problems.

name	# of atoms	# of plane-waves	description
InGaP512	512	6603	512-atom InGaP semiconductor alloy
InGaAs9k	9000	137919	9000-atom InAs quantum dot system
InGaAs93k	93000	1342479	93000-atom InAs quantum dot system
InGaAs250k	250000	3683087	250000-atom InAs quantum dot system

system is constructed from the superposition of atomic screened pseudopotentials $v_\alpha(r)$ of atom type α. As a result, the Hamiltonian of the system can be written as:

$$\hat{H} = -\frac{1}{2}\nabla^2 + \sum_{R_\alpha} v_\alpha(\mathbf{r} - \mathbf{R}_\alpha) \tag{8}$$

where $\{\mathbf{R}_\alpha\}$ are the atomic positions of atom type α, which are obtained *via* a valence force field calculation [14]. The empirical pseudopotential $v_\alpha(\mathbf{r})$ is fitted to bulk band structures and deformation potentials. The electronic structure of the system is obtained by solving the Schrodinger's equation

$$\hat{H}\psi(\mathbf{r}) = \epsilon\psi(\mathbf{r}), \tag{9}$$

where the wavefunction $\psi(\mathbf{r})$ is expanded using a plane wave basis.

This non-selfconsistent empirical pseudopotential scheme has been used to study quantum wells, superlattices, disordered superlattices, quantum wires, colloidal quantum dots, embedded quantum dots and composition modulations in alloys. Excellent agreements with the experiment have been obtained for single particle levels [25], exchange splitting [7], optical absorption spectra [23] and the magnitudes of Γ-X coupling [20].

As in most electronic structure calculations of semiconductor materials, the eigenvalues of the matrices fall into two distinct groups, the smaller ones form a group known as the valence band and the larger ones the conduction band. Typically, the eigenvalues of interests are those near the band gap because they are directly related to observable electronic properties [8]. Using the empirical pseudopotential schemes, it is possible to directly compute these eigenvalues and their corresponding eigenvectors without computing all the valence band states. Since the goal of this paper is demonstrate the capability of the eigenvalue method, we have decided to only report the timing results for computing a number of lowest conduction band states. In the cases where the valence band states are also computed, we observe similar performance characteristics as reported here.

Brief descriptions of the test problems used are list in Table 1. All InAs quantum dots listed are embedded in a GaAs lattice matrix. Let H denote the discrete form of the Hamiltonian given by the empirical pseudopotential method. We compute the conduction band states by computing the smallest eigenvalues of $(H - E_{ref})^2$ [22] with E_{ref} chosen to be $-4.4eV$ which is in the band gap and is near the top of the gap. The matrix H is Hermitian. The eigenvectors are represented as plane-waves and all calculations are done at the gamma point. Because of the gamma point symmetry, only half of the plane-wave coefficients need to be stored. The number of plane-waves reported in Table 1 are the number of plane-wave coefficients that are actually stored in computer memory.

Table 2: Time (seconds) used to find 5 lowest conduction states of InGaP512.

method	MATVEC	time (sec)
PLANSO	2578	473.6
PLANSO-lock	> 20000	> 530
max γ-3	3512	109.7
max γ	2936	84.3
max μ	2737	78.0

Timing results report here are obtained on a massively parallel computer, the Cray T3E 900, located at National Energy Research Supercomputer Center[1]. The matrix-vector multiplication uses parallel, three dimensional FFTs optimized for the Cray T3E [26].

Our first set of tests is performed on the smallest test problem, InGaP512. It is used to identify the restarting scheme that works well for this type of problems. Table 2 shows the time (seconds) used by a number of different Lanczos methods on 8 Processing Elements (PE) of the T3E. In addition to the thick-restart Lanczos method, we also used a package called PLANSO [28] in two different ways. The PLANSO package implements the Lanczos method with partial re-orthogonalization [12, 16]. The row headed by PLANSO uses PLANSO without restarting. Because this is a very small test problem, we are able to store as many (2578) Lanczos vectors as necessary to compute the five smallest eigenvalues of $(H - E_{ref})^2$. For larger matrices, the non-restarted Lanczos method usually requires more memory than is available on the T3E thus, it is not a widely available option. The other four methods each store 25 Lanczos vectors. PLANSO-lock represents a common way of restarting the Lanczos algorithm. The program has allocated enough space to store 25 Lanczos vectors. When this workspace is filled, the Rayleigh-Ritz projection is invoked to compute five approximate solutions. If any of them have converged, it will be locked and only used in orthogonalizing new Lanczos vectors. We can either restart the Lanczos method by taking one of the Ritz vectors or taking a linear combination of the Ritz vectors. However, neither of the two were successful in reaching desired accuracies within 20000 matrix-vector multiplications. The 530 seconds recorded in Table 2 is the time used to run the algorithm for 20000 steps (20000 matrix-vector multiplications).

The last three rows of Table 2 are from using the thick-restart Lanczos method with different restarting strategies. Row three (max γ-3) uses the dynamic restarting scheme used earlier [18] which always saves $m-3$ vectors when restarting. Row four (max γ) shows the time used when the thick-restart Lanczos method uses our new implementation to maximize the effective gap ratio γ. The main difference between these two is that less Ritz pairs are saved in the latter one. Because it computes less Ritz vectors, the restarting process is cheaper than before. In addition, each restart loop can carry out more matrix-vector multiplications and therefore generate more new information for the subsequent Rayleigh-Ritz projection. This leads to better approximate solutions with the newer scheme. In this particular example, 3512 Lanczos steps are taken with the former restarting strategy and 2936 steps, or, 16% less steps, are used with the latter strategy, and 23% less time is needed using the latter one. The time used by the Lanczos method with the strategy of maximizing residual norm reduction of the whole restarted loop (max μ), see last row of Table 2, is the smallest in the table. It uses almost 30% less time than restarting with the (max $\gamma - 3$) method and it is significantly better than the naive restart scheme (PLANSO-lock).

[1]NERSC can be accessed through the web at `http://www.nersc.gov`.

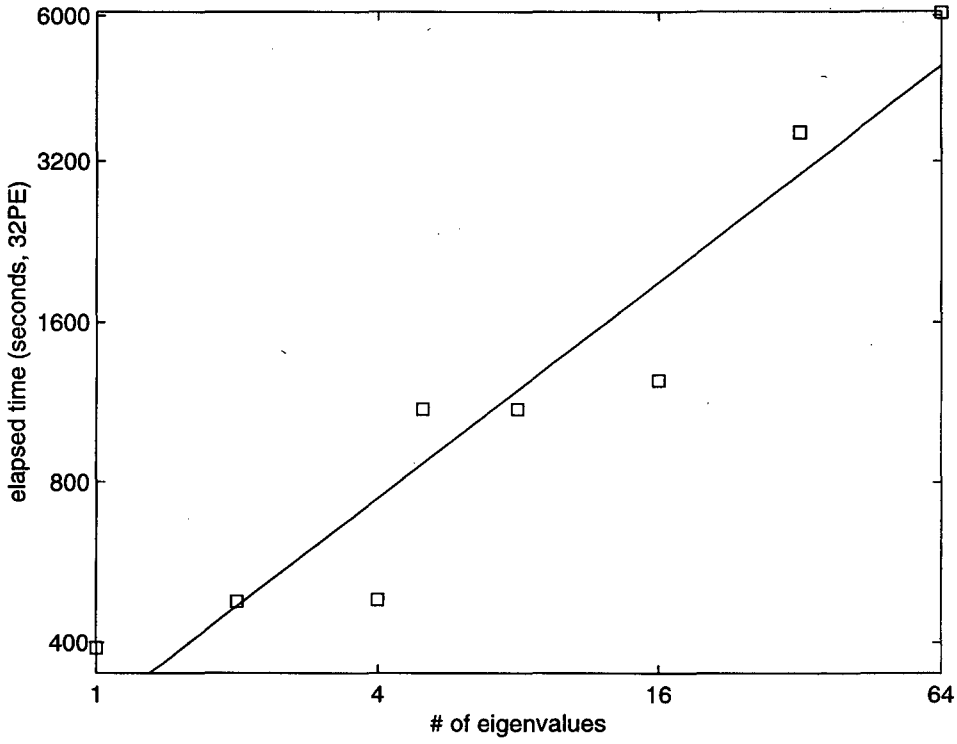


Figure 1: Time (seconds) used to compute different numbers of conduction band states of InGaAs9k.

Earlier, we mentioned that the restarted Lanczos methods use more iterations than the non-restarted versions that perform re-orthogonalization. Since each Lanczos iteration needs one matrix-vector multiplication, the data shown in Table 2 confirms the observation. However, the Lanczos method using the new restarting strategies needs less iterations than using the older strategies. In fact, the thick-restart Lanczos method that maximizes μ only uses six per cent more iterations than PLANSO. However, it only uses one sixth of the time of PLANSO. This difference in time is mostly due to the difference in time spend in re-orthogonalization. PLANSO saves all Lanczos vectors it ever computed, when it performs a re-orthogonalization it orthogonalizes against all of them. Each re-orthogonalization is very expensive near the end of the iterations. The restarted Lanczos method only saves a small number of vectors so that each re-orthogonalization is much cheaper. Even though it uses more matrix-vector multiplications and more re-orthogonalizations it still uses significantly less time.

The timing results shown in Table 2 are fairly representative of other tests we have conducted on this type of eigenvalue problems. In many cases, the new restarting scheme of maximizing μ is more effective than others. For this reason, we will only show results using this restarting strategy with the thick-restart Lanczos method, in the rest of this paper. Next, we will show how the new method scales with the number of eigenvalues and the matrix sizes.

Figure 1 shows the time used to compute different numbers of conduction band states of the InGaAs9k test problem on 32 processors of the Cray T3E. The eigenvalues and eigenvectors of $(H - E_{ref})^2$ are computed using the thick-restart Lanczos method that tries to maximize μ when restarting. When computing n_{eig} eigenvalues, the Lanczos basis size is $m = n_{eig} + 50$. In other words, the timing results shown in Figure 1 are generated by allowing the Lanczos method to use the fixed amount of workspace in addition to the space required to store the eigenvectors. The line

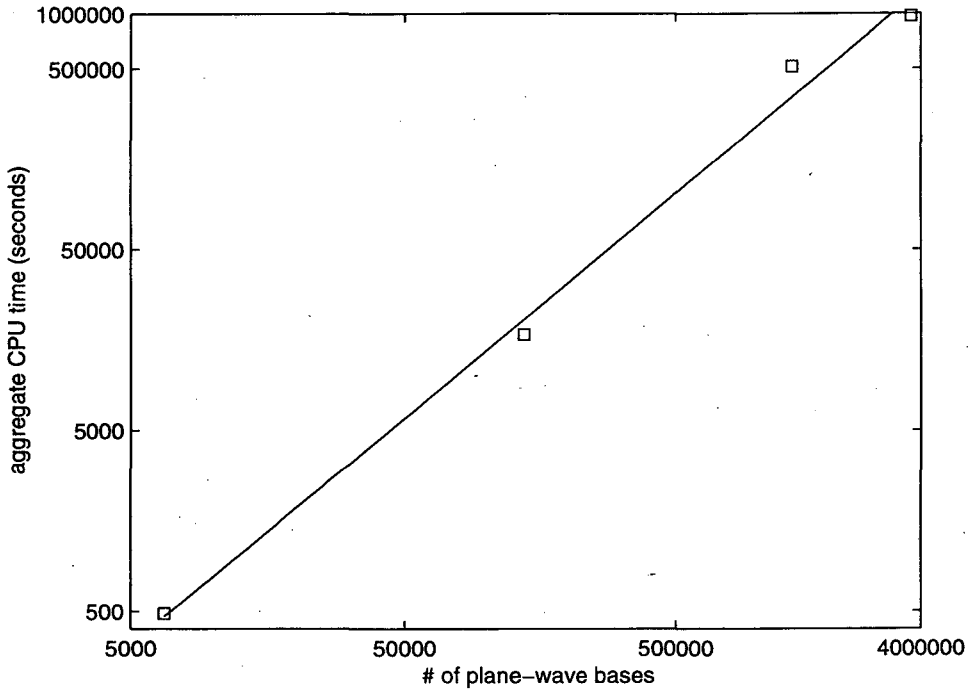


Figure 2: Aggregate time (seconds) used to compute conduction band states of different size test problems.

Table 3: Elapsed time (seconds) used to compute five conduction band states of the test problems.

name	m	# of PE	MATVEC	time
InGaP512	25	8	2737	78.0
InGaAs9k	50	32	5458	1096.2
InGaAs93k	100	64	4021	8021.8
InGaAs250k	200	256	3107	3782.4

going through the data points represents a linear regression of the log of time versus the log of n_{eig} and the slope indicates that to compute twice as many eigenvalues and eigenvectors the restarted Lanczos method used about 60 per cent more time ($t \propto n_{eig}^{0.7}$). The exact difference in time is a function of the spectrum distribution as well as the method used to compute the eigenvalues. Given a different type of eigenvalue problem, the exact scaling factor may change. Here we offer an intuitive explanation for the sub-linear scaling observed here and a more precise analysis can be found elsewhere [10]. While computing λ_1 and x_1, the thick-restart scheme also saves the nearby Ritz pairs . When λ_1 and x_1 reach convergence, λ_2 and x_2 are nearly converged too. After the first eigenvalue is computed, much less time may be needed in order to compute the second one.

The second type of scaling studied here is to see how the new method behaves when the problem size increases. Figure 2 shows the aggregate time used by the thick-restart Lanczos method to solve the different sized empirical pseudopotential calculations. All four problems listed in Table 1 are used. The time shown in the figure is the aggregate time used by all processors to compute the five lowest conduction band states. Table 3 shows the number of processors and the elapsed time. As

the problem size increases, we use more processors and larger Lanczos bases. The line in Figure 2 is a linear regression of the data, more precisely, the log of time versus the log of problem size, and its slope indicates a scaling factor of roughly of 1.2, i.e., the aggregate time used is proportional to $n^{1.2}$, where n is the number of plane-wave bases used. The time used by the Lanczos method grows faster than linear because most of its components scale super-linearly. For example, the time to apply the Hamiltonian on a vector scales as $n\log(n)$, the Gram-Schmidt procedure used to perform re-orthogonalization scales as mn, and the time needed to compute k Ritz vectors during restarting procedure scales as kmn. In addition, as more processors are used there is more communication overhead which is also contributing to the total time growing faster than linear. Of course, as the problem size changes, the spectrum also changes which affects the total time because different numbers of Lanczos steps are needed. Typically, as problem size increases, more steps are needed to compute the same number of eigenvalues, and therefore more time will be used.

We have also performed a series of tests by directly computing the smallest eigenvalues of H. The scaling factors observed for these calculations were close to those observed for computing the conduction band edge states. On this set of test problems, the thick-restart Lanczos method scales well with both the number of eigenvalues and the matrix size. Many eigenvalue problems from electronic structure calculations have similar characteristics to the test problems and we expect the thick-restart Lanczos method to work well for these cases.

5 Quality of solutions and workspace requirement

In the previous section we have demonstrated that the new method uses less time than some of the older versions of the Lanczos method and the new one scales well as the problem size increases. This section addresses two issues that worry the application programmers particularly those who perform electronic structure calculations: the Lanczos method is not able to compute all eigenvectors of a degenerate eigenvalue and it requires more workspace than other methods such as CG.

Electronic structure calculations often give rise to degenerate or near degenerate eigenvalues and it is crucial that all eigenvectors are found. In exact arithmetic, the Lanczos method can only compute one eigenpair from each degenerate set. In order to reliably compute multiple eigenvectors of a degenerate eigenvalue, one either uses a block version of the Lanczos method or adds locking to the standard Lanczos method. To see how the thick-restart Lanczos method computes degenerate eigenvalues, we start by examining its convergence history.

Figure 3 shows the convergence history of solving the InGaP512 test problem which has higher degeneracy than the others. The top plot shows the five smallest Ritz values of $(H - E_{ref})^2$ (in natural units: Rydberg^2) and the bottom plot shows their corresponding residual norms. Initially, the five smallest Ritz values are distinct. After about 700 Lanczos steps, the two smallest Ritz values have converged to the two smallest eigenvalues but the residual norms are only of the order of 10^{-5}. After about 1400 steps, the third Ritz value drops below the second one and approaches the smallest one. This indicates that the second eigenvector corresponding to the smallest eigenvalue of $(H - E_{ref})^2$ has been identified. After about 2200 steps, the third smallest Ritz value converges to the first two and the third eigenvector of the smallest eigenvalue appears. It takes roughly the same number of Lanczos steps to identify one eigenvector corresponding to the smallest eigenvalue of $(H-E_{ref})^2$. In this case, about 700 Lanczos steps are needed to identify each eigenvector. Similar observation have also been made in the case where the Lanczos algorithm is used with the partial re-orthogonalization but without restart, see Figure 4. Previously, similar convergence history has been observed in Lanczos methods without re-orthogonalization [6, 21]. However, the difference is that without re-orthogonalization the Lanczos method repeatedly generates the same eigenvector

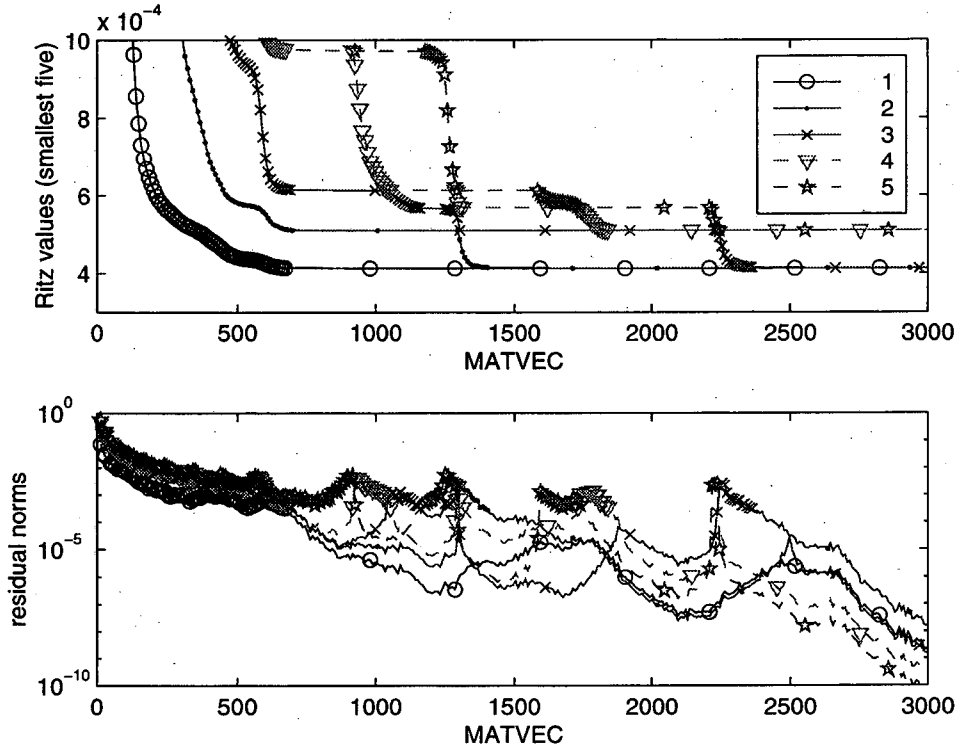


Figure 3: The convergence history of the thick-restart Lanczos method.

while with re-orthogonalization the eigenvectors computed are distinct. Our explanation of the similarities is as follows. Because of the floating-point round-off error, the Lanczos basis is likely to contain a small component in the direction of any eigenvector. It takes the Lanczos method about the same number of steps to compute each eigenvector because the convergence rates are dictated by the eigenvalues which are the same for different eigenvectors of a degenerate eigenvalue. In addition, the initial starting points can be regarded as the same for most eigenvectors since every one, except the first, starts as a round-off error. Note that locking is not used in generating either Figure 3 or 4.

The above arguments show that the thick-restart Lanczos method is almost certain to find all eigenvectors of a degenerate eigenvalue. To ensure that no eigenvector is missed in the solution, we suggest two strategies, to compute more eigenvalues than needed and to ask for more accuracy than

Table 4: The smallest five Ritz values of $(H - E_{ref})^2$ computed when asking for different n_{eig} ($\|r_i\| < 10^{-5}$, $m = 25$).

n_{eig}	MATVEC	time (sec)	λ_1	λ_2	λ_3	λ_4	λ_5
			($\times 10^{-4}$)				
5	2144	60.8	4.1	4.1	5.1	5.1	5.7
8	2123	63.4	4.1	4.1	5.1	5.1	5.7
9	3575	107.4	4.1	4.1	4.1	5.1	5.1

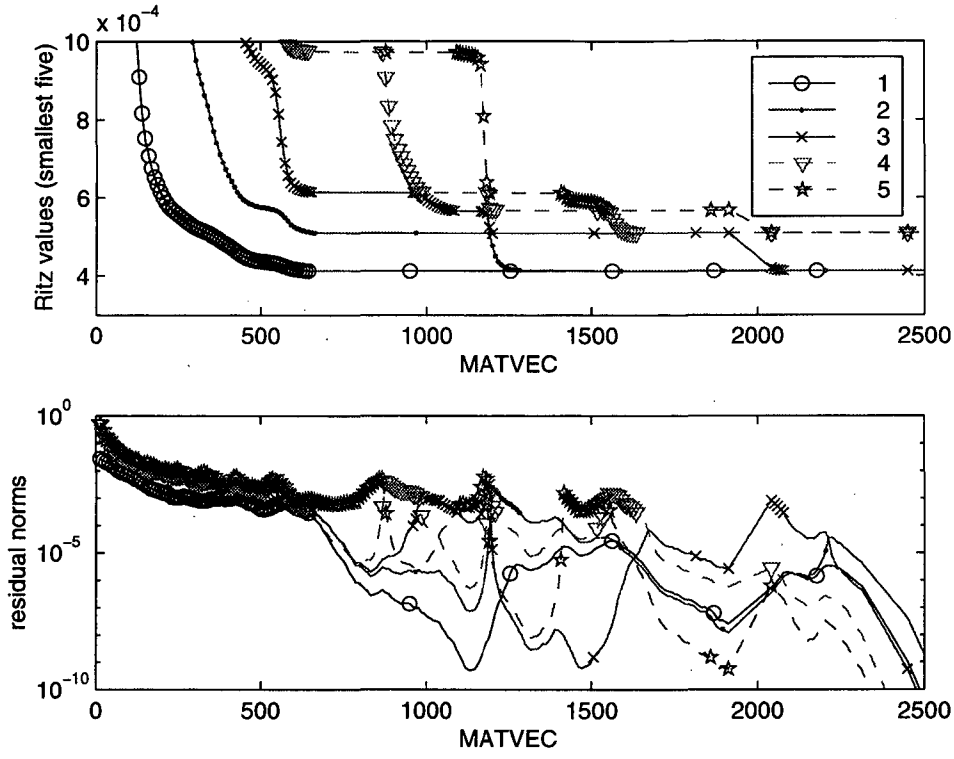


Figure 4: The convergence history of the Lanczos method without restart.

needed. From Figure 3 and 4, we can identify five distinct horizontal lines in the Ritz value history and each of the lines represents an eigenvalue of $(H - E_{ref})^2$. If they were simple eigenvalues, the five smallest eigenvalues would be 4.1×10^{-4}, 5.1×10^{-4}, 5.7×10^{-4}, 6.1×10^{-4}, and 9.7×10^{-4}. Table 4 shows the five smallest Ritz values computed when different number of eigenvalues are requested. As more and more eigenvalues are requested, the five smallest Ritz values become closer and closer to the five smallest eigenvalues. When requesting nine eigenvalues, the five smallest ones displays the correct degeneracy. Table 5 shows how the five smallest Ritz values change as the tolerance changes. In this particular case, we need to set τ to something less than 10^{-5} in order to get the correct solutions. The time to generate the solutions with the correct degeneracy are within 20% of each other in Table 4 and 5. This indicates that the two schemes are almost equally effective. Both schemes need additional research to make them more rigorous. We offer the following rule-of-thumb for choosing parameters:

- when choosing the option of computing more eigenvalues than needed, compute at least five more eigenvalues or if computing a large number of eigenvalues compute 10% more;
- when using the residual tolerance as the control, make sure the value of τ is less than $\sqrt{\epsilon}\|A\|$, where ϵ is the machine precision and $\|A\|$ is the two-norm or Frobenius norm of the matrix.

Between the two schemes, we believe the second one, controlling τ, is more effective. This is based on the observation that when all eigenvectors of a degenerate eigenvalue are identified the residual norms decrease rapidly and monotonically. This means that requiring addition accuracy does not cost a significant amount of extra time. In Table 5, $\tau = 10^{-7}$ and $\tau = 10^{-8}$ both lead to the correct solutions, but requiring $\tau = 10^{-8}$ only takes 5% more time than requiring $\tau = 10^{-7}$,

Table 5: The smallest five Ritz values of $(H - E_{ref})^2$ computed when different residual tolerances are used ($\|r_i\| < \tau$, $m = 25$).

τ	MATVEC	time (sec)	λ_1	λ_2	λ_3	λ_4	λ_5
			($\times 10^{-4}$)				
10^{-3}	712	20.3	4.1	5.1	6.1	9.7	15.0
10^{-4}	1979	56.5	4.1	4.1	5.1	5.1	5.7
10^{-5}	2144	60.8	4.1	4.1	5.1	5.1	5.7
10^{-6}	2737	78.0	4.1	4.1	4.1	5.1	5.1
10^{-7}	2956	84.4	4.1	4.1	4.1	5.1	5.1
10^{-8}	3109	88.5	4.1	4.1	4.1	5.1	5.1

Table 6: Time (seconds) used to compute the five lowest conduction band states of InGaAs9k using different size bases.

m	50	60	75	100	200
time	1096.2	1040.5	1063.1	1107.2	1299.2

similarly requiring $\tau = 10^{-7}$ only needs 8% more time than requiring $\tau = 10^{-6}$. There are many cases where two eigenvalues are distinct but are near to each other, e.g., eigenvalue 5.7×10^{-4} and 6.1×10^{-4}, where the Lanczos method may have similar difficulty to computing degenerate eigenvalues. The two schemes suggested here should be reasonable approaches to deal with this case as well.

One parameter the user needs to choose when using a restarted Lanczos method is the basis size, m. Next we will show that it is reasonably easy to pick a good value for m. Table 6 shows the time required with different m to compute the five lowest conduction band states. From the table we see that the difference in time caused by different m is relatively small compared to the difference between using the thick-restart Lanczos method and other versions of the Lanczos method, see Table 2. Typically, when m is small, as m increases, the time decreases. After m increases to the optimal value, the minimum time is achieved. If m increases further, the time increases slowly as shown in Table 6. The user usually has to perform a small number of tests in order to identify a reasonable m to use. For computing n_{eig} eigenvalues and eigenvectors, we suggest testing $m = n_{eig} + 10$ and $m = n_{eig} + 20$. If one of the two test cases fail to compute the solutions in a reasonable amount of time or the larger basis size works considerably better than the smaller one, a even larger m should be used. The basis sizes reported in table 3 can be used as a reference for solving similar types of problems. However, the values reported here are probably larger than necessary if one is to compute the smallest eigenvalues of H rather than those of $(H - E_{ref})^2$.

One of the common complaints against the Lanczos method is that it uses more workspace than CG. This is true in some cases. However, because a larger workspace, i.e., a larger m, often leads to a faster convergence rate, it is worthwhile to use more workspace if there is enough memory available. In addition, the thick-restart Lanczos method works well with a constant amount of workspace as the number of eigenvalues increases as shown Figure 1. If a large number of eigenvalues and eigenvectors are required, the thick-restart Lanczos method may still need more workspace than

some band-by-band versions of the CG method, however, it may actually need less workspace than some implementations of all-band CG methods.

6 Concluding remarks

In this paper, we have given a practical version of the thick-restart Lanczos method for symmetric and Hermitian eigenvalue problems and described two restarting strategies that we found to be effective. Through numerical examples, we have demonstrated that the thick-restart Lanczos method uses less time than older versions of the Lanczos method and the new method scales well as the problem size increases. This method is well suited for computing a large number of eigenvalues and eigenvectors of very large matrices.

Many electronic structure calculations need to compute solutions of a set of related eigenvalue problems [13]. In these cases, it is important to take advantage of the existing solutions when solving the next eigenvalue problem. One way to do this is to use a linear combination of the eigenvectors from the previous step as the starting vector for the Lanczos method [1, 15]. However, a version of dynamic thick-restart Davidson method [18] might be more appropriate than the thick-restart Lanczos method. Even in this case, the restarting strategies described in this paper are still useful for the Davidson method.

Through the study of the convergence history, we conclude that the thick-restart Lanczos method can compute all eigenvectors of degenerate eigenvalues. There is no easy way to detect that all eigenvectors are found, however, the two strategies, computing more eigenvalues and requiring more accuracy, appear to work well in practice.

The Lanczos method often needs more workspace than some versions of the CG method. However, if there is a large amount of computer memory available, it is worthwhile to let the thick-restart Lanczos method use more workspace as this often leads to less time being used. Clearly, the thick-restart Lanczos method is not for every type of eigenvalue problem. However, in the cases where it is appropriate, for example, when tens of eigenvalues are required, when there is reasonable amount of space to store some extra vectors ($m - n_{eig} > 10$), or when there isn't a large number of good starting vectors, we have demonstrated that the thick-restart Lanczos is an effective method.

7 Acknowledgment

This work was supported by the Director, Office of Science, Office of Laboratory Policy and Infrastructure Management, of the U.S. Department of Energy under Contract No. DE-AC03-76SF00098, and by the Office of Basic Energy Sciences, Division of Material Sciences, under contract No. DE-AC36-83CH10093.

This research used resources of the National Energy Research Scientific Computing Center, which is supported by the Office of Science of the U.S. Department of Energy.

We would also like to thank Osni Marques and Alex Zunger for useful discussion during the preparation of this paper.

References

[1] A. Alavi, J. Kohanoff, M. Parrinello, and D. Frenkel. *Ab Initio* molecular dynamics with excited electrons. *Phys. Rev. Lett.*, 73(19):2599–2602, 1994.

[2] D. Calvetti, L. Reichel, and D. Sorensen. An implicitly restarted Lanczos method for large symmetric eigenvalue problems. *Electronic Transactions on Numerical Analysis*, 2:1–21, 1994.

[3] J. Cullum and R. A. Willoughby. *Lanczos Algorithms for Large Symmetric Eigenvalue Computations: Theory*, volume 3 of *Progress in Scientific Computing*. Birkhauser, Boston, 1985.

[4] Ernest R. Davidson. The iterative calculation of a few of the lowest eigenvalues and corresponding eigenvectors of large real-symmetric matrices. *J. Comput. Phys.*, 17:87–94, 1975.

[5] Ernest R. Davidson. Super-matrix methods. *Computer Physics Communications*, 53:49–60, 1989.

[6] James W. Demmel. *Applied Numerical Linear Algebra*. SIAM, Philadelphia, PA, 1997.

[7] A. Franceschetti, L. W. Wang, H. Fu, and A. Zunger. Short-range versus long-range electron-hole exchange interactions in semiconductor quantum dots. *Phys. Rev. B*, 58(20):13367–13370, 1998.

[8] Charles Kittel. *Introduction to Solid State Physics*. John Wiley & Sons, New York, 1986.

[9] R. Lehoucq, D. Sorensen, and C. Yang. *ARPACK USERS GUIDE: solution of large scale eigenvalue problems with implicitly restarted Arnoldi methods*. SIAM, Philadelphia, PA, 1998. ARPACK Software is available at `http://www.caam.rice.edu/software/ARPACK/`.

[10] Ronald B. Morgan. On restarting the Arnoldi method for large nonsymmetric eigenvalue problems. *Mathematics of Computation*, 65(215):1213–1230, July 1996.

[11] B. N. Parlett and D. Scott. The Lanczos algorithm with selective orthogonalization. *Math. Comp.*, 33:217–2388, 1979.

[12] Beresford N. Parlett. *The symmetric eigenvalue problem*. Classics in Applied Mathematics. SIAM, Philadelphia, PA, 1998.

[13] M. C. Payne, M. P. Teter, D. C. Allan, T. A. Arias, and J. D. Joannopoulos. Iterative minimization techniques for *ab initio* total-energy calculations: molecular dynamics and conjugate gradients. *Review of Modern Physics*, 64:1045–1097, 1992.

[14] C. Pryor, J. Kim, L.-W. Wang, A. J. Williamson, and A. Zunger. Comparison of two methods for describing the strain profiles in quantum dots. *Journal of Applied Physics*, 83(5):2548–2554, 1998.

[15] Yousef Saad. *Numerical Methods for Large Eigenvalue Problems*. Manchester University Press, 1993.

[16] Horst D. Simon. Analysis of the symmetric Lanczos algorithm with reorthogonalization methods. *Lin. Alg. Appl.*, 61:101–131, 1984.

[17] D. S. Sorensen. Implicit application of polynomial filters in a K-step Arnoldi method. *SIAM J. Matrix Anal. Appl.*, 13(1):357–385, 1992.

[18] A. Stathopoulos, Y. Saad, and K. Wu. Dynamic thick restarting of the Davidson and the implicitly restarted Arnoldi methods. *SIAM J. Sci. Comput.*, 19(1):227–245, 1998.

[19] I. Stich, R. Car, M. Parrinello, and S. Baroni. Conjugate gradient minimization of the energy functional: a new method for electronic structure calculation. *Phys. Rev. B*, 39(8):4997–5004, 1989.

[20] L.-W. Wang, A. Franceschetti, and A. Zunger. Million-atom pseudopotential calculation of gamma -X mixing in gaas/alas superlattices and quantum dots. *Phys. Rev. Lett.*, 78(14):2819–2822, 1997.

[21] L.-W. Wang and A. Zunger. Large scale electronic structure calculations using the Lanczos method. *Computational Materials Science*, 2:326–340, 1994.

[22] L.-W. Wang and A. Zunger. Solving Schrodinger's equation around a desired energy: application to silicon quantum dots. *Journal of Chemical Physics*, 100(3):2394–7, February 1994.

[23] L.-W. Wang and A. Zunger. Pseudopotential calculations of nanoscale cdse quantum dots. *Phys. Rev. B*, 53(15):9579–9582, 1996.

[24] L.-W. Wang and A. Zunger. Pseudopotential theory of nanometer silicon quantum dots. In P. V. Kamat and D. Meisel, editors, *Semiconductor Nanoclusters*. Elsevier Science, 1996.

[25] L.-W. Wang and A. Zunger. High-energy excitonic transitions in cdse quantum dots. *Journal of Physical Chemistry B*, 102(34):6449–6454, 1998.

[26] A. J. Williamson, A. Zunger, and A. Canning. Prediction of a strain-induced conduction-band minimum in embedded quantum dots. *Phys. Rev. B*, 57(8):4253–4256, 1998.

[27] Kesheng Wu. *Preconditioned Techniques for Large Eigenvalue Problems*. PhD thesis, University of Minnesota, 1997. An updated version also appears as Technical Report TR97-038 at the Computer Science Department.

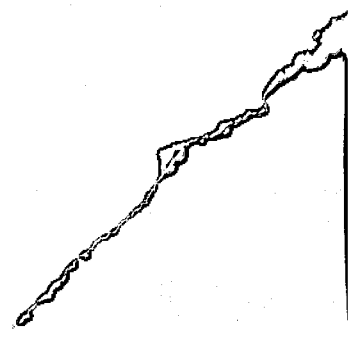
[28] Kesheng Wu and Horst Simon. A parallel Lanczos method for symmetric generalized eigenvalue problems. Technical Report 41284, Lawrence Berkeley National Laboratory, 1997.

[29] Kesheng Wu and Horst Simon. Thick-restart Lanczos method for symmetric eigenvalue problems. Technical Report 41412, Lawrence Berkeley National Laboratory, 1998.

[30] Kesheng Wu and Horst D. Simon. *Dynamic Retarting Schemes For Thick-Restart Lanczos Method*, 1998. in preparation.

[31] Alex Zunger. Electronic-structure theory of semiconductor quantum dots. *MRS Bulletin*, 23(2):35–42, February 1998.

Ernest Orlando Lawrence Berkeley National Laboratory
One Cyclotron Road | Berkeley, California 94720



Prepared for the U.S. Department of Energy under Contract No. DE-AC03-76SF00098