A Density Matrix-based Algorithm for Solving Eigenvalue Problems

Eric Polizzi

Department of Electrical and Computer Engineering, University of Massachusetts, Amherst
(Dated: January 17, 2009)

A new numerical algorithm for solving the symmetric eigenvalue problem is presented. The

technique deviates fundamentally from the traditional Krylov subspace iteration based techniques
(Arnoldi and Lanczos algorithms) or other Davidson-Jacobi techniques, and takes its inspiration

from the contour integration and density matrix representation in quantum mechanics. It will be

shown that this new algorithm - named FEAST - exhibits high efficiency, robustness, accuracy

and scalability on parallel architectures. Examples from electronic structure calculations of Carbon

nanotubes (CNT) are presented, and numerical performances and capabilities are discussed.

PACS numbers: 02.60.-x,02.70.Hm,02.70.-c,02.10.Ud,31.15.-p,71.15.Dx

Keywords: diagonalization technique, contour integration, electronic structure calculations

I. INTRODUCTION

The generalized eigenvalue problem, that is the determination of nontrivial solutions (λ, x) of

Ax = λBx

with A and B square matrices, is a central topic in numerical linear algebra and arises from a wide range of

applications in sciences and engineering. In electronic structure calculations, in particular, the eigenvalue

problem is one of the most challenging applied numerical problem - also called diagonalization procedure or spectral decomposition. In these calculations, the electron density can be formally calculated by summation of the amplitude square of the wave functions Ψm solution of the Schrödinger-like eigenvalue problem HΨm = EmSΨm with different discrete energies Em (where H represents the Hamiltonian Hermitian matrix and S is a symmetric positive matrix obtained using non-orthogonal basis functions). This procedure can be quite computationally challenging for large-scale simulations of systems containing more than a hundred of atoms and/or where a large number of eigenpairs (Em, Ψm) are needed. Progress in electronic structure calculations as for other large-scale modern applications, are then much likely dependent on advances in diagonalization methods.

In the past decades, the eigenvalue problem has led to many challenging numerical questions and a central problem [1]: how can we compute eigenvalues and eigenvectors in an efficient manner and how accurate are they? Powerful tools have then been developed from Jacobi method and power iterations, to iterative Krylov subspace techniques (including Arnoldi, and Lanczos methods), or other Davidson-Jacobi techniques [2]. Traditional numerical algorithms and library packages are yet facing new challenges for addressing the current large-scale simulations needs for ever higher level of efficiency, accuracy and scalability in modern parallel architectures.

This article presents a new robust and scalable algorithm design for solving the eigenvalue problem- named FEAST- which deviates fundamentally from the traditional techniques above, and takes its inspiration from the density matrix representation and contour integration in quantum mechanics. Section II summarizes the electronic structure and contour integration problems which have motivated the development of the new algorithm. The FEAST algorithm is then described in detail in Section III, and numerical examples and performance results are presented in Section IV. Finally, Section V presents some discussions regarding the efficiency, robustness and scalability of the algorithm.

II. THE CONTOUR INTEGRATION TECHNIQUE IN ELECTRONIC STRUCTURE CALCULATIONS

Although new fast sparse solvers have allowed considerable time saving for obtaining the eigenpairs (Em, Ψm) in electronic structure calculations, such as the Rayleigh-quotient multigrid [3] developed for the MIKA package, or the parallel Chebyshev subspace iteration technique developed for the PARSEC package [4,5], these calculations are still considered computationally extremely challenging and linear scalability is not easily achievable.

An alternative approach to the Schrödinger picture for obtaining the electron density consists in performing a contour integration of the diagonal elements of the Green’s function matrix G(Z) = (ZS − H)−1 over the complex energy space [6]. At zero temperature, the resulting expression for the electron density in real-space is:

\[
n(r) = -\frac{1}{2\pi i} \int_{C} dZ \, G(r, r, Z) = \sum_{m} |\Psi_{m}(r)|^{2},
\]

where the complex contour C includes all the eigenvalues Em below the Fermi level EF, and where the spin factor is not considered. It should be noted that at non zero temperatures, this expression would also include the contribution of the residues of all poles of the Fermi-Dirac distribution function on the imaginary axis at the position of the Fermi level [7]. For transport problems
and open systems, in turn, the contour integration is often used to compute the equilibrium part of the electron density where self-energy boundary conditions need to be included in the Hamiltonian matrix $H$. The contour integration technique represents a priori an attractive alternative approach to the traditional eigenvalue problem for computing the electron density since the number of alternative approach to the traditional eigenvalue problem integration technique represents a priori an attractive algorithm for computing the equilibrium part of the electron and open systems, in turn, the contour integration is of generalized eigenvalue problems of this form containing directly the eigenpairs solutions using the density numerical algorithm design FEAST is proposed for obtaining directly the Green’s function and perform instead of solving linear systems (i.e. shifting strategy). More recently, Sakurai et al. have proposed a root finding technique which consists of a contour integration of a projected Laurent series-type decomposition of the Green’s function. In principle, a set of complex moments is clear from equation (1) that the contour integration technique can be expressed in terms of the eigenvectors present in the contour as follows:

$$\rho = \frac{1}{2\pi i} \int_{C} dZ G(Z) = \sum_{m=1}^{M} |x_m\rangle \langle x_m|.$$  \hfill (3)

In matrix notations the second term of the equation reads $XX^T$ where $X_{N \times M} = \{x_1, x_2, \ldots, x_M\}$ ($X_{N \times M}$ being the vector $\{x_1, x_2, \ldots, x_M\}$ obtained by solving linear systems along the contour

$$Q_{N \times M} = \frac{1}{2\pi i} \int_{C} dZ G(Z)Y_{N \times M},$$  \hfill (4)

while the second expression in (3), implies these vectors $Q$ can be formally generated by the eigenfunctions $X$ inside the contour

$$Q_{N \times M} = X_{N \times M} W_{M \times M} \text{ with } W_{i,j} = x_i^T y_j.$$  \hfill (5)

In other words, each $Q$ column vector obtained in $\{4\}$ represents a different linear combination of unknown basis functions $X$ in (3). Using a Rayleigh-Ritz procedure, the problem (2) is now equivalent to computing the eigenpairs $(\epsilon_m, \Phi_m)$ of the following reduced generalized eigenvalue problem of size $M$:

$$A_Q \Phi = \epsilon B_Q \Phi$$  \hfill (6)

with $A_Q = QT^TA$ and $B_Q = QT^TB$. \hfill (7)

The Ritz values and vectors are then given by:

$$\lambda_m = \epsilon_m, \quad m = 1, \ldots, M$$  \hfill (8)

$$X_{N \times M} = Q_{N \times M} \Phi_{M \times M}$$  \hfill (9)

where $\Phi_{M \times M} = \{\Phi_1, \Phi_2, \ldots, \Phi_M\}$. One can show that the obtained eigenvectors $X$ are naturally $B$-orthonormal i.e. $x_i^T B x_j = \delta_{i,j}$, if the eigenvectors of the reduced problem (9) are $B$-orthonormal i.e. $\Phi_i^T B \Phi_j = \delta_{i,j}$.

### III. FEAST

#### A. Introduction

In this section, a new algorithm is presented for solving generalized eigenvalue problems of this form

$$Ax = \lambda Bx,$$  \hfill (2)

within a given interval $[\lambda_{\text{min}}, \lambda_{\text{max}}]$, where $A$ is real symmetric or Hermitian and $B$ is a symmetric positive definite (s.p.d). One common way to accelerate the convergence rate of traditional iterative techniques consists in performing a factorization of the Green’s function $G(\sigma) = (\sigma B - A)^{-1}$ for some reasonable shift $\sigma$ close to the eigenvalues in the search interval which leads to solving linear systems (i.e. shifting strategy). More recently, Sakurai et al. have proposed a root finding technique which consists of a contour integration of a projected Laurent series-type decomposition of the Green’s function. In principle, a set of complex moments can be obtained by solving few linear systems along the contour, which can generate an identical subspace to the one spanned by the eigenvectors present inside the contour. In practice, however, robustness and accuracy are not easily achievable. In our approach, we avoid decomposing directly the Green’s function and perform instead an exact mathematical factorization of its contour integration - which represents the reduced density matrix $\rho$ in quantum mechanics. One can show that this factorization can be expressed in terms of the eigenvectors present inside the contour as follows:

$$\rho = \frac{1}{2\pi i} \int_{C} dZ G(Z) = \sum_{m=1}^{M} |x_m\rangle \langle x_m|.$$  \hfill (3)

In practice, the vectors $Q$ are computed by performing a numerical integration of each vectors $G(Z)Y$ along the complex contour $C$. Let us consider a circle centered in the middle of the search interval $[\lambda_{\text{min}}, \lambda_{\text{max}}]$, it should be noted that the expression of the contour integration
can be further simplified since $G(Z) = G^\dagger(Z)$. Denoting $C^+$ the positive half circle of the complex contour, it comes if $A$ is Hermitian:

$$\rho = -\frac{1}{2\pi i} \int_{C^+} dZ \{ G(Z) - G^\dagger(Z) \},$$

(10)

and if $A$ is real symmetric:

$$\rho = -\frac{1}{\pi} \int_{C^+} dZ \Im\{ G(Z) \},$$

(11)

where $\Im\{ \}$ stands for the imaginary part. Using a $N_e$-point Gauss-Legendre quadrature on the positive half circle $C^+$ (see Figure 1), with $x_e$ the $e^{th}$ Gauss node associated with the weight $\omega_e$, one obtains if $A$ is Hermitian and $Y, Q \in \mathbb{C}^{N \times M}$:

$$Q = -\sum_{e=1}^{N_e} \frac{1}{4} \omega_e r(\exp(\im\theta_e) G(z_e) + \exp(-\im\theta_e) G^\dagger(z_e)) Y,$$

with

$$r = \frac{\lambda_{\max} - \lambda_{\min}}{2}, \quad \theta_e = -\frac{\pi}{2}(x_e - 1),$$

(13)

$$Z_e = \frac{\lambda_{\max} + \lambda_{\min}}{2} + r \exp(\im\theta_e).$$

(14)

If $A$ is real symmetric, $Y, Q \in \mathbb{R}^{N \times M}$ and one can use:

$$Q = -\sum_{e=1}^{N_e} \frac{1}{2} \omega_e \Re\{ r \exp(\im\theta_e) G(z_e) Y \},$$

(15)

where $\Re\{ \}$ stands for the real part.

In order to reduce the numerical quadrature error of the contour integral, one may consider the two following improvements:

(i) Performing outer-iterative refinement steps. Once the eigenvectors $X$ are obtained (9), a new set of initial guess vectors $Y = BX$ can be used. Postmultiplying the density matrix $Q$ by $Y$, one now obtains from (10) that $Q$ converges to $X$ since $X^T BX = I$ (i.e. $W_{ij} = \delta_{i,j}$ and then $\rho BX = X$). A fast test for convergence can be obtained by checking the trace of the eigenvalues $\lambda_i$.

(ii) Postmultiplying the density matrix $Q$ by $M_0$ random vectors (rather than $M$) where $M_0$ is greater than $M$. The reduced dense generalized eigenvalue problem (8) of size $M_0$ can be solved using standard eigenvalue LAPACK routines [13]. Since we do not perform the orthogonalization of the vectors $Q$, one has to make sure that $BQ_{M0}M_0$ is symmetric positive definite i.e. $M_0$ does not exceed an upper limit which can easily be obtained a posteriori.

FIG. 1: Schematic representation of the complex contour integral defined by the positive half circle $C^+$. In practice, the vectors $Q$ are computed via a numerical integration (e.g. Gauss-Legendre quadrature) where only few linear systems $G(Z)Y$ needs to be solved at specific points $Z_e$ along the contour.

1- Select $M_0 > M$ random vectors $Y_{N \times M_0} \in \mathbb{R}^{N \times M_0}$

2- Set $Q = 0$ with $Q \in \mathbb{R}^{N \times M_0}$; $r = (\lambda_{\max} - \lambda_{\min})/2$;

For $e = 1, \ldots N_e$

compute $\theta_e = -\frac{\pi}{2}(x_e - 1)$,

compute $Z_e = (\lambda_{\max} + \lambda_{\min})/2 + r \exp(\im\theta_e)$,

solve $(Z_eB - A)Q_e = Y$ to obtain $Q_e \in \mathbb{C}^{N \times M_0}$

compute $Q = Q - (\omega_e/2) \Re\{ r \exp(\im\theta_e) Q_e \}$

End

3- Form $A_0Q_{M0 \times M_0} = Q^\dagger AQ$ and $B_0Q_{M0 \times M_0} = Q^\dagger BQ$

4- Solve $A_Q \Phi = \im\epsilon B_Q \Phi$ to obtain the $M_0$ eigenvalue $\epsilon_m$, and eigenvectors $\Phi_{M0 \times M_0} \in \mathbb{R}^{M0 \times M0}$

5- Set $\lambda_m = \epsilon_m$ and compute $X_{N \times M_0} = Q_{N \times M_0} \Phi_{M0 \times M_0}$

If $\lambda_m \in [\lambda_{\min}, \lambda_{\max}]$, $\lambda_m$ is an eigenvalue solution and its eigenvector is $X_m$ (the $m^{th}$ column of $X$).

6- Check convergence for the trace of the eigenvalues $\lambda_m$

If iterative refinement is needed, compute $Y = BX$ and go back to step 2

FIG. 2: FEAST pseudocode (sequential version) for solving the generalized eigenvalue problem $AX = \lambda BX$ where $A$ is real symmetric and $B$ is s.p.d., and obtaining all the M eigenpairs within a given interval $[\lambda_{\min}, \lambda_{\max}]$. The numerical integration is performed using $N_e$-point Gauss-Legendre quadrature with $x_e$ the $e^{th}$ Gauss node associated with the weight $\omega_e$. For the case $N_e = 8$, one can use:

$$(x_1, \omega_1) = (0.183434642495649, 0.362683738738361),$$

$$(x_2, \omega_2) = (0.25532409916328, 0.31370664587887),$$

$$(x_3, \omega_3) = (0.79666477413626, 0.222381034453374),$$

$$(x_4, \omega_4) = (0.96028956497536, 0.101228536290376),$$

and $$(x_{2i}, \omega_{2i})_{i=1,\ldots,4} = (-x_{2i-1}, \omega_{2i-1})$$

The performances of the basic FEAST algorithm will then depend on a trade off between the choices of the number of Gauss quadrature points $N_e$, the size of the subspace $M_0$, and the number of outer refinement loops. So far, using $M_0 \geq 1.5M$, $N_e = 8$, and with at most 2 refinement loops, we have consistently obtained a relative residual equal or smaller than $10^{-10}$ seeking up to 1000 eigenpairs for a variety of problems. The basic pseudocode for the FEAST algorithm is given in Figure 2 in the case of $A$ real symmetric. In the case of $A$ complex Hermitian, we note the following changes:
Y, Q ∈ \mathbb{C}^{N \times M_0}, \Phi ∈ \mathbb{C}^{M_0 \times M_0}, and the construction of the vectors Q in step-2 of the pseudocode must be modified to satisfy \cite{12}.

IV. NUMERICAL EXPERIMENTS

In this section, we propose to demonstrate the numerical stability, robustness and scalability of the FEAST algorithm using three examples derived from electronic structure calculations of Carbon nanotube (CNT).

A. Example I

Let us first consider a family of eigenvalue problems, Test-CNT, obtained using a 2D FEM discretization of the DFT/Kohn-Sham equations at a given cross-section of a (13,0) CNT – the 2D atomistic potential is derived from the mode approach used in the CMB strategy for solving the full 3D problem presented in \cite{9}. In Test-CNT, A is real symmetric, and B is s.p.d., the size of both matrices is N = 12, 450 and their sparsity pattern is identical with a number of non-zero elements nnz = 86, 808.

In Table I we report the times and relative residual obtained by the public domain eigenvalue solver package ARPACK \cite{14} (using the shift-invert strategy) and the FEAST algorithm presented in Figure 2 for solving the Test-CNT example seeking up to M = 800 (lowest) eigenpairs. The inner linear systems in ARPACK and FEAST are solved using the shared-memory parallel direct solver PARDISO \cite{15}. It should be noted, however, that FEAST benefits more than ARPACK from the PARDISO solver, as the inner linear systems have multiple right-hand sides. Although both algorithms could benefit from a parallel distributed implementation (e.g. using the P_ARPACK package), the simulation runs are here restricted to a given node of a 8-cores Intel Clovertown system (16Gb,2.66GHz) where the linear systems in FEAST are factorized and solved one after another. The performances of ARPACK and FEAST can also depend on fine tunings parameters such as the choices of the size of the subspace M_0 (M_0 = 1.5M here for both algorithms), the inner systems solvers, the number of contour points N_c for FEAST, or the stopping criteria for obtaining the residual. The simulation results in this section are then not intended to compare quantitatively the two solvers but rather to point out the potentialities of the FEAST algorithm.

In our experiments, the convergence criteria on the relative residual for FEAST is obtained when the relative error on the trace of the eigenvalues \sum m \lambda_m in the search interval is smaller or equal to 10^{-13}. Table I shows the variation of the relative error on the trace with the number of outer-iterative refinement for FEAST. These results demonstrate that only 2 to 3 refinement loops are necessary to obtain the small relative residuals for the different cases reported in Table I. It should be noted that only one loop is necessary to obtain the eigenvalues with an accuracy of \sim 10^{-5} or below.

| TEST-CNT | ARPACK | FEAST |
|----------|--------|-------|
| N = 12,450 | Time(s) | Resid. | Time(s) | Resid. |
| M = 100 | 12.2 | 2.0 \times 10^{-11} | 7.8 | 4.5 \times 10^{-10} |
| M = 200 | 31 | 2.0 \times 10^{-11} | 14 | 5.5 \times 10^{-10} |
| M = 400 | 86 | 1.4 \times 10^{-11} | 21 | 1.8 \times 10^{-10} |
| M = 800 | 213 | 4.5 \times 10^{-9} | 58 | 3.4 \times 10^{-11} |

B. Example II

In another set of numerical experiments, we intend to demonstrate the robustness of FEAST in capturing the
multiplicity of the eigenvalues. We propose to create artificially new TEST-CNT systems called \( k(N, M) \) where the matrices \( A \) and \( B \) are repeated \( k \) times along the main diagonal (the new system matrix is block diagonal with \( k \) blocks). Physically, these systems can describe the cross section of a bundle composed by \( k \) CNTs, where we do not consider the interactions between the different tubes such that each eigenvalue is now \( k \) times degenerate.

If we keep the same search interval used to obtain \( M = 100 \) eigenpairs for \( k = 1 \) (where the size of the matrices \( A \) and \( B \) is \( N \)), \( 100k \) eigenpairs must now be found for \( k \geq 1 \), where each one of them have the multiplicity \( k \).

In Table IV, we report the simulation times and relative residuals obtained using ARPACK and FEAST on these \( k(N, M) \) TEST-CNT systems. For the case \( 8(N, M) \), for example, the size of the new system matrix is 99,600 and the first 100 eigenvalues have all the multiplicity 8 (so 800 eigenpairs are found in total). The simulation results show linear scalability performances with the size of the system and the number of eigenpairs. In contrast to ARPACK where the number of matrix-vector multiplications and linear system solves would keep increasing with \( k \), the number of operations in FEAST stays the same for all these cases. The scalability of the algorithm depends then mainly on the scalability of the linear system solver.

### Table III: Performance results obtained by FEAST seeking \( M = 100 \) eigenpairs for different values of \( N \).

| \( N_e \) | Time(s) | Resid. | # loops |
|---------|---------|--------|---------|
| 4       | 7.0     | 8.3 \( \times 10^{-8} \) | 6       |
| 8       | 7.8     | 4.5 \( \times 10^{-10} \) | 4       |
| 16      | 10.2    | 3.4 \( \times 10^{-12} \) | 3       |

C. Example III

We have shown that FEAST can re-use the computed subspace as suitable initial guess for performing iterative refinements. This capability can also be of benefit to modern applications in science and engineering where it is often necessary to solve a series of eigenvalue problems that are close one another. In bandstructure calculations, in particular, many eigenvalue problems of the form \( (A + S_k)x_k = \lambda_kBx_k \) need to be solved at different locations in the \( k \)-space (i.e. for different values of \( \mathbf{k} \) and where \( S \) is Hermitian with \( S_0 = I \)). Let us consider the eigenvalue sparse system of size \( N = 492,982 \) obtained for a (5,5) metallic CNT using our in-house DFT/real-space mesh technique framework for bandstructure calculations of nanowires-type structure [16]. In Figure 3, we propose to solve this eigenvalue problem using the same search interval for the eigenvalues \( \lambda \) for different locations of \( \mathbf{k} \) where the subspace computed by FEAST at the point \( \mathbf{k} - 1 \) is successively used as initial guess for the neighboring point \( \mathbf{k} \). In addition, the inner linear systems in

![Figure 3: Bandstructure calculations of a (5,5) metallic CNT. The eigenvalue problems are solved successively for all the \( \mathbf{k} \) points (from \( \Gamma \) to \( \mathbf{X} \), while the computed subspace of size \( M_0 = 25 \) at the point \( \mathbf{k} \) is used as initial guess for the point \( \mathbf{k} + 1 \). The number of eigenvalues found ranges from 13 to 20, and by the third \( \mathbf{k} \) point, the FEAST convergence is obtained using only one refinement loop. The convergence is obtained with the relative error on trace of the eigenvalues smaller or equal to \( 10^{-8} \), while the inner linear systems are solved using an iterative method with an accuracy of \( 10^{-5} \). The final relative residuals on the eigenpairs range from \( 10^{-3} \) to \( 10^{-5} \). FEAST are solved using an iterative method with preconditioner where a modest relative residual of \( 10^{-3} \) is used (e.g. a suitable banded preconditioner can be obtained using a mode approach [2]). It should be noted that the convergence criteria for the relative error on the trace of the eigenvalues is chosen much smaller at \( 10^{-8} \), while the eigenvectors are expected to be obtained within the same (or a smaller) order of accuracy that the one used.

\[
\text{max}(|Ax - \lambda Bx|_1 ||Ax||_1)
\]

which artificially reproduce \( k \) times the original TEST-CNT system. The \( kM \) (lowest) eigenpairs are found where each eigenvalue has a multiplicity of \( k \).

![Diagram](image-url)
for the solutions of the inner systems. Figure 3 shows that 13 to 20 eigenvalues (i.e. energies) are found within the selected search interval along the different k points (from the Γ to the X point in the graph). Although the size of the subspace stays identical at M = 0, after the first initial point at k = 0 (Γ point in the graph) FEAST converges within only one refinement loop for almost all the other k points.

V. DISCUSSIONS

In comparison to iterative Krylov subspace techniques, FEAST can be cast as a "direct" technique which is based on an exact mathematical derivation \(3\). FEAST does naturally then capture all the multiplicity and no-orthogonalization procedure is necessary (such as Gram-Schmidt orthogonalization process). As described above, the main computational tasks in FEAST consist of solving \(N_0\) independent linear systems along the contour with \(M_0\) right-hand-sides and a reduced dense generalized eigenvalue problem of size \(M_0\). Since FEAST has the ability to re-use the basis from the previously computed subspace, an outer-iterative refinement procedure is proposed to improve the accuracy of the solutions. The capability to take advantage of suitable initial guess can also be of benefit to modern applications in sciences and engineering where it is often necessary to solve a series of eigenvalue problems that are close one another (e.g. Bandstructure calculations in Example III Section IV).

In one sense, the difficulty of solving an eigenvalue problem has been replaced by the difficulty of solving a linear system with multiple right-hand sides. For large sparse systems, this latter can be solved using either a direct system solver such as PARDISO \(15\) (as proposed in Section IV), or an iterative system solver with preconditioner. In turn, for banded systems or banded preconditioner, FEAST can be seen as an outer-layer for the author’s SPIKE parallel system solver \(17\). It should be noted that the inner linear systems arising from standard eigenvalue solvers (using the shift-strategy), need often to be solved highly accurately via direct methods. Direct system solvers, however, are not always suited for addressing large-scale modern applications because of memory requirements. In Example III of Section IV we have shown that FEAST can take advantage of iterative solvers for solving the inner linear systems with modest relative residual and obtaining the eigenvectors solution within the same order of accuracy. The resulting subspace could also be used as a very good initial guess for a one step more accurate refinement procedure (i.e. using more accurate relative residual for the inner systems).

FEAST exhibits important potentialities for parallelism at three different levels: (i) many search interval \([\lambda_{\min}, \lambda_{\max}]\) can be run independently, (ii) each linear systems can be solved simultaneously (e.g. on each node of parallel architecture where the factorization of the linear system can be done only once for all the refinement loops), (iii) the linear system solver can be parallel (e.g. within a given node as in Section IV). Depending on the parallel architecture at hand, the local memory of a given node and the properties of the matrices of the eigenvalue problems, one may preferably select one parallel option among the others, or just take advantage of a combination of those. In particular, there will be a trade off between how many search intervals to consider and how many eigenpairs FEAST can handle by intervals. For example if \(M_0\) is more than few thousands, one could either (i) solve the obtained reduced system of size \(M_0\) using efficient dense parallel symmetric eigenvalue solvers \(18\), or (ii) propose to divide the initial search interval into two or more to be processed in parallel. In addition, it should be noted that the orthonormalization step is absent from FEAST which will drastically reduce the communication overhead for performing scalar products on high-end parallel architectures (the scalar product in step-3 in Fig. 2 has to be done only once per iterative refinement). Given the recent advances in parallel architectures and parallel linear system solvers, it is reasonable to envision using FEAST in a near future for obtaining up to millions of eigenpairs of large sparse symmetric eigenvalue problems. Finally the capabilities of FEAST could potentially be enhanced for addressing non-symmetric eigenvalue problems where the contour integration would then be performed in a given region of the complex space.

Acknowledgments

The author wishes to acknowledge helpful discussions with Dr. Ahmed Sameh and Dr. Massimo Fischetti. This material is supported by NSF under Grant #CCF-0635196.

[1] G. Golub and H. A. van der Vorst, J. of Comput. and Appl. Math. 123, 35 (2000).
[2] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst, Templates for the solution of Algebraic Eigenvalue Problems: A Practical Guide (Society for Industrial and Applied Mathematics, Philadelphia, PA, 2000).
[3] T. Torsti, M. Heiskanen, M. J. Puska, and R. M. Nieminen, Int. J. Quantum Chem 91, 171 (2003).
[4] Y. Saad, Y. Zhou, C. Bekas, M. L. Tiago, and J. Chelikowsky, physica status solidi (b) 243, 2188 (2006).
[5] Y. Zhou, Y. Saad, M. L. Tiago, and J. Chelikowsky, Phys. Rev. E. 74, 066704 (2006).
[6] R. Zeller, J. Deutz, and P. Dederichs, Solid state communications 44, 993 (1982).
[7] J. Taylor, H. Guo, and J. Wang, Phys. Rev. B. 63, 245407 (2001).
[8] M. Brandbyge, J.-L. Mozos, P. Ordejon, J. Taylor, and K. Stokbro, Phys. Rev. B. 65, 165401 (2002).
[9] D. Zhang and E. Polizzi, J. Comput. Elec. 7, 427 (2008).
[10] D. Zhang and E. Polizzi, 2008 NSTI Nanotechnology Conference and Trade Show. Technical Proceedings. 1, 12 (2008).
[11] T. Sakurai and H. Sugiura, J. of Comput. and Appl. Math. 159, 119 (2003).
[12] T. Sakurai, Y. Kodaki, H. Tadano, D. Takahashi, M. Sato, and U. Nagashima, Future Generation Computer Systems 24, 613 (2008).
[13] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. D. C. J. Dongarra, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, LAPACK Users Guide (Society for Industrial and Applied Mathematics, Philadelphia, PA, 1999), third ed. ed.
[14] R. B. Lehoucq, D. C. Sorensen, and C. Yang., ARPACK Users Guide: Solution of Large Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods. (Society for Industrial and Applied Mathematics, Philadelphia, PA, 1998).
[15] O. Schenk and K. Gärtner, Journal of Future Generation Computer Systems 20, 475 (2004).
[16] D. Zhang and E. Polizzi, in preparation (2009).
[17] E. Polizzi and A. Sameh, Parallel Computing 32, 177 (2006).
[18] L. Blackford, J. Choi, A. Cleary, E. D’Azevedo, J. Demmel, I. Dhillon, J. Dongarra, S. Hammarling, G. Henry, A. Petitet, et al., ScaLAPACK Users Guide (Society for Industrial and Applied Mathematics, Philadelphia, PA, 1997).