Solving complex band structure problems with the FEAST eigenvalue algorithm

S. E. Laux*

Semiconductor Research and Development Center (SRDC), IBM Research Division, T. J. Watson Research Center, P.O. Box 218, Yorktown Heights, New York 10598-0218, USA

(Received 13 June 2012; published 3 August 2012)

With straightforward extension, the FEAST eigenvalue algorithm [Polizzi, Phys. Rev. B **79**, 115112 (2009)] is capable of solving the generalized eigenvalue problems representing traveling-wave problems—as exemplified by the complex band-structure problem—even though the matrices involved are complex, non-Hermitian, and singular, and hence outside the originally stated range of applicability of the algorithm. The obtained eigenvalues/eigenvectors, however, contain spurious solutions which must be detected and removed. The efficiency and parallel structure of the original algorithm are unaltered. The complex band structures of Si layers of varying thicknesses and InAs nanowires of varying radii are computed as test problems.

DOI: 10.1103/PhysRevB.86.075103 PACS number(s): 02.60.−x, 02.70.Hm, 31.15.−p

I. INTRODUCTION

The FEAST algorithm of Polizzi[1] is a novel and highly efficient means to solve generalized eigenvalue problems (GEPs) $\boldsymbol{A}x = \lambda \boldsymbol{B}x$, derived for square matrices $\boldsymbol{A}$ (real symmetric or complex Hermitian) and $\boldsymbol{B}$ (real symmetric positive definite). The algorithm essentially replaces the difficulty of solving an eigenvalue problem with the difficulty of solving linear systems with multiple right-hand sides. Optionally, the algorithm benefits significantly from a suitable initial guess which accelerates, for example, band structure calculations which seek solutions for incrementing energy. Eigenvalue multiplicity is naturally captured and no orthogonalization procedure is required. The algorithm can be readily parallelized. Only those eigenvalues residing within a given contour in the solution space are obtained. All of these features are well documented in the original paper,[1] but here we consider the question of whether the algorithm can be used with matrices $\boldsymbol{A}$, $\boldsymbol{B}$—complex, non-Hermitian and singular—obtained from traveling-wave problems such as the complex band structure eigenvalue problem.

The question has a favorable answer: every desirable feature of the original algorithm is retained. The original algorithm need only be modified in straightforward ways. Unfortunately, in the specific case considered here the algorithm can produce spurious solutions. Fortunately, the spurious eigenvalues can be detected and discarded efficiently, leaving behind the desired solutions.

The outline of this paper is as follows. Section II clarifies the properties of matrices $\boldsymbol{A}$, $\boldsymbol{B}$ considered here. In Sec. III, the changes to the original FEAST algorithm are summarized. Section IV presents calculation examples to demonstrate that the favorable features of the original algorithm are retained. Section V discusses the performance of the algorithm. Conclusions are found in Sec. VI. Notation will follow Ref. 1 wherever possible; any undefined symbol can be resolved there.

II. PROPERTIES OF *A* AND *B* MATRICES

The essential property to be exploited in solving the GEP $\boldsymbol{A}x = \lambda \boldsymbol{B}x$ representing the complex band problem is that the eigenvalues must come in quadruples. Specifically, the complex band structure problem $\boldsymbol{A}x = \lambda \boldsymbol{B}x$ as formulated by Boykin[2] reads

$$\begin{bmatrix} H_{(\ell,\ell)} & H^{\dagger}_{(\ell,\ell+1)} \\ \mathbf{1} & \mathbf{0} \end{bmatrix} \begin{bmatrix} C_{\ell} \\ C_{\ell-1} \end{bmatrix} = \lambda \begin{bmatrix} -H_{(\ell,\ell+1)} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \begin{bmatrix} C_{\ell} \\ C_{\ell-1} \end{bmatrix}, \tag{1}$$

where ℓ indexes repeating unit cells in the propagation direction, $H_{(\ell,\ell)}$ is the Hamiltonian within the unit cell ℓ, $H_{(\ell,\ell+1)}$ is the coupling between unit cells ℓ and ℓ+1, and C_{ℓ}, $C_{\ell-1}$ are the expansion coefficients in unit cells ℓ and ℓ–1, respectively. Boykin shows that eigenvalues come in quadruples,

$$\lambda_i = 1/\lambda_j = \lambda_k^* = 1/\lambda_l^* \text{, with } i,j,k,l \text{ all different,} \tag{2}$$

corresponding to the four combinations of forward and backward propagation of growing and decaying waves. Note that $H_{(\ell,\ell+1)}$ is generally extremely sparse and likely has many columns which are all zero. As a result, $\boldsymbol{A}$, $\boldsymbol{B}$ can be singular non-Hermitian matrices even if the Hamiltonian matrix $H_{(\ell,\ell)}$ itself is Hermitian.

The formulation of the complex eigenvalue problem summarized above is completely analogous to the "open boundary condition" problem[3] and hence the FEAST algorithm as extended here can be used to obtain such traveling-wave solutions as well. In this paper, only examples drawn from the complex band structure problem will be discussed.

III. EXTENDING THE FEAST ALGORITHM

A. Subspaces, spurious solutions, and contour determination

Figure 2 of Ref. 1 lists the six steps of the FEAST algorithm, herewith termed the "single subspace" (SS) algorithm. A "dual subspace" (DS) algorithm is appropriate for the non-Hermitian, complex problem, and is summarized in Fig. 1. The DS moniker derives from the need to simultaneously solve the given problem $\boldsymbol{A}|x\rangle = \lambda \boldsymbol{B}|x\rangle$ and the adjoint problem $\boldsymbol{A}^{\dagger}|\hat{x}\rangle = \hat{\lambda}\boldsymbol{B}^{\dagger}|\hat{x}\rangle$ in order to build dual vector spaces which are $\boldsymbol{B}$ biorthonormal, i.e., $\langle \hat{x}_i|\boldsymbol{B}|x_j\rangle = \delta_{i,j}$, where $\delta_{i,j}$ is the Kronecker δ and Dirac notation has been adopted.[4] Note that throughout the circumflex (ˆ) is used to denote a term arising from the adjoint problem. Corresponding to (3) of Ref. 1 is the

1. Select $M_0 > M$ random vectors $\boldsymbol{Y}_{N\times M_0}$, $\hat{\boldsymbol{Y}}_{N\times M_0} \in \mathbb{C}^{N\times M_0}$
1′. Normalize columns of $\boldsymbol{Y}$, $\hat{\boldsymbol{Y}}$
2. Set $\boldsymbol{Q}_{N\times M_0} = \boldsymbol{0}$, $\hat{\boldsymbol{Q}}_{N\times M_0} = \boldsymbol{0}$
   Fix N_e quadrature points with nodes z_e, weights ω_e
   For $e = 1, ..., N_e$
      solve $(z_e\boldsymbol{B} - \boldsymbol{A})\boldsymbol{Q}_e = \boldsymbol{Y}$ to obtain $\boldsymbol{Q}_e \in \mathbb{C}^{N\times M_0}$
      $\boldsymbol{Q} = \boldsymbol{Q} + \omega_e\boldsymbol{Q}_e$
      solve[a] $(z_e\boldsymbol{B}^\dagger - \boldsymbol{A}^\dagger)\hat{\boldsymbol{Q}}_e = \hat{\boldsymbol{Y}}$ to obtain $\hat{\boldsymbol{Q}}_e \in \mathbb{C}^{N\times M_0}$
      $\hat{\boldsymbol{Q}} = \hat{\boldsymbol{Q}} + \omega_e\hat{\boldsymbol{Q}}_e$
   End
3. Form $\boldsymbol{A}_{\mathbb{Q}_{M_0\times M_0}} = \hat{\boldsymbol{Q}}^\dagger\boldsymbol{A}\boldsymbol{Q}$, $\boldsymbol{B}_{\mathbb{Q}_{M_0\times M_0}} = \hat{\boldsymbol{Q}}^\dagger\boldsymbol{B}\boldsymbol{Q}$
4. Solve $\boldsymbol{A}_{\mathbb{Q}}\boldsymbol{\Phi} = \epsilon\boldsymbol{B}_{\mathbb{Q}}\boldsymbol{\Phi}$ to obtain M_0 eigenvalues ϵ_m and eigenvectors $\boldsymbol{\Phi}_{M_0\times M_0} \in \mathbb{C}^{M_0\times M_0}$
   Solve[b] $\boldsymbol{A}^\dagger_{\mathbb{Q}}\hat{\boldsymbol{\Phi}} = \hat{\epsilon}\boldsymbol{B}^\dagger_{\mathbb{Q}}\hat{\boldsymbol{\Phi}}$ to obtain M_0 eigenvalues $\hat{\epsilon}_m$ and eigenvectors $\hat{\boldsymbol{\Phi}}_{M_0\times M_0} \in \mathbb{C}^{M_0\times M_0}$
5. Set $\lambda_m = \epsilon_m$ and compute $\boldsymbol{X}_{N\times M_0} = \boldsymbol{Q}_{N\times M_0}\boldsymbol{\Phi}_{M_0\times M_0}$
   Set $\hat{\lambda}_m = \hat{\epsilon}_m$ and compute $\hat{\boldsymbol{X}}_{N\times M_0} = \hat{\boldsymbol{Q}}_{N\times M_0}\hat{\boldsymbol{\Phi}}_{M_0\times M_0}$
   λ_m within the contour are *trial* solutions to the original problem, with eigenvector $\boldsymbol{X}_m$ (the mth column of $\boldsymbol{X}$)
   $\hat{\lambda}_m$ within the contour are *trial* solutions to the adjoint problem, with eigenvector $\hat{\boldsymbol{X}}_m$ (the mth column of $\hat{\boldsymbol{X}}$)
6. Check convergence. If iterative refinement needed, compute $\boldsymbol{Y} = \boldsymbol{B}\boldsymbol{X}$, $\hat{\boldsymbol{Y}} = \boldsymbol{B}^\dagger\hat{\boldsymbol{X}}$ and go back to step 1′.

[a] The matrix factorization $LU = (z_e^*\boldsymbol{B} - \boldsymbol{A})$, available from the preceding factorization step since both z_e and z_e^* are quadrature points, may be reused here to solve the equivalent $(LU)^\dagger\hat{\boldsymbol{Q}}_e = \hat{\boldsymbol{Y}}$ if supported.
[b] This is equivalent to the left-handed eigenvalue problem $\hat{\boldsymbol{\Phi}}^\dagger\boldsymbol{A}_Q = \hat{\epsilon}^*\hat{\boldsymbol{\Phi}}^\dagger\boldsymbol{B}_Q$. Routines like zggev[Ref. 15] would permit this left-handed eigenvalue problem to be combined with the preceding right-handed eigenvalue problem.

FIG. 1. Dual subspace FEAST algorithm for solving the generalized eigenvalue problem $\boldsymbol{A}x = \lambda\boldsymbol{B}x$, with $\boldsymbol{A}$, $\boldsymbol{B}$ $N \times N$ complex, non-Hermitian matrices. Compare to the single subspace algorithm for real symmetric or Hermitian complex matrices given in Fig. 2 of Ref. 1. A circumflex denotes (ˆ) objects unique to the dual subspace algorithm.

DS factorization

$$\boldsymbol{\rho} = \frac{1}{2\pi\imath}\int_{\mathcal{C}} dz\,\boldsymbol{G}(z) = \sum_{i=1}^{M}\frac{|x_i\rangle\langle\hat{x}_i|}{\langle\hat{x}_i|\boldsymbol{B}|x_i\rangle}, \tag{3}$$

where eigenvalues λ_i, $i = 1,2,\ldots,M$, are interior to the contour $\mathcal{C}$, indices are arranged so that $\lambda_i = \hat{\lambda}_i^*$ (*denotes the complex conjugate), and $|x_i\rangle$, $|\hat{x}_i\rangle$ are unnormalized. Equation (3) derives, in turn, from[5]

$$\boldsymbol{G}(z) \equiv (z\boldsymbol{B} - \boldsymbol{A})^{-1} = \sum_{i=1}^{N}\frac{|x_i\rangle\langle\hat{x}_i|}{(z-\lambda_i)\langle\hat{x}_i|\boldsymbol{B}|x_i\rangle}, \tag{4}$$

where N is the order of the problem. Note that deriving (4) assumes the $|x_i\rangle$ form a complete basis.[5] This is not formally true in the present case as follows. For most Hamiltonians, $H_{(\ell,\ell+1)}$ of Sec. II has multiple columns of zeros, as is the case for the tight-binding Hamiltonian[6,7] adopted here, and as a result $\boldsymbol{A}|x\rangle = \lambda\boldsymbol{B}|x\rangle$ will prove to have both $\lambda = 0$ and

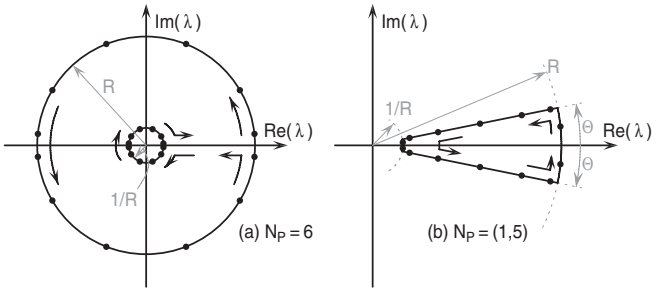

FIG. 2. Two contours in the complex λ plane. (a) A multiply connected domain described by the points between two concentric circles at the origin of radii R and $1/R$. Integration proceeds in a counter-clockwise direction along the outer circle and clockwise along the inner circle so that FEAST will resolve the eigenvalues between the circles. This contour allows $|\mathrm{Im}(k)|$ to be bounded. As shown there are $N_P = 6$ Gauss-Legendre quadrature points along each *semicircle* $\mathrm{Im}(\lambda) \geqslant 0$ and $\mathrm{Im}(\lambda) \leqslant 0$. The total number of quadrature points is $N_e = 4N_P = 24$. Contours of this type are denoted by a single value N_P. (b) A simply connected domain obtained from (a) after introducing the addition angular constraint $\mathrm{Im}(\lambda)/\mathrm{Re}(\lambda) < |\tan\Theta|$. As shown there are $N_{P1} = 1$ quadrature points in the angular direction and $N_{P2} = 5$ quadrature points in the radial direction *in upper and lower half planes*; the total number of quadrature points is $N_e = 4N_{P1} + 2N_{P2} = 14$. Contours of this type are denoted by the ordered pair $N_P = (N_{P1}, N_{P2})$.

$\lambda = \pm\infty$ (unphysical) solutions; the infinite eigenvalues preclude a complete basis. However, we never seek these eigenvalues; $\mathcal{C}$ never includes these eigenvalues in its interior. And so, without rigorous proof but supported by extensive experience, the FEAST process fortuitously carries through and obtains any desired (finite) eigenvalues without breakdown.

The DS algorithm of Fig. 1 reveals two changes (cf. Fig. 2 of Ref. 1): (i) analogous operations for the adjoint system have been added to every step; (ii) a normalization step 1′ must be introduced.

Note the suggestion in Fig. 1 that only *trial* solutions are identified. Experience shows this to be the case in our problem, as spurious eigenvalues can reside within the contour at convergence. This is likely the result of the lack of a complete basis in the theory as already mentioned, but regardless, it can be handled quite easily and expeditiously. Since it is known (2) that eigenvalues come in quadruples, define $\mathcal{C}$ so as to include them all. Then, genuine eigenvalues can be identified by the presence of their membership in a quadruple. In practice, only pairings are tested. Hence each λ_i pairs with a sufficiently close $1/\lambda_j$ or $1/\lambda_k^*$ (whichever is closer) or remains unpaired. This *ad hoc* procedure has proven extremely robust and becomes insensitive to the exact pairing tolerance adopted at convergence. In addition, convergence can be determined by testing just how close paired eigenvalues are. This idea is now made precise.

Eigenvalues λ_i and λ_j are considered paired by condition $\mathcal{P}(\delta)$ if

$$P(\lambda_i,\lambda_j) \equiv \frac{\min(|\lambda_i - 1/\lambda_j|, |\lambda_i - 1/\lambda_j^*|)}{|\lambda_i|} < \delta \quad i \neq j, \tag{5}$$

and unpaired otherwise. Note that pair identification occurs *in a very specific order*: the first pair designated minimizes $P(\lambda_i, \lambda_j)$ for all $i \neq j$, the second pair minimizes $P(\lambda_i, \lambda_j)$ for all unpaired $i \neq j$, etc., until pairing ceases because (5) is no longer true for any remaining $i \neq j$. The strictness of this pairing is determined by the tolerance δ. Note that $P(\lambda_i, \lambda_j) = P(\lambda_j, \lambda_i)$. $P(\lambda_i, \lambda_j)$ measures the relative "pairing distance" between λ_i and $1/\lambda_j$ or $1/\lambda_j^*$, and grows smaller as they grow closer.

Convergence can be determined by so-called "pair-δ convergence," which occurs when every eigenvalue within the given contour is either labeled unpaired according to $\mathcal{P}(\delta_1)$, $\delta_1 \sim 10^{-3}$, or paired according to $\mathcal{P}(\delta_2)$ for the convergence criterion $\delta_2 \ll \delta_1$. Pairing, pair-δ convergence, and spurious eigenvalues are elaborated upon in Sec. III B. While convergence obtained by monitoring the trace of all eigenvalues within the contour[1] also continues to be tested, it is rarely achieved because spurious eigenvalue(s) within the contour never settle to a limiting value during FEAST iterations. The "pair-δ error" which is monitored for convergence is simply the maximum of $P(\lambda_i, \lambda_j)$ for all eigenvalues paired within the contour according to $\mathcal{P}(\delta_1)$. By convention, no pairing is done when starting from a random initial guess. As a result, pair-δ convergence is only possible at the second or later iteration.

In this paper, the FEAST contour is set in order to seek a "few" propagating solutions at a given energy which have the smallest nonzero $|\mathrm{Im}(k)|$. These are the evanescent states which spatially decay or grow most slowly. For energies within the band gap of a semiconductor, these states are of greatest interest as they represent states with the greatest tunneling probabilities. Given this goal, corresponding bounds on the eigenvalue λ must be prescribed. Recall that the relationship between $\lambda \equiv re^{i\theta}$ and k is[2]

$$\lambda = e^{ikL} \quad \rightarrow \quad k = -i\ln(\lambda)/L = \theta/L - i\ln(r)/L\,, \quad (6)$$

where L is the distance between repeating unit cells in the propagating direction.[2] FEAST resolves only eigenvalues within a contour defined in the complex λ plane. Figure 2 shows two contours relevant to the task at hand. The contour of Fig. 2(a) resolves solutions with the smallest $|\mathrm{Im}(k)|$; specifically, $|\mathrm{Im}(k)| < \ln(R)/L$. $\mathrm{Re}(k)$ is constrained as $|\mathrm{Re}(k)| < \pi/L$; however, after postprocessing k to unfold the Brillouin zone,[8,9] $\mathrm{Re}(k)$ in fact occupies the entire zone. To integrate around the area $1/R < r < R$, integration proceeds counterclockwise along $r = R$ and clockwise around $r = 1/R$ as indicated. Contours of this type are referred to by a single value of N_P which represents the number of quadrature points *on every semicircle*; here $N_P = 6$. The total number of quadrature points is therefore $N_e = 4N_P$. A second contour has proven even more useful for this work, however.

Figure 2(b) shows a similar contour but with limited angular extent, i.e., $-\Theta < \theta < \Theta$. This leaves unresolved states with large $\mathrm{Re}(k)$ which prove unnecessary for computing the loci of complex k corresponding to the path of least action in the band gap.[10] Figure 2(b) depicts one quadrature point along the $r =$ const boundary and five quadrature points along the $\theta =$ const boundaries in the upper *and* lower half planes. Contours of this type are referred to here by an ordered pair $N_P = (N_{P1}, N_{P2})$; as shown $N_P = (1,5)$. The total number of quadrature points is $N_e = 4N_{P1} + 2N_{P2}$.

Note that for either contour, it might seem clever to compute over a smaller region: either the portion $r > 1$ or $r < 1$. But this is not feasible as it undermines the $\mathcal{P}(\delta)$ operation (5) proposed since all quadruple values (2) would not be within the contour.

B. Practical considerations

Besides the matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, the FEAST algorithm requires selecting the integration contour [Fig. 2(b) is considered throughout], the number of the quadrature points and the size of the subspace M_0 as defined in Ref. 1. Thus R, Θ, M_0, and $N_P = (N_{P1}, N_{P2})$ must be determined. Creating a robust eigenvalue solver depends on the quality of these selections. Care must be taken here, and the experiences related are specific to the complex eigenvalue problem. While it is expected that the general principles will apply broadly, this is merely hoped and has not been demonstrated.

FEAST generally proceeds to convergence, frequently in just a few iterations. It can be shown that, if a complete basis exists, and if contour integrations of the form (3) can be performed exactly, FEAST will always converge in one iteration (see the Appendix). This can even be observed, especially if a good initial guess is available for the computation. In practice, and depending on the quality of the selections made—to be subsequently discussed—FEAST will either (i) converge to fewer eigenvalues than desired; (ii) converge to more eigenvalues than desired; or (iii) stall. In the first case, the contour is enlarged and the computation restarted. In the second case, the contour is shrunk should subsequent computation be desired. The interesting third case is a signal that the contour and M_0 are not in sync. That is, the contour encloses more than M_0 eigenvalues. Either the contour is shrunk, M_0 is increased, or both. In the rare case where this fails to produce convergence, any initial guess is discarded and the algorithm is restarted from scratch with a random initial guess. Note here any change in contour size is done via R alone (Θ fixed) for simplicity.

Polizzi's recommendation is $M_0 \gtrsim 1.5M$, where M is the number of eigenvalues within the contour (paired or not). Since M is not known *a priori*, the algorithm implemented here includes the ability to adjust M_0 as a part of step 6 in Fig. 1 (but not explicitly shown there). Of course, M_0 is only adjusted if its value is to change significantly to avoid needless churn. The danger of dynamically adjusting M_0 is seen in Fig. 3; it is possible to erroneously conclude that there are no eigenvalues within the contour if the initial value of M_0, M_0^{INIT} is too small. Convergence indicating no eigenvalues within the contour may need to be viewed skeptically; M_0 and/or R may need to be increased, and any initial guess abandoned.

Here, FEAST is invoked for increments in energy to create a complex band structure diagram. After the first solution, R is adjusted by extrapolating from prior R value(s), using a constant, linear, parabolic or optimum-order diagonal rational[11] extrapolation scheme (depending on the depth of prior values). R is adjusted after every converged FEAST invocation. While not a part of FEAST *per se*, R selection demands care for large problems with tightly spaced eigenvalues. A slight overestimate of R can mistakenly challenge the algorithm to resolve many eigenvalues, thwarting convergence.

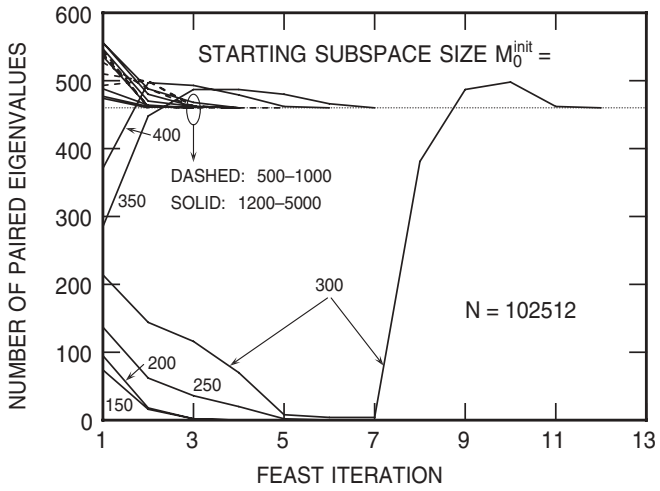

FIG. 3. Number of paired eigenvalues versus FEAST iteration for various values of the starting subspace size M_0^{INIT}. The same problem, contour and quadrature points are used throughout. Choosing $M_0^{\text{INIT}} \geqslant 300$ ultimately obtains the correct solution of 460 paired eigenvalues (indicated with a thin, horizontal dashed line). Below this, the algorithm erroneously converges to find zero paired eigenvalues. This possibility must be detected and corrected by restarting the algorithm with a larger M_0^{INIT} value. Here, the problem considered is a Si layer 640 atomic layers thick at a midgap energy. The contour is given by Fig. 2(b) with $R = 2.4597$ and $\Theta = 2\pi/100$. Quadrature points used are $N_P = (3,11)$. The matrices $\boldsymbol{A}, \boldsymbol{B}$ are of order $N = 102\,512$.

Last, the Gauss-Legendre quadrature order $N_P = (N_{P1}, N_{P2})$ must be chosen. One can surmise that a larger R would demand large N_{P1}, N_{P2} to maintain constant integration accuracy, and this is indeed the case.

It is possible to establish a qualitative understanding of the relationship between M_0 and N_P as follows. Consider quadrature of order N_e with weights w_e and locations z_e. Then, evaluating (3) and assuming (4) holds,

$$\boldsymbol{\rho} \approx \sum_{e=1}^{N_e} w_e \boldsymbol{G}(z_e) = \sum_{e=1}^{N_e} w_e \sum_{i=1}^{N} \frac{|x_i\rangle\langle\hat{x}_i|}{(z_e - \lambda_i)\langle\hat{x}_i|\boldsymbol{B}|x_i\rangle}$$
$$= \sum_{i=1}^{N} \left[\sum_{e=1}^{N_e} \frac{w_e}{(z_e - \lambda_i)} \right] \frac{|x_i\rangle\langle\hat{x}_i|}{\langle\hat{x}_i|\boldsymbol{B}|x_i\rangle}, \tag{7}$$

where the quantity in brackets defines a *selection function* $S(\lambda_i)$, which depends only on the contour and quadrature scheme adopted. $S(\lambda_i)$ is seen to be a weighting factor applied to the ith term of the sum. If the contour integral were performed exactly, $S = 1$ if λ_i is within $\mathcal{C}$ and $S = 0$ otherwise. As defined, S represents precisely how this FEAST eigenvalue selection process is approximated by the quadrature scheme adopted. As shown in Fig. 4, S does a credible job of approximating the exact result, with two caveats: (i) S does not fall to zero abruptly outside $\mathcal{C}$; and (ii) S is not exactly unity within $\mathcal{C}$. The consequences of (i) is that M_0 must be as large as the number of convergent eigenvalues within $\mathcal{C}$ and its surrounding fringe. As will be seen in Sec. V, S does indeed modulate convergency of each eigenvalue $i = 1, \ldots, N$: for $|S| \approx 1$, maximum error reduction per iteration is obtained, and for $|S| \ll 1$, no error reduction results. Increasing N_e sharpens this transition thereby enclosing fewer eigenvalues within $\mathcal{C}$ and its now smaller fringe which allows M_0 to be decreased. The consequences of (ii) seem ominous: should eigenvalues accidentally reside too close to quadrature points, $|S| \to \infty$. However, in our experience, this proves an elusive failure to witness. Occasionally, a given $N_P = (N_{P1}, N_{P2})$ might provide disappointing convergency, while $N_P' = (N_{P1} \pm 1, N_{P2} \pm 1)$ yields better behavior. These rare occurrences have always been viewed as owing to this effect, but have not been otherwise investigated. It has just not been sufficiently vexing to motivate an inquiry.

One final choice involves setting the tolerance δ_1 for $\mathcal{P}(\delta_1)$ in testing for pair-δ convergence. In Sec. III A the notion of using a relatively large tolerance δ_1 to define paired and

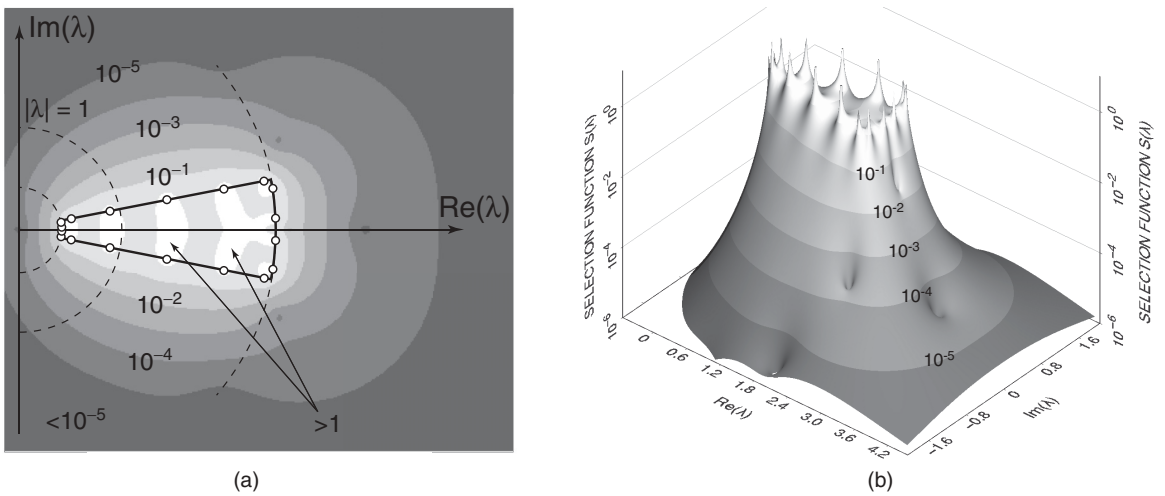

FIG. 4. The magnitude of the selection function $|S(\lambda)|$ for a contour with $R = 2.5$, $\Theta = 0.2$ and $N_P = (2,5)$. (a) $|S|$ values are shown in gray scale contours. The unit circle $|\lambda| = 1$ is shown dashed for reference, as are portions of the circles $|\lambda| = R$ and $|\lambda| = 1/R$. Within the contour, $|S| \approx 1$; regions where $|S| > 1$ can be seen. (b) The quadrature points are also the pole locations of S, as seen in the definition of S in (7). This can be inferred from the upward spikes present in the surface. Note that logarithmic scaling is used in both depictions.

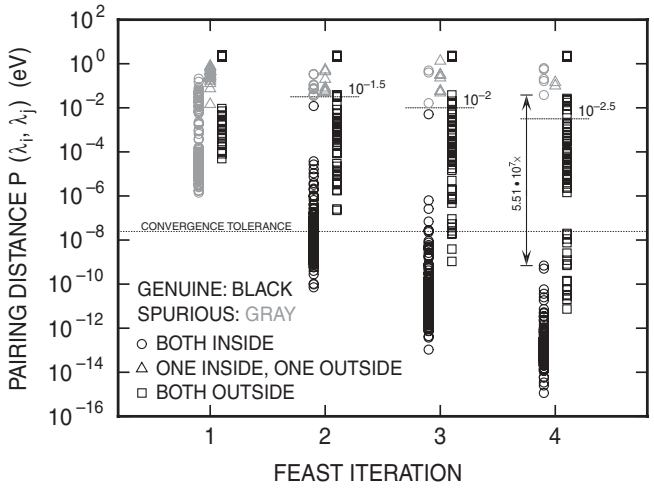

FIG. 5. Pairing distance $P(\lambda_i,\lambda_j)$ versus FEAST iteration for the same problem as Fig. 3 at a fixed energy of 0.598 eV (near midgap) with subspace size $M_0 = 800$ fixed. At each iteration, all eigenvalues are paired by invoking $\mathcal{P}(\infty)$. Each symbol represents a pair of eigenvalues with the ordinate giving their proximity to each other. Symbols are shaded, however, as to whether they should have been labeled genuine (black) or spurious (gray) according to $\mathcal{P}(\delta_1)$ where δ_1 varies with iteration as described in Sec. III B. The value of δ_1 is indicated by a horizontal line for iterations 2–4. (Note: spurious eigenvalues are unpaired eigenvalues within the contour.) At each iteration, circles, triangles, and squares indicate whether eigenvalue pairs have two, only one, or no eigenvalues inside the given integration contour. Convergence is obtained at the fourth iteration: all genuine pairs within the contour (black circles) are below the convergence tolerance, while all spurious pairs within or straddling the contour (gray) are far away. The plot clearly shows (i) the value for δ_1 proves not critical. Spurious and genuine populations are separated by $>10^7\times$ at convergence. (ii) An apparently mislabeled pair at iterations 2 and 3 (a black circle where gray might have been favored) proves ultimately irrelevant. Also note that eigenvalue pairs outside the integration contour span a more continuous range of $P(\lambda_i,\lambda_j)$, unlike the pairs inside the contour.

unpaired eigenvalues was introduced. Choosing this tolerance proves noncritical, as demonstrated in Fig. 5. Here, for the same problem considered in Fig. 3, convergence behavior at a single midgap energy 0.598 eV is examined. For this plot only, all eigenvalues are paired via $\mathcal{P}(\infty)$. Thus each symbol indicates a pair of eigenvalues ($\lambda_i, \lambda_j \approx 1/\lambda_i$ or $1/\lambda_i^*$) with the ordinate giving the pairing distance P between them. However, pairs considered unpaired according to $\mathcal{P}(\delta_1)$ (with δ_1 described subsequently) and with at least one eigenvalue in the contour are colored gray; otherwise, pairs are black. Gray shading indicates a spurious eigenvalue, defined as *any eigenvalue in the contour which is unpaired according to* $\mathcal{P}(\delta_1)$. Circles, triangles, and squares indicate that two, only one, or no eigenvalues of the pair reside within the integration contour. Throughout this work, δ_1 is chosen as follows: when no restart subspace matrices $\boldsymbol{Y}$, $\hat{\boldsymbol{Y}}$ are available, $\delta_1 = \infty$, $10^{-1.5}$, 10^{-2}, $10^{-2.5}$, and 10^{-3} are chosen in turn for the first five iterations. For all subsequent invocations, $\delta_1 = 10^{-4}$. When initial guesses for $\boldsymbol{Y}$, $\hat{\boldsymbol{Y}}$ are available, $\delta_1 = 10^{-3}$ for all iterations. Figure 5 shows pair-δ convergence is obtained at the fourth iteration (convergence criterion $\delta_2 = 2 \times 10^{-8}$) and with a substantial separation ($\sim 10^7\times$) between genuine and spurious eigenvalue pairs at convergence. The precise value for δ_1 at convergence proves not critical given this large separation. Note how one eigenvalue pair within the contour (black circle) at the second and third iteration would better have been labeled spurious (gray), but that this has no effect on the ultimate outcome of the algorithm. Finally, note how the eigenvalue pairs outside the contour span a more continuous range of pair-δ values. But then, they are outside the contour and should not really be resolved at all.

IV. EXAMPLES

Examples are drawn from a tight-binding formulation of the semiconductor complex band structure problem.[2] The first problem consists of determining the complex band structure of a [001] Si layer of variable thickness t, with transport in the $\langle 100\rangle$ direction. Tight-binding parameters are taken from Ref. 7. The second problem consists of determining the complex band structure of an InAs nanowire of variable radius r, with transport in the $\langle 110\rangle$ direction. Tight-binding parameters are taken from Ref. 12. In both cases, dangling bonds are terminated with hydrogen,[13] spin orbit interaction is included, and bond lengths are fixed at their bulk values.

Figure 6 shows both physical and numerical results for two Si layers, one $10a$ thick and one $1280a$ thick ($a = 0.543$ nm). A total of 11 layer thicknesses spanning these extremes were examined; the results shown here are fully indicative of all results obtained. The loci of solutions tracing out the path of least action for band-to-band tunneling[10] has been successfully obtained, meeting the goal set out in the design of the Fig. 2(b) contour. In Fig. 6(a), the thin layer shows minimal quantization, with a $\sim$1.18-eV band gap, while in Fig. 6(b), the thick layer essentially obtains the bulk 1.12-eV band gap. With rather mundane results physically, a "bulklike" Si layer proves a challenging computational test given that wave vectors are very tightly spaced at a given energy. FEAST is able to extract just a few eigenvalues from a near continuum of states. Very provocative is the observation that FEAST converges in one iteration within the band gap. This occurs because a good initial guess was available, the heuristic to predict the next contour size was adequately accurate, and enough quadrature points were employed. Note that more quadrature points needed to be used for Fig. 6(b) in the upper part of the band gap, because of the extremely small eigenvalue spacings there. Also reported in Fig. 6 are the number of quadrature points used and the number of eigenvalues found at each energy (see caption).

Figure 7 shows both physical and numerical results for two InAs nanowires, one of radius a and one of radius $12a$ ($a = 0.605\,83$ nm). A total of nine radii spanning these extremes were examined; again the results shown here are fully indicative of all results obtained. In contrast to the layer problem, significant quantization is clearly visible here. In Fig. 7(a), the smaller wire has a $\sim$2.7-eV band gap; in Fig. 7(b), the larger wire has a $\sim$0.50-eV band gap. This compares to a bulk InAs band gap of 0.37 eV.[12] As a result, eigenvalues are more widely spaced and relatively fewer quadrature points are required to still obtain outstanding convergency. Again,

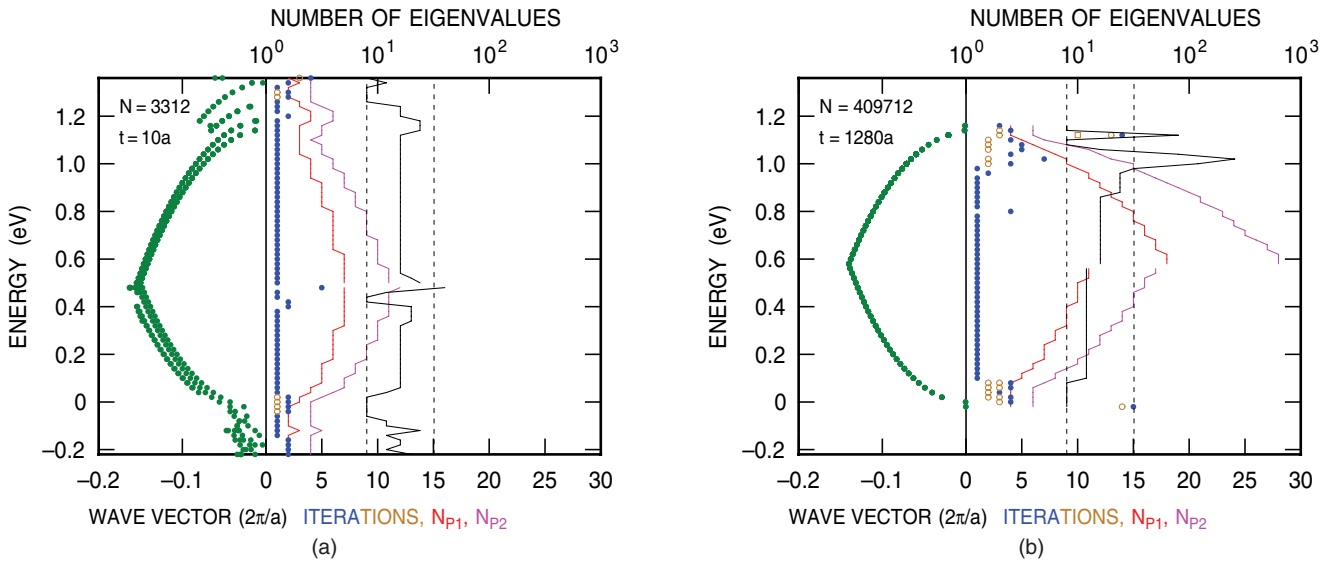

FIG. 6. (Color online) Physical and numerical results for a Si layer of (a) thickness $10a$ ($a = 0.543$ nm), matrix order $N = 3312$; and (b) thickness $1280a$, matrix order $N = 409\,712$. These represent the extremes of layer thickness explored for this example. *Within each plot*, left: The energy vs Im(k) result is shown on the negative abscissa. The loci of solutions begins and ends near the band edges of bulk Si owing to the insignificant quantization present in either layer, and traces out the path of least action for band-to-band tunneling (Ref. 10). Right: The lower positive abscissa gives the number of FEAST iterations required (blue dots/brown circles/brown squares), as well as the quadrature order $N_P = (N_{P1}, N_{P2})$ (N_{P1} red, N_{P2} magenta lines). At a given energy, brown dots signify the iteration when the algorithm adjusted the contour size larger, brown squares indicate an adjustment smaller, while the blue dot indicates the iteration of final convergence. Within the band gap, once a proper initial guess is established and the contour size prediction algorithm becomes accurate, one FEAST iteration is required for convergence in all but a few cases. The (N_{P1}, N_{P2}) values are seen to track the contour's size: as |Im(k)| increases, R must be increased and hence also N_{P1} and N_{P2}. Since the calculation is performed in two steps (computing from lowest energy to midgap, then computing from highest energy to midgap), the lines may not join at midgap. Larger N_{P1}, N_{P2} were needed for (b), especially the upper half of the gap, to maintain convergency in the presence of very small eigenvalue separations present (often in the fifth significant digit). Finally, a black solid line refers to the upper positive abscissa and shows logarithmically the actual number of eigenvalues obtained at each energy. The target was at least 8, and hopefully, no more than 32 (these limits shown as black, vertical, dashed lines).

within the band gap, a good initial guess, predictable contour size estimation, and sufficient quadrature points conspire to obtain convergence with (a) a few or (b) just one FEAST iteration.

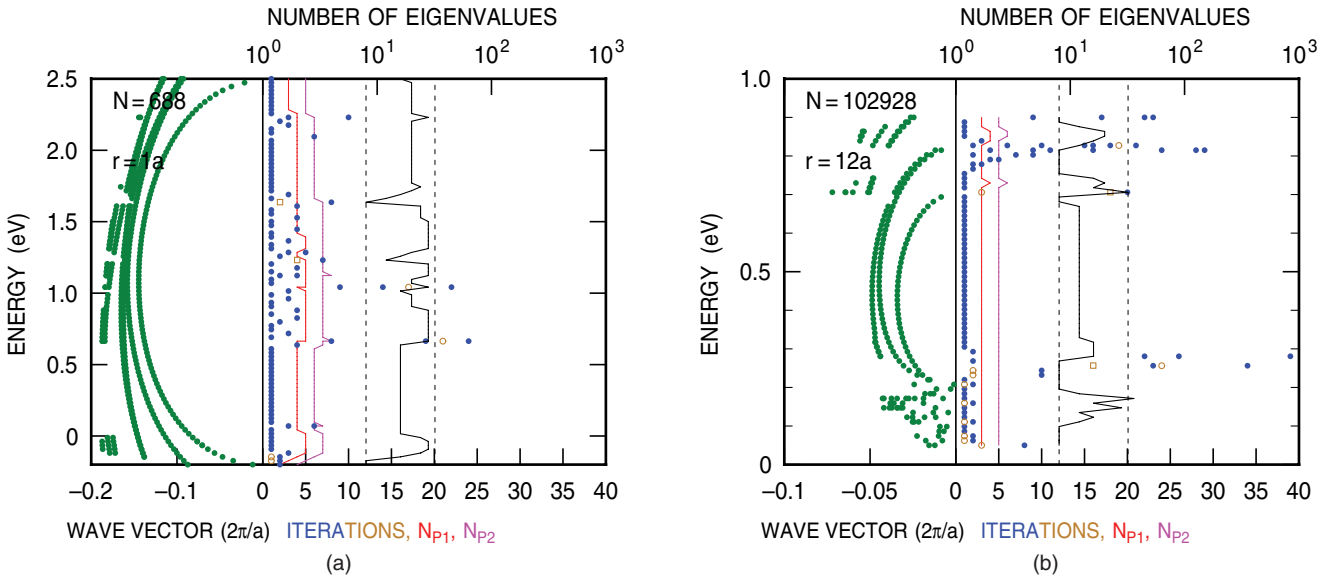

FIG. 7. (Color online) Physical and numerical results for an InAs nanowire of (a) radius a ($a = 0.605\,83$ nm), matrix order $N = 688$; and (b) radius $12a$, matrix order $N = 102\,928$. These represent the extremes of nanowire radius explored for this example. Plot description as for Fig. 6.

V. PERFORMANCE

As noted by Polizzi,[1] there are multiple opportunities to parallelize the FEAST algorithm. Also discussed is the observation that the computational effort required by linear solves (step 2, Fig. 1) dominates that required for the reduced eigenvalue problems (step 4, Fig. 1), which in turn dominates all others parts of the algorithm. This, of course, is what makes the algorithm so attractive, since the growth rate for eigenvalue solution is much faster than for linear solves. Here, linear solves are done with sparse LU factorization and multiple right-hand-side solution using y12m[14] modified for complex double precision arithmetic. The reduced eigenvalue problem (step 4) is solved using zggev.[15]

In a fully parallel implementation (done here), the duration of a single FEAST iteration is given by the sum of one linear solve time (one sparse LU factorization followed by M_0 right-hand side solutions), and one reduced GEP solution. Although $2N_e$ linear solves and two eigenvalue solves are required per FEAST iteration, each are fully independent solves and hence each may be assigned to its own parallel processor. For the second and subsequent FEAST iterations, duration is even less since LU factorization can be skipped, provided contour size and hence quadrature point locations have not changed. Figure 8 verifies that linear solve times dominate GEP times for all 11 Si layer problems described previously. A cubic dependence on M_0 for the GEP times is expected and observed. Also shown are times for LU factorization and multiple right-hand side (RHS) solutions of one linear system. These elapsed times (ETs) depend on N, the matrix size, as follows:[16]

$$\mathrm{ET}_{LU} = 10^{-3.90} N^{1.20}; \quad \mathrm{ET}_{\mathrm{RHS}} = 10^{-5.75} M_0 N^{1.09}. \quad (8)$$

The very modest superlinear dependence of these times on N is consistent with the block-diagonal nature of the matrix factored, since essentially a line of atoms is used to represent a layer. Analogous results (not shown) are obtained for the nine nanowire problems described previously. However, in that case, elapsed times are fitted by

$$\mathrm{ET}_{LU} = 10^{-7.60} N^{2.27}; \quad \mathrm{ET}_{\mathrm{RHS}} = 10^{-7.73} M_0 N^{1.61}. \quad (9)$$

Representing the atoms in a nanowire cross section yields a different matrix structure and hence stronger dependence on N as indicated.

Finally, the influence—or, more properly, the lack thereof—of spurious eigenvalues on convergence will be described for the largest layer $N = 409\,712$ at a single energy $E = 0.98$ eV. Figure 9(a) shows how convergence proceeds by plotting pair-δ error vs eigenvalue position along the Re(λ) axis. Note that in this case, all eigenvalues happen to represent pure imaginary wave vectors and so they are all located along the Re(λ) axis. The point $(R,0)$ resides on the contour's boundary and defines the center of the abscissa; the distance from the boundary either inside or outside the contour is indicated. Converging eigenvalues are paired with another (open symbols) with decreasing error as FEAST iterates. This occurs both inside and outside the contour. Convergence outside the contour occurs because the selection function S does not abruptly go to zero outside the contour. In Fig. 9(a), eigenvalues in the upper right corner (outside, unpaired, large error, far from contour) fail to converge as might be expected. However, eigenvalues

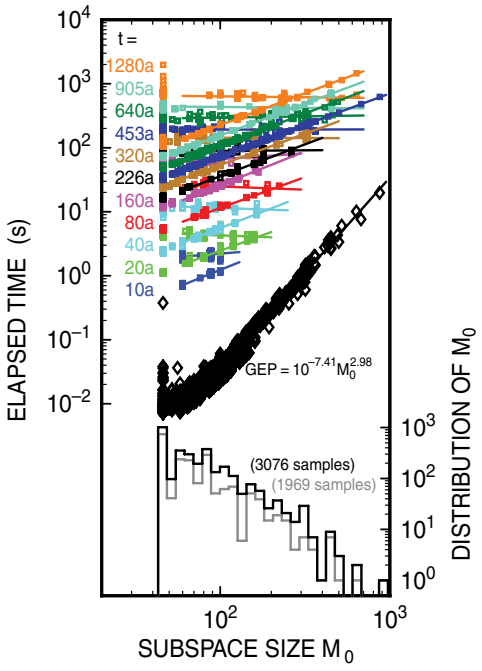

FIG. 8. (Color online) Timing and distribution of M_0 vs subspace size M_0 for computation of a full complex band diagram (i.e., Fig. 6) of 11 Si layers ranging from $10a$ to $1280a$ in thickness. In the plot center, the elapsed time to solve one reduced GEP of size M_0 is indicated with least-squares fit $\mathrm{ET} = 10^{-7.41} M_0^{2.98}$; a cubic dependence on M_0 is expected. Above these data are shown for various layer thicknesses, in various colors, pairs of lines as follows: a horizontal line giving the time of one LU factorization and a line of unity slope giving the time to solve once M_0 right-hand sides. The LU factorization time is independent of M_0 and is hence horizontal; it depends only on N which grows with layer thickness. The time to solve M_0 right-hand sides grows linearly with M_0 and also grows with N. Finally, referred to the right ordinate, two distributions of M_0 are shown. In black, the M_0 value for every FEAST iteration that included an LU factorization is shown. In gray, the M_0 value for every FEAST iteration that did not include an LU factorization is shown. Both indicate the M_0 values used do indeed remain sufficiently small to ensure that linear solve times alone set the overall algorithm performance.

in the upper left corner (inside, unpaired, large error) are the spurious eigenvalues, stubbornly resisting convergence yet remaining inside the contour. If the selection function S is considered to "allow error reduction" (equivalently, $1/S$ "blocks error reduction") an interesting observation follows: S gives a quantitative indication of how error reduction proceeds both inside and outside of the contour, as a function of iteration number, with the addition of just two constants. One constant sets how much error improves per iteration. A second constant sets the absolute error reduction. The colored curves of Fig. 9(a) show this result. Note that S is evaluated at each eigenvalue—just the abscissa value—to form the curves shown. Just how to estimate these two constants remains unknown at this writing. Figure 9(b) reports the overall pair-δ error reduction of this calculation and indicates that convergence could have been declared at iteration 9, at which time three spurious eigenvalues existed. While quite unnecessary, it would require 21 iterations to remove

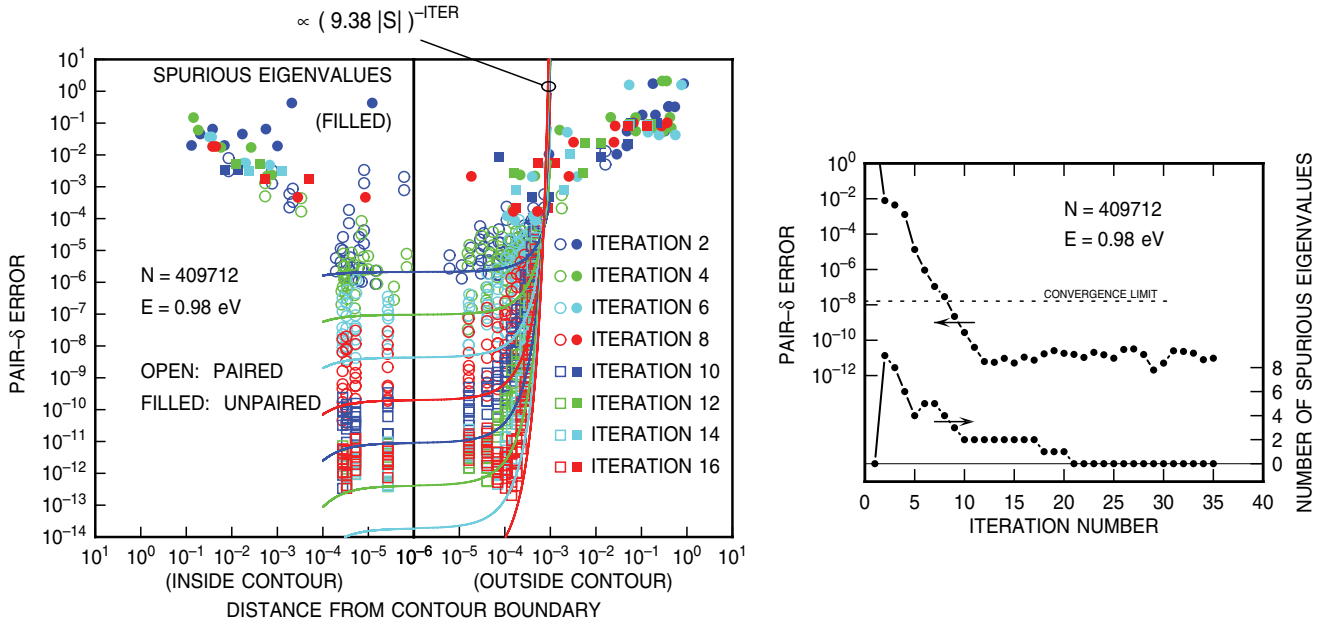

FIG. 9. (Color online) Convergence in the presence of spurious eigenvalues for the largest Si layer with $N = 409\,712$ at energy 0.98 eV. (a) Using no initial guess, eigenvalue pair-δ error is shown vs eigenvalue distance from the contour, either inside (left) or outside (right). The abscissa corresponds to the Re(λ) axis, with the contour boundary residing at (R,0) [see Fig. 2(b)]. Colors and shapes correspond to FEAST iterations 2, 4, ..., 16. Open symbols indicate an eigenvalue which is paired with another according to $\mathcal{P}(\delta_1)$ (see Sec. III B); filled symbols are unpaired. As FEAST iterates, inside the contour four paired eigenvalues (open symbols) are obtained with ever-decreasing error until a noise floor is hit ($\sim 10^{-12}$). Since they are converging to fixed positions within the contour, this gives the impression of four columns of open symbols suspended on the noise floor. If FEAST worked in an idealized fashion, only these eigenvalues (converging, inside contour) would be seen. However, others columns are forming outside the contour indicating converged, paired eigenvalues are appearing there. In the upper right (outside, far from contour, large error), unpaired roots are randomly distributed as noise. But also, in the upper left (inside, large error), unpaired roots are also seen; these are the spurious roots. The colored curves, proportional to the indicated function of the selection function magnitude $|S|$ evaluated at eigenvalue locations, give a startlingly good indication of how error reduction proceeds both inside and outside the contour, as a function of the iteration number. (b) The pair-δ error and number of spurious eigenvalues vs iteration number indicates that convergence could have been declared at iteration 9, where there were three spurious eigenvalues. Allowing FEAST to continue shows that spurious eigenvalues persist until iteration 21.

the spurious eigenvalues entirely from the computation. No degradation due to the presence of spurious eigenvalues is in evidence here, and generally, has not been observed over the course of many calculations.

VI. CONCLUSIONS

The FEAST algorithm has been extended to a dual subspace variant to solve the generalized eigenvalue problem $\boldsymbol{A}x = \lambda \boldsymbol{B}x$, when $\boldsymbol{A}$, $\boldsymbol{B}$ are complex values, non-Hermitian and singular. The matrices derive from the complex band structure problem. While formally outside the range of applicability of the original, single subspace FEAST algorithm,[1] the dual subspace extension prevails in solving the problem even in the presence of spurious eigenvalues. Here, spurious eigenvalues are eigenvalues inside the FEAST contour which refuse to converge. Convergence, or its refusal, is judged by exploiting a property of the eigenvalues of this specific problem; namely, that eigenvalues comes in quadruples λ, $1/\lambda$, λ^*, and $1/\lambda^*$. While other possibilities certainly exist to identify spurious solutions, this technique proved simple, robust, and inexpensive.

Two classes of test problems were described: Si layers with thicknesses up to $1280a$, and InAs nanowires of radii up to $12a$. In both cases, a few wave vectors with smallest, nonzero imaginary parts were computed in the semiconductor's band gap with the use of an appropriate FEAST contour. The problem is solved far more efficiently than simply invoking a complex matrix generalized eigenvalue solver like zggev[15] while computational effort using zggev is $\propto N^3$, this work demonstrated $\propto N^{1.09-1.20}$ (layer) or $\propto N^{1.61-2.27}$ (nanowire) dependencies, depending on the subspace size M_0 employed.

All of the advantages of the original FEAST algorithm[1] remain intact for the dual subspace variant described here, i.e., degeneracies resolved; able to exploit an initial guess; can obtain interior eigenvalues; easy to program and parallelize.

ACKNOWLEDGMENTS

This work spanned serious illness and an associated extended absence from technical activity. The patience shown by IBM management during this time was exemplary. This work benefited from conversation with E. Polizzi (University of Massachusetts Amherst), who introduced the author to dual subspaces and their applicability to this problem.

APPENDIX: DOES FEAST CONVERGE IN ONE ITERATION WITH EXACT INTEGRATION?

A proof in the affirmative of this assertion is sketched. Solve

$$\beta_i \boldsymbol{A}|x_i\rangle = \alpha_i \boldsymbol{B}|x_i\rangle, \tag{A1}$$

where α_i, β_i are complex scalars, $\lambda_i = \alpha_i/\beta_i$ is the complex eigenvalue, and $i = 1,\ldots,N$. The associated adjoint system is

$$\hat{\beta}_i \boldsymbol{A}^\dagger|\hat{x}_i\rangle = \hat{\alpha}_i \boldsymbol{B}^\dagger|\hat{x}_i\rangle, \tag{A2}$$

where $\hat{\alpha}_i$, $\hat{\beta}_i$ are complex scalars and $\hat{\lambda}_i \equiv \hat{\alpha}_i/\hat{\beta}_i$. It is possible to arrange indices such that $\lambda_i = \hat{\lambda}_i^*$ and normalize eigenvectors such that they are $\boldsymbol{B}$ biorthonormal:[5]

$$\langle\hat{x}_i|\boldsymbol{B}|x_j\rangle = \delta_{ij}, \tag{A3}$$

where δ_{ij} is the Kronecker δ. The ability to assert this derives from the assumption that $|x\rangle$, $|\hat{x}\rangle$ form complete N-dimensional bases, which is assumed for this Appendix. The Green's function can then be factored as

$$\boldsymbol{G}(z) \equiv (z\boldsymbol{B} - \boldsymbol{A})^{-1} = \sum_{i=1}^{N} \frac{|x_i\rangle\langle\hat{x}_i|}{(z-\lambda_i)\langle\hat{x}_i|\boldsymbol{B}|x_i\rangle}, \tag{A4}$$

where eigenfunctions are assumed unnormalized. Also,

$$\boldsymbol{A} = \sum_{i=1}^{N} \lambda_i \frac{\boldsymbol{B}|x_i\rangle\langle\hat{x}_i|\boldsymbol{B}}{\langle\hat{x}_i|\boldsymbol{B}|x_i\rangle}, \quad \boldsymbol{B} = \sum_{i=1}^{N} \frac{\boldsymbol{B}|x_i\rangle\langle\hat{x}_i|\boldsymbol{B}}{\langle\hat{x}_i|\boldsymbol{B}|x_i\rangle}. \tag{A5}$$

For the adjoint problem,

$$\hat{\boldsymbol{G}}(z) \equiv (z\boldsymbol{B}^\dagger - \boldsymbol{A}^\dagger)^{-1} = \sum_{i=1}^{N} \frac{|\hat{x}_i\rangle\langle x_i|}{(z-\hat{\lambda}_i)\langle x_i|\boldsymbol{B}^\dagger|\hat{x}_i\rangle}. \tag{A6}$$

Integrating exactly using Cauchy's integral formula,[17]

$$\begin{aligned} \boldsymbol{\rho} &= \frac{1}{2\pi\imath}\int_{\mathcal{C}} dz\, \boldsymbol{G}(z) = \sum_{i=1}^{M} \frac{|x_i\rangle\langle\hat{x}_i|}{\langle\hat{x}_i|\boldsymbol{B}|x_i\rangle}, \\ \hat{\boldsymbol{\rho}} &= \frac{1}{2\pi\imath}\int_{\mathcal{C}} dz\, \hat{\boldsymbol{G}}(z) = \sum_{i=1}^{\hat{M}} \frac{|\hat{x}_i\rangle\langle x_i|}{\langle x_i|\boldsymbol{B}^\dagger|\hat{x}_i\rangle}, \end{aligned} \tag{A7}$$

where the $\lambda_i = 1,\ldots,M \leqslant N$ are interior to a simple closed contour $\mathcal{C}$, taken in a positive sense. Since $\mathcal{C}$ is constructed to contain $z^*, 1/z, 1/z^*$ if it contains z, then $M = \hat{M}$. Now define $\boldsymbol{Y}_{N\times M_0} = [\,|y_1\rangle\,|y_2\rangle\cdots|y_{M_0}\rangle\,]$, $M_0 \geqslant M$, with $|y_i\rangle$ linearly independent random vectors. Also define a second random matrix $\hat{\boldsymbol{Y}}_{N\times M_0} = [\,|\hat{y}_1\rangle\,|\hat{y}_2\rangle\cdots|\hat{y}_{M_0}\rangle\,]$, $\boldsymbol{Q}_{N\times M_0} = \boldsymbol{\rho}_{N\times N}\boldsymbol{Y}$ and $\hat{\boldsymbol{Q}}^\dagger_{N\times M_0} = \hat{\boldsymbol{Y}}^\dagger\hat{\boldsymbol{\rho}}^\dagger_{N\times N}$. Then

$$\boldsymbol{Q} = \boldsymbol{\rho}\boldsymbol{Y} = \boldsymbol{X}_{N\times M}\boldsymbol{W}_{M\times M_0} \equiv \left[\frac{|x_1\rangle}{\langle\hat{x}_1|\boldsymbol{B}|x_1\rangle}\ \frac{|x_2\rangle}{\langle\hat{x}_2|\boldsymbol{B}|x_2\rangle}\cdots\frac{|x_M\rangle}{\langle\hat{x}_M|\boldsymbol{B}|x_M\rangle}\right]_{N\times M}\boldsymbol{W}_{M\times M_0}, \tag{A8}$$

$$\hat{\boldsymbol{Q}}^\dagger = \hat{\boldsymbol{Y}}^\dagger\hat{\boldsymbol{\rho}}^\dagger = \hat{\boldsymbol{W}}^\dagger_{M_0\times M}\hat{\boldsymbol{X}}^\dagger_{M\times N} \equiv \hat{\boldsymbol{W}}^\dagger_{M_0\times M}\begin{bmatrix} \frac{\langle\hat{x}_1|}{\langle\hat{x}_1|\boldsymbol{B}|x_1\rangle} \\ \frac{\langle\hat{x}_2|}{\langle\hat{x}_2|\boldsymbol{B}|x_2\rangle} \\ \vdots \\ \frac{\langle\hat{x}_M|}{\langle\hat{x}_M|\boldsymbol{B}|x_M\rangle} \end{bmatrix}_{M\times N}, \tag{A9}$$

where $\boldsymbol{W}_{l,m} = \langle\hat{x}_l|y_m\rangle$ and $\hat{\boldsymbol{W}}^\dagger_{l,m} = \langle\hat{y}_l|x_m\rangle$. The matrices $\boldsymbol{A}_{\boldsymbol{Q}_{M_0\times M_0}} = \hat{\boldsymbol{Q}}^\dagger\boldsymbol{A}\,\boldsymbol{Q}$ and $\boldsymbol{B}_{\boldsymbol{Q}_{M_0\times M_0}} = \hat{\boldsymbol{Q}}^\dagger\boldsymbol{B}\,\boldsymbol{Q}$ can now be written as

$$\boldsymbol{A}_{\boldsymbol{Q}_{M_0\times M_0}} = \hat{\boldsymbol{W}}^\dagger{}_{M_0\times M}\begin{bmatrix} \frac{\lambda_1}{\langle\hat{x}_1|\boldsymbol{B}|x_1\rangle} & 0 & \ldots & 0 \\ 0 & \frac{\lambda_2}{\langle\hat{x}_2|\boldsymbol{B}|x_2\rangle} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \ldots & 0 & \frac{\lambda_M}{\langle\hat{x}_M|\boldsymbol{B}|x_M\rangle} \end{bmatrix}_{M\times M}\boldsymbol{W}_{M\times M_0}, \tag{A10}$$

$$\boldsymbol{B}_{\boldsymbol{Q}_{M_0\times M_0}} = \hat{\boldsymbol{W}}^\dagger{}_{M_0\times M}\begin{bmatrix} \frac{1}{\langle\hat{x}_1|\boldsymbol{B}|x_1\rangle} & 0 & \ldots & 0 \\ 0 & \frac{1}{\langle\hat{x}_2|\boldsymbol{B}|x_2\rangle} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \ldots & 0 & \frac{1}{\langle\hat{x}_M|\boldsymbol{B}|x_M\rangle} \end{bmatrix}_{M\times M}\boldsymbol{W}_{M\times M_0}, \tag{A11}$$

where use has been made of $\boldsymbol{B}$ biorthogonality and (A5).

The system $\boldsymbol{A}_{\boldsymbol{Q}}\boldsymbol{\Phi}_{M_0\times M_0} = \epsilon\boldsymbol{B}_{\boldsymbol{Q}}\boldsymbol{\Phi}_{M_0\times M_0}$ can then be written

$$\hat{\boldsymbol{W}}^\dagger{}_{M_0\times M}\hat{\boldsymbol{C}}_{M\times M}\begin{bmatrix} \lambda_1 & 0 & \ldots & 0 \\ 0 & \lambda_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \ldots & 0 & \lambda_M \end{bmatrix}_{M\times M}\boldsymbol{W}_{M\times M_0}\boldsymbol{\Phi}_{M_0\times M_0} = \boldsymbol{\epsilon}\,\hat{\boldsymbol{W}}^\dagger{}_{M_0\times M}\hat{\boldsymbol{C}}_{M\times M}\boldsymbol{W}_{M\times M_0}\boldsymbol{\Phi}_{M_0\times M_0}, \tag{A12}$$

where

$$\hat{\boldsymbol{C}}_{M\times M} \equiv \begin{bmatrix} \frac{1}{\langle \hat{x}_1|\boldsymbol{B}|x_1\rangle} & 0 & \dots & 0 \\ 0 & \frac{1}{\langle \hat{x}_2|\boldsymbol{B}|x_2\rangle} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \frac{1}{\langle \hat{x}_M|\boldsymbol{B}|x_M\rangle} \end{bmatrix} \tag{A13}$$

and $\boldsymbol{\epsilon}$ is the diagonal matrix diag(ϵ_i). The GEP (A12) is the desired result and yields eigenvalues $\epsilon_i = \lambda_i$ and eigenvectors $|\Phi_i\rangle$ which map back to $|x_i\rangle$. One way this can be seen follows.

Since the product $\hat{\boldsymbol{W}}^\dagger{}_{M_0\times M}\hat{\boldsymbol{C}}_{M\times M}$ seen in (A12) is a matrix of rank M, premultiplying this pair by a nonsingular matrix $\boldsymbol{E}_{M_0\times M_0}$ [obtained as a product of elementary matrices as shown in Ref. 18 (Sec. 5.5)] can be rewritten

$$\boldsymbol{E}_{M_0\times M_0}\hat{\boldsymbol{W}}^\dagger{}_{M_0\times M}\hat{\boldsymbol{C}}_{M\times M} \equiv \begin{bmatrix} \boldsymbol{D}_{M\times M} \\ \boldsymbol{0}_{(M_0-M)\times M} \end{bmatrix}, \tag{A14}$$

where $\boldsymbol{D}_{M\times M}$ is also nonsingular. Hence premultiplying (A12) by $\boldsymbol{E}_{M_0\times M_0}$ obtains

$$\boldsymbol{\epsilon_\beta}\begin{bmatrix} \boldsymbol{D}_{M\times M} \\ \boldsymbol{0}_{(M_0-M)\times M} \end{bmatrix}\begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \lambda_M \end{bmatrix}_{M\times M}\boldsymbol{W}_{M\times M_0}\boldsymbol{\Phi}_{M_0\times M_0} = \boldsymbol{\epsilon_\alpha}\begin{bmatrix} \boldsymbol{D}_{M\times M} \\ \boldsymbol{0}_{(M_0-M)\times M} \end{bmatrix}\boldsymbol{W}_{M\times M_0}\boldsymbol{\Phi}_{M_0\times M_0}, \tag{A15}$$

where the diagonal matrix $\boldsymbol{\epsilon}$ has been split into two diagonal matrices appearing on both sides of the equality ($\boldsymbol{\epsilon_\beta}$ left, $\boldsymbol{\epsilon_\alpha}$ right) as is common in computer routines solving the GEP (e.g., zggev[15]). After numerical experimentation, (A15) is "readily" solved by inspection: the M_0 (eigenvalue, eigenvector) solution pairs are

$$(\epsilon_{\alpha_i}/\epsilon_{\beta_i},\ \boldsymbol{W}|\Phi_i\rangle) = \begin{cases} (\lambda_i,\ |i\rangle_{M\times 1}), & i \leqslant M \\ (0/0,\ |\xi\rangle_{M\times 1}, & M < i \leqslant M_0 \end{cases}, \tag{A16}$$

where $|i\rangle_{M\times 1}$ is a vector of zeros with unity in the ith position and $|\xi\rangle_{M\times 1}$ is an arbitrary complex vector. This expression assumes distinct λ_i. If they are not distinct, a slightly more complicated form is required for the eigenfunctions which is not shown. Equation (A16) is seen to represent exactly the anticipated solution pairs to (A1) within the contour $\mathcal{C}$ together with $(M_0 - M)$ additional solutions as follows:

$i \leqslant M$: Eigenvalues $\epsilon_i = \lambda_i$; eigenvectors $\boldsymbol{X}_i$ (the ith column of $\boldsymbol{X}$) are computed as

$$\boldsymbol{X}_i = \boldsymbol{Q}|\Phi_i\rangle = \boldsymbol{X}\,\boldsymbol{W}|\Phi_i\rangle = \begin{bmatrix} \frac{|x_1\rangle}{\langle \hat{x}_1|\boldsymbol{B}|x_1\rangle} & \frac{|x_2\rangle}{\langle \hat{x}_2|\boldsymbol{B}|x_2\rangle} & \cdots & \frac{|x_M\rangle}{\langle \hat{x}_M|\boldsymbol{B}|x_M\rangle} \end{bmatrix}_{N\times M}|i\rangle_{M\times 1} = \frac{|x_i\rangle}{\langle \hat{x}_i|\boldsymbol{B}|x_i\rangle}. \tag{A17}$$

$M < i \leqslant M_0$: Additional solutions are obtained, identifiable by $\epsilon_{\alpha_i} = \epsilon_{\beta_i} = 0$.

The additional solutions do not solve the original problem (A1) and are discarded. The original problem's solutions $(\lambda_i, |x_i\rangle)$ are thus obtained, and without any iteration.

*laux@us.ibm.com

[1] E. Polizzi, Phys. Rev. B **79**, 115112 (2009).

[2] T. B. Boykin, Phys. Rev. B **54**, 8107 (1996).

[3] M. Luisier, A. Schenk, W. Fichtner, and G. Klimeck, Phys. Rev. B **74**, 205323 (2006).

[4] A. Messiah, *Quantum Mechanics (Two Volumes Bound as One)* (Dover, Mineola, NY, 1999).

[5] P. M. Morse and H. Feshbach, *Methods of Theoretical Physics Part I: Chapters 1 to 8* (McGraw-Hill, New York, 1953).

[6] J.-M. Jancu, R. Scholz, F. Beltram, and F. Bassani, Phys. Rev. B **57**, 6493 (1998).

[7] T. B. Boykin, G. Klimeck, and F. Oyafuso, Phys. Rev. B **69**, 115201 (2004).

[8] T. B. Boykin and G. Klimeck, Phys. Rev. B **71**, 115215 (2005).

[9] A. Ajoy, K. V. R. M. Murali, and S. Karmalkar, J. Phys.: Condens. Matter **24**, 055504 (2012).

[10] S. E. Laux, in *Proceedings of the 13th International Workshop on Computational Electronics* (IEEE, Piscataway, NJ, 2009), pp. 5–8.

[11] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in FORTRAN: The Art of Scientific Computing*, 2nd ed. (Cambridge University Press, Cambridge, England, 1992).

[12] T. B. Boykin, G. Klimeck, R. C. Bowen, and F. Oyafuso, Phys. Rev. B **66**, 125207 (2002).

[13] Y. Zheng, C. Rivas, R. Lake, K. Alam, T. B. Boykin, and G. Klimeck, IEEE Trans. Electron Dev. **52**, 1097 (2005).

[14] Z. Zlatev, J. Wasniewski, and K. Schaumburg, *Y12M, Solution of Large and Sparse Linear Algebraic Equations*, Lecture Notes in Computer Science No. 121 (Springer-Verlag, New York/ Heidelberg, 1981), code available at www.netlib.org/y12m.

[15] *Lapack—linear algebra package (version 3.0+updates)*, 2000, http://www.netlib.org/lapack/.

[16] *IBM Power 755 server*, 2010, http://www-03.ibm.com/systems/power/hardware/755/.

[17] R. V. Churchill, J. W. Brown, and R. F. Verhey, *Complex Variables and Applications*, 3rd ed. (McGraw-Hill, New York,, 1974).

[18] B. Noble, *Applied Linear Algebra* (Prentice-Hall, Englewood Cliffs, NJ, 1969).