Enhanced matrix spectroscopy: The preconditioned Green-function block Lanczos algorithm

Todd J. Minehardt, J. David Adcock, and Robert E. Wyatt
Department of Chemistry and Biochemistry, Institute for Theoretical Chemistry, The University of Texas at Austin, Austin, Texas 78712-1167
(Received 14 April 1997)

We present herein the results of a doubly filtered block Lanczos code, which effectively uses shifted simultaneous inverse iteration to make a block tridiagonal representation of H, a Hamiltonian matrix. The first filter preconditions the starting block of Lanczos vectors through one or more applications of $p(E)$, the preconditioning operator. This block is then used to seed the Lanczos recursion, driven with the Green-function filter $G(E)=(EI-H)^{-1}$. Each set of successively generated Lanczos vectors is orthogonalized against all previous ones. These steps allow us to converge eigenvalues near E, in the interior region of the eigenspectrum, with extreme accuracy. Degenerate eigenvalues are reported; ''ghost'' eigenvalues (multiple copies of eigenvalues that have already converged) are avoided. For a set number of Lanczos recursions, the use of a preconditioner effectively doubles the number of converged eigenvalues than are resolved without its use. The computation time is increased almost imperceptibly. The $2e_g \leftarrow a_g$ torsional transition for the water trimer, $(H_2O)_3$, is examined in an application of our method. We resolve the quantities needed for this calculation is less than one-fifth the time required to directly diagonalize the matrices, with no loss of accuracy. [S1063-651X(97)03310-2]

PACS number(s): 02.70.−c

I. INTRODUCTION

Central to the theme of quantum physics is the reality that sooner or later, one is faced with the daunting task of extracting eigenvalues and eigenvectors from an often prohibitively large Hamiltonian matrix, H. Even with today's impressive computing power and technology, we must resort to using efficient algorithms in order to tackle the most challenging applications. Recently, a strategy has been invoked under the name *matrix spectroscopy* [1–3]: we can peer into any part of the eigenspectrum and resolve eigenvalues and eigenvectors to machine precision by coupling the Lanczos algorithm with a spectral filter, denoted $f(E)$. Because H is not directly diagonalized (in fact, it need not ever be computed in full), we can then work on extremely large problems without compromising accuracy. The filter should be designed to tune out the undesirable portion of the spectrum, while amplifying the desired portion.

The single-vector Lanczos algorithm (SLA) [4] for partially tridiagonalizing large, sparse matrices has been widely used in a variety of applications [5–11]. This approach is attractive for several reasons: the large Hamiltonian matrix enters into the calculation only in the formation of matrix-vector products, which is how the tridiagonal representation (or Krylov subspace) is constructed, only a few iterations are needed to rapidly converge the most widely separated eigenvalues, and the basic algorithm is quite easily coded [12].

The SLA does have its shortcomings, however. A significant deficiency is the inherent inability of the algorithm to pinpoint multiplicities in the eigenspectrum unless deflation techniques are employed [13]. The form of the SLA most widely used in the literature is usually implemented without these safeguards. As a result, only single copies of eigenvalues can be reported with confidence. More than one copy of an eigenvalue is a ''ghost'' and indicates that the eigenvalue in question has converged to machine precision [14] (it should be noted that this is only a problem if global orthogonality is not preserved among the Lanczos vectors [15]). If the eigenvalue happens to be one of several truly degenerate eigenvalues, we are unaware of this fact unless further tests are performed. And even if these steps are taken, the eigenvectors corresponding to degenerate eigenvalues are not correctly reported. Further, the Lanczos method usually converges eigenvalues and eigenvectors at the opposite ends of the eigenspectrum, and is slow to resolve the important interior region, unless the local gap is relatively large [1,3].

Degeneracy is a common feature of many applications that are studied in molecular physics. This degeneracy gerally arises because the molecule possesses some form of symmetry, whether it be permutational symmetry associated with sets of identical particles, inversion symmetry through the center of mass, symmetry associated with rotations about the center of mass, or point group symmetry associated with rotations, reflections, and inversions of an equilibrium structure. In each case, there is a set of operators forming a mathematical group, and each group operator commutes with the molecular Hamiltonian. Degeneracy arises when the irreducible representations for the appropriate group have dimensions greater than one. For the point groups, this happens when there is a threefold or higher axis of rotation: irreducible representations of at least dimension two are produced.

Given that many problems of interest can only be studied by extracting eigensystem information from Hamiltonian matrices that describe degenerate systems, we are fortunate that perhaps the most serious limitation of the SLA is easily remedied by utilizing an extension of the algorithm that can resolve and report multiplicities: the block Lanczos algorithm (BLA) [16–19]. Mechanistically identical to the SLA, this approach forms the Krylov subspace by applying the Hamiltonian onto a block matrix (having at least as many columns as there are expected multiplicities), which preserves the ''degenerate character'' of the original problem. The Cullum-Willoughby test for spurious eigenvalues is tailored specifically for the SLA, and handles the ghost problem quite nicely for those cases [14]. In the block version, however, the ghost problem is commonly handled by avoid-

 © 1997 The American Physical Society

ing doing too many iterations so that the loss of global orthogonality is not great enough to warrant the appearance of ghost eigenvalues. The correct number of eigenvalues, and the corresponding eigenvectors, are reported for parts of the eigenspectrum that are well separated.

We are still concerned with resolving the part of the eigenspectrum in the interior, dense part of the problem. The Lanczos approaches can be coupled with a way to ''pry open'' these target areas, dilating the spacing between adjacent eigenvalues while compressing the uninteresting eigenvalues and then taking advantage of the convergence properties of either the SLA or BLA. Most recently, Wyatt has implemented these matrix spectroscopic techniques by driving the Lanczos algorithm with what he terms the *Green function filter*, $G(E)=(EI-H)^{-1}$ [1,2]. Following Ericcson and Ruhe's *shift and invert* method [3], the original problem is recast by shifting the diagonal elements of the Hamiltonian matrix by some ''test energy'' E, which corresponds to an approximation of an eigenvalue lying in some part of the interior region of interest. In the fortunate case that H can be factored, there is a straightforward way to implement *shift and invert*: by factoring the shifted matrix, and solving for a new matrix with which the recursion is driven, we can invert the Hamiltonian and force the gaps between eigenvalues near E to become greatly dilated and thus converge rapidly. Eigenvalues too far from the test energy are mapped to an uninteresting cluster near zero, which is a fixed point of the nonlinear map. A more detailed explanation of this approach is given below.

The current work details our procedure for coupling the BLA with a Green function filter (GFBLA) and a preconditioning loop. In a method similar to Roy and Carrington's *guided Lanczos* procedure [20], we first prime the Lanczos process by starting it with a block of orthonormal vectors that are already ''leaning in the direction of'' a few eigenvectors near E. This is accomplished before entering the Lanczos loop by simply performing a few iterations where we apply a preconditioning operator $p(E)$ to a set of vectors. We then start the Lanczos process with the new, preconditioned block. We see a marked increase in the number of converged eigenvalues for a very modest increase in computational time. Each newly generated set of Lanczos vectors is explicitly reorthogonalized against all previously generated ones. The result of this procedure is that a Cullum-Willoughby-like test for ghost eigenvalues becomes invalid. We determine which eigenvalues are good, and which are bad, by comparing two lists of eigenvalues generated by one GFBLA run near a test eigenvalue. Test cases for matrices of dimension $N \leqslant 3000$ are presented. We review both Lanczos algorithms and the Green-function filter (with an example of its use), describe our approach for determining good and bad eigenvalues, provide an example that illustrates when and why to use the block variant of Lanczos, outline the preconditioning loop, and detail our computational methods in Sec. II. The results of our tests for the convergence and stability properties of our code and its application to the $2e_g \leftarrow a_g$ torsional transition in $(H_2O)_3$ are presented in Sec. III. Conclusions are given in Sec. IV.

II. ALGORITHMS

A. The single-vector Lanczos algorithm

The single-vector Lanczos algorithm (SLA) has been used as an effective tool in reducing enormous sparse matrices to small, tridiagonal representations of the system space (Krylov subspace) while retaining accurate information about the extreme upper and lower eigenvalues of the original matrix. Poulin and co-workers [9] used the Lanczos algorithm to extract information about vibrational excitation energies of formaldehyde from extremely large Hamiltonian matrices [the recursive residue generation method (RRGM) [5,6] was used to tackle the problem]. The photoionization of hydrogen in an electric field was probed by Karlsson and Goscinski [10], who modified the RRGM process (which is driven by the Lanczos algorithm) to facilitate larger calculations that do not tax computational resources as much as unoptimized approaches do. Jolicard and Atabek investigated the rate of H^{2+} photodissociation in laser fields by coupling the Lanczos method with a new wave-operator technique [11]. In practice, the Lanczos algorithm must be used with care. The propagation of roundoff error (finite precision) has prevented some potential users from treating it as a ''black box'' solution to computational problems. We outline the procedure below and discuss what to some inexperienced users is the most troubling aspect of SLA: that the loss of orthogonality among Lanczos vectors is both a blessing and a curse.

First, consider the unrealistic case: an ideal computer with infinite precision arithmetic. The SLA develops a tridiagonal matrix T through a three-term recurrence relation by the successive generation of orthonormal column vectors q_j. The set of $\{q_j\}$ vectors satisfies the three-term relation

$$Hq_j = b_{j-1}q_{j-1} + a_j q_j + b_j q_{j+1},$$

where H is any symmetric matrix and $\{q_j\}$ form the columns in the matrix Q for which $Q^TQ=I$ and $Q^THQ=T$. In practice, the SLA is implemented as an iterative process in the following way [12].

Initialization:

$$j=0, \quad q_0=0, \quad b_0=1 \quad \text{and} \quad r_0=q_1.$$

Iteration:

$$q_{j+1}=r_j/b_j,$$

$$j \rightarrow j+1,$$

$$a_j=q_j^T H q_j,$$

$$r_j=Hq_j-a_jq_j-b_{j-1}q_{j-1},$$

$$b_j=\|r_j\|_2.$$

Each pass through the iteration loop produces the matrix elements a_j and b_j in the tridiagonal representation T_j:

$$T_j=\begin{bmatrix} a_1 & b_1 & & \\ b_1 & a_2 & b_2 & \\ & \ddots & \ddots & b_{j-1} \\ & & b_{j-1} & a_j \end{bmatrix}.$$

Because $Q^THQ=T$, the eigenvalues of H are equal to those of T. Any number of efficient subroutines can then be used to diagonalize T with ease.

The situation is more interesting in finite precision arithmetic. In this case, one must use care when implementing the

SLA without taking precautions to protect against the loss of orthogonality among successively generated vectors $\{q_j\}$. Although the algorithm provides for *local* orthogonalization of each q_{j+1} against the previous $\{q_j\}$, roundoff error quickly leads to a set of $\{q_j\}$ in which not all the vectors are linearly independent. *Global* orthogonalization, in which each new q_j is orthogonalized against every previous $q_{j-1,j-2,\ldots}$, is usually avoided because in the past it has been computationally demanding and expensive.

Two methods for preserving orthogonality among Lanczos vectors are *selective* and *partial* reorthogonalization. In *selective* reorthogonalization [13], the Lanczos process is stopped after a few recursions and the $L\times L$ matrix T_j is diagonalized. We examine the error bounds of an uncomputed Ritz vector (the eigenvalues and eigenvectors of the Lanczos tridiagonal matrix are called Ritz values and Ritz vectors, respectively, and are approximations to the true eigenvalues and eigenvectors of H), which is given by the product of $b_j|s_{Lj}|$ (where s_{Lj} is the element in the Lth row and column j of the eigenvector matrix obtained by diagonalizing T_j): if the quantity is sufficiently small, this indicates that the Ritz vector (and its corresponding Ritz value) has converged. We compute the Ritz vector, store it and the Ritz value, and continue the Lanczos process. Now, we orthogonalize all successively generated Lanczos vectors against the converged Ritz vector. This method is attractive because we need only store converged Ritz values and vectors and a small subset of the Lanczos vectors. Further, multiple Ritz values can and do converge: after the Ritz vector corresponding to one Ritz value is ''banished'' (in the words of Parlett and Scott [13]) from successive Lanczos vectors, roundoff error will allow the next Ritz vector (orthogonal to the first one) to surface and thus the multiple copy of the Ritz value in question is realized. This method requires multiple runs to be made when more than one Ritz value converges, therefore increasing computational demands.

The *partial* reorthogonalization [15] scheme monitors the loss of orthogonality of the most recently computed Lanczos vector, q_j, among previously computed Lanczos vectors. When this tolerance is exceeded, q_j is orthogonalized against a set of Lanczos vectors within which the loss of orthogonality was greatest. It has been shown that both approaches are effective in terms of convergence and cost [15].

It turns out that we can still determine which eigenvalues are multiple copies in the event that we do not reorthogonalize at all. Cullum and Donath [19] proposed two methods, both of which require additional computations. One result was that Ritz vectors corresponding to degenerate Ritz values will be linearly independent only up to the order of that degeneracy. The rest of the Ritz vectors associated with this Ritz value will be linearly dependent over the first set and they correspond to ghosts in the eigenspectrum. Whichever method is eventually used, the SLA is most accurate when computing extreme eigenvalues of the eigenvalue spectrum.

The SLA is attractive in solving large eigenproblems because one only needs to partially construct T in order to yield excellent approximations to (at least) $\lambda_{\min}$ and $\lambda_{\max}$. Accurate determination of the extreme eigenvalues of a matrix of dimension 10^3 can usually be achieved after only 5 or so SLA iterations. Interior eigenvalues also converge quickly if the local gaps are relatively large.

B. The block Lanczos algorithm

All of the procedural details of the SLA are preserved in the block Lanczos algorithm. In effect, matrix-vector multiples are replaced by matrix-matrix multiples, and the resultant tridiagonal representation of H is now block tridiagonal. The main advantage of the BLA over the SLA is that multiplicities in the eigenvalue spectrum are reported. A_j and B_j are both square matrices, as opposed to the scalar quantities a_j and b_j generated by the SLA. The dimension of A is equal to that of B and should be determined by the minimum number of multiplicities expected for an eigenvalue [16–18].

The salient features of the BLA are given below, following the work of Grimes and co-workers [18].

Initialization:

$$Q_0=0 \quad \text{and} \quad B_1=0;$$

$$\text{choose the } N\times M \text{ matrix } R_1\text{; orthonormalize}$$

$$\text{its columns to yield } Q_1.$$

Iteration:

$$j=1,$$

$$U_j=HQ_j-Q_{j-1}B_j^T,$$

$$A_j=Q_j^TU_j,$$

$$R_{j+1}=U_j-Q_jA_j,$$

$$\text{do the orthogonal factorization } Q_{j+1}B_{j+1}=R_{j+1},$$

$$j\rightarrow j+1,$$

where H is $N\times N$, A and B are $M\times M$, and Q, R, and U are $N\times M$ dimensional matrices. Further, when R is QR factored into Q and B, B is upper triangular and Q is orthonormal. We thus arrive at the block tridiagonal matrix T_j, which looks like

$$T_j=\begin{bmatrix} A_1 & B_2^T & & & \\ B_2 & A_2 & B_3^T & & \\ & \ddots & \ddots & \ddots & \\ & & B_{j-1} & A_{j-1} & B_j^T \\ & & & B_j & A_j \end{bmatrix},$$

where, as in the case of the SLA, the Ritz values of T are approximations to eigenvalues of H.

As with the SLA, the loss of orthogonality among each successively generated Q with the previous one is also a problem in the BLA. Again, selective reorthogonalization [13,15,18] has been implemented in practice to preserve local, if not global, orthogonality in the desire to extract eigenvalues of interest in their most accurate approximations to the eigenvalues of the original matrix H.

We chose a block size of $M=2$ for this study; all results are for this case unless noted otherwise. The decision to use a particular block size is detailed in the introduction to Sec. III.

C. The Green-function filter

Several filters have been used in conjunction with the SLA: an *exponential filter*, where $H \rightarrow f(H) = \exp[-\beta H]$ was used to drive the Lanczos algorithm [21], a *Gaussian derivative filter*, where $H \rightarrow f(E) = (H - EI)\exp[-\beta(H-EI)^2]$ [22], and an *auxiliary operator*, $f(E) = (EI - H)^2$ [23], have all been shown to be effective for certain problems (β controls the width of the filter). The block version of the filtered Lanczos algorithm has been used with a filter that approximates $\exp[-H/\Delta]$ (Δ is the interval that encompasses the part of the eigenspectrum of interest) with a Chebyshev polynomial [24].

The Green-function filter, $G(E) = (EI - H)^{-1}$, provides the Lanczos algorithm with a matrix whose spectrum has been greatly dilated around E. The result of using this filter is that these eigenvalues are resolved first and most accurately; eigenvalues far from E are mapped to a cluster near zero while eigenvalues near E are mapped to very large positive or negative values [1,3,18]. Because of these properties, the Green function is a superb filter (however, that does not mean that it is easy to implement in the computer code). For this study, we apply the Green function to H by factoring the inverse filter $f(E)^{-1} = (EI - H)$ and then solving the linear algebra equation $f(E)^{-1} Q_{\text{new}} = Q_{\text{old}}$ for Q_{new} at the start of each Lanczos step. Q_{new} thus replaces the matrix-matrix product HQ_j in the first term of the BLA.

The eigenvalues of the block tridiagonal matrix are related to eigenvalues of H by

$$\lambda_H \approx \lambda_R = E - 1/\lambda_L , \tag{1}$$

where λ_H is an eigenvalue of H, λ_R is a Ritz value, and λ_L is an eigenvalue of T_j.

D. Testing for good and bad Ritz values

The single-vector Lanczos algorithm is known to have converged a Ritz value when more than one copy of that Ritz value appears in the ''final list'' of eigenvalues of T_j [14]. In fact, this property of the algorithm is naturally exploited by the loss of global orthogonality of successively generated vectors, due to the propagation of roundoff error. An approach for separating the good and bad eigenvalues was developed by Cullum and Willoughby [14]. It is important to note that this test need only be invoked when successively generated Lanczos vectors Q_j are not fully reorthogonalized against all previous ones. Further, this test was developed for the SLA case: its implementation for the BLA would be quite complex, as will be shown below.

The Cullum-Willoughby test, as implemented for the SLA with no reorthogonalization, goes as follows: After j Lanczos steps, we have the $j \times j$ tridiagonal matrix, denoted T_j. If we zero out the first row and column of T_j, then we define a new matrix called T_j^*:

$$T_j = \begin{bmatrix} a_1 & b_1 & & \\ b_1 & a_2 & b_2 & \\ & \ddots & \ddots & b_{j-1} \\ & & b_{j-1} & a_j \end{bmatrix},$$

$$T_j^* = \begin{bmatrix} 0 & 0 & & \\ 0 & a_2 & b_2 & \\ & \ddots & \ddots & b_{j-1} \\ & & b_{j-1} & a_j \end{bmatrix}.$$

We diagonalize both matrices and examine the eigenvalues. The list of eigenvalues from T_j is called the good list here, and that from T_j^* the bad list. (Good and bad are simply labels and reflect the fact that T_j is a more complete matrix than T_j^*.) If there exist multiple copies of an eigenvalue on the good list and there is a copy (to within machine precision) of the same eigenvalue on the bad list, we keep the eigenvalue as good. However, if there is only one copy of an eigenvalue on the good list that matches (again, to within machine precision) one on the bad list, we throw away that eigenvalue. Finally, lone eigenvalues on the good list that have no match on the bad list are kept as good. Applying this test to a block implementation of the Lanczos algorithm would be tricky, given that we would expect truly good eigenvalues to be present as single and multiple copies in both the good and bad lists.

Our implementation of the BLA with full reorthogonalization sidesteps the problem of ghost eigenvalues. However, we are left with the matter of determining the following: at what distance from E do the eigenvalues cease to be sufficiently good approximations to the real eigenvalues of H?

Our test follows exactly that proposed by Barbour and co-workers [25]. At the end of j BLA recursions, we have the $(2j) \times (2j)$ dimensional block tridiagonal matrix, which we denote T_j. Because we have dilated the spectrum around E and explicitly reorthogonalized all of the successively generated sets of Lanczos vectors that constitute T_j, we are assured that ghosts are not present and that the Ritz values that have converged so far are very close to real eigenvalues of H, but only within a small region around E. This statement is not true for eigenvalues that lie at a distance from E. We define a truncated Lanczos matrix, T_{j-1} as

$$T_{j-1} = \begin{bmatrix} A_1 & B_2^T & & \\ B_2 & A_2 & B_3^T & \\ & \ddots & \ddots & B_{j-1}^T \\ & & B_{j-2} & A_{j-1} \end{bmatrix}.$$

Diagonalizing T_j and T_{j-1} yields good and bad lists, which both contain degenerate and nondegenerate eigenvalues. These numbers correspond to the shifted and inverted Ritz values; the backtransformed Ritz values are given by Eq. (1), where λ_L is an eigenvalue from either T_j or T_{j-1}.

Given two lists of Ritz values, we then proceed in testing for good eigenvalues. We first determine which eigenvalues in each list are degenerate, given a tolerance ϵ_d (this is for practical reasons, as we demonstrate in Sec. III: the algorithm can resolve Ritz values that differ by less than 1 part in 10^9). If there exists more than one eigenvalue that differs from another by less than ϵ_d, then we call those eigenvalues degenerate. Each list is then collapsed to contain one copy of each eigenvalue; the degeneracy of each is stored in another array to be referenced later. Finally, the good and bad lists are compared. If an eigenvalue from the good list differs by less than ϵ_k from an eigenvalue on the bad list, we report

both the good eigenvalue and its degeneracy. Specific choices for ϵ_d and ϵ_k will be presented in Sec. III.

E. The Hückel matrix for benzene: a simple example

Perhaps the easiest way to illustrate the advantages of using GFBLA over GFSLA is to examine the eigenvalues for the system of π -electron molecular orbitals in the benzene (C_6H_6) molecule, as treated by Hückel molecular orbital theory. No preconditioning steps were taken for these examples.

Here, a system of π electrons is considered to be delocalized over the rigid framework of the molecule as dictated by the σ electrons. It is assumed that each carbon atom contributes some amount of its π character to the molecular orbital, which can be expressed as

$$\Psi_j = \sum_{i=1}^{6} c_{ij}\phi_i ,$$

where ϕ_i represents carbon atom i's p_z atomic orbital and c_{ij} is analogous to a ''mixing coefficient.'' The atomic orbital basis set $\{\phi_i\}$ is assumed to be orthonormal.

The 6×6 Hamiltonian matrix is parametrized in terms of two quantities, α and β. The diagonal elements of the Hamiltonian matrix are denoted α. In the case of benzene, all the carbons are equivalent and thus all the diagonal matrix elements are equal. The off-diagonal elements of H are given by β, but only for nearest-neighbor interactions. Otherwise, the off-diagonal element is set to zero. For example, the (1,2) and (1,6) elements are both β but the (1,4) element is zero. The resultant Hamiltonian matrix is real and symmetric. Direct diagonalization gives 4 energy levels, the middle two of which are both doubly degenerate. The model Hamiltonian matrix is symbolically represented as

$$H=\begin{bmatrix} \alpha & \beta & 0 & 0 & 0 & \beta \\ \beta & \alpha & \beta & 0 & 0 & 0 \\ 0 & \beta & \alpha & \beta & 0 & 0 \\ 0 & 0 & \beta & \alpha & \beta & 0 \\ 0 & 0 & 0 & \beta & \alpha & \beta \\ \beta & 0 & 0 & 0 & \beta & \alpha \end{bmatrix}.$$

1. SLA results

When the GFSLA is used to reduce the Hückel matrix for benzene to a tridiagonal form, it does so by a series of matrix-vector multiplications and thus does not build in any information about the degeneracy of the energy levels in the system. We let $\alpha=4$ and $\beta=2$ in this example, knowing that the corresponding eigenvalues should be 0, 2, 2, 6, 6, and 8. We shifted the eigenvalue spectrum by letting $E=3.5$. Implementing the variation given above, we did 6 iterations ''by hand'' with MATHEMATICA. Because this example was done without reorthogonalization of each newly generated Lanczos vector against all previous ones, we must invoke the Cullum-Willoughby test for spurious eigenvalues before reporting our final list of converged eigenvalues. The results of diagonalizing T_j and T_j^* are given in Table I, where the eigenvalues of T_j and T_j^* have been backtransformed according to the prescription given in Eq. (1). The eigenvalues that pass the Cullum-Willoughby test are 8, 6, 2, and 0 (rounded). Each one is accurately represented in value, but information about degeneracies is missing. The implications of using the SLA to solve eigenproblems where multiplicities in the eigenvalue spectrum exist are clear: not only will these degenerate states go undetected, there is also the possibility of misidentifying a bad eigenvalue for that of a true one unless a test like that of Cullum-Willoughby is used. Thus, the SLA fails to properly preserve the most important information about systems where degeneracies exist.

TABLE I. Eigenvalues of the Hückel matrix for C_6H_6. Eigenvalues are for T_j and T_j^*. ‡ represents a backtransformed eigenvalue of $T_j^*=-\infty$.

λ, T_j	λ, T_j^*
8.0	-12.2538
6.21803	6.21803
6.0	6.93602
2.0	1.31776
1.98837	1.98837
-2.22045×10^{-15}	‡

However, if a full reorthogonalization is done at each step, we are left with the task of imposing criteria that determine whether or not multiple copies of an eigenvalue are to be called degenerate. MATHEMATICA gives 0, 1.97, 2, 6, 6.01, and 8 as the eigenvalues of T_j. We would most likely not accept the pairs (1.97, 2) and (6, 6.01) as degenerate: this decision is shown to be valid upon diagonalizing T_{j-1}, which yields 0, 1.91, 2, 6, and 8 as eigenvalues. Applying our test for accepting Ritz values results in reporting 0, 2, 6, and 8 as ''good.''

We see that either method is problematic where degeneracies are concerned: in the worst case (no reorthogonalization), degeneracies are missed entirely. In the best case (complete reorthogonalization), degeneracies are alluded to, but not resolved with acceptable accuracy.

2. BLA results

Conversely, the block Lanczos algorithm identifies the degenerate eigenvalues correctly, both in number and in quantity. We used a block size of 6×2 for the matrix Q, knowing in advance that the maximum degeneracy to be expected in this case was 2. After 3 iterations, the block algorithm yields the correct values for energies E_1 through E_4, giving (rounded) $\lambda_1=0$, $\lambda_2=\lambda_3=2$, $\lambda_4=\lambda_5=6$, and $\lambda_6=8$. Note that because only three iterations were done, it was unnecessary to reorthogonalize each successively generated block Q.

As long as Q is at least of dimension $N\times M$ (where N is the dimension of the original matrix and M is the maximum number of multiple eigenvalues expected), the BLA returns the proper number of eigenvalues as well as their values.

F. Preconditioning the initial Lanczos vectors

Roy and Carrington [20] proposed a method where the Lanczos process is ''guided'' to accelerate the convergence of Ritz values near an eigenvalue of interest. The authors investigated a spectral transform method [which uses a Gaussian filter for $f(H)$ to drive the Lanczos method] and a

scheme for computing an optimized starting vector by using a Gaussian-windowed Fourier-transform function for q_1. The latter of these methods was found to be the most attractive. The approach detailed in our work couples the spectral transform driver with a *block* of optimized initial Lanczos vectors.

The BLA is initiated with a set of orthonormal column vectors [18], which we refer to as Q_1. A common choice is one where each basis function is equally represented, and the inner product of the vectors is zero. We initially worked with the $N\times 2$ block,

$$Q_1=\begin{bmatrix} 1/\sqrt{N} & 1/\sqrt{N} \\ 1/\sqrt{N} & -1/\sqrt{N} \\ 1/\sqrt{N} & 1/\sqrt{N} \\ \vdots & \vdots \end{bmatrix},$$

noting that this combination is suitable only where N is even (which can easily be arranged for any case). Filling Q_1 with randomly generated numbers and then doing a QR factorization is also acceptable. However, *any way* we arbitrarily fill Q_1 is unfavorable for accelerating the convergence of the Lanczos process.

Therefore, it makes sense to precondition Q_1 so that, in a sense, the vectors it contains are already ''leaning'' in the direction of the eigenvectors around E. We can then invest a portion of the computational time and resources saved by reducing the number of Lanczos iterations in other tasks, such as factoring $EI-H$ or performing detailed analyses on the backtransformed Ritz values and vectors.

The preconditioning operator $p(E)$ acts upon a primitive initial block, X_1, P times:

$$p(E)^P X_1=Q_1,$$

where an excellent choice would be $p(E)\equiv G(E)$ in cases where $EI-H$ can be factored and X_1 is any block of orthonormal column vectors. When this process is finished, we then start the Lanczos loop with the preconditioned block Q_1. For a given number of Lanczos steps, even one loop through the preconditioner yields more converged eigenvalues than without it. We refer to this scheme as PC-GFBLA.

However, there will certainly be cases where we just cannot factor $EI-H$ because of its size. We can estimate $G(E)$ and then apply it several times to X_1 through a series of matrix-matrix multiplications to arrive at a preconditioned starting block (this method is called the estimated inverse GFBLA, or EI-GFBLA), now denoted Q_1. One method for generating a model Green function is as follows. Define a submatrix S of $EI-H$ of dimension p, which is centered around the diagonal elements closest to E. We approximate the diagonal elements of $G(E)$ that fall outside the borders of S by $(EI-H_{ii})^{-1}$. Any element of $(EI-H)$ that is not within S or on the diagonal is set to zero; S is directly inverted. The resultant estimated inverse, $G^0(E)$, becomes $p(E)$ and is used for the preconditioning. The matrix-matrix product $G^0(E)Q_j$ in the Lanczos loop would be constructed by using a linear system solver [such as GMRES (generalized minimun residual) [26] or DIIS (direct inversion in the iterative subspace) [27]] or by using perturbative matrix partitioning techniques [1] at the beginning of each loop. Below, we show a sparse 10×10 matrix from which $G^0(E)$ is built for $E=3.5$ and $p=4$. The submatrix S encompasses the $(5,5)\rightarrow(8,8)$ elements of $f(E)$:

$$H=\begin{bmatrix} 1 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0.25 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0.1 & 0 & 0 & 0 & 0.1 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.5 & 0 & 0 & 0 & 3 & 0.05 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.1 & 0 & 0.05 & 3 & 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.1 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 5 & 0 \\ 0.25 & 0 & 0.1 & 0 & 0 & 0 & 0 & 0 & 0 & 5 \end{bmatrix},$$

$$G^0(E)=\begin{bmatrix} 0.4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.66 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.66 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2.02 & 0.19 & 0 & -0.04 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.19 & 1.94 & 0 & -0.38 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -0.04 & -0.38 & 0 & -1.92 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.66 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.66 \end{bmatrix}.$$

The quality of the approximation is largely a function of the size of the matrix elements of H that are excluded from S, the smaller block that we choose to directly invert. This is illustrated for the present case by examining $G^0(E)(EI-H)$:

$$G^0(E)(EI-H)=\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.99 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.99 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.01 & -0.0095 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -0.0095 & 0.97 & 0 & 0.038 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.038 & 0 & 0.96 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.99 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.99 \end{bmatrix}.$$

G. Computational details

We followed the basic outline of Grimes and co-workers [18] in implementing the block Lanczos code. The Green-function filter scheme follows that of Wyatt [1] and Ericsson and Ruhe [3]. The entire Hamiltonian matrix is factored in the present work: we were able to run test cases for $N \leqslant 3000$ on our IBM RS-6000/370 workstation. All matrix operations are handled by subroutines from BLAS (basic linear algebra subprograms) and LAPACK (linear algebra package) [28], with the exception of the modified Gram-Schmidt and QR routines, which were hand coded for future ease in parallelizing the code. An outline of our code is given below. To initialize the procedure, we set $Q_0=B_1=0$ and define X_1 to have orthonormal columns.

A. Form $(EI-H)$.

B. Do an LU factorization of $(EI-H)$.
Start preconditioning loop: $P=1,2,\ldots$.

C. Solve $p(E)^P X_1=Q_1$ for Q_1.
End preconditioning loop.
Start Lanczos loop: $j=1,2,\ldots$.

D. Given Q_j, form $V_j=G(E)Q_j$ by solving $(EI-H)V_j=Q_j$.

E. $U_j=V_j-Q_{j-1}B_j^T$.

F. $A_j=Q_j^T U_j$.

G. $R_{j+1}=U_j-Q_jA_j$.

H. Do a QR factorization: $R_{j+1}=Q_{j+1}B_{j+1}$.

I. Orthogonalize columns of Q_{j+1} against all previous Q_j.
End Lanczos loop.

J. Compute eigenvalues for both T_j and T_{j-1}.

K. Test each eigenvalue list for degeneracies: tolerance factor ϵ_d.

L. Collapse each list to one copy of each eigenvalue, retaining degeneracy.

M. Keep eigenvalues that appear on both lists: tolerance factor ϵ_k.

With each pass through the Lanczos loop, we create a set of vectors Q_j. In order to perform our complete reorthogonalization at each step, we store the new set after step I above.

III. COMPUTATIONAL RESULTS

The focus of this study is to find an efficient way to accurately extract eigensystem information from a Hamiltonian matrix that may contain degeneracies (here, limited to a two-fold degeneracy). We are therefore concerned with the following issues: how well do the Ritz values and Ritz vectors agree with direct diagonalization results? How many eigenpairs do we converge (given ϵ_k and ϵ_d) for a given number of Lanczos steps? Is there a significant advantage to preconditioning? And finally, how much time does it take to converge an acceptable number of eigenpairs?

The choice of an approprite block size M is a consideration when implementing the block Lanczos algorithm. Conceivably, one could reduce the number of Lanczos steps by increasing the size of M: for instance, instead of doing 10 Lanczos steps with $M=2$, do 5 steps with $M=4$. We are then faced with solving a linear system with more right-hand sides than before, but we do not have to do it as many times as with $M=2$. The matrix-matrix multiplications within the Lanczos loop take more time, but again, we do not do as many passes through the loop as with the smaller block size.

Golub and Underwood [16] conclude ''it is best to choose the block size at least as large as the largest multiplicity possessed by any eigenvalue'' of H. Ruhe [17] agreed, saying that the block size ''should be chosen as the number of independent eigenvectors sought to any cluster of interesting eigenvalues.'' Grimes and co-workers [18] state that ''in general, it is best to choose a block size as large as the largest expected multiplicity if eigenvalues of moderate multiplicities are expected. This is particularly important if many clusters of eigenvalues are expected,'' as in our test cases. They found that the cost of a Lanczos run initially decreases with increasing block size (fewer factorizations, linear system solves, and matrix-matrix multiplications), but then *increases* because the dimension of the matrix containing the Lanczos vectors also increases. It is also noted that maintaining a degree of orthogonality (semiorthogonality in [18], full orthogonality in the present study) and backtransforming the Ritz vectors are substantial contributors to the overall cost. They concluded that they did not ''see an optimal choice for

block size.'' Because we will always know the maximum expected multiplicity for a Hamiltonian matrix generated to describe a system of a given symmetry, we can follow the advice presented above and restrict M to equal that quantity.

The following subsections describe the model Hamiltonian, the convergence of Ritz values, and how we compute Ritz vectors of H. We demonstrate that Ritz values computed with the GFBLA agree extremely well with direct results. When preconditioning steps are invoked, the number of converged Ritz values increases substantially at an almost unnoticeable increase in computational time. The PC-GFBLA is shown to be superior to the EI-GFBLA (for cases where $EI-H$ can be factored). The effects of preconditioning are evident in plots of expansion coefficients for several Lanczos vectors in the eigenvector basis of H (obtained from a direct diagonalization). Ritz vectors are tested by calculating the overlap coefficients for each Ritz vector in the entire eigenvector matrix of H. We show that preconditioning greatly enhances not only the number of converged good Ritz values, but also their corresponding Ritz vectors [20].

Finally, we comment on results presented in the following sections concerning the timing of our code on the RS-6000. The program readable system clock reports execution times to 0.1 s. For this reason, the execution times reported in the present study are only valid to within ± 0.1 s.

A. Model Hamiltonian

In practice, we will be extracting eigenvalues from Hamiltonian matrices that are very sparse: typically, up to about 6% of the matrix elements will be nonzero. Further, the off-diagonal elements are usually between 0.01% and 0.1% of the smallest diagonal element. Preliminary tests of the code showed that the number of converged Ritz values was not a strong function of coupling strength. We chose the following model for this study:

$$H_{i,i}=H_{i+1,i+1}=i,$$

where $i=1,N$ in increments of 2

$$H_{i,j}=e^{-10R},$$

where $i\neq j$ and $R\in[0,1)$ as generated by a random number generator. E and the placement of off-diagonal elements are also randomly determined. Because we are only interested in the interior region of the eigenspectrum, we enforce the shift to be $E=(N/2)R$, $R\in[0,1)$ as generated by a random number generator.

B. Convergence of Ritz values

For this study, we ran the GFBLA code as described above with the option of comparing Ritz values and Ritz vectors with those from a direct diagonalization of H, which is obtained by calling DSPEV (LAPACK [28]). The direct list is subjected to the same sorting criteria as the GFBLA lists. Hence, we can see how well our computed eigenvalues and degeneracies match the results of a direct diagonalization. Presented in Table II are the results of a small test case ($N=500$) where we have reported several converged eigenvalues around E for the matrices T_j, T_{j-1}, and H. The full and truncated Lanczos eigenvalue lists have been subjected to the test for multiplicity; the direct results, in the last column, have only been sorted. The preconditioning step was not invoked for this example. Clearly, degeneracy is in the eye of the beholder: only the imposition of ϵ_d results in our code declaring two Ritz values as degenerate. Without this condition, the code resolves the nearly degenerate Ritz values with the same accuracy as it does the other Ritz values. When we impose ϵ_k (for this study, we chose 10^{-10}), the list in Table II yields 10 acceptable eigenvalues, all of which agree with direct diagonalization results to at least 8 decimal places.

TABLE II. Comparison of select Ritz values near $E=56.289$. $N=500$, $L=20$, $\rho=1\%$, and $\epsilon_d=10^{-5}$. Ritz values are nondegenerate except where noted by ‡, which are twofold degenerate as determined by ϵ_d. All lists were subjected to the same sorting criteria. Eigenvalues for H are from a direct diagonalization. No preconditioning was used for these cases.

λ, T_j	λ, T_{j-1}	λ, H
51.013 538 70	51.013 538 70	51.013 538 70
52.996 856 31	52.996 856 31	52.996 856 31
52.999 667 62	52.999 667 62	52.999 667 62
54.998 804 31	54.998 804 31	54.998 804 31
54.999 981 14	54.999 981 14	54.999 981 14
‡56.999 998 04	‡56.999 998 04	56.999 998 04
56.999 999 96	56.999 999 96	56.999 999 96
‡58.999 999 94	‡58.999 999 94	58.999 999 94
58.999 999 99	58.999 999 99	58.999 999 99
60.999 988 41	60.999 988 41	60.999 988 41
61.003 857 44	61.003 857 44	61.003 857 44
62.995 561 38	62.995 562 11	62.995 561 30

Four variations of the block Lanczos code were tested to illustrate the advantages of filtering and preconditioning: (i) no filter and no preconditioning; (ii) filter and no preconditioning; (iii) preconditioning and no filter; and (iv) preconditioning and filter. We chose $N=1000$, $\rho=1\%$, $\epsilon_k=10^{-10}$, and $\epsilon_d=10^{-5}$ for the tests. Each case was timed based on the first appearance of converged Ritz values. The results are given in Table III. For each case given above, every pass through the Lanczos loop involves a minimum of 6 matrix-matrix multiplications, a QR factorization, and a modified Gram-Schmidt operation. The last two operations scale as N^2; the matrix-matrix multiplications are much more costly, scaling as N^3 [29]. If we precondition or filter, we introduce

TABLE III. Timing for different implementations of preconditioning and filtering with the block Lanczos algorithm. The number of Lanczos steps L is given for the first appearance of converged Ritz values. $N=1000$, $\rho=1\%$, $\epsilon_d=10^{-5}$, and $\epsilon_k=10^{-10}$ for all cases. $P=10$ for (iii) and (iv). Time is user time, given by executing /usr/bin/time-p a.out on the RS-6000; error is ± 0.1 s.

Method	L	t (s)
(i) No filter, no preconditioning	114	94.8
(ii) Filter, no preconditioning	8	13.7
(iii) No filter, preconditioning	3	12.4
(iv) Filter, preconditioning	2	12.4

TABLE IV. Timing and convergence: PC-GFBLA vs EI-GFBLA. $N=500$, $L=20$, $\rho=1\%$, $\epsilon_d=10^{-5}$, and $\epsilon_k=10^{-10}$. The directly inverted block size for the EI-GFBLA was fixed at 100 and $E=250.056$. Time is user time, given by executing /usr/bin/time-p a.out on the RS-6000; error is ± 0.1 s. Values in parentheses are for the EI-GFBLA cases.

P	No. converged	t (s)
0	10	3.8
1	12 (12)	3.8 (8.6)
2	12 (12)	3.9 (8.6)
3	12 (12)	3.9 (8.5)
4	14 (12)	3.8 (8.5)
5	14 (12)	4.0 (8.6)
6	16 (12)	4.0 (8.7)
7	16 (12)	4.0 (8.7)
8	16 (12)	4.0 (8.8)
9	16 (12)	4.1 (8.8)
10	18 (11)	4.2 (8.8)

a factorization step and a linear system solve step, each of which require N^3 operations. It should be noted, however, that we only need to factor once. Further, each pass through the Lanczos loop is as costly as 7 preconditioning steps.

Given that we are interested in some interior region of the eigenspectrum, we can dismiss methods (i) and (iii) because only the extreme Ritz values and vectors will converge when we do relatively few Lanczos iterations (interior eigenvalues will converge eventually, but the cost of doing so many iterations makes the approach unattractive). We are left with deciding which of methods (ii) (GFBLA) or (iv) (PC-GFBLA) is more effective. As seen in Table III, 10 preconditioning steps reduce the number of Lanczos iterations required to resolve the first appearing converged Ritz values by a factor of 4. The savings in terms of computational time amounts to slightly more than 10%. But why, one might ask, should we precondition at all? If we target a certain quantity of Ritz values we want converged, certainly we can just do more Lanczos steps. For example, it would seem reasonable to demand 50 converged Ritz values for a single run of the code. For the $N=1000$ case used above, the PC-GFBLA converges 50 Ritz values in 43 s ($L=50$ here); the GFBLA needs 7 more Lanczos steps to reach the same goal, for a total cost of 48 s. The time demand is now at more than 11%. When this savings is taken with the improvement seen in the convergence of Ritz vectors (Sec. III D, below), the PC-GFBLA is clearly superior to the GFBLA.

We also see a noticeable change in the extent of convergence when preconditioning steps are invoked (we denote the number of preconditioning steps by P). When using the GFBLA with small ($N \leqslant 3000$) matrices, it makes sense to factor the whole shifted matrix and precondition by solving the linear algebra problem that is also solved with each Lanczos step (the PC-GFBLA approach). The EI-GFBLA scheme, useful for situations where factoring is not possible, is slower than the PC-GFBLA and does not increase convergence as quickly as PC-GFBLA. Table IV illustrates this point. We actually see a *decrease* in the number of converged Ritz values for the $P=10$ step for the EI-GFBLA

TABLE V. Block size vs convergence for EI-GFBLA. $L=10$, $\rho=1\%$, $\epsilon_d=10^{-5}$, and $\epsilon_k=10^{-10}$. For $N=1000$, $E=500.056$. For $N=2000$, $E=1000.056$. $P=10$ for all cases. Time is user time, given by executing /usr/bin/time-p a.out on the RS-6000; error is ± 0.1 s.

p	No. converged	t (s)
	$N=1000$	
100	4	49.5
200	4	49.7
300	4	50.8
400	4	52.8
500	4	55.3
600	4	57.2
700	4	61.7
800	5	67.7
900	6	74.9
1000	8	86.1
	2000	
1000	4	139.4
1200	4	149.3
1400	4	185.9
1600	4	259.7
1700	4	261.7
1800	5	295.0
1900	8	331.8
2000	8	369.2

with a constant block size p (accompanied by an increase in computational time). The PC-GFBLA converges 80% more Ritz values with $P=10$ than for $P=0$, with only an 11% increase in computational time. A 42% increase in time for the EI-GFBLA leads to a gain of only one more Ritz value.

However, this is not generally the case, since we would make p, the dimension of the submatrix S that gets directly inverted, greater than $N/5$ for practical purposes. Still, the number of converged Ritz values is not as strong a function of the number of preconditioning steps, P, for the EI-GFBLA as it is of the dimension of S. In Table V, we show that no advantage is gained by using the EI-GFBLA until S is of dimension (at least) $p \leqslant 0.8N$. If one is willing (or is forced) to invest the time required for the EI-GFBLA, one need not sacrifice accuracy: Table VI compares Ritz values obtained with this method with those from a direct diagonalization. Given the small return on the time investment, and the fact that the full reorthogonalization scheme already gives us an acceptable number of converged Ritz values, the EI-GFBLA is hardly an attractive preconditioner.

The PC-GFBLA is clearly superior if the problem is small. The sparsity of the matrix appears to have little, if any, effect on convergence (Table VII) (the same was true for EI-GFBLA). Test cases for $500 \leqslant N \leqslant 3000$ show that for 10 preconditioning steps, the most we increase user time is by 16% (which, for the case in question, converges four times the number of Ritz values that we would get if we did not precondition). The largest return on our investment was for the $N=3000$ case, where time increased 5% and we converged 8 Ritz values instead of 2 (Table VIII).

TABLE VI. Converged eigenvalues: EI-GFBLA vs direct. $N=1000$, $L=10$, $\rho=1\%$, $\epsilon_d=10^{-5}$, $\epsilon_k=10^{-10}$, $E=500.056$, and $P=10$. The six Ritz values converged for the EI-GFBLA in 75.0 s. The block size $p=900$ for the submatrix S. Ritz values noted by ‡ converged when the preconditioning steps were left out ($t=47.8$ s). Time is user time, given by executing /usr/bin/time-p a.out on the RS-6000: error is ± 0.1 s.

EI-GFBLA	Direct
497.001 403 05	497.001 403 05
‡498.999 074 54	498.999 074 54
‡499.001 448 02	499.001 448 02
‡500.999 788 41	500.999 788 41
‡501.001 406 92	501.001 406 92
502.984 602 22	502.984 602 22

The excellent agreement between Ritz values returned from the PC-GFBLA and a direct diagonalization of H attest to the advantage of using both the Green-function filter and full reorthogonalization at each Lanczos step. The ghost eigenvalue problem is avoided, while the block structure of the BLA allows us to resolve degenerate (according to our criteria), nearly degenerate, and nondegenerate eigenvalues of H near E. If $EI-H$ can be factored, then the advantages of preconditioning are maximized in terms of the number of converged Ritz values. If we fix the number of Ritz values we want to see converged, we must invest 10% more time in resolving these quantities without preconditioning than with preconditioning. If $EI-H$ cannot be factored, preconditioning can still be done, but the return on user time invested per converged Ritz values makes this method (EI-GFBLA) less attractive than PC-GFBLA.

C. Eigenvector composition of the Lanczos vectors

The effect of using a preconditioner is most clearly illustrated when we plot the expansion coefficients of select Lanczos vectors in the eigenvector basis of H. For a given number of Lanczos recursions, L, we have an $N\times 2L$ array of orthonormal Lanczos vectors, $\{Q_L\}$. For Lanczos vector j, we let

TABLE VII. Converged Ritz values for the PC-GFBLA: number of preconditioning steps (P) vs sparsity (ρ) of H. $N=1000$, $L=20$, $\epsilon_d=10^{-5}$, and $\epsilon_k=10^{-10}$.

P/ρ	1%	5%	10%
0	10	10	10
1	10	10	10
2	12	12	12
3	12	12	12
4	14	14	14
5	14	14	14
6	14	15	14
7	16	16	16
8	16	16	16
9	18	18	18
10	18	18	18

TABLE VIII. Timing and convergence: PC-GFBLA. $L=10$, $\rho=1\%$, $\epsilon_d=10^{-5}$, $\epsilon_k=10^{-10}$, and $P=10$. Numbers in parentheses are for GFBLA with no preconditioning. Time is user time, given by executing /usr/bin/time-p a.out on the RS-6000; error is ± 0.1 s.

N	Converged	t (s)
500	8 (2)	2.9 (2.5)
1000	8 (2)	14.6 (13.2)
1500	6 (2)	39.8 (36.2)
2000	6 (2)	84.9 (78.1)
2500	8 (2)	157.1 (140.2)
3000	8 (2)	323.6 (308.5)

$$c_i=(Q_j^T Z_i)^2 \quad (i=1,2,...,N), \tag{2}$$

where Q_j is a Lanczos vector and Z_i is an eigenvector of H.

If we do not precondition, then Q_1 is simply

$$Q_1=\begin{bmatrix} 1/\sqrt{N} \\ 1/\sqrt{N} \\ 1/\sqrt{N} \\ \vdots \end{bmatrix}.$$

If we use Eq. (2) to plot c_i for Q_1, as is shown in Fig. 1(a), we see that the expansion coefficient is more or less a constant over the entire range of eigenvectors $\{Z_i\}$. That is, there are almost equal amounts of every eigenvector of H in Q_1, which is not a good way to start if we want to make the most of a few Lanczos steps. The case shown in Fig. 1(a) is replotted in Fig. 1(b), with the important modification that we have done one preconditioning step. Not only has c_i grown in the region around eigenvectors near E, but c_i is now much smaller for every eigenvector of H some distance away from E than it was previously: $-15\leqslant \ln c_i \leqslant -12$ now, whereas $\ln c_i \approx -7$ without preconditioning.

We can also examine other Lanczos vectors from $\{Q_L\}$. Each Lanczos step gives us a set of two new vectors, Q_{new}. The vectors are orthogonal to each other and to all previously generated Lanczos vectors. There should be no component of any converged eigenvector (which would be associated with a previous vector or set of vectors Q_{previous}) in Q_{new}. As successive Q_{new}'s are formed, we should be working our way out from E and into the surrounding eigenspectrum. Figures 2(a)–2(d) further illustrate this point: expansion coefficients for Q_1, Q_5, Q_{10}, and Q_{40} in the eigenvector basis are shown, where $N=1000$, $L=20$, $\rho=1\%$ and the number of preconditioning steps, P, is fixed at 10. A sharp spike in Fig. 2(a) corresponds to Q_1 having significant components of an eigenvector whose eigenvalue is closest to E (here, $E=112.579$). Figure 2(b) shows the peak around E widening as Q_5 now ''picks out'' eigenvectors a little further from the previously converged ones. This broadening continues as we look at Q_{10} [Fig. 2(c)] and Q_{40} [Fig. 2(d)], the last Lanczos vector for this test case. Besides the growing area being covered around E, we also see the growth of a deep trough in the large peak. This feature is a remnant of our full reorthogonalization scheme: the width of

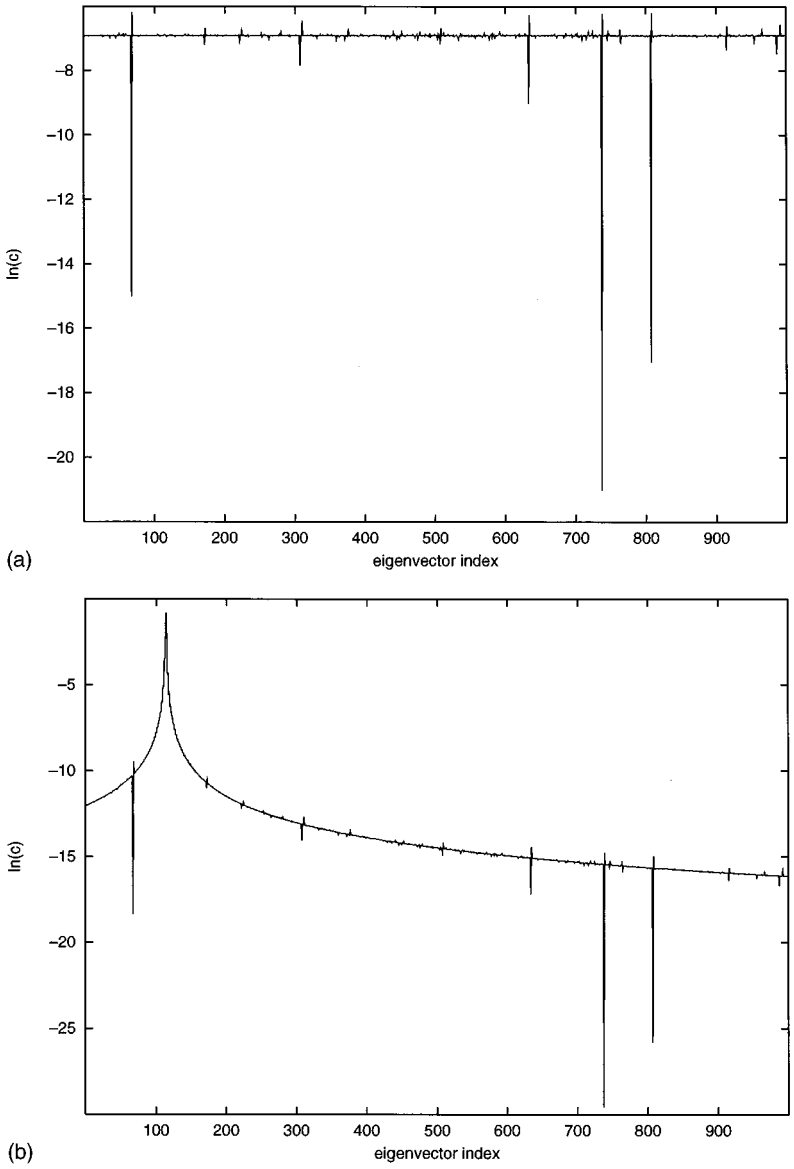

FIG. 1. Expansion coefficients for Lanczos vector Q_1 in the eigenvectors of H: (a) no preconditioning ($P=0$), (b) one preconditioning step ($P=1$). $N=1000$, $L=20$, and $\rho=1\%$.

the trough describes a bandwidth, within which we have converged all the eigenvectors closest to E for a given number of Lanczos steps.

D. Computation of Ritz vectors from the Lanczos vectors

After a number of Lanczos steps (L), we have constructed the $(2L)\times(2L)$ projection of $G(E)$ in the Lanczos basis:

$$Q^T G(E) Q = T. \tag{3}$$

Diagonalizing the block tridiagonal matrix T yields

$$S^T T S = \lambda \Rightarrow T = S \lambda S^T$$

and, using Eq. (3), we see that

$$(QS)^T G(E)(QS) = \lambda.$$

Thus, $V=QS$ is the $N\times 2L$ matrix of Ritz vectors of $G(E)$.

It is instructive to look at the overlap of a Ritz vector (denoted V_i) versus the corresponding eigenvector of H (denoted Z_i), as determined from a direct diagonalization. The better our methods, the closer to unity will be what we define as s_i, the overlap coefficient:

$$s_i = \sum_{j=1}^{N} V_i^T Z_j. \tag{4}$$

Further, if V_i is very close to one eigenvector and nearly orthogonal to the others,

$$|s_i| \approx 1 \tag{5}$$

for any i (we take the absolute value because the phase of the eigenvector varies). Because of the global reorthogonalization, we would expect most s_i to be close to unity for some distance away from Z_{shift}.

The effect of preconditioning is immediately noticeable if we plot $|s_i|$ for different Ritz vectors V_i. In Fig. 3(a), we show such a plot for a case where $N=1000$, $L=20$, $\rho=1\%$, $\epsilon_d=10^{-5}$, $\epsilon_k=10^{-10}$, and $P=0$ (that is, no preconditioning). Working our way in from the ends of the plot, we

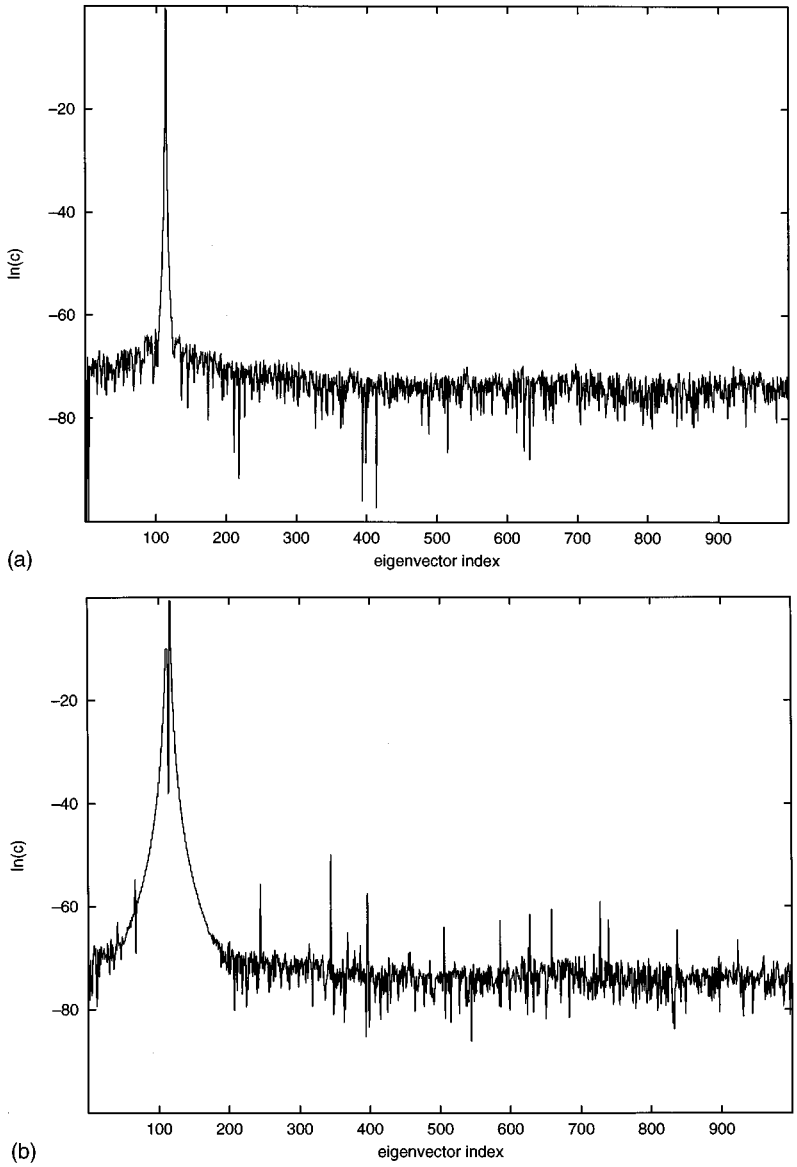

FIG. 2. Expansion coefficients of preconditioned Lanczos vectors Q_i in the eigenvectors of H: (a) Q_1, (b) Q_5, (c) Q_{10}, (d) Q_{40}. $N=1000$, $L=20$, $\rho=1\%$, and $P=10$.

see that $|s_i|\approx 1$ for only about 17 of our backtransformed Ritz vectors (the ends of the plot correspond to converged eigenvectors closest to the shift). The width of the middle region of the plot indicates that many (slightly over 20) of our backtransformed Ritz vectors have significant components of the other eigenvectors: predictably, these Ritz vectors are at some distance away from Z_{shift}. The picture changes when we look at Fig. 3(b), which was produced under the same conditions as Fig. 3(a) with the exception that 10 preconditioning steps were taken ($P=10$). Now, about 25 backtransformed eigenvectors satisfy Eq. (5), a direct result of preconditioning. There is now a wider interval around E within which our backtransformed Ritz vectors are ''pure'' (i.e., they have very small components of the other eigenvectors that encompass the space of H).

E. Error bounds on eigenpairs of T

There is (at least) one more test that illustrates the quality of our preconditioning method. Similar to the calculations performed before computing a Ritz vector in the selective reorthogonalization scheme of Parlett and Scott [13], we can use the block variant [18] to monitor the convergence of Ritz values and their associated Ritz vectors. Recalling the notation used above, we denote a backtransformed Ritz pair as λ_R, V. The quality of the approximation of a Ritz pair to the true, corresponding eigenpair of H is [18]

$$\|HV_j - V_j\lambda_{j(R)}\|_2 = \|B_{j+1}S_j\|_2, \tag{6}$$

where the last M elements of the jth column of S are used in Eq. (6) (here, $M=2$). The 2×2 matrix B_{j+1} is not used in forming T but is computed during the last Lanczos recursion before we terminate the run. Equation (6) is more useful if we implement the Lanczos algorithm without shifting and inverting: we introduce error by backtransforming Ritz values and vectors and need to account for these sources accordingly. Following Ericsson and Ruhe [3], we can instead use the eigenvalue error bound:

$$\gamma \equiv |\lambda_H - \lambda_R| \leqslant \frac{\|B_{j+1}S_j\|_2}{\lambda^2_{j(T)}}, \tag{7}$$

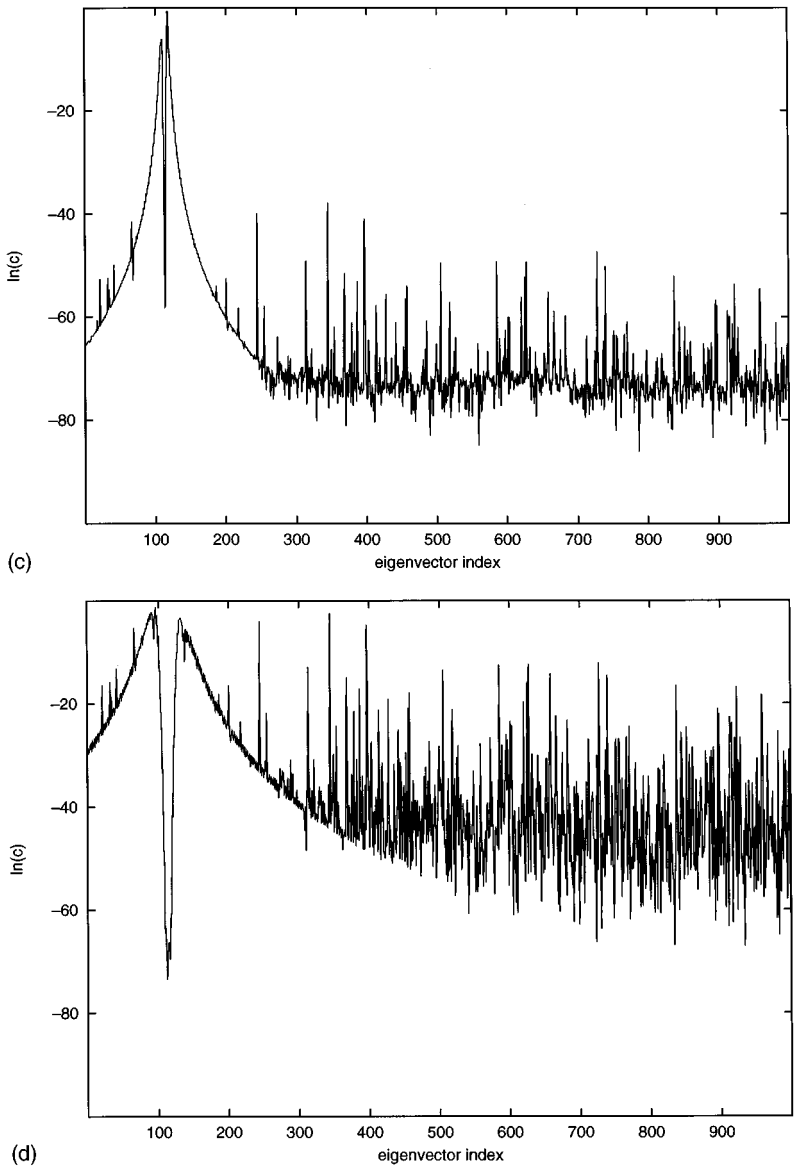

FIG. 2 (*Continued*).

where λ_T is an eigenvalue of T (not yet backtransformed). Equation (7) tells us that either large values of λ_T (which are located near E) or small values of $\|B_{j+1}S_j\|_2$ will minimize γ, the difference between an eigenvalue of H and a computed Ritz value.

In Sec. III D, we plotted $|s_i|$ for $\{V_i\}$, the set of Ritz vectors, and showed that the Ritz vectors near the ''edges'' of the plots [Figs. 3(a) and 3(b)] correspond to very good approximations of eigenvectors of H. Based upon the relationship expressed in Eq. (6), this is one way of saying that if the last M components of the eigenvector S_j are small, then the Ritz vector that is constructed with that S_j and the corresponding set of Lanczos vectors has converged. We now ask the following: what are the minimum and maximum values that the left-hand side of Eq. (7) can take? If we know the answer to this question, then we have yet another tool available for determining the accuracy of our methods and the efficacy of our preconditioning strategy.

The two cases used to plot Figs. 3(a) and (b) are now used to show how the error bound, γ, is improved upon preconditioning the starting block of Lanczos vectors. As in the previous sections, the effect is clearly seen: Figure 4(a) shows a plot of $\ln \gamma_i$ versus the index i on the eigenvalue $\lambda_i(T)$ for the case where no preconditioning steps were taken, and Fig. 4(b) illustrates how much γ_i decreases when $P=10$. Because we save all our Lanczos vectors and can quickly and easily compute the eigenvalues and eigenvectors of T, the test outlined above is available for our use at almost no extra computing cost. If we choose a γ that is acceptable for the range of eigenpairs of H we wish to compute, then we have a control on convergence that double checks our choices of ϵ_d and ϵ_k. For instance, if we were to accept as converged a Ritz eigenpair for which $\ln \gamma \leqslant -10$, then the 10 preconditioning steps yield 18 acceptable eigenvalues λ_T that would be backtransformed and reported as good approximations (along with their corresponding Ritz vectors) to an eigenpair of H. If we neglect to precondition, only 14 eigenvalues of T (and thus 14 eigenvectors of T) meet our criterion. Recalling the results presented in Table VII, in which 10 and 18 Ritz values were reported as acceptable for the $P=0$ and $P=10$ cases, respectively, we see that our choice of γ is too big. If we make $\ln \gamma \leqslant -11$, then we have agree-

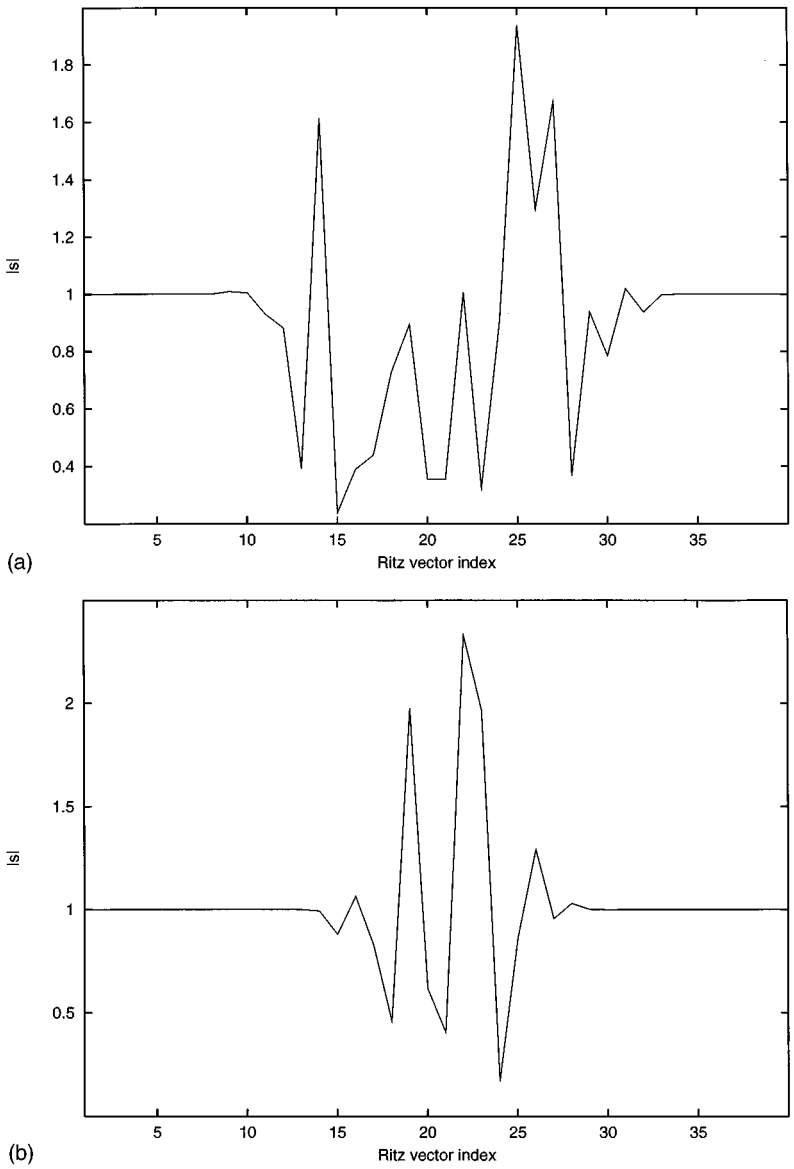

FIG. 3. Overlap coefficients of Ritz vectors $\{V\}$: (a) $P=0$; (b) $P=10$. $N=1000$, $L=20$, $\rho=1\%$, $\epsilon_k=10^{-10}$, and $\epsilon_d=10^{-5}$.

ment with the results presented in Table VII. This procedure can be used either in-line as in the selective reorthogonalization scheme [13] or, as presented here, as a final check on how many eigenpairs of T are acceptable Ritz pairs.

F. The water trimer

1. Introduction and model

Recently, a number of experimental and theoretical studies have focused upon the spectra of water clusters [30–40]. These studies provide valuable insights into the transition between small water clusters, characterized by a few vibrational degrees of freedom, and bulk water, whose properties are frequently modelled by statistical mechanics. The recent far-infrared studies by Saykally and co-workers on the vibration-rotation-torsion transitions of water clusters in jet-cooled expansions have provided a wealth of data that may lead to a greatly improved understanding of the intermolecular potential surfaces governing the cluster dynamics [30–36]. The low-frequency quantum tunneling and torsional excitations of the water trimer, $(H_2O)_3$, have been the focus of some of the studies [36,37].

In its equilibrium geometry, three OH bonds tend to lie in the plane defined by the three O atoms, which lie near the vertices of an equilateral triangle. In the equilibrium structure [38,39,41], the other three H atoms lie with two on one side and one on the opposite side of the oxygen plane, such as the up-up-down geometry, denoted uud. One of the H atoms, say the first one that is pointed up, can flip to the down position by rotating about the planar O-H bond, leading to the isoenergetic structure denoted dud. Other hydrogen atom flips about the planar O-H bonds lead to a multiple-minima potential energy surface. These low-frequency hydrogen-atom torsional energy levels are labeled a_g, a_u, e_g, or e_u, according to the irreducible representations of the group C_{3h}.

As an application of the PC-GFBLA to a nontrivial problem in molecular spectroscopy, we will calculate the $2e_g \leftarrow a_g$ torsional transition for the water trimer. The e_g representation is doubly degenerate; a_g is nondegenerate. Details concerning the coordinates, Hamiltonian operator, pseudospectral basis sets, and symmetry adaptation are presented in the study by Guiang and Wyatt [42]. However, for

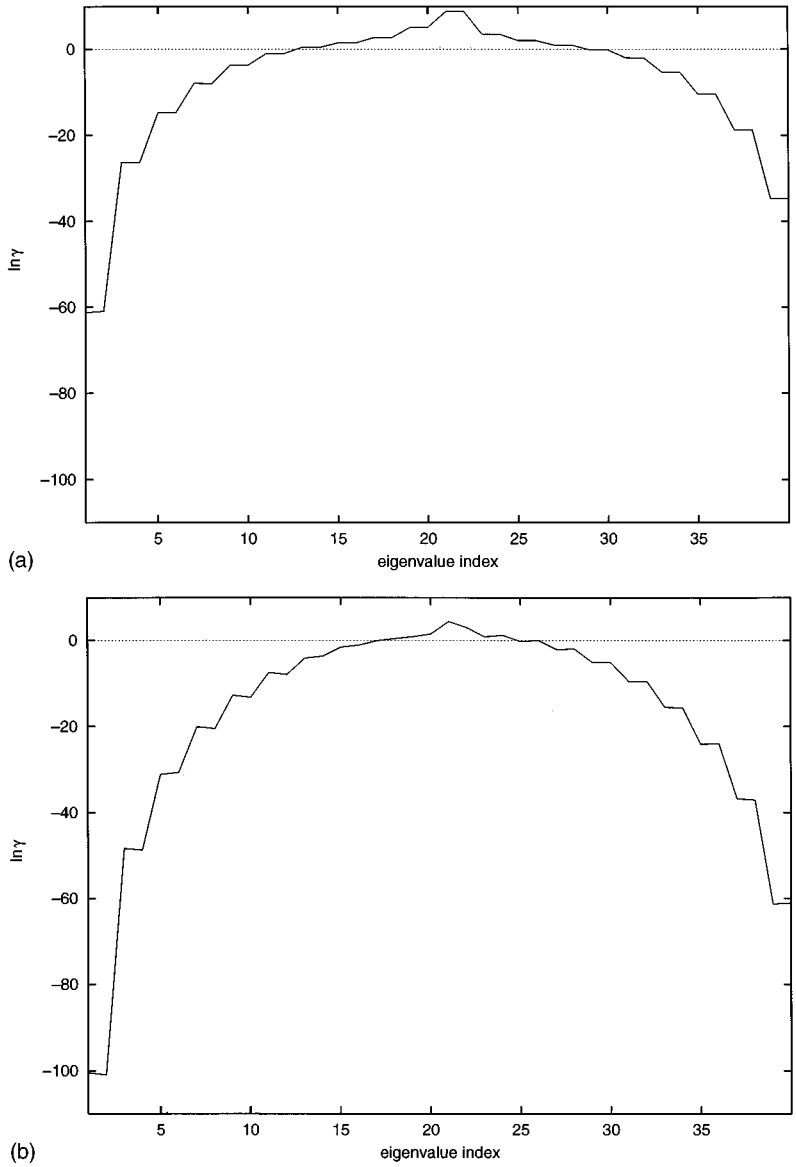

FIG. 4. Error bounds for λ_T: (a) $P=0$; (b) $P=10$. $N=1000$, $L=20$, $\rho=1\%$, $\epsilon_k=10^{-10}$, and $\epsilon_d=10^{-5}$.

completeness, a brief overview is presented here.

The three torsional coordinates are denoted $\{\chi_1,\chi_2,\chi_3\}$ and the torsional Hamiltonian operator is given by [40]

$$H=-B\left[\frac{\partial^2}{\partial\chi_1^2}+\frac{\partial^2}{\partial\chi_2^2}+\frac{\partial^2}{\partial\chi_3^2}\right]+V(\chi_1,\chi_2,\chi_3),$$

where B is an effective torsional constant and $V(\chi_1,\chi_2,\chi_3)$ is the three-dimensional torsional potential energy surface. In order to develop this potential surface, a series of *ab initio* electronic structure calculations were performed [42] and the results were corrected for basis set superposition error by employing the counterpoise correction method [43,44]. The corrected energies were then fit to an 11-parameter functional form, the CKL (Cieplak-Kollman-Lybrand) potential [45]. The basis set was constructed from the direct product of three one-dimensional basis sets, each one of which applies to a single torsional coordinate. The pseudospectral technique was used to construct the basis set for each torsional coordinate [46]. Along each of these coordinates, an odd number of grid points (denoted N_p) was used, and associated with each of these points is a pseudospectral basis function, $f_j(\chi_k)$. The functions that were used are defined in Vincke *et al.* [46]. Finally, from the large direct product basis set (of dimension N_p^3), projection operators [47] were used to build four symmetry adapted subspaces, one for each of the irreducible representations of C_{3h}. Only the e_g and a_g subspaces are considered in the present study.

2. Results

Each of the subspaces was examined for two basis set sizes. The resulting Hamiltonian matrices were of dimension $N=2280,3080$ for the two e_g cases and $N=1150,1551$ for the corresponding a_g cases. The PC-GFBLA parameters were set to $P=10$, $\epsilon_k=10^{-10}$, and $\epsilon_d=10^{-5}$; Ritz values and Ritz vectors were computed. We were able to converge the needed eigenvalues for all four matrices with only 3 Lanczos steps ($L=3$). A small shift, $E=0.05$, was chosen because we were interested in only the first few smallest Ritz values (matrix elements are generated in units of kcal/mol). All matrices were directly diagonalized (eigenvalues and eigenvectors) to assess the accuracy and efficiency of the PC-GFBLA.

TABLE IX. Time is adjusted user time, given by executing /usr/bin/time-p a.out on the RS-6000; error is ± 0.1 s. Disk I/O time for reading the Hamiltonian matrix has been subtracted from the quantities given in this table.

Matrix	PC-GFBLA (s)	Direct (s)
a_g, $N=1150$	20.9	66.6
a_g, $N=1551$	52.4	208.3
e_g, $N=2280$	139.2	633.1
e_g, $N=3080$	288.7	1697.7

Because the Hamiltonian matrices were stored on disk and read into the PC-GFBLA code, most of the real and user time (reported by executing /usr/bin/time-p a.out on the RS-6000) is accounted for by disk I/O operations. We timed the same runs for estimates of I/O time by stopping each job immediately after the disk file was completely read into core memory. The difference is then reported as the time spent for executing operations other than I/O (i.e., Lanczos iterations, diagonalizing the resulting matrix T, and backtransforming Ritz values and Ritz vectors). Adjusted times for each case are reported in Table IX. The $2e_g \leftarrow a_g$ transition is calculated as 113.6 cm^{-1} for both basis set sizes. The direct results and the PC-GFBLA results agree exactly for up to (at least) 8 decimal places. The assignment of this transition to a band at 98.1 cm^{-1} has been reported by Klopper and Schütz [48]; Sabo and co-workers [40] calculated this transition as 96.9 cm^{-1} using a three-dimensional DVR approach. Calculations made using a more refined potential than was used in the present study are presented in Guiang and Wyatt [42].

The calculations made for the water trimer present the most compelling argument in favor of using a method such as the PC-GFBLA: even though we were only interested in the lowest eigenvalue of the a_g matrices and the second lowest eigenvalue of the e_g matrices (and, for other applications, the corresponding eigenvectors), we would be forced to do a direct diagonalization of H to get the information. For the larger basis set calculations, this translates to a more than fivefold increase in user time. The PC-GFBLA is clearly a more efficient way to obtain the same information with the same accuracy.

IV. CONCLUSIONS

We have reported on an enhanced matrix spectroscopic technique for resolving the interior, possibly degenerate, eigenspectrum of H, which is effectively probed by using the Green-function filter $G(E)=(EI-H)^{-1}$ to drive the block Lanczos recursion. Preconditioning the initial block of Lanczos vectors increases the number of converged Ritz values and Ritz vectors substantially without significantly increasing computational time for cases where $EI-H$ [the inverse filter, $f(E)^{-1}$] can be factored (PC-GFBLA). When factoring $f(E)^{-1}$ is not possible, preconditioning can still be accomplished by estimating the Green-function filter by $G^0(E)$ (the EI-GFBLA method). The block Lanczos procedure would be driven by solving for $G^0(E)Q_j$ at each step by using an iterative linear system solver (such as GMRES [26] or DIIS [27]) or by using matrix partitioning and perturbative techniques [1,2].

The PC-GFBLA approach is clearly superior to the EI-GFBLA: any number of preconditioning steps greatly enhance the rate of convergence of Ritz values near the shift. For small cases ($N \leqslant 3000$), a 16% investment in user time (at most) can result in converging almost one Ritz value per Lanczos step. We store all Lanczos vectors and do a complete reorthogonalization of every newly generated set of Lanczos vectors against every previously generated set. Ghost eigenvalues are not reported. Good eigenvalues are determined by comparing two lists of Ritz values from diagonalizing the full and truncated Lanczos matrices, then retaining values that appear on both lists. Degenerate and nondegenerate eigenvalues are reported in excellent agreement with results from directly diagonalizing H.

Eigenvectors computed with the PC-GFBLA are linearly independent within a larger interval around E than those computed without preconditioning. Error bounds on Ritz values can be computed at no extra cost and are shown to be a reliable check on the convergence of a Ritz pair when used in conjunction with user-specified sorting and retention criteria ϵ_d and ϵ_k.

ACKNOWLEDGMENTS

This work was supported in part by the National Science Foundation and the Robert Welch Foundation. T.J.M. thanks the Dreyfus Foundation for additional support. The authors thank Chona Guiang for providing the water trimer test cases. We also thank Tucker Carrington, Jr. and John Frederick for useful comments and suggestions.

[1] R. E. Wyatt, Phys. Rev. E **51**, 3643 (1995).
[2] R. E. Wyatt, J. Chem. Phys. **103**, 8433 (1995).
[3] T. Ericsson and A. Ruhe, Math. Comput. **35**, 1251 (1980).
[4] C. Lanczos, J. Res. Nat. Bur. Stds. **45**, 255 (1950).
[5] A. Nauts and R. E. Wyatt, Phys. Rev. Lett. **51**, 2238 (1983).
[6] A. Nauts and R. E. Wyatt, Phys. Rev. A **30**, 872 (1984).
[7] M. Kaluža, Comput. Phys. Commun. **79**, 425 (1994).
[8] M. J. Bramley and T. Carrington, Jr., J. Chem. Phys. **101**, 8494 (1994).
[9] N. M. Poulin, M. J. Bramley, T. Carrington, Jr., H. G. Kjaergaard, and B. R. Henry, J. Chem. Phys. **104**, 7807 (1996).
[10] H. O. Karlsson and O. Goscinski, J. Phys. B **27**, 1061 (1994).
[11] G. Jolicard and O. Atabek, Phys. Rev. A **46**, 5845 (1992).
[12] G. Strang, *Introduction to Applied Mathematics* (Wellesley-Cambridge Press, Cambridge, 1986).
[13] B. N. Parlett and D. S. Scott, Math. Comput. **33**, 217 (1979).
[14] J. K. Cullum and R. A. Willoughby, *Lanczos Algorithms for Large Symmetric Eigenvalue Computations: Vol. 1 Theory* (Birkhaüser, Boston, 1985).
[15] H. D. Simon, Math. Comput. **42**, 115 (1984).

[16] G. H. Golub and R. Underwood, in *Mathematical Software III*, edited by J. Rice (Academic Press, New York, 1977).
[17] A. Ruhe, Math. Comput. **33**, 680 (1979).
[18] R. G. Grimes, J. G. Lewis, and H. D. Simon, SIAM J. Matrix Anal. Appl. **15**, 228 (1994).
[19] J. Cullum and W. E. Donath, in *Proceedings of the 1974 IEEE Conference on Decision and Control* (IEEE Computer Society, Bellingham, 1974), pp. 505–509.
[20] P. N. Roy and T. Carrington, Jr., J. Chem. Phys. **103**, 5600 (1995).
[21] F. Webster, P. J. Rossky, and R. A. Friesner, Comput. Phys. Commun. **63**, 494 (1991).
[22] H. Kono, Chem. Phys. Lett. **214**, 137 (1993).
[23] A. Cordelli, G. Grosso, and G. P. Parravicini, Comput. Phys. Commun. **83**, 255 (1994).
[24] M. Braun, S. A. Sofianos, D. G. Papageorgiqu, and I. E. Lagaris, J. Comput. Phys. **126**, 315 (1996).
[25] I. M. Barbour, N.-E. Behilil, P. E. Gibbs, G. Schierholz, and M. Teper, in *The Recursion Method and its Applications*, Springer Series in Solid-State Sciences Vol. 58, edited by D. G. Pettifor and D. L. Weaire (Springer-Verlag, Berlin, 1985).
[26] Y. Saad and M. Schultz, SIAM (Soc. Ind. Appl. Math.) J. Sci. Stat. Comput. **7**, 856 (1986).
[27] P. Pulay, Chem. Phys. Lett. **73**, 393 (1980); T. P. Hamilton and P. Pulay, J. Chem. Phys. **84**, 5728 (1986).
[28] E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, and D. Sorensen, *LAPACK User's Guide* (SIAM, Philadelphia, 1992).
[29] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed. (Johns Hopkins Press, Baltimore, 1996).
[30] N. Pugliano and R. J. Saykally, J. Chem. Phys. **96**, 1832 (1992).
[31] N. Pugliano, J. D. Cruzan, J. G. Loeser, and R. J. Saykally, J. Chem. Phys. **98**, 6600 (1993).
[32] K. Liu, J. D. Cruzan, and R. J. Saykally, Science **271**, 929 (1996).
[33] J. D. Cruzan, L. Braly, K. Liu, M. G. Brown, J. G. Loeser, and R. J. Saykally, Science **271**, 59 (1996).
[34] K. Liu, M. G. Brown, J. D. Cruzan, and R. J. Saykally, Science **271**, 62 (1996).
[35] K. Liu, M. G. Brown, C. Carter, R. J. Saykally, J. K. Gregory, and D. C. Clary, Nature (London) **381**, 501 (1996).
[36] N. Pugliano and R. J. Saykally, Science **257**, 1937 (1992).
[37] K. Liu, J. G. Loeser, M. J. Elrod, B. C. Host, J. A. Rzepiela, N. Pugliano, and R. J. Saykally, J. Am. Chem. Soc. **116**, 3507 (1994).
[38] D. J. Wales, J. Am. Chem. Soc. **115**, 11 180 (1993).
[39] M. Schütz, T. Bürgi, S. Leutwyler, and H. B. Bürgi, J. Chem. Phys. **99**, 5228 (1993).
[40] D. Sabo, Z. Bacic, T. Bürgi, and S. Leutwyler, Chem. Phys. Lett. **244**, 283 (1995).
[41] T. R. Walsh and D. J. Wales, J. Chem. Soc. Faraday Trans. **92**, 2505 (1996).
[42] C. Guiang and R. E. Wyatt (unpublished).
[43] N. R. Kestner, J. Chem. Phys. **48**, 252 (1968).
[44] S. F. Boys and F. Bernardi, Mol. Phys. **19**, 553 (1970).
[45] P. Cieplak, P. Kollman, and T. Lybrand, J. Chem. Phys. **92**, 6755 (1990).
[46] M. Vincke, L. Malegat, and D. Baye, J. Phys. B **26**, 811 (1993).
[47] P. R. Bunker, *Molecular Symmetry and Spectroscopy* (Academic Press, New York, 1979), pp. 55–112.
[48] W. Klopper and M. Schütz, Chem. Phys. Lett. **237**, 536 (1995).