Journal of Computational Physics **172,** 472–514 (2001)
doi:10.1006/jcph.2001.6828, available online at http://www.idealibrary.com on IDEAL®

The Subspace Projected Approximate Matrix (SPAM) Modification of the Davidson Method

Ron Shepard,* Albert F. Wagner,* Jeffrey L. Tilson,† and Michael Minkoff†

**Theoretical Chemistry Group, Chemistry Division, Argonne National Laboratory, Argonne, Illinois 60439;*
†*Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, Illinois 60439*
E-mail: shepard@tcg.anl.gov; wagner@tcg.anl.gov; jtilson@ccr.buffalo.edu; minkoff@mcs.anl.gov

Received December 15, 2000; revised May 1, 2001

A modification of the iterative matrix diagonalization method of Davidson is presented that is applicable to the symmetric eigenvalue problem. This method is based on subspace projections of a sequence of one or more approximate matrices. The purpose of these approximate matrices is to improve the efficiency of the solution of the desired eigenpairs by reducing the number of matrix–vector products that must be computed with the exact matrix. Several applications are presented. These are chosen to show the range of applicability of the method, the convergence behavior for a wide range of matrix types, and also the wide range of approaches that may be employed to generate approximate matrices.

Key Words: Davidson; matrix; eigenvalue; eigenvector; iterative; subspace; projection; Ritz; symmetric; tensor; chemistry.

1. INTRODUCTION

The symmetric eigenvalue problem

$$(\mathbf{H} - \lambda_j)\mathbf{v}_j = \mathbf{0} \tag{1.1}$$

is familiar in many application areas [1]. In some of these, the computation of the entire spectrum of eigenvalues and associated eigenvectors is necessary, and in others, only selected eigenpairs are desired. In the former case, particularly with dense unstructured matrices, the overall computational effort scales as $O(N^3)$ where N is the dimension of the matrix; these are called *direct* or *dense* methods. When only a few vectors are required, they may sometimes be determined using *iterative* methods, and the overall effort may be much less, particularly if some structure of the matrix (e.g., banded, blocked, sparse, outer-product, tensor-product, and so forth) may be exploited. The largest eigenvalue problems correspond to N as large as 10^8 or 10^9; for these situations, dense methods cannot even be considered, and iterative methods are the only practical choice.

0021-9991/01 $35.00

The method that will be described in this work is a modification of the Davidson iterative method [2–6]. The Davidson method has the following features, all of which are shared by the method described in this work.

1. Only matrix–vector products (or linear transformations) of the matrix with arbitrary trial vectors are needed. For structured or sparse matrices, this allows the products to be computed efficiently, with less computational effort, fewer floating point operations, and/or less I/O than the usual matrix–vector product. The matrix is not modified during the procedure, so sparse fill-in does not occur. Furthermore, it is not necessary to actually compute and store the matrix elements explicitly. There are many examples of applications for which it is more efficient to either recompute the elements "on-the-fly" as needed (either from formal expressions for the individual matrix elements or from underlying simpler, compact, data structures) or for which the matrix structure itself may be exploited in some way in order to compute the matrix–vector products in "operator" form. A few examples of this are discussed in detail below.

2. The Davidson method is a *subspace* method. As a trial vector is added to the subspace during the iterative procedure, the new computed approximate eigenvalues from this subspace (called the *Ritz* values) bracket those of the previous iteration. This is particularly beneficial when computing the lowest roots because the intermediate computed eigenvalues are always upper bounds to the final converged lowest eigenvalues. Similarly, the highest roots of an intermediate subspace representation are lower bounds to the final converged highest eigenvalues.

3. The Davidson method can be used to find the lowest eigenpair, several of the lowest eigenpairs, the highest eigenpair, several of the highest eigenpairs, or selected interior eigenpairs.

4. A benefit of a subspace method is that convergence is generally more robust than for a single-vector (or update) method. In general, given any single-vector iterative method, a corresponding subspace method may be devised, and this subspace method will always converge better than the original single-vector method. In fact, the subspace method may sometimes converge rapidly even when the single-vector method upon which it is based oscillates, diverges, exhibits false convergence, or otherwise converges problematically. However, the subspace method typically requires more resources (memory, disk space, and so forth) than the corresponding single-vector method, and the manipulation of the multiple vectors is computationally more demanding than for the single-vector method. (These comments regarding convergence may not apply necessarily to *sequential relaxation* [7], also called *continuous update*, single-vector methods. Each iteration of such a method consists of N individual updates, usually applied in sequential order to the elements of the trial eigenvector. A subspace analog of these types of single-vector methods is impractical because the subspace dimension would grow too large. Although these methods can converge efficiently, particularly for isolated eigenpairs, the sequential update process requires ordered access to the matrix elements, and this limits the range of applicability of these methods.)

5. It is possible for the Davidson method to converge to the wrong root, or, when several roots are requested, to "skip" over roots and converge to nearby roots instead. This places some importance on the choice of initial vectors.

One disadvantage of the Davidson method is that it can be slowly convergent for some matrices. These include matrices that are not diagonally dominant. Slower convergence means that more matrix–vector products are required, resulting in greater computational

effort. This is particularly problematic for matrices of very large dimension for which each matrix–vector product requires a major computational effort. It is primarily this situation that is addressed by the method described in this work.

2. THE SPAM METHOD

The original Davidson Method is outlined in Fig. 1. During the iterative procedure, a set of expansion vectors $\{\mathbf{x}_j;\ j=1, n\}$ is available. These vectors may be collected together to form the columns of a matrix $\mathbf{X}^{[n]}$, where the superscript denotes the number of vectors. The details of the methods used to generate the new expansion vectors are discussed in Appendix B. There are also the corresponding matrix–vector products $\mathbf{W}^{[n]} = \mathbf{H}\mathbf{X}^{[n]}$ that are stored. The representation of $\mathbf{H}$ within this subspace is given by

$$\langle \mathbf{H} \rangle^{[n]} = \mathbf{X}^{[n]T}\mathbf{H}\,\mathbf{X}^{[n]} = \mathbf{W}^{[n]T}\mathbf{X}^{[n]}, \tag{2.1}$$

in which the superscript T denotes the transpose. A projection matrix may be defined as

$$\mathbf{P}^{[n]} = \mathbf{X}^{[n]}\left(\mathbf{X}^{[n]T}\mathbf{X}^{[n]}\right)^{-1}\mathbf{X}^{[n]T}. \tag{2.2}$$

The method described here may be implemented in terms of general nonorthogonal expansion vectors. However, for simplicity, it will be assumed hereafter that the expansion vectors are chosen to satisfy the relation $(\mathbf{X}^{[n]T}\mathbf{X}^{[n]}) = 1$. This allows the projection matrix to be written simply as $\mathbf{P}^{[n]} = \mathbf{X}^{[n]}\mathbf{X}^{[n]T}$. There is also the orthogonal projector defined as $\mathbf{Q}^{[n]} = (\mathbf{1} - \mathbf{P}^{[n]})$. These projectors result in the identity

$$\mathbf{H} = \left(\mathbf{P}^{[n]} + \mathbf{Q}^{[n]}\right)\mathbf{H}\left(\mathbf{P}^{[n]} + \mathbf{Q}^{[n]}\right) \tag{2.3}$$

$$= \mathbf{P}^{[n]}\mathbf{H}\,\mathbf{P}^{[n]} + \mathbf{P}^{[n]}\mathbf{H}\,\mathbf{Q}^{[n]} + \mathbf{Q}^{[n]}\mathbf{H}\,\mathbf{P}^{[n]} + \mathbf{Q}^{[n]}\mathbf{H}\,\mathbf{Q}^{[n]} \tag{2.4}$$

$$= \left(\mathbf{X}^{[n]}\langle \mathbf{H} \rangle^{[n]}\mathbf{X}^{[n]T} + \mathbf{X}^{[n]}\mathbf{W}^{[n]T}\mathbf{Q}^{[n]} + \mathbf{Q}^{[n]}\mathbf{W}^{[n]}\mathbf{X}^{[n]T}\right) + \mathbf{Q}^{[n]}\mathbf{H}\mathbf{Q}^{[n]}. \tag{2.5}$$

Outline of the Davidson Method

```
Generate an initial vector x_1
MAINLOOP: DO n =1
    Compute and save  w_n = Hx_n
    Compute the n-th row and column of ⟨H⟩:  ⟨H⟩_{1:n,n} = w_n^T X^[n]
    Compute the subspace eigenvector and value: (⟨H⟩ – ρ)c = 0
    Compute the residual:  r=W_{1:n}c_{1:n} – ρ X_{1:n}c_{1:n}
    Check for convergence using |r|, c, ρ, etc.
    IF (converged) THEN
        EXIT MAINLOOP
    ELSE
        Generate a new expansion vector x_{n+1} from r, ρ, v=Xc, etc.
    ENDIF
ENDDO MAINLOOP
```

FIG. 1. Outline of the Davidson method.

An arbitrary matrix–vector product $\mathbf{Hy}$ may therefore be computed as four separate contributions, the first three of which involve only operations with the subspace vectors $\mathbf{X}$ and $\mathbf{W}$.

The crucial idea of the method described here is that an approximate matrix $\mathbf{H}^{(1)}$ is available, that matrix–vector products $\mathbf{H}^{(1)}\mathbf{y}$ require less effort to compute than the exact products $\mathbf{Hy}$, and that these approximate matrix–vector products are used to reduce the overall computational effort. This reduced effort could be because $\mathbf{H}^{(1)}$ is less dense than $\mathbf{H}$, or because $\mathbf{H}^{(1)}$ is generated from some formal or algebraic approximation to $\mathbf{H}$ (e.g., simpler basis, a smaller basis, a lower-order approximation, an outer-product approximation, a tensor-product approximation, a coarser computational grid, and so on). With this approximate matrix available, a subspace projected approximate matrix (SPAM) $\bar{\mathbf{H}}^{[n]}$ is defined:

$$\bar{\mathbf{H}}^{[n]} \equiv \mathbf{P}^{[n]}\mathbf{H}\,\mathbf{P}^{[n]} + \mathbf{P}^{[n]}\mathbf{H}\,\mathbf{Q}^{[n]} + \mathbf{Q}^{[n]}\mathbf{H}\,\mathbf{P}^{[n]} + \mathbf{Q}^{[n]}\mathbf{H}^{(1)}\mathbf{Q}^{[n]} \tag{2.6}$$

$$= \left(\mathbf{X}^{[n]}\langle\mathbf{H}\rangle^{[n]}\mathbf{X}^{[n]T} + \mathbf{X}^{[n]}\mathbf{W}^{[n]T}\mathbf{Q}^{[n]} + \mathbf{Q}^{[n]}\mathbf{W}^{[n]}\mathbf{X}^{[n]T}\right) + \mathbf{Q}^{[n]}\mathbf{H}^{(1)}\mathbf{Q}^{[n]}. \tag{2.7}$$

Note that the first three terms in Eqs. (2.6) and (2.7) are "exact" when compared to Eqs. (2.4) and (2.5). It is only the last term that is affected by the approximation. For a given subspace of dimension $[n]$, the eigenpair from this approximate matrix is computed

$$\left(\bar{\mathbf{H}}^{[n]} - \lambda_j^{[n]}\right)\mathbf{v}_j^{[n]} = 0. \tag{2.8}$$

This eigenvector is then appended to the subspace (after orthonormalization) to form $\mathbf{X}^{[n+1]}$. An exact matrix–vector product is computed to form $\mathbf{W}^{[n+1]}$. This expanded subspace then defines a new projector $\mathbf{P}^{[n+1]}$ and a corresponding new approximate matrix $\bar{\mathbf{H}}^{[n+1]}$, and the process is repeated until convergence is achieved. Although the underlying approximate matrix $\mathbf{H}^{(1)}$ remains the same during this process, the SPAM $\bar{\mathbf{H}}^{[n]}$ changes as the iterations proceed. Both the eigenvector and the eigenvalue from Eq. (2.8) are approximations to the converged results. The accuracy of the approximation is quantified in Appendix A. However, the approximate eigenvalue does not enjoy the upper (or lower, as relevant) bound property that holds for the subspace eigenvalue computed from the exact matrix–vector products only.

When a vector $\mathbf{y}$ is a member of $\{\mathbf{x}_j;\ j = 1, n\}$, or if it is a general linear combination of these vectors, $\mathbf{y} = \mathbf{X}^{[n]}\mathbf{c}$, then Eq. (2.6) results in the relation

$$\bar{\mathbf{H}}^{[n]}\mathbf{y} = \mathbf{Hy}; \quad \text{when } \mathbf{y} \in \mathit{Span}\left(\mathbf{X}^{[n]}\right). \tag{2.9}$$

It is only vectors $\mathbf{y}$ orthogonal to $\mathbf{X}^{[n]}$, or that contain orthogonal components, that are approximated by $\bar{\mathbf{H}}^{[n]}\mathbf{y}$ relative to the exact matrix–vector product $\mathbf{Hy}$. As the procedure converges to the eigenpair of interest, the subspace $\mathbf{X}^{[n]}$ contains the eigenvector. When this occurs, the converged eigenpair of $\bar{\mathbf{H}}^{[n]}$ is also an eigenpair of the exact $\mathbf{H}$.

This leads to the question of how to solve the eigenvector equation of Eq. (2.8). It is the same dimension as the original equation, so it is appropriate that an iterative method should be used. In the current work, the iterative Davidson method outlined in Fig. 1 is used. Equation (2.9) suggests that an initial subspace consisting of $\mathbf{X}^{[n]}$ could be used for this iterative solution. Because the exact matrix–vector products $\mathbf{W}^{[n]}$ are already available, the first $n \times n$ subblock of the subspace matrix $\langle\bar{\mathbf{H}}^{[n]}\rangle_{1:n,1:n}$ has already been computed and is

available. Furthermore, all of the new expansion vectors that are added during the iterative eigensolution can be chosen to be orthogonal to $\mathbf{X}^{[n]}$. In this case, a matrix–vector product takes the simple form

$$\bar{\mathbf{H}}^{[n]}\mathbf{x}_\perp = \mathbf{X}^{[n]}\mathbf{W}^{[n]T}\mathbf{x}_\perp + \mathbf{Q}^{[n]}\mathbf{H}^{(1)}\mathbf{x}_\perp \tag{2.10}$$

$$= \mathbf{w}^{(1)} + \mathbf{X}^{[n]}\left(\mathbf{W}^{[n]T}\mathbf{x}_\perp - \mathbf{X}^{[n]T}\mathbf{w}^{(1)}\right), \tag{2.11}$$

where $\mathbf{w}^{(1)} = \mathbf{H}^{(1)}\mathbf{x}_\perp$ is the inexpensive matrix–vector product. Furthermore, as a result of Eq. (2.9), a subspace matrix element between a vector $\mathbf{y}$ within $\mathbf{X}^{[n]}$ and a vector $\mathbf{x}_\perp$ orthogonal to $\mathbf{X}^{[n]}$ is exact:

$$\mathbf{x}_\perp^T\bar{\mathbf{H}}^{[n]}\mathbf{y} = \mathbf{y}^T\bar{\mathbf{H}}^{[n]}\mathbf{x}_\perp = \mathbf{x}_\perp^T\mathbf{H}\mathbf{y}; \quad \mathbf{y} \in Span\left(\mathbf{X}^{[n]}\right),\ \mathbf{X}^{[n]T}\mathbf{x}_\perp = 0. \tag{2.12}$$

It is only matrix elements in the diagonal subblock of $\langle\bar{\mathbf{H}}^{[n]}\rangle$ between two vectors in the orthogonal space that are not exact relative to the matrix $\langle\mathbf{H}\rangle$ in the same vector subspace.

This suggests the SPAM implementation in Fig. 2. This is basically the same as the original Davidson method, except that a flag, *wtype*, is toggled between 0 and 1 to denote

Outline of the SPAM Method

```
Generate an initial vector x_1
Set wtype_1=1  ! Start the iterations with approximate products
Set n_0=0; n=1
MAINLOOP: DO
    Compute and save w_n = H(wtype_n, n_0) x_n
    Compute the n-th row and column of <H>:  <H>_{1:n,n} = w_n^T X^[n]
    Compute the subspace eigenvector and value: (<H> - ρ)c = 0
    Compute the residual:  r=W_{1:n}c_{1:n} - ρ X_{1:n}c_{1:n}
    Check for convergence using |r|, c, ρ, etc.
    IF (converged .AND. wtype_n.EQ.0) then
        EXIT MAINLOOP ! Final convergence is achieved
    ELSEIF (converged .AND. wtype_n≠0) then
        Contract x_{n_0+1} ← X_{(n_0+1):n} c_{(n_0+1):n} / |c_{(n_0+1):n}|
        Set n_0←n_0+1; n= n_0
        Set wtype_n=0  ! The next product will be exact
    ELSE
        Set n←n+1
        Generate a new expansion vector x_n from r, ρ, v=Xc, etc.
        Set wtype_n=1  ! The next product will be approximate
    ENDIF
ENDDO MAINLOOP
```

FIG. 2. Outline of the SPAM method.

the type of matrix–vector product for each expansion vector. Furthermore, the convergence criteria are slightly more complicated. Basically, there are two kinds of convergence. When convergence is achieved with $wtype_n = 0$, then all of the matrix–vector products have been computed with the exact matrix $\mathbf{H}$, and the desired eigenpair has been found. When convergence is achieved with $wtype_n = 1$, this means that the current SPAM eigensolution of Eq. (2.8) has been found. At this time, the new expansion vectors (the n_1 vectors corresponding to the $wtype_k = 1$ vectors) are contracted using the coefficients $\mathbf{c}$ from the current subspace eigenvector, this vector is saved in the $[n_0 + 1]$ position in $\mathbf{X}$, and *wtype* is then set to 0 for that vector to ensure that the next matrix–vector product will be computed exactly.

One way to view the overall SPAM iterative procedure is to monitor the subspace dimension and to note the number of exact (with $wtype_k = 0$) products computed and the number of approximate (with $wtype_k = 1$) vectors. In the following discussion, such a mixed subspace will be denoted $[n_0, n_1]$. As outlined above, the number of exact products n_0 in the subspace never decreases during the iterative procedure, but the number of approximate products n_1 is an irregular sawtooth function during the iterative procedure. The number of approximate products increases for a few iterations, then upon intermediate convergence of Eq. (2.8), the count n_1 is reset to zero, and it then begins to increase again from that point. Examples of this convergence behavior are given below.

3. THE MULTILEVEL SPAM METHOD

During the SPAM iterative method, the iterative solution to the eigenvector equation (Eq. (2.8)) is required. Matrix–vector products with the approximate matrix $\mathbf{H}^{(1)}$ are assumed to require less effort than the exact products involving $\mathbf{H} \equiv \mathbf{H}^{(0)}$. However, what if convergence of Eq. (2.8) (for a given projection rank $[n_0]$) is slow and there are many of these $\mathbf{H}^{(1)}$ matrix–vector products required, the total cost of which is excessive? The answer to this problem is to temporarily treat the matrix $\bar{\mathbf{H}}^{[n_0]}$ as "exact," and to apply the SPAM method to this problem with yet another "approximate" matrix $\mathbf{H}^{(2)}$:

$$\begin{aligned}\bar{\mathbf{H}}^{[n_0,n_1]} = {} & \mathbf{P}^{[n_0,n_1]}\bar{\mathbf{H}}^{[n_0]}\mathbf{P}^{[n_0,n_1]} + \mathbf{P}^{[n_0,n_1]}\bar{\mathbf{H}}^{[n_0]}\mathbf{Q}^{[n_0,n_1]} \\ & + \mathbf{Q}^{[n_0,n_1]}\bar{\mathbf{H}}^{[n_0]}\mathbf{P}^{[n_0,n_1]} + \mathbf{Q}^{[n_0,n_1]}\mathbf{H}^{(2)}\mathbf{Q}^{[n_0,n_1]}. \end{aligned} \tag{3.1}$$

In order to reduce the computational effort, matrix–vector products with $\mathbf{H}^{(2)}$ must require even less effort than those of $\mathbf{H}^{(1)}$. The eigenvector solution from the equation

$$\left(\bar{\mathbf{H}}^{[n_0,n_1]} - \lambda_j^{[n_0,n_1]}\right)\mathbf{v}_j^{[n_0,n_1]} = \mathbf{0} \tag{3.2}$$

is converged using the Davidson procedure. At this point, the vector space is denoted $[n_0, n_1, n_2]$, which means that there are n_0 vectors for which exact matrix–vector products with $\mathbf{H}^{(0)}$ are available, n_1 vectors for which $\bar{\mathbf{H}}^{[n_0]}$ matrix–vector products using the approximate matrix $\mathbf{H}^{(1)}$ have been computed, and n_2 vectors for which $\bar{\mathbf{H}}^{[n_0,n_1]}$ matrix–vector products using the approximate matrix $\mathbf{H}^{(2)}$ have been computed and are available. Upon convergence of Eq. (3.2) the $[n_0, n_1, n_2]$ subspace is contracted in order to define a new $[n_0, n_1 + 1, 0]$ subspace, and the process is continued until convergence is achieved. When convergence is achieved eventually for the sequence of level-1 SPAM approximations, the

current $[n_0, n_1] \equiv [n_0, n_1, 0]$ space is contracted down to form a $[n_0+1] \equiv [n_0+1, 0, 0]$ space, as described in Section 2. Analogous to Eqs. (2.9) and (2.11), matrix–vector products satisfy

$$\bar{\mathbf{H}}^{[n_0,n_1]}\mathbf{y} = \bar{\mathbf{H}}^{[n_0]}\mathbf{y}; \quad \text{when } \mathbf{y} \in \mathit{Span}\left(\mathbf{X}^{[n_0,n_1]}\right) \tag{3.3}$$

$$\bar{\mathbf{H}}^{[n_0,n_1]}\mathbf{x}_\perp = \mathbf{w}^{(2)} + \mathbf{X}^{[n_0,n_1]}\left(\mathbf{W}^{[n_0,n_1]T}\mathbf{x}_\perp - \mathbf{X}^{[n_0,n_1]T}\mathbf{w}^{(2)}\right); \quad \textit{when } \mathbf{X}^{[n_0,n_1]T}\mathbf{x}_\perp = 0. \tag{3.4}$$

These equations suggest a generalization of the SPAM method to an arbitrary number of approximation levels based on a modification of the subspace procedure described in the previous section. The SPAM at a given approximation level, labeled by $(m+1)$ and dependent on the current expansion vector subspace $[n_0, n_1 \ldots n_m]$, is defined in terms of the previous m-level SPAM along with a new approximate matrix $\mathbf{H}^{(m+1)}$:

$$\begin{aligned}\bar{\mathbf{H}}^{[n_0,n_1,\ldots n_m]} = {} & \mathbf{P}^{[n_0,n_1,\ldots n_m]}\bar{\mathbf{H}}^{[n_0,n_1,\ldots n_{m-1}]}\mathbf{P}^{[n_0,n_1,\ldots n_m]} + \mathbf{P}^{[n_0,n_1,\ldots n_m]}\bar{\mathbf{H}}^{[n_0,n_1,\ldots n_{m-1}]}\mathbf{Q}^{[n_0,n_1,\ldots n_m]} \\ & + \mathbf{Q}^{[n_0,n_1,\ldots n_m]}\bar{\mathbf{H}}^{[n_0,n_1,\ldots n_{m-1}]}\mathbf{P}^{[n_0,n_1,\ldots n_m]} + \mathbf{Q}^{[n_0,n_1,\ldots n_m]}\mathbf{H}^{(m+1)}\mathbf{Q}^{[n_0,n_1,\ldots n_m]}\end{aligned} \tag{3.5}$$

This procedure is outlined in Fig. 3. In this multilevel SPAM method, the $wtype_k$ variable

Outline of the MultiLevel SPAM Method

Generate an initial vector $\mathbf{x}_1$
Set $wtype_1$=*MaxSpamLevel* *! Start with approximate products*
Set $n_{0:MaxLevel}$=0; n=1
MAINLOOP: DO
  Compute and save $\mathbf{w}_n = \mathbf{H}(wtype_n, n_{0:MaxLevel})\,\mathbf{x}_n$
  Compute the n-th row and column of $\langle\mathbf{H}\rangle$: $\langle\mathbf{H}\rangle_{1:n,n} = \mathbf{w}_n^T\mathbf{X}^{[n]}$
  Compute the subspace eigenvector and value: $(\langle\mathbf{H}\rangle - \rho)\mathbf{c} = \mathbf{0}$
  Compute the residual: $\mathbf{r}=\mathbf{W}_{1:n}\mathbf{c}_{1:n} - \rho\,\mathbf{X}_{1:n}\mathbf{c}_{1:n}$
  Check for convergence using $|\mathbf{r}|$, $\mathbf{c}$, ρ, etc.
  IF (*converged* .AND. $wtype_n$.EQ.0) THEN
    EXIT MAINLOOP *! Final convergence is achieved*
  ELSEIF (*converged* .AND. $wtype_n \neq 0$) then
    Contract $\mathbf{x}_k \leftarrow \mathbf{X}_{k:n}\mathbf{c}_{k:n}/|\mathbf{c}_{k:n}|$ *! contract $wtype_n$ vectors*
    Reset n, n_{wtype}, $n_{wtype\text{-}1}$
    Set $wtype_n \leftarrow wtype_n - 1$ *! one step more accurate*
  ELSE
    Set $n \leftarrow n+1$
    Generate a new expansion vector $\mathbf{x}_n$ from $\mathbf{r}$, ρ, $\mathbf{v}=\mathbf{X}\mathbf{c}$, etc.
    Set $wtype_n$= *MaxSpamLevel* *! most approximate level*
  ENDIF
ENDDO MAINLOOP

FIG. 3. Outline of the multiLevel SPAM method.

is set to the approximation level of the corresponding matrix–vector product: 0 for exact matrix–vector products, 1 for the first level of approximation, 2 for the second level of approximation, and so forth. When the maximum SPAM level is set to 0, then the multilevel SPAM method outlined in Fig. 3 is equivalent to the simple Davidson method outlined in Fig. 1. When the maximum SPAM level is set to 1, then the multilevel SPAM method in Fig. 3 is equivalent to the method outlined in Fig. 2.

This general idea is entirely consistent with the usual approach taken in various applications involving eigenvalue problems. The "exact" problem is too difficult to solve, so it is approximated, in some way, by a model problem that is formally, conceptually, or computationally simpler. If this simpler problem is itself too difficult to solve, then yet further approximations are invoked. The SPAM method allows this series of approximations to be incorporated directly into the numerical solution of the original "exact" eigenproblem.

4. DISCUSSION

The single-level and the general multilevel SPAM methods described above have been implemented in a standard Fortran 90 subroutine. In this section, the features of this implementation are discussed. Several types of test matrices are used in these discussions, and several different ways of formulating approximate matrices are demonstrated. Additional details of the implementation are discussed in the context of these examples.

Banded Matrix Examples: The first examples are based on a banded matrix of the general form

$$\begin{aligned} H_{k,k} &= k; && \text{for } k = 1 \ldots N \\ H_{k,l} &= \Delta^{|k-l|}; && \text{for } |k-l| \le W \text{ and } k \ne l \\ H_{k,l} &= 0; && \text{otherwise.} \end{aligned} \tag{4.1}$$

These matrices are characterized by three scalar parameters, the matrix dimension N, the bandwidth W, and Δ, which determines the diagonal dominance of the matrix. In the following test calculations, a matrix with a large bandwidth will be approximated by a matrix with a smaller bandwidth. By using recursion, matrix–vector products with this matrix may be computed with $O(N)$ floating point operations, independent of W. The SPAM method is not the best approach for this matrix because the exact matrix–vector products are just as expensive to compute as the approximate ones, but this is an excellent matrix to use as a model for general matrices that display similar convergence characteristics because the degree of diagonal dominance and the accuracy of the successive approximate matrices is easily controlled.

The first column of results in Table I shows the convergence of the regular Davidson iterative method, with a diagonal–preconditioned residual (DPR) expansion vector, for the lowest eigenpair of a matrix characterized by $N = 10{,}000$, $W = 64$, and $\Delta = 0.75$. The initial vector is $\mathbf{e}_1$, the first column of a unit matrix of dimension N. The dimension is chosen so that this problem is nontrivial, yet the structure of the matrix results in model test problems that are readily solved. The convergence criterion for this test case is $|\mathbf{r}| < 10^{-8}$ (see Appendix A), which is a typical convergence requirement. For this matrix, the lowest eigenvalue, $\lambda_1 = 0.585510562346823$, is converged to approximately machine precision ($\sim 10^{-15}$) with this convergence tolerance, which is consistent with the bound in Eq. (A14). Twelve iterations, each of which require an $\mathbf{H}^{(0)}$ matrix–vector product, are required to

TABLE I
Comparison of DPR and SPAM Convergence

	DPR		SPAM Fixed Tolerance		SPAM Dynamic Tolerance	
Iteration	$[n_0]$	$\|\mathbf{r}\|$	$[n_0, n_1]$	$\|\mathbf{r}\|$	$[n_0, n_1]$	$\|\mathbf{r}\|$
1	[1]	1.13E+00	[0, 1]	1.13E+00	[0, 1]	1.13E+00
2	[2]	3.23E-01	[0, 2]	3.23E-01	[0, 2]	3.23E-01
3	[3]	1.05E-01	[0, 3]	1.05E-01	[0, 3]	1.05E-01
4	[4]	2.73E-02	[0, 4]	2.73E-02	[0, 4]	2.73E-02
5	[5]	5.41E-03	[0, 5]	5.41E-03	[0, 5]	5.41E-03
6	[6]	8.66E-04	[0, 6]	8.66E-04	[0, 6]	8.66E-04
7	[7]	1.16E-04	[0, 7]	1.16E-04	[0, 7]	1.16E-04
8	[8]	1.35E-05	[0, 8]	1.35E-05	[1, 0]	1.40E-04
9	[9]	1.37E-06	[0, 9]	1.37E-06	[1, 1]	2.11E-05
10	[10]	1.24E-07	[0, 10]	1.24E-07	[1, 2]	2.78E-06
11	[11]	1.02E-08	[0, 11]	1.02E-08	[1, 3]	3.47E-07
12	[12]	7.59E-10	[0, 12]	7.59E-10	[1, 4]	7.97E-08
13			[1, 0]	7.12E-05	[1, 5]	2.13E-08
14			[1, 1]	4.34E-06	[1, 6]	7.30E-09
15			[1, 2]	2.06E-07	[2, 0]	7.35E-09
16			[1, 3]	1.69E-08		
17			[1, 4]	6.30E-09		
18			[2, 0]	6.28E-09		
$N_{product}$	[12]		[2, 16]		[2, 13]	

Note. Convergence trajectories of the lowest root of the banded test matrix with $N = 10{,}000$, $W_0 = 64$, and $\Delta = 0.75$. For the SPAM calculations, $W_1 = 32$. The convergence criterion is $|\mathbf{r}| < 10^{-8}$.

achieve convergence with the traditional Davidson DPR method. This convergence rate is typical of many eigenproblems that occur in various applications. The largest off-diagonal element in this matrix is 0.75, and the smallest nonzero off-diagonal element is $1.01 \cdot 10^{-8}$.

An approximate matrix $\mathbf{H}^{(1)}$ with half the bandwidth of $\mathbf{H}^{(0)}$ is characterized by $N = 10{,}000$, $W = 32$, and $\Delta = 0.75$, and a single-level SPAM is applied. As seen in Table I, only two $\mathbf{H}^{(0)}$ matrix–vector products are required along with 16 approximate $\mathbf{H}^{(1)}$ matrix–vector products. The total number of iterations has increased, but almost all of them are with the approximate matrix $\mathbf{H}^{(1)}$ rather than the exact matrix $\mathbf{H}^{(0)}$. The smallest nonzero off-diagonal element in $\mathbf{H}^{(1)}$ is $1.00 \cdot 10^{-4}$. The largest element in the difference matrix $(\mathbf{H}^{(1)} - \mathbf{H}^{(0)})$ has the magnitude $7.5 \cdot 10^{-5}$. If the $\mathbf{H}^{(1)}$ products were 10 times cheaper to compute than the $\mathbf{H}^{(0)}$ products (an effort ratio of $1/10$) for an actual application with similar convergence properties, then already the SPAM method would have resulted in an overall savings of $12 : (2 + 1.6)$, or an overall 70% reduction of effort.

Inspection of the convergence trajectory of the SPAM calculation in Table I suggests that too many level-1 iterations are performed during the generation of the [1, 0] subspace. Basically, no matter how well the $[0, n_1]$ iterations are converged, the residual of the [1, 0] iteration (immediately after contraction of the n_1 vectors) will have a vector norm of at least $\sim 1 \cdot 10^{-4}$. This suggests that instead of the final residual norm convergence tolerance, a dynamic adjustment of the intermediate residual norms would result in improved efficiency. The accuracy of residual norms is quantified in Appendix A. Equation (A22) suggests that the convergence of the $\bar{\mathbf{H}}^{[0]}$ matrix during this first SPAM iteration needs to be converged

to at least $\|\mathbf{H}^{(1)} - \mathbf{H}^{(0)}\|$. Two estimates of this matrix norm were considered, one based on the Gerschgorin disk bound [8], and the other based on the eigenvalue bound in Eq. (A13) along with a coordinate unit vector. The expression in Eq. (A13) was found empirically both to be smaller in magnitude and to result in the more accurate residual norm estimate, with a value of $1.61 \cdot 10^{-4}$ for this particular matrix. Because this estimate is based on a bound, and is not necessarily an accurate estimate of either the matrix difference norm or of the residual vector of the [1,0] iteration, an additional scale factor of $\alpha = 0.95$ is used, and the first SPAM iteration is converged to $|\mathbf{r}| < 1.54 \cdot 10^{-4}$. This may be regarded as a prediction, before contraction, of the actual [1, 0] residual norm after contraction. If this scale factor α is chosen to be too small, then a few extra approximate $\mathbf{H}^{(1)}$ matrix–vector products may be computed. However, if the scale factor is too generous, and the first sequence of SPAM iterations is not sufficiently converged, then the penalty is that too many expensive $\mathbf{H}^{(0)}$ matrix–vector products may be computed. Because the penalty for overconverging the approximate SPAM sequence is less than the penalty for underconverging the SPAM sequence, it is better generally to err on the side of caution than to err on the side of optimism. In the general case, if $n = n_0 + n_1$ is the number of expansion vectors during an iteration, then

$$\left|Sin\left(\psi^{[n_0]}\right)\right| = \left|c_{(n_0+1):n}\right| = \sqrt{c^T_{(n_0+1):n} c_{(n_0+1):n}}. \tag{4.2}$$

This value involves only the expansion coefficients of the basis vectors in the n_1 subspace. The iterative solution of the SPAM eigenpair is terminated when the residual norm satisfies

$$\left|\mathbf{r}^{[n_0,n_1]}\right| \le \alpha \left|\mathrm{Sin}\left(\psi^{[n_0]}\right)\right| \cdot \left\|\mathbf{H}^{(1)} - \mathbf{H}^{(0)}\right\|. \tag{4.3}$$

Comparing the residual norms for the [0, 7] and the [1, 0] iterations in Table I, it is seen that a choice of $\alpha = 0.95$, along with the above estimate of $\|\mathbf{H}^{(1)} - \mathbf{H}^{(0)}\|$, is sufficiently accurate for this particular matrix. The final result of adjusting the convergence dynamically during the SPAM iterative process according to Eq. (4.3) for this test case is that two exact matrix–vector products are required and 13 approximate $\mathbf{H}^{(1)}$ matrix–vector products are required to achieve convergence. That is, three approximate matrix–vector products were skipped compared to the previous fixed-tolerance convergence trajectory. For a matrix–vector product effort ratio of $1/10$, the overall effort, compared to the reference DPR expansion vector procedure, would be $12 : (2 + 1.3)$, or an overall 73% reduction in effort.

The convergence characteristics for several SPAM calculations are shown in Table II. Each row corresponds to a different choice of approximate matrix $\mathbf{H}^{(1)}$. For each approximate $\mathbf{H}^{(1)}$, the matrix difference norm $\|\mathbf{H}^{(1)} - \mathbf{H}^{(0)}\|$ is estimated from Eq. (A13), and this estimate is used to dynamically adjust the intermediate convergence tolerances as described above. In all cases, a choice of $\alpha = 0.95$ was used. For each convergence trajectory, the maximum subspace dimension n_{max} that is required to achieve convergence is listed, along with the total number of matrix–vector products for each of the two matrices, $\mathbf{H}^{(0)}$ and $\mathbf{H}^{(1)}$. Two separate final convergence tolerances are imposed on the computed residual norms, a looser value of 10^{-5} and a tighter value of 10^{-8}. These span the range of "typical" convergence criteria for various applications. Both the maximum subspace dimension and the total number of products can be important in determining the overall efficiency of a calculation, and even whether the calculation fits within the memory or disk space limitations. A more detailed effort model is discussed below. Comparing the two sets of calculations for the two residual norm tolerances shows that smaller values of n_{max} and fewer matrix–vector

TABLE II
Comparison of SPAM convergence

		$\|\mathbf{r}\| < 10^{-5}$		$\|\mathbf{r}\| < 10^{-8}$	
$[W]$	$\|\mathbf{H}^{(1)} - \mathbf{H}^{(0)}\|$	n_{max}	$N_{product}$	n_{max}	$N_{product}$
[64, 64]	0.0	9	[1, 9]	12	[1, 12]
[64, 56]	$1.608 \cdot 10^{-7}$	9	[1, 9]	10	[2, 12]
[64, 48]	$1.614 \cdot 10^{-6}$	9	[1, 9]	9	[2, 12]
[64, 40]	$1.612 \cdot 10^{-5}$	8	[2, 9]	8	[2, 12]
[64, 32]	$1.611 \cdot 10^{-4}$	7	[2, 9]	8	[2, 12]
[64, 24]	$1.609 \cdot 10^{-3}$	6	[2, 9]	7	[3, 15]
[64, 16]	$1.602 \cdot 10^{-2}$	6	[3, 11]	7	[4, 16]
[64, 8]	$1.605 \cdot 10^{-1}$	6	[4, 12]	7	[6, 17]
[64, 1]	$1.203 \cdot 10^{0}$	9	[9, 9]	12	[12, 12]
[64, 0]	$1.604 \cdot 10^{0}$	9	[9, 9]	12	[12, 12]

Note. Convergence of the lowest root of the banded test matrix with $N = 10{,}000$ and $\Delta = 0.75$. The $\|\mathbf{H}^{(1)} - \mathbf{H}^{(0)}\|$ values are estimated from the residual norm bound. The intermediate convergence tolerance is adjusted dynamically.

products are required for the looser convergence criteria. This is consistent with the convergence of the usual Davidson DPR method. The other general trend is that the better the $\mathbf{H}^{(1)}$ approximation, the fewer exact $\mathbf{H}^{(0)}$ products are required. In particular, the $W_1 = 64$, $W_1 = 56$ and $W_1 = 48$ calculations demonstrate that convergence can be achieved with a single exact matrix–vector product in the most favorable situations. Convergence is always achieved with a single "exact" matrix–vector product with $W_0 = W_1$, and this is demonstrated in the first row in Table II; this has no practical consequence, but it demonstrates that the implementation satisfies this formal boundary condition in the limit $\mathbf{H}^{(1)} \to \mathbf{H}^{(0)}$.

The last two rows of Table II, with $W_1 = 1$ and with $W_1 = 0$ should also be mentioned. The $W_1 = 1$ row uses a tridiagonal $\mathbf{H}^{(1)}$ matrix. For this test case, because of the dynamical adjustment of the intermediate convergence, each DPR expansion vector generated for $\bar{\mathbf{H}}^{[n_0]}$ is "contracted" immediately and used to form an exact $\mathbf{H}^{(0)}$ matrix–vector product. The result is that there is an equal number of $\mathbf{H}^{(0)}$ and $\mathbf{H}^{(1)}$ products for both convergence tolerances, and the $\mathbf{H}^{(0)}$ convergence trajectory is identical to the DPR trajectory. The last row, with $W_1 = 0$, employs a diagonal $\mathbf{H}^{(1)}$. The convergence trajectory of this row is also equivalent to the DPR trajectory. This is examined in more detail below. It is somewhat disappointing that a tridiagonal $\mathbf{H}^{(1)}$ does not perform significantly better than a diagonal $\mathbf{H}^{(1)}$; linear equation solutions with a tridiagonal matrix require only slightly more effort than those with a diagonal matrix, and combined with the IIGD/GJD method (see Appendix B), this would have been a good alternative way to generate improved expansion vectors with minimal additional effort.

Table III shows the convergence trajectory for level-2 and level-3 SPAM convergence with the same $W = 32$ $\mathbf{H}^{(1)}$ matrix described above, along with a $W = 16$ $\mathbf{H}^{(2)}$ and a $W = 8$ $\mathbf{H}^{(3)}$ matrix. The dynamical adjustment of the intermediate convergence tolerance used previously is generalized to the multilevel case. After each subspace diagonalization, the coefficient vector is decomposed into contributions from the various *wtype* levels. These individual contributions are accumulated to define

$$\left|\mathrm{Sin}\left(\psi^{[n_0, n_1 \ldots n_k]}\right)\right| = \left|c_{(n_0+n_1\ldots+n_k+1):n}\right| \tag{4.4}$$

TABLE III
MultiLevel SPAM Convergence

	2-level SPAM Dynamic Tolerance		3-level SPAM Dynamic Tolerance	
Iteration	$[n_0, n_1, n_2]$	$\|\mathbf{r}\|$	$[n_0, n_1, n_2, n_3]$	$\|\mathbf{r}\|$
1	[0, 0, 1]	1.13E+00	[0, 0, 0, 1]	1.13E+00
2	[0, 0, 2]	3.23E-01	[0, 0, 0, 2]	3.22E-01
3	[0, 0, 3]	1.05E-01	[0, 0, 0, 3]	1.08E-01
4	[0, 0, 4]	2.73E-02	[0, 0, 1, 0]	1.53E-01
5	[0, 0, 5]	5.41E-03	[0, 0, 1, 1]	4.00E-02
6	[0, 1, 0]	1.02E-02	[0, 0, 1, 2]	1.05E-02
7	[0, 1, 1]	1.63E-03	[0, 0, 2, 0]	1.05E-02
8	[0, 1, 2]	3.14E-04	[0, 1, 0, 0]	1.35E-02
9	[0, 1, 3]	1.05E-04	[0, 1, 0, 1]	6.15E-03
10	[0, 2, 0]	1.08E-04	[0, 1, 0, 2]	1.37E-03
11	[1, 0, 0]	1.27E-04	[0, 1, 0, 3]	4.17E-04
12	[1, 0, 1]	5.37E-05	[0, 1, 0, 4]	5.65E-05
13	[1, 0, 2]	1.52E-05	[0, 1, 1, 0]	3.25E-04
14	[1, 0, 3]	3.77E-06	[0, 1, 1, 1]	5.63E-05
15	[1, 0, 4]	5.32E-07	[0, 1, 2, 0]	5.63E-05
16	[1, 0, 5]	7.06E-08	[0, 2, 0, 0]	5.64E-05
17	[1, 1, 0]	2.74E-07	[1, 0, 0, 0]	8.87E-05
18	[1, 1, 1]	2.46E-08	[1, 0, 0, 1]	2.57E-05
19	[1, 1, 2]	4.43E-09	[1, 0, 0, 2]	6.78E-06
20	[1, 2, 0]	4.37E-09	[1, 0, 0, 3]	2.46E-06
21	[2, 0, 0]	5.47E-09	[1, 0, 0, 4]	5.15E-07
22			[1, 0, 1, 0]	1.55E-06
23			[1, 0, 1, 1]	2.56E-07
24			[1, 0, 1, 2]	9.90E-08
25			[1, 0, 2, 0]	9.72E-08
26			[1, 1, 0, 0]	1.16E-07
27			[1, 1, 0, 1]	4.43E-08
28			[1, 1, 0, 2]	1.35E-08
29			[1, 1, 0, 3]	3.89E-09
30			[1, 1, 1, 0]	4.98E-09
31			[1, 2, 0, 0]	5.02E-09
32			[2, 0, 0, 0]	5.29E-09
$N_{product}$	[2, 4, 15]		[2, 4, 7, 19]	

Note. Convergence trajectories of the lowest root of the banded test matrix with $N = 10{,}000$, $W_0 = 64$, and $\Delta = 0.75$. For the SPAM calculations, $W_1 = 32$, $W_2 = 16$, $W_3 = 8$. The convergence criterion is $|\mathbf{r}| < 10^{-8}$.

for each approximation level k. This factor, along with the estimates of the matrix difference norms, provides a prediction for the residual norm after the next contraction at the k-th level according to Eq. (A13). The current intermediate residual norm is compared to these estimates according to

$$\left|\mathbf{r}^{[n_0,n_1\ldots]}\right| \leq \alpha \cdot Max\left\{\left|\mathrm{Sin}\left(\psi^{[n_0,n_1\ldots n_k]}\right)\right| \cdot \left\|\mathbf{H}^{(k+1)} - \mathbf{H}^{(k)}\right\| : k = 0 \ldots wtype_n\right\}. \quad (4.5)$$

The *Max* in this comparison picks out the weakest link in the approximation sequence for

the current set of expansion vectors. Just as in the single-level SPAM discussed above, it does not improve efficiency to converge the intermediate results beyond this value because a larger residual norm will be computed later after some subsequent contraction step. As seen in Table III, this results in an acceptable convergence trajectory without any apparent wasted effort. It should be mentioned that an incorrect estimate of the scale factor α or of a matrix difference norm does not result in incorrect results, it simply results in too much effort required to achieve the correct results. Furthermore, just as for the single-level case, the penalty for choosing an α (or a matrix difference norm estimate) too large is greater than that for choosing an α too small, so, in general, it is better to be too conservative than too optimistic.

In all of the above examples, the traditional Davidson DPR vector has been used to define the new expansion vectors. Before examining other SPAM convergence trajectories, various choices for trial expansion vectors within the SPAM method will be compared. Olsen *et al.* [26] have proposed the Inverse-Iteration Generalized Davidson (IIGD) method for generating expansion vectors within the Davidson subspace method. As discussed in Appendix B, this is equivalent to the Generalized Jacobi–Davidson (GJD) method of Sleijpen *et al.* [27, 28] when applied to the symmetric eigenvalue problem with unit metric matrix and with the same (diagonal) approximate preconditioner. For essentially the same effort, and using the same diagonal preconditioner, the IIGD/GJD method results in an improved expansion vector that sometimes converges better than the traditional Davidson DPR method. Another choice of expansion vector is the residual vector itself. As discussed in Appendix B, this results in the well-known Lanczos method.

The convergence of the Davidson method using these three expansion vector choices is compared in Table IV for the same $W = 64$ banded matrix described above and with the same convergence tolerance. The convergence trajectory for the DPR expansion vector has already been given in Table I. The convergence using the IIGD/GJD expansion vector is essentially identical for this matrix. This is, in part, because the starting vector is the first column of the unit matrix; other starting vector choices would show larger iteration-by-iteration differences. Both the DPR expansion vector and the IIGD/GJD expansion vector require 12 iterations to converge. The Lanczos expansion vector, by contrast, requires 68 iterations to

TABLE IV
Comparison of Various Expansion Vectors

Expansion vector type	$[W]$	n_{max}	$N_{product}$
DPR	[64]	12	[12]
IIGD/GJD	[64]	12	[12]
Lanczos	[64]	68	[68]
SPAM+DPR	[64, 0]	12	[11, 21]
SPAM+IIGD	[64, 0]	12	[11, 21]
SPAM+Lanczos	[64, 0]	28	[13, 131]
SPAM+DPR	[64, 32, 0]	7	[2, 13, 20]
SPAM+IIGD	[64, 32, 0]	7	[2, 13, 20]
SPAM+Lanczos	[64, 32, 0]	77	[2, 14, 289]

Note. Convergence of the lowest root of the banded test matrix with $N = 10{,}000$ and $\Delta = 0.75$. The convergence criterion is $|\mathbf{r}| < 10^{-8}$.

converge. As discussed in Appendix B, this is typical of convergence comparisons between the preconditioned gradient expansion vectors, which selectively converge the eigenpair of interest, and the underlying Krylov subspace that is used in the Lanczos method, which does not converge selectively. Because there are no contractions or restarts in these calculations, the maximum subspace is the same as the number of products for these calculations. Although the subspace diagonalization is still trivial for these cases, even for the slowly convergent Lanczos case, the vector manipulations can become significant, particularly for very large matrix dimensions N.

For comparison purposes, rows 4–6 of Table IV show the convergence results for level-1 SPAM calculations in which $\mathbf{H}^{(1)}$ is chosen to be the same diagonal matrix as the preconditioners used in the DPR and in the IIGD/GJD methods. The three rows correspond to the three different choices for expansion vectors: DPR, IIGD/GJD, and Lanczos. The convergence is identical, iteration by iteration, for the DPR and IIGD/GJD expansion vectors: 11 exact $\mathbf{H}^{(0)}$ matrix–vector products are required and 28 diagonal matrix–vector products are required to converge the highest-level SPAM eigenvalue problem. The fact that 11, rather than 12 (as before), $\mathbf{H}^{(0)}$ products achieves convergence for this problem is an insignificant discretization artifact; as seen in Table I, the residual norm on the 11th DPR iteration is just slightly larger than the convergence tolerance, and for these SPAM convergence cases, it is just slightly below the tolerance on the 11th iteration. This demonstrates that there is no significant advantage of the SPAM method over these other preconditioned expansion vector procedures for this choice of $\mathbf{H}^{(1)}$. As discussed in Appendix B, it is expected that this result will be general. This is because the formal advantages of SPAM are not significant compared to the coarseness of the diagonal $\mathbf{H}^{(1)}$ approximation. Furthermore, although SPAM requires, in principle, several iterations to solve the $\bar{\mathbf{H}}^{[n]}$ eigenvalue equation, in practice it is observed usually that a single DPR (or IIGD/GJD) iteration is sufficient to achieve convergence with the dynamically adjusted tolerance. Forcing convergence beyond this value does not improve significantly the overall efficiency. When only a single iteration is performed to solve the SPAM equation, the expansion vector is exactly the same as that for the DPR method (or whichever expansion vector method is used to generate expansion vectors for the iterative solution of the highest SPAM level). This was demonstrated already in Table II. The results in Table IV were generated by adjusting the estimate for $\|\mathbf{H}^{(1)} - \mathbf{H}^{(0)}\|$ in order to artificially prevent this from occurring for this particular comparison; two or three iterations where required to solve for each SPAM eigenvector for the SPAM/DPR and SPAM/IIGD expansion vectors.

Row 6 of Table IV shows the results for the Lanczos expansion vector. For this expansion vector, 13 exact $\mathbf{H}^{(0)}$ matrix–vector products are required and 131 diagonal matrix–vector products are required to converge the highest-level SPAM eigenvalue problem with the same adjusted estimate for $\|\mathbf{H}^{(1)} - \mathbf{H}^{(0)}\|$ as before. This is an interesting result for several reasons. First, it demonstrates the general principle that the SPAM method isolates the number of exact matrix–vector products that are required to achieve convergence from the quality of the individual expansion vectors. This is true for arbitrary $\mathbf{H}^{(1)}$ approximations, the diagonal approximation here is simply the most extreme example. Second, this example shows that the number of high-level (i.e., more approximate) matrix–vector products generally increases as new SPAM approximation levels are added. This is compensated by a reduced number of low-level (i.e., more exact) products. Whether or not this is beneficial depends on the relative costs of the products with the two different approximations and on the number of products of each that are required. This is discussed in more detail below. Finally, another

TABLE V
Total Effort Model for Various SPAM Levels

$[W]$	$N_{product}$	n_{max}	$\mu=1$	$\mu=3/4$	$\mu=1/2$	$\mu=1/4$	$\mu=1/10$	$\mu=1/100$
[64]	[12]	12	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
[64, 32]	[2, 13]	7	1.2500	0.9792	0.7083	0.4375	0.2750	0.1775
[64, 32, 16]	[2, 4, 15]	6	1.7500	1.1198	0.6458	0.3281	0.2125	0.1701
[64, 32, 16, 8]	[2, 4, 7, 19]	5	2.6667	1.4128	0.6771	0.3112	0.2074	0.1701
[64, 32, 16, 8, 4]	[2, 4, 7, 11, 22]	4	3.8333	1.7116	0.7083	0.3079	0.2069	0.1701
[64, 32, 16, 8, 4, 2]	[2, 4, 7, 12, 15, 21]	4	5.0833	1.9775	0.7370	0.3087	0.2070	0.1701
[64, 32, 16, 8, 4, 2, 1]	[2, 4, 7, 10, 15, 18, 23]	4	6.5833	2.1889	0.7383	0.3063	0.2068	0.1701

Note. Convergence of the lowest root of the banded test matrix with $N = 10{,}000$ and $\Delta = 0.75$. The convergence criterion is $|\mathbf{r}| < 10^{-8}$.

advantage of SPAM in this situation is that the maximum subspace dimension reached during the entire process is only $n = 28$ compared to the $n = 68$ with the straight Lanczos method in row 3.

Rows 7–9 of Table IV show the results for a 2-level SPAM convergence with each of the three choices for expansion vectors. The number of products required are 2, 13, and 20, respectively, for the three matrices with bandwidths of 64, 32, and 0 for the DPR and for the IIGD/GJD expansion vectors. The Lanczos expansion vector requires 2, 14, and 289 matrix–vector products, respectively, for the three matrices. The same general trend is seen as for the previous Lanczos rows in Table IV. Namely, the slow convergence of the Lanczos expansion vector is isolated to the highest approximation level. It is also worth noting that the maximum subspace dimension has increased to $n = 77$, which is larger even than the straight Lanczos convergence in row 3. Although this increase is somewhat artificial because of the adjusted convergence tolerance, the increase is interesting even when compared in a relative way to row 6, which has the same adjusted convergence tolerance.

Multilevel SPAM convergence is examined in Table V. The bandwidths of the various approximation levels are shown in the first column, and the corresponding number of matrix–vector products required to achieve convergence is shown in the second column. The expansion vector in all cases is the DPR procedure, but the IIGD expansion produces identical results. Except for small variations in the matrix–vector product counts resulting from threshold discretization, it is generally observed that as new SPAM levels are added, the counts for the previous levels remain constant. It is only the highest level that is changed when a new level is added. The overall effort required to achieve convergence is given by the sum of the total efforts required for each level. This can be modeled by assuming that the ratio of the effort required,

$$\mu_k = \mathit{Effort}\left(\mathbf{H}^{(k)}\mathbf{x}\right) / \mathit{Effort}\left(\mathbf{H}^{(k-1)}\mathbf{x}\right), \tag{4.6}$$

for each approximation level is the same for all levels. This will not be true in actual applications, but it gives an idea of the general trend of overall efficiency as a function of μ and of the number of approximation levels.

The first column of Table V corresponds to $\mu = 1$, which means that all of the matrix–vector products require the same effort. As expected, it is seen that the overall effort increases with the number of SPAM levels. This is actually the situation for the banded test matrices

used in this section—they all require the same effort regardless of the bandwidth, so no efficiency is gained by approximating one matrix by another. The second column corresponds to $\mu = 3/4$. That is, each successive SPAM level requires 75% of the effort of the previous one. In this case, it is seen that for the convergence rates for this model problem, the minimum overall effort decreases for one SPAM level, and then begins to increase as more approximation levels are added. The third column corresponds to $\mu = 1/2$. For this case, adding one SPAM level reduces the overall effort by about 30%, adding a second level reduces the overall effort by an additional 5%, but adding more SPAM levels causes the overall effort to increase. The next column corresponds to $\mu = 1/4$. For this case, the overall effort decreases down to about 31% with three approximate matrices, and remains fairly constant after that. For $\mu = 1/10$, the overall effort minimizes with three SPAM levels at 21%, and then remains roughly constant beyond that. The last column corresponds to $\mu = 1/100$, and the effort minimizes at 17% with two SPAM levels. The general conclusion from this effort model is that there is some optimum SPAM level for each problem, and increasing the SPAM level beyond that either increases the overall effort, or leaves the overall effort approximately the same so that nothing further is gained. The optimal approximation level at which that minimum effort occurs depends on the accuracy of the sequence of matrix approximations and on the effort required for each matrix–vector product at each approximation level.

It is also observed in Table V that the maximum subspace dimension tends to decrease as the number of SPAM levels increases. This effect is not included into the simple effort model described above, but for very large matrix dimensions, where either memory or external storage is a limiting factor, this can be an important aspect of overall efficiency.

All of the above discussion has concerned convergence of the lowest eigenpair. Convergence of several of the lowest eigenpairs is examined next. There are several ways to converge excited states with the Davidson method. One approach is to converge the lowest vector completely and then save that converged vector and the corresponding matrix–vector product. Then a new trial vector is generated for the second root, and the procedure is restarted with two initial trial vectors ($\mathbf{x}_1$, $\mathbf{x}_2$) and one product vector ($\mathbf{w}_1$). Because the lowest vector satisfies its convergence criteria, all of the expansion vectors in this second step will be directed toward convergence of the second eigenpair. Upon convergence, both of the lowest two vectors and products are saved, a new trial vector is generated for the third root, and the process is continued until all of the desired eigenpairs have been computed. The lowest 10 eigenpairs of the banded test matrix described above are computed in this "one at a time" approach, and the convergence summary is given in the first row of Table VI. The residual norm for each vector is converged to $|\mathbf{r}_j| < 10^{-8}$, the same as the previous calculations. For this particular matrix, the convergence of each new vector requires 11 or 12 iterations, and convergence of all 10 roots requires 118 matrix–vector products total. The maximum subspace dimension reaches its maximum value of $n = 21$ on convergence of the 10th vector. At this time, nine converged vectors for the lower roots have been computed and stored, and while iterating the last vector, twelve additional subspace vectors are required to achieve convergence.

The above "one at a time" procedure for excited states is appropriate when it is not known in advance how many vectors are needed. After each vector is converged, it may be examined to determine if another vector needs to be computed. This characterization is, of course, very problem-specific. If it is known ahead of time how many vectors will be required, then another procedure may be employed. In this approach, all of the requested

TABLE VI
Convergence Results for Multiple Eigenvectors

Method	[W]	n_{max}	$N_{product}$
DPR			
One vector at a time	[64]	21	[118]
Simultaneous/lowest	[64]	42	[42]
Simultaneous/cycle	[64]	28	[28]
Simultaneous/largest $\|\mathbf{r}_j\|$	[64]	28	[28]
SPAM			
One vector at a time	[64, 32]	17	[20, 138]
Simultaneous/lowest	[64, 32]	42	[20, 62]
Simultaneous/cycle	[64, 32]	36	[20, 50]
Simultaneous/largest $\|\mathbf{r}_j\|$	[64, 32]	34	[20, 52]

Note. Convergence of the lowest 10 roots of the banded test matrix with $N = 10{,}000$ and $\Delta = 0.75$. The convergence criterion is $|\mathbf{r}_j| < 10^{-8}$. The matrix-vector product counts are the totals for all 10 roots. The converged computed eigenvalues are: $\lambda_1 = 0.585510562346823$, $\lambda_2 = 1.723295074298214$, $\lambda_3 = 2.808750052512915$, $\lambda_4 = 3.867329659136034$, $\lambda_5 = 4.908652636212611$, $\lambda_6 = 5.937892192171621$, $\lambda_7 = 6.958397150707880$, $\lambda_8 = 7.972562750803514$, $\lambda_9 = 8.982177511445222$, $\lambda_{10} = 9.988585488303615$.

vectors are converged simultaneously, from the same set of expansion vectors. One or more initial vectors are generated, and at each step, one or more unconverged vectors are chosen to define one or more new expansion vectors. If the effort involved in the computation of a matrix–vector product is dominated by processing of the matrix itself (e.g., generation of the matrix elements, indexing of the elements in a sparse data structure, or performing the associated I/O on the matrix elements), then it is beneficial to compute simultaneously several new trial vectors. This is because the cost of the matrix processing is amortized over several vector products. This is the basis of the blocked version of the Davidson method proposed by Liu [4, 9]. However, if the effort is dominated by the multiplications with the vector elements, then it is more efficient to add a single new vector at a time to the subspace. This latter situation is assumed in the SPAM implementation described in this section. There are three ways that this addition of a single expansion vector is done.

The first way is to select the lowest unconverged vector, and use the corresponding Ritz value and residual vector to define the new expansion vector. Once this vector is added to the space, it may benefit not only the selected eigenpair, but also all of the other eigenpairs. In this way, the total effort required for convergence of several eigenpairs is reduced compared to the "one at a time" approach. The total matrix–vector product count for this method is given in the second row of Table VI. The total number of products is reduced to 42, which is a significant reduction compared to the "one at a time" approach. On average, the number of matrix–vector products has been reduced from 11.8/eigenpair down to only 4.2/eigenpair. However, it is also seen that the maximum subspace dimension has increased from $n = 21$ to $n = 42$, so, compared to the "one at a time" approach, there is a tradeoff between reducing the number of expansion vectors and increasing the maximum subspace dimension.

A second way that individual expansion vectors may be selected is to cycle among the unconverged vectors. It is perhaps not obvious why this should result in an improvement,

but experience shows that this is the case for some problems. The qualitative reason for this is that the final expansion vectors computed for a particular almost-converged eigenvector are rather selective for that particular vector and do not benefit the other vectors within the expansion space. In contrast, the vectors that are added early for the poorly converged eigenvectors tend to benefit other nearby poorly converged eigenpairs. By cycling over the roots early, rather than picking one and iterating it to convergence, all the vectors within the space are benefited. The results of this approach are given in the third row of Table VI. It is seen that the total number of products is reduced to 28 vectors, which is, on average, less than three matrix–vector products per converged eigenpair. This improved overall convergence also reduces the maximum subspace dimension down to $n = 28$. This is not as good as the "one at a time" value, but it is better than the second row results.

A third way that individual expansion vectors may be selected is to improve the unconverged vector that has the largest residual norm. The advantage of this approach is that the intermediate Ritz values tend to maintain to the extent possible the same order throughout the convergence process. The convergence results for this method are given in the fourth row of Table VI. For this test case, the convergence is comparable to that for the cycling option.

These same four general approaches to convergence of multiple eigenpairs in the traditional Davidson method also apply to the SPAM method. The matrix–vector product counts are reported in rows 5–8 of Table VI. In all four cases, the SPAM procedure requires only 20 exact $\mathbf{H}^{(0)}$ matrix–vector products to converge all 10 eigenpairs. The number of approximate $\mathbf{H}^{(1)}$ products required shows the same trend as discussed above for the traditional Davidson method. Namely, the "one at a time" approach is least efficient and requires 138 $\mathbf{H}^{(1)}$ products, the "lowest unconverged vector" approach is significantly better with 62 $\mathbf{H}^{(1)}$ products, the "cycle among the unconverged vectors" approach is best and requires only 50 $\mathbf{H}^{(1)}$ products, and the "largest residual" approach is almost as good with 52 $\mathbf{H}^{(1)}$ products. In all cases, the average number of exact products required is reduced to only two per converged eigenpair, which is significantly better than even the best performance that is achieved with the traditional Davidson/DPR procedure. Furthermore, if the $\mathbf{H}^{(1)}$ products are very much cheaper than the $\mathbf{H}^{(0)}$ products, then the use of the SPAM method allows the practical use of the "one at a time" approach to convergence in those cases where the number of converged eigenpairs is unknown at the beginning, and each converged vector must be examined. The maximum subspace dimensions for these four SPAM cases follow the same trend as for the analogous four DPR cases. Namely, the "one at a time" approach has the smallest subspace requirements, whereas the simultaneous convergence options, cycling among the unconverged vectors, choosing the largest residual, and iterating on the lowest unconverged vector, result in larger subspace requirements.

The previous discussion has assumed that the lowest eigenpairs within the spectrum are desired. All of these vector choices apply also to the convergence of the highest eigenpairs within the spectrum. An example of this is given below.

In the above simultaneous convergence examples, all the requested vectors are converged relative to their own dynamical convergence tolerances at each SPAM level before contraction of the vectors to the next lower (more accurate) SPAM level occurs. This contraction involves the projection operator $\mathbf{Q}^{[n]}$, followed by orthonormalization of all the vectors, and this projection may introduce linear dependencies in the set of contracted vectors. Early during the iterations at some level, none of the vectors are converged, so a new expansion vector is computed for each requested eigenpair. However, not all of the vectors converge on the same iteration; some will converge before others and new expansion vectors are added

only for unconverged eigen pairs. Consequently, there are several situations that can occur during contraction of the vectors: (1) There are more vectors than roots, and each root has at least one expansion vector computed for it. In this case the projected vectors will be linearly independent. (2) There are more new expansion vectors than roots sought, but some roots do not have expansion vectors because they are already converged at that level. The projected vectors may be linearly dependent in this case. (3) There are fewer new expansion vectors than roots. There may be linear dependencies in the projected vectors in this case. In the SPAM implementation described here, all three of these situations are treated with singular value decomposition (SVD). The subblock of the coefficient matrix is decomposed according to

$$\mathbf{c}_{(n_m+1):n,1:n_r} = \mathbf{U}\sigma\mathbf{V}^T, \tag{4.7}$$

where n_r is the number of requested roots (or the current subspace dimension as appropriate); n_m is the number of expansion vectors up through the mth SPAM level (i.e., the ones *not* being contracted); **U** and **V** are orthogonal square matrices; and σ is the "diagonal" matrix of singular values. In general, the subblock of **c** is rectangular, not square, and σ has the same dimensions as the subblock of **c**. All three of the above situations may be treated by examining the ratio of the singular values σ_j/σ_1. When this ratio becomes too small, less than about 0.1 in most situations, then the corresponding vector in **U** may be safely ignored without affecting convergence. In case (1) above, there will be n_r singular value ratios that are very close to 1.0. In case (2) above, there will be one or more singular values close to 1.0, and the remaining singular values will be small (usually 0.001 or smaller). In case (3), there will be one or more ratios close to 1.0, but there may be also small singular values that must be deleted. Once the number of "large" singular values are identified, the corresponding columns of **U** define the appropriate contraction coefficients and the expansion vectors are contracted accordingly. In the special case of a single root, this procedure is always equivalent to the contraction described in Figs. 2 and 3. It is only for simultaneous convergence of several states that the SVD procedure is used to recognize linear dependencies. There are several features of this SVD transformation that are important. Because the columns of **U** are orthonormal, and the underlying expansion space is already orthonormal, the vectors may be contracted without further orthonormalization. Also, all of the above SVD operations occur just within the subspace; manipulations within the large vector space $\mathbf{X}^{[n]}$ are therefore simplified or eliminated entirely. The matrix **V** is not used in this procedure, so it need not be computed or stored. In situations for which loose convergence criteria are specified for some eigenpairs, and tight convergence criteria are specified for others, it is convenient to weight correspondingly the columns of **c** prior to the SVD procedure.

There are two other forms of excited state convergence that are implemented within the SPAM procedure discussed in this section. In some situations, a single interior eigenpair is desired of some unknown index j within the entire spectrum ($1 \ldots N$), but a good estimate of the final converged eigenvalue is known. After the Ritz values within the subspace are determined, the vector associated with the approximate eigenvalue closest to this reference value is used to define the next expansion vector. This is called the *root-homing* mode. In order to converge to the correct eigenpair, a good initial guess for the vector in addition to the eigenvalue is required, and the target eigenvalue should be well-separated from other nearby eigenvalues. An example of root-homing convergence is given in Table VII. An estimate of

TABLE VII
Interior Eigenpair Convergence

Method	$[W]$	n_{max}	$N_{product}$
Root-homing, DPR expansion vector	[64]	20	[20]
Root-homing, IIGD expansion vector	[64]	16	[16]
Root-homing, SPAM+DPR expansion vector	[64, 32]	16	[2, 25]
Root-homing, SPAM+IIGD expansion vector	[64, 32]	16	[2, 19]
Vector-following, DPR expansion vector	[64]	19	[18]
Vector-following, IIGD expansion vector	[64]	20	[20]
Vector-following, SPAM+DPR expansion vector	[64, 32]	15	[2, 24]
Vector-following, SPAM+IIGD expansion vector	[64, 32]	16	[2, 22]

Note. Convergence of an interior root of the banded test matrix with $N = 10{,}000$ and $\Delta = 0.75$. The convergence criterion is $|\mathbf{r}| < 10^{-8}$. In root-homing mode, $\rho_{ref} = 10.0$ and $\mathbf{x}_1 = \mathbf{e}_{11}$. In vector-following mode, $\mathbf{z} = \mathbf{x}_1 = \mathbf{e}_{11}$ and $\mathbf{v}^T\mathbf{z} = 0.7439$. In all cases, the converged eigenpair corresponds to $\lambda_{10} = 9.988585488303615$.

the eigenvalue is $\rho_{ref} = 10.0$ and the starting vector is $\mathbf{x}_1 = \mathbf{e}_{11}$, the 11th column of the unit matrix. The Davidson procedure with the DPR expansion vector converges to the appropriate root in 20 iterations. In this case, the IIGD expansion vector converges in only 16 iterations. Applying a single-level SPAM to this requires only two exact matrix–vector products to converge, a significant reduction. SPAM using the IIGD expansion vector requires the same number of exact products, but it reduces the number of approximate products compared to the DPR expansion vector.

The other excited state method applies to the situation in which the index j within the entire spectrum $(1 \ldots N)$ is unknown, but it is the character of the eigenvector that determines the appropriate eigenpair. Suppose that there is some reference vector $\mathbf{z}$, perhaps that results from some simplified model problem, or the solution of an eigenvalue equation that is "similar" to the current problem in some general sense. After the determination of the Ritz vectors, the overlaps $(\mathbf{z}^T\mathbf{v}_j)$ may be computed for $(j = 1 \ldots n)$, the current subspace dimension. Then the approximate vector with the largest absolute overlap is chosen to define the next correction vector. This is called the *vector-following mode*. An example of vector-following convergence is given in Table VII. The same starting vector $\mathbf{x}_1 = \mathbf{e}_{11}$ is used as before for the root-homing mode, and this same vector is used also to define the reference vector. The convergence trajectory is slightly different for vector-following than for root-homing, and in this particular case the DPR expansion vector performs slighltly better than the IIGD expansion vector for the straight Davidson method, but the IIGD expansion vector performs slightly better for the SPAM method. In both of the SPAM calculations, only two exact matrix–vector products are required to achieve convergence, and these are significant improvements over the straight Davidson DPR and IIGD results.

Inspection of the converged eigenvector in both the root-homing and vector-following calculation reveals that the initial vector overlap with the final converged eigenvector is only 0.7439. If a better starting vector is used, then convergence improves for the DPR and IIGD methods. The excellent convergence results of SPAM in this case demonstrate the inherent advantage of isolating the quality of the initial vector from the $\mathbf{H}^{(0)}$ convergence rate. In this case, even starting with a relatively poor starting vector, the SPAM method converges in

the same number of iterations as does a SPAM ground state calculation with the same $\mathbf{H}^{(1)}$. By contrast, the Davidson DPR and IIGD methods require almost twice as many iterations for this interior eigenpair (with a poor starting vector) as they require for the ground state calculation (with a better starting vector).

Tensor-Product Examples: Tensor-product (also called direct-product, or Kronecker product) matrices occur in many application areas. Examples include separable differential equations, boundary value problems, translational and rotational operators in many-body problems, and symmetry operators in group theory. A tensor product of two matrices is defined by

$$(\mathbf{A} \otimes \mathbf{B})_{(ij)(kl)} = A_{ik} B_{jl}. \tag{4.8}$$

If the dimensions of the component matrices $\mathbf{A}$ and $\mathbf{B}$ are $N_A \times M_A$ and $N_B \times M_B$, respectively, then in the tensor-product "matrix," (ij) is treated as a single row index that ranges from 1 to $N_A N_B$, and (kl) is treated as a single column index with range 1 to $M_A M_B$. Equation (4.8) may be used to demonstrate the following useful relations with tensor-product matrices:

$$\begin{aligned}
&a.\ (\mathbf{A}+\mathbf{B}) \otimes \mathbf{C} = \mathbf{A} \otimes \mathbf{C} + \mathbf{B} \otimes \mathbf{C} \\
&b.\ (\mathbf{A} \otimes \mathbf{B}) \otimes \mathbf{C} = \mathbf{A} \otimes (\mathbf{B} \otimes \mathbf{C}) \\
&c.\ (\mathbf{AB}) \otimes (\mathbf{CD}) = (\mathbf{A} \otimes \mathbf{C})(\mathbf{B} \otimes \mathbf{D}) \\
&d.\ (\mathbf{A} \otimes \mathbf{B})^{-1} = \mathbf{A}^{-1} \otimes \mathbf{B}^{-1} \\
&e.\ Rank(\mathbf{A} \otimes \mathbf{B}) = Rank(\mathbf{A}) \cdot Rank(\mathbf{B}) \\
&f.\ Tr(\mathbf{A} \otimes \mathbf{B}) = Tr(\mathbf{A}) \cdot Tr(\mathbf{B}) \\
&g.\ \{\lambda(\mathbf{A} \otimes \mathbf{B})\} = \{\lambda_j(\mathbf{A}) \cdot \lambda_k(\mathbf{B}) : j = 1 \ldots N_A, k = 1 \ldots N_B\} \\
&h.\ Det(\mathbf{A} \otimes \mathbf{B}) = Det(\mathbf{A})^{N_B} Det(\mathbf{B})^{N_A}
\end{aligned} \tag{4.9}$$

All of these relations generalize in the obvious way for tensor-products of three or more component matrices. Consider a general matrix–vector product of a tensor-product matrix with a vector: $\mathbf{w} = (\mathbf{A} \otimes \mathbf{B})\mathbf{x}$. In the general dense case, this would appear to require $N_A N_B M_A M_B$ floating point multiplications (and an equal number of additions). However, Eq. (4.8) allows the matrix-vector product to be rewritten in the form

$$w_{ij} = \sum_{(kl)} (\mathbf{A} \otimes \mathbf{B})_{(ij)(kl)} x_{(kl)} = \sum_k A_{ik} \left(\sum_l B_{jl} x_{kl} \right) \tag{4.10}$$

The term in parentheses is a matrix–matrix product that requires $M_A N_B M_B$ floating point multiplications. The second summation, over k is a second matrix–matrix product that requires $N_A M_A N_B$ floating point multiplications. For rectangular matrices $\mathbf{A}$ and $\mathbf{B}$, a different operation count may result if the summation order is interchanged. Particularly for square component matrices of large dimension, matrix–vector products involving tensor-product matrices are much easier to compute in this "operator" form than those of a general matrix of the same dimensions. Sparseness and symmetry of the component matrices can reduce the operation counts even below those given above. In the more general case, matrix–vector

products of tensor-product matrices may be computed as

$$w_{(i_1 i_2 \ldots i_m)} = \sum_{(j_1 j_2 \ldots j_m)} \left(\mathbf{A}^{(1)} \otimes \mathbf{A}^{(2)} \cdots \otimes \mathbf{A}^{(m)}\right)_{(i_1 i_2 \ldots i_m)(j_1 j_2 \ldots j_m)} x_{(j_1 j_2 \ldots j_m)}$$
$$= \sum_{j_1} A^{(1)}_{i_1 j_1} \left(\sum_{j_2} A^{(2)}_{i_2 j_2} \cdots \left(\sum_{j_m} A^{(m)}_{i_m j_m} x_{(j_1 j_2 \ldots j_m)} \right) \right). \tag{4.11}$$

In other words, each component matrix $\mathbf{A}^{(k)}$ is used to transform one index in the "vector" $\mathbf{x}$, and there are m such nested one-index transformations. With a suitable arrangement of the subscript indices, each one-index transformation is a matrix–matrix product, and for rectangular component matrices, the total operation count depends on the order of the summations. If each component matrix is square and of dimension N, then Eq. (4.11) requires only $mN^{(m+1)}$ floating point multiplications. This should be compared to the N^{2m} multiplications that are required for a general matrix–vector product involving a matrix of dimension N^m. Therefore, when treated in "operator" form as in Eq. (4.11), matrix–vector products involving tensor-product matrices can require much less effort than a general matrix–vector product of the same dimension.

Eqs. (4.9) may be used to show that the eigenpairs of a tensor-product matrix are given by

$$\begin{aligned} \left((\mathbf{A} \otimes \mathbf{B}) - \lambda_{(ij)}\right) \mathbf{v}_{(ij)} &= 0 \\ \mathbf{v}_{(ij)} &= \mathbf{v}^A_i \otimes \mathbf{v}^B_j \\ \lambda_{(ij)} &= \lambda_i(\mathbf{A}) \cdot \lambda_j(\mathbf{B}) \end{aligned} \tag{4.12}$$

Murray *et al.* [25] have proposed the use of tensor-product matrices as test problems for iterative diagonalization methods because the exact eigenvalues and eigenvectors may be determined in this product manner and compared to the results from the iterative calculation.

In the present work, the reduced computational effort for the matrix–vector products involving tensor-product matrices is exploited in a different manner. Suppose the eigenpairs are required for some large matrix $\mathbf{H}^{(0)}$. It is assumed that $\mathbf{H}^{(0)}$ is not a tensor-product matrix, but a good approximation $\mathbf{H}^{(1)}$ exists that is of tensor-product form. Such approximations often occur naturally, for example, from low-order operator expansions or truncations, combined with an appropriate formal expansion basis. The goal in the present work is to exploit the tensor-product nature of $\mathbf{H}^{(1)}$ in order to improve the efficiency of the eigenvector determination of $\mathbf{H}^{(0)}$.

In order to model this general kind of matrix decomposition, a perturbed-tensor-product matrix $\mathbf{H}^{(0)}$ will be defined as

$$\mathbf{H}^{(0)} = \mathbf{H}^{(1)} + \beta \boldsymbol{\Delta}, \tag{4.13}$$

in which $\mathbf{H}^{(1)} = \mathbf{A}^{(1)} \otimes \mathbf{A}^{(2)} \otimes \cdots \otimes \mathbf{A}^{(m)}$ is the m-fold tensor-product of the 4×4 matrices used by Murray *et al.* [25]:

$$\mathbf{A}^{(k+1)} = \begin{pmatrix} 3+k/10 & 1/10 & 2/10 & 3/10 \\ 1/10 & 4+k/10 & 0 & 0 \\ 2/10 & 0 & 5+k/10 & 0 \\ 3/10 & 0 & 0 & 6+k/10 \end{pmatrix}; \quad \text{for } k = 0 \ldots (m-1). \tag{4.14}$$

The perturbation matrix $\mathbf{\Delta}$ is defined with the elements

$$\begin{aligned} \Delta_{jk} &= -1/2; \quad \text{for } |j-k| = 1 \\ \Delta_{1N} &= \Delta_{N1} = -1/2 \\ \Delta_{jk} &= 0; \quad \text{otherwise.} \end{aligned} \tag{4.15}$$

This matrix occurs in the Hückel theory of the molecular electronic structure of cyclic polyenes[10], and both the eigenvalues and the eigenvectors have closed-form, analytic solutions. The lowest eigenvalue is $\lambda_1 = -1$, and the corresponding (unnormalized) eigenvector is given by $v_k = 1$ for $k = 1 \ldots N$; the largest eigenvalue is $\lambda_N = +1$, and the corresponding eigenvector is $v_k = (-1)^k$ for $k = 1 \ldots N$; the remaining eigenvalues are doubly degenerate and are otherwise distributed evenly about zero in between these extreme values. This results in the norms $\|\mathbf{\Delta}\| = 1$ and $\|\mathbf{H}^{(1)} - \mathbf{H}^{(0)}\| = \beta$. The perturbation is not of a tensor-product form, and this ensures that $\mathbf{H}^{(0)}$ is not an exact tensor-product. However, β will be chosen appropriately in order to ensure that $\mathbf{H}^{(1)}$ is a good approximation that can be used to accelerate convergence of the eigenvectors. The sparse form of $\mathbf{\Delta}$ allows for efficient computation of matrix–vector products, and this combination results in good test problems for the SPAM method. The test cases in [25] involve the 8-fold and the 10-fold tensor products. The corresponding matrix dimensions are $4^8 = 65{,}536$ and $4^{10} = 1{,}048{,}576$, respectively. Ordinarily, dense matrix–vector products with matrices of these dimensions would require $4^{16} = 4.3 \cdot 10^9$ and $4^{20} = 4.1 \cdot 10^{12}$ floating point multiplications respectively; by contrast the tensor-product contributions, computed according to Eq. (4.11), require only $8 \cdot 4^9 = 2.1 \cdot 10^6$ and $10 \cdot 4^{11} = 4.2 \cdot 10^7$ floating point multiplications, respectively (ignoring the sparseness and symmetry in the component matrices), and the operation count for the perturbation matrix is insignificant. If these test matrices are taken as models of general matrices of the same dimensions, then these operation counts would results in effort ratios of $\mu_8 = 4.9 \cdot 10^{-4}$ and $\mu_{10} = 1.0 \cdot 10^{-5}$. These effort ratios are typical of tensor-product approximations, and these examples show the tremendous advantage this type of approximation offers in improving efficiency when combined with the SPAM method. Although these test cases are nontrivial, they do provide a relatively inexpensive (a few seconds for each matrix–vector product on current desktop computers) model for testing the behavior of SPAM for tensor-product matrices.

The convergence summaries are given in Table VIII for the lowest few roots of the $m = 8$ and $m = 10$ matrices. For both matrices, the perturbation parameter β was chosen to result in 10 to 20 DPR iterations with the usual Davidson method to converge the lowest eigenpair. This level of perturbation is representative of operator approximations in many applications. The same four convergence approaches are taken as before: the vectors are converged either sequentially or simultaneously, and the three possible choices to determine the next expansion vector are compared for the simultaneous convergence cases. In all cases, the initial vectors were chosen to be the appropriate $\mathbf{H}^{(1)}$ eigenvector, which was computed as a tensor product of the component matrix eigenvectors according to Eq. (4.12).

The DPR convergence summaries for the lowest 10 roots are given in the first four rows of Table VIII. As with the previous banded matrix examples, the simultaneous convergence options result in fewer matrix–vector products than the "one at a time" convergence approach, and the simultaneous convergence options require larger maximum subspaces.

The convergence summaries for the SPAM calculations, with the usual DPR expansion vector, are given in rows 4–8 in Table VIII. Significant reductions in the numbers of $\mathbf{H}^{(0)}$

TABLE VIII
Perturbed-Tensor-Product Convergence Results for Multiple Eigenvectors

Method	$m=8$			$m=10$		
	n_{max}	$N_{product}$	*Effort*	n_{max}	$N_{product}$	*Effort*
DPR						
One vector at a time	40	[203]	1.000	26	[145]	1.000
Simultaneous/lowest	82	[82]	1.000	99	[99]	1.000
Simultaneous/cycle	94	[94]	1.000	94	[94]	1.000
Simultaneous/largest $\|\mathbf{r}_j\|$	95	[95]	1.000	93	[93]	1.000
SPAM+DPR			$\mu = 4.9 \cdot 10^{-4}$			$\mu = 1.0 \cdot 10^{-5}$
One vector at a time	42	[20, 164]	0.099	26	[20, 132]	0.138
Simultaneous/lowest	83	[19, 87]	0.232	99	[20, 104]	0.202
Simultaneous/cycle	86	[19, 90]	0.203	94	[20, 101]	0.213
Simultaneous/largest $\|\mathbf{r}_j\|$	95	[19, 99]	0.201	94	[20, 99]	0.215
SPAM+IIGD			$\mu = 2.0 \cdot 10^{-3}$			$\mu = 4.0 \cdot 10^{-5}$
One vector at a time	11	[20, 20]	0.099	11	[20, 20]	0.138
Simultaneous/lowest	20	[19, 21]	0.232	20	[20, 21]	0.202
Simultaneous/cycle	20	[19, 21]	0.203	20	[20, 21]	0.213
Simultaneous/largest $\|\mathbf{r}_j\|$	20	[19, 21]	0.200	20	[20, 21]	0.215

Note. Convergence summaries of the lowest 10 roots of the $m=8$ and $m=10$ perturbed-tensor-product matrices described in the text. The initial vectors in all cases are the eigenvectors of the tensor-product matrices, which were computed as tensor-products of the eigenvectors of the 4×4 component matrices. The matrix–vector product counts are the totals for all 10 roots. For the $m=8$ calculations, $N=65{,}536$, $\beta=10$, and $|\mathbf{r}_j| < 10^{-1}$. For the $m=10$ calculations, $N=1{,}048{,}576$, $\beta=100$, and $|\mathbf{r}_j| < 10^{0}$.

products are achieved for the "one at a time" convergence mode and for the simultaneous convergence modes. The total relative effort is given for each convergence mode relative to the corresponding DPR convergence using the μ effort ratios discussed above. The reduction in effort is significant for all of the SPAM cases, but largest for the "one at a time" convergence mode, resulting in a 91% reduction of effort relative to the DPR "one at a time" calculation for the $m=8$ matrix, and an 86% reduction of effort for the $m=10$ matrix.

In addition to using the diagonal elements of $\mathbf{H}^{(0)}$ as the preconditioner in the DPR method, the tensor-product nature of $\mathbf{H}^{(1)}$ allows for a significant improvement when using the IIGD/GJD procedure to determine the expansion vectors:

$$\left(\mathbf{H}^{(1)} - \rho\right)\delta^{IIGD} = -\mathbf{r} + \varepsilon\mathbf{x}. \tag{4.16}$$

The spectral form, $(\mathbf{H}^{(1)} - \rho) = \mathbf{U}(\mathbf{D} - \rho)\mathbf{U}^T$, with

$$\mathbf{U} = \mathbf{U}^{(1)} \otimes \mathbf{U}^{(2)} \otimes \cdots \otimes \mathbf{U}^{(m)} \tag{4.17}$$

in which $\mathbf{U}^{(k)}$ is the set of eigenvectors of the component matrix $\mathbf{A}^{(k)}$, allows for the efficient computation of the inverse. This is called a *fast inverse* procedure. For the component matrices in Eq. (4.14), the eigenvectors are the same for each component matrix, and the corresponding component eigenvalues are related by a uniform shift of 1/10 from the previous component matrix. This leads to some simplifications in computing the fast inverse, but it does not result in any significant additional performance advantage. The eigenvalues $\mathbf{D}$ are products of the component matrix eigenvalues, the generalization of Eq. (4.12). This spectral decomposition allows the IIGD/GJD expansion vector to be computed

with the steps

$$\begin{aligned} \mathbf{r}_U &= \mathbf{U}^T\mathbf{r} \\ \mathbf{x}_U &= \mathbf{U}^T\mathbf{x} \\ (\mathbf{D}-\rho)\delta_U &= -\mathbf{r}_U + \varepsilon\mathbf{x}_U \\ \delta^{IIGD} &= \mathbf{U}\delta_U . \end{aligned} \tag{4.18}$$

During the SPAM iteration, each IIGD/GJD expansion vector requires four total matrix–vector products (two with $\mathbf{U}^T$, one with $\mathbf{U}$, and one, after orthonormalization, with $\mathbf{H}^{(1)}$) compared to the single $\mathbf{H}^{(1)}$ matrix–vector product each iteration with the simple diagonal preconditioner. The effort ratio μ is four times larger for this procedure than that for a SPAM iteration involving the simple diagonal preconditioner. Therefore, in order to improve efficiency, the IIGD/GJD procedure should converge in $1/4$, or fewer, of the number of SPAM iterations required with the simple diagonal preconditioner. The optimal choice of expansion vector method is therefore problem-specific.

The convergence summary of SPAM using the IIGD/GJD expansion vectors is given in the last four rows of Table VIII. These results may be compared directly with the previous four rows, which used the DPR expansion vectors in the SPAM procedure. For both the $m=8$ and $m=10$ matrices, the SPAM + IIGD expansion vectors result in significant reduction in the number of $\mathbf{H}^{(1)}$ products that are required, but, because of the larger effort ratios μ, only modest overall efficiency improvements compared to the DPR expansion vectors. However, the maximum subspace dimension is reduced significantly for the IIGD/GJD expansion vector choice compared to the DPR expansion vector choice.

For future reference, the lowest computed eigenvalues are given in Table IX for both of these tensor-product test matrices. The unperturbed $\beta=0$ eigenvalues are the tensor-product eigenvalues, the lowest of which are given by Murray *et al.* (note the typographical error for λ_2 in Ref. 25). These may also be computed by taking the appropriate products of the eigenvalues of the component matrices of Eq. (4.14). The perturbed $\beta \neq 0$ eigenvalues have no simple or closed-form solution.

MRSDCI Examples: The multireference single- and double-excitation configuration interaction (MRSDCI) code in the COLUMBUS Program System [11, 12] is a "direct-CI"

TABLE IX
Eigenvalues of the Perturbed-Tensor-Product Matrices

	$m=8;\ N=65{,}536$		$m=10;\ N=1{,}048{,}576$	
	$\beta=0$	$\beta=10$	$\beta=0$	$\beta=100$
λ_1	13517.53848	13518.20621	194306.6355	194313.3266
λ_2	17479.77431	17479.04546	248296.3451	248289.0640
λ_3	17591.64848	17592.44787	249739.2683	249747.2806
λ_4	17710.02377	17710.67916	251261.4363	251268.0025
λ_5	17835.48382	17836.15370	252869.5615	252876.2711
λ_6	17968.68428	17969.35303	254571.1364	254577.8342
λ_7	18110.36428	18111.03326	256374.5505	256381.2504
λ_8	18261.36016	18262.02920	258289.2285	258295.9287
λ_9	18422.62196	18423.29105	260325.7948	260332.4955
λ_{10}	21228.42326	21228.60963	262496.2714	262502.9724

method, which means that the Hamiltonian matrix is treated in operator form—the required matrix–vector products are computed "directly" from the underlying repulsion integrals and coupling coefficients. The repulsion integrals are partitioned based on the number of "internal" and "external" orbital indices, and the coupling coefficients are partitioned and computed correspondingly. The most important contributions to the eigenvalue are from the repulsion integrals indexed by four "internal" orbital indices, g_{pqrs}; these include both the reference configuration state function (CSFs) and those related to the reference CSFs by rearrangements of the electrons within the internal orbitals. In the graphical unitary group approach used in the COLUMBUS Program System, these CSFs are called the "z-walks." The next most important contributions to the eigenvalue are those that involve the interactions of the z-walks with the other expansion CSFs. These involve only the small subset of the integrals with three internal, g_{apqr}, and two internal, g_{abpq} and g_{apbq}, orbital indices. These interactions are sufficient to determine the first-order wave function and the second-order energy contributions in the perturbation expansion. An approximate Hamiltonian matrix may be defined that consists only of the diagonal elements and of the rows and columns corresponding to the z-walks. This is called a "B_k" approximate Hamiltonian matrix. Matrix–vector products with the B_k and exact Hamiltonian matrices correspond to typical effort ratios of $\mu = 10^{-1}$ to $\mu = 10^{-3}$.

This suggests the use of the B_k Hamiltonian as the $\mathbf{H}^{(1)}$ matrix, and $\mathbf{H}^{(0)}$ as the exact matrix in the SPAM procedure [13]. The convergence summaries for two test calculations are given in Table X. The first calculation is for a single-reference wave function for the $^3B_1$ ground state of the CH_2 molecule. This small calculation consists of 2,036 expansion CSFs with two z-walks, and this results in a measured effort ratio of $\mu = 1.03 \cdot 10^{-1}$. The second calculation is for a multireference wave function for the ground state of the CH_3 radical. This is a larger test case, but is still modest, with 70,254 expansion CSFs and 188 z-walks, and this results in a measured effort ratio of $\mu = 6.04 \cdot 10^{-2}$. In both cases, the initial vector is the column of a unit matrix corresponding to the lowest diagonal element. As seen in Table X, the SPAM procedure only improves the overall efficiency by a modest factor of 10%–30%, depending on the convergence tolerance. This is because the B_k Hamiltonian is a rather poor approximation to the exact Hamiltonian matrix, and leads to a large $\|\mathbf{H}^{(1)} - \mathbf{H}^{(0)}\|$ (which was empirically estimated in these calculations for the dynamic tolerance). This

TABLE X
Convergence Summary for MRSDCI Calculations

Convergence tolerance	Calculation type	$CH_2(^3B_1)$		$CH_3(^2A_2'')$	
		n_{max}	$N_{product}$	n_{max}	$N_{product}$
$\|\mathbf{r}\| < 10^{-3}$	DPR	6	[6]	6	[6]
	SPAM	5	[4, 7]	6	[5, 9]
$\|\mathbf{r}\| < 10^{-5}$	DPR	9	[9]	10	[10]
	SPAM	8	[7, 12]	9	[9, 14]
$\|\mathbf{r}\| < 10^{-7}$	DPR	12	12	15	[15]
	SPAM	10	[10, 16]	13	[13, 18]

Note. The SPAM calculations use the B_k Hamiltonian for $\mathbf{H}^{(1)}$. $N = 2{,}036$ and $N_z = 2$ for the CH_2 calculations; $N = 70{,}254$ and $N_z = 188$ for the CH_3 calculations. The initial vector in all cases is the column of the unit matrix corresponding to the lowest diagonal element.

is also evident from the convergence trajectories in which a single $\mathbf{H}^{(1)}$ iteration often is followed immediately by a subsequent $\mathbf{H}^{(0)}$ iteration. Future effort will be directed toward finding more accurate approximate $\mathbf{H}^{(1)}$ Hamiltonian matrices.

Rational-Function Direct-SCF Examples: Self-Consistent Field (SCF) wave function optimization involves the optimization of a trial electronic structure wave function with respect to the essential subset of orbital rotations [14]. One approach to this nonlinear optimization problem involves a sequence of rational function approximations. Optimization of an intermediate rational function approximation results in the eigenvalue equation

$$\begin{pmatrix} \mathbf{B}-\lambda & \mathbf{w} \\ \mathbf{w}^T & -\lambda \end{pmatrix} \begin{pmatrix} \mathbf{k} \\ 1 \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ 0 \end{pmatrix}. \tag{4.19}$$

In this equation, the matrix $\mathbf{B}$ is the orbital-rotation Hessian matrix (the matrix of second derivatives) evaluated with the current reference wave function, the vector $\mathbf{w}$ is the orbital-rotation gradient, and $\mathbf{k}$ is the vector that defines the optimal orbital rotations within the local rational-function approximation. The vector $\mathbf{k}$ is used to update the wave function and to define a new reference wave function expansion point for the next rational function approximation; this sequence of wave function updates constitutes an "outer" iteration. For each "outer" iteration, a single eigenpair of Eq. (4.19) is required, and it is the one that corresponds to the lowest eigenvalue. The iterative solution of this eigenvector is the "inner" iteration. Further details of this kind of wave function optimization may be found in [14, 15]. For the present discussion, the form of the matrix $\mathbf{B}$ is of interest:

$$B_{(ia)(jb)} = 2F_{ab}^{[MO]}\delta_{ij} - 2F_{ij}^{[MO]}\delta_{ab} + 8\left(2g_{aibj}^{[MO]} - \frac{1}{2}g_{ajbi}^{[MO]} - \frac{1}{2}g_{abij}^{[MO]}\right). \tag{4.20}$$

During the eigenpair solution, the Fock matrix elements F_{ab} and F_{ij} are available, but the remaining repulsion integral contributions to the matrix ($2g_{aibj} - 1/2g_{ajbi} - 1/2g_{abij}$) are relatively expensive to include. For large molecular problems in which the "direct-SCF" approach is used, these contributions must be recomputed on-the-fly as the matrix–vector products are needed during the iterative solution to the eigenvalue equation [16]. This suggests the SPAM procedure using the approximation

$$B_{(ia)(jb)}^{(0)} = B_{(ia)(jb)}^{(1)} + 8\left(2g_{aibj}^{[MO]} - \frac{1}{2}g_{ajbi}^{[MO]} - \frac{1}{2}g_{abij}^{[MO]}\right) \tag{4.21}$$

$$B_{(ia)(jb)}^{(1)} = 2F_{ab}^{[MO]}\delta_{ij} - 2F_{ij}^{[MO]}\delta_{ab} = 2\left(\mathbf{1}_{dd} \otimes \mathbf{F}_{vv}^{[MO]} - \mathbf{F}_{dd}^{[MO]} \otimes \mathbf{1}_{vv}\right)_{(ia)(jb)}. \tag{4.22}$$

The tensor-product form of the $\mathbf{B}^{(1)}$ matrix is shown explicitly in Eq. (4.22). The relative effort between a $\mathbf{B}^{(0)}$ and the simpler $\mathbf{B}^{(1)}$ matrix–vector product ranges from $\mu = 10^{-2}$ to $\mu = 10^{-4}$ or better [15, 17]. This approximation to the Hessian matrix has also been used successfully by Wong and Harrison [18] in a preconditioned-conjugate-gradient optimization. Table XI summarizes the DPR and SPAM convergence using this tensor-product approximation to the Hessian matrix for the $Fe(CO)_5$ molecule. This calculation requires three or four "outer" iterations (each of which requires a new eigenvector solution) to converge, depending on the overall convergence tolerance. The number of matrix–vector products required to achieve convergence is given for each of the "outer" iterations, along with the overall totals. The efficiency improvements are modest for this calculation, ranging from 10% to 30% reductions in the total effort.

TABLE XI
Convergence Summary for Rational-Function SCF Optimizations for $Fe(CO)_5$

Convergence tolerance	Calculation type	$N_{product}$ total	$N_{product}$ "Outer" iterations 1	2	3	4
$\|\mathbf{r}\| < 10^{-3}$	DPR	[8]	[3]	[4]	[1]	
	SPAM	[7, 13]	[2, 3]	[4, 9]	[1, 1]	
$\|\mathbf{r}\| < 10^{-5}$	DPR	[13]	[3]	[6]	[3]	[1]
	SPAM	[10, 20]	[2, 3]	[4, 9]	[3, 7]	[1, 1]
$\|\mathbf{r}\| < 10^{-7}$	DPR	[16]	[3]	[6]	[6]	[1]
	SPAM	[13, 33]	[2, 3]	[4, 9]	[6, 20]	[1, 1]

There are two other important features of this particular optimization problem that should be mentioned because they apply generally to other similar optimization problems. First, because the eigenvalue equation is embedded within an "outer" level optimization process, the convergence criteria for the individual eigensolutions changes as the overall optimization process converges; in particular in the present application, during the initial outer iterations, the eigensolution involving $\mathbf{B}^{(1)}$ alone is often sufficiently accurate, and the costs for the $\mathbf{B}^{(0)}$ products increases as the repulsion integral thresholds are tightened toward convergence. Secondly, the above equations are written in the molecular orbital [*MO*] basis. However, the actual calculations are done in the atom-centered atomic-orbital [*AO*] basis where computation of the repulsion integrals $\mathbf{g}^{[AO]}$ is easiest; $\mathbf{B}^{(1)}$ is also a tensor-product matrix in this basis [14–17]. This is typical of such tensor-product approximations. Because the tensor-product nature of the matrix is maintained after such basis transformations, the individual component matrices may be treated in the most convenient or most efficient manner.

Ill-Conditioned Eigenproblem Examples: Because of the finite precision used in computations, the computed eigenvalue ρ_j and eigenvector $\mathbf{v}_j$ of the matrix $\mathbf{H}$ almost satisfy [1] the exact equation

$$(\mathbf{H}+\mathbf{E}-\rho_j)\mathbf{v}_j = 0 \quad [\textit{exact arithmetic}]. \tag{4.23}$$

That is, the computed eigenpair is almost the exact eigenpair of a matrix $(\mathbf{H}+\mathbf{E})$ that is close to the matrix $\mathbf{H}$. Using backward-error-analysis [1], the error matrix $\mathbf{E}$ satisfies

$$\|\mathbf{E}\| \le p(N)\varepsilon\|\mathbf{H}\|, \tag{4.24}$$

where $p(N)$ is a modestly growing polynomial of the matrix dimension N. The term ε is the relative precision of the floating point representation and is called the *machine epsilon*. For simplicity, the polynomial will be approximated hereafter as $p(N) \approx 1$. The bound Eq. (A1) may be used to estimate the absolute error of the computed eigenvalue:

$$|\lambda_j - \rho_j| \le \|\mathbf{E}\| \approx \varepsilon\|\mathbf{H}\|. \tag{4.25}$$

The relative error of the eigenvalue is then bounded by

$$e_j \equiv \frac{|\lambda_j - \rho_j|}{|\lambda_j|} \le \frac{\|\mathbf{E}\|}{|\lambda_j|} \approx \varepsilon\frac{\|\mathbf{H}\|}{|\lambda_j|} \le \varepsilon\frac{Max|\lambda_{1:N}|}{Min|\lambda_{1:N}|}. \tag{4.26}$$

The ratio of the largest exact eigenvalue magnitude and the smallest exact eigenvalue magnitude on the right-hand side of Eq. (4.26) is called the *matrix condition number*. Eq. (A2) gives a similar bound on the accuracy of the computed eigenvector

$$|\mathrm{Sin}(\psi_j)| \leq \frac{\|\mathbf{E}\|}{Gap(\lambda_j,\, j, \mathbf{H})} \approx \varepsilon \frac{\|\mathbf{H}\|}{Gap(\lambda_j,\, j, \mathbf{H})}. \tag{4.27}$$

From Eqs. 4.26 and 4.27 it is seen that the accuracy with which an eigenvalue and eigenvector may be computed using finite precision arithmetic depends on the machine epsilon, the condition number of the matrix, on the eigenvalue being computed, and on the gap of the eigenvalue being computed. An ill-conditioned eigenproblem is one in which the accuracy of the desired eigenpair of a given problem is limited because of an unfortunate combination of these factors.

In order to examine the convergence behavior of the SPAM method with ill-conditioned eigenproblems, a model matrix $\mathbf{H}$ is defined in spectral form according to

$$\mathbf{H} = \mathbf{U}\mathbf{D}\mathbf{U}^T \tag{4.28}$$

$$D_{jk} = \Delta^{k-1}\delta_{jk}; \quad \text{for all } j, k \tag{4.29}$$

$$\mathbf{U} = (\mathbf{1}+\mathbf{Y})(\mathbf{1}-\mathbf{Y})^{-1} \tag{4.30}$$

$$Y_{k,k+1} = -Y_{k+1,k} = -Y_{1N} = Y_{N1} = \alpha; \quad \text{for all } k \tag{4.31}$$

$$Y_{jk} = 0; \quad \text{otherwise.}$$

The exact eigenvalues of $\mathbf{H}$ are the elements of the diagonal matrix $\mathbf{D}$, and the corresponding eigenvectors are the columns of the orthogonal cyclic Toeplitz matrix $\mathbf{U}$. Specifically, there is an eigenvalue with the positive value $\lambda_k = \Delta^{k-1}$ and with the corresponding eigenvector $\mathbf{v}_k = \mathbf{U}\mathbf{e}_k$ where $\mathbf{e}_k$ is the unit vector corresponding to the kth coordinate. The scalar parameter Δ, along with the matrix dimension N, determines the condition number of the matrix and the eigenvalue gaps. The scalar parameter α defines the skew-symmetric matrix $\mathbf{Y}$, which may be regarded as a generator for the orthogonal rotation matrix $\mathbf{U}$. The parameter α corresponds roughly to a rotation angle, with smaller angles α corresponding to smaller rotations which, in turn, result in smaller off-diagonal elements of the matrix $\mathbf{H}$. With these scalar parameters, the condition number, the eigenvalue gaps, and the diagonal dominance of the matrix may be controlled.

For large matrix dimension N, it is not practical to compute the matrix $\mathbf{H}$ explicitly. However, matrix–vector products may be computed efficiently in operator form as

$$\mathbf{H}\,\mathbf{x} = (\mathbf{1}+\mathbf{Y})(\mathbf{1}-\mathbf{Y})^{-1}\,\mathbf{D}(\mathbf{1}-\mathbf{Y})(\mathbf{1}+\mathbf{Y})^{-1}\,\mathbf{x}, \tag{4.32}$$

in which the individual factors operate on the trial vector $\mathbf{x}$ in right-to-left order. Because of the special form of the skew-symmetric matrix $\mathbf{Y}$, both the matrix–vector products and the linear equation solutions for the individual factors may be computed with only $O(N)$ arithmetic operations.

The rotation matrix $\mathbf{U}$ may be approximated by truncation of the series expansion:

$$\mathbf{U}_m = \mathbf{1} + 2\mathbf{Y} + 2\mathbf{Y}^2 + 2\mathbf{Y}^3 + \cdots + 2\mathbf{Y}^m \tag{4.33}$$

$$= \mathbf{1} + \mathbf{Y}(2 + \cdots (2 + \mathbf{Y}(2 + \mathbf{Y}(2 + 2\mathbf{Y})))) \tag{4.34}$$

This allows an approximate matrix to be defined as

$$\mathbf{H}^{[1]} = \mathbf{U}_m\mathbf{D}\,\mathbf{U}_m^T. \tag{4.35}$$

TABLE XII
Convergence Summaries for Ill-Conditioned Eigenproblems

			$k=1:5$				$k=(N-4):N$			
Δ	λ_N/λ_1	$[m]$	$\lambda_2-\lambda_1$	n_{max}	$N_{product}$	e_k	$\lambda_N-\lambda_{N-1}$	n_{max}	$N_{product}$	e_k
1.01	2.1E4	$[\infty]$	1.0E-02	19	[19]	5.2E-13	2.1E+02	13	[13]	3.3E-15
		$[\infty, 16]$		23	[10, 39]	6.7E-13		14	[5, 14]	3.3E-15
		$[\infty, 16, 12]$		21	[10, 18, 68]	6.7E-13		16	[5, 10, 30]	3.3E-15
$(1.01)^{-1}$	2.1E4	$[\infty]$	4.8E-05	27	[27]	1.9E-13	9.9E-03	13	[13]	4.4E-16
		$[\infty, 16]$		29	[11, 54]	1.8E-14		14	[5, 14]	2.2E-16
		$[\infty, 16, 12]$		43	[12, 23, 152]	1.9E-14		16	[5, 10, 27]	5.5E-16
1.05	1.5E21	$[\infty]$	5.0E-02	—	—	2.6E+4	7.0E+19	12	[12]	2.7E-14
		$[\infty, 16]$		—	—	2.6E+4		12	[5, 12]	2.6E-14
		$[\infty, 16, 12]$		—	—	2.6E+4		13	[5, 10, 21]	2.7E-14
$(1.05)^{-1}$	1.5E21	$[\infty]$	3.4E-23	—	—	1.5E+5	4.8E-02	12	[12]	1.0E-15
		$[\infty, 16]$		—	—	1.5E+5		12	[5, 12]	6.7E-16
		$[\infty, 16, 12]$		—	—	1.5E+5		13	[5, 10, 12]	6.7E-16

Note. For all matrices $N=1000, \alpha=0.1$, and the final convergence criteria are set to guarantee that $Sin(\psi_k) \leq 10^{-8}$. The matrix–vector product counts are for all five computed eigenpairs. The relative errors e_k are the maximum for the five computed eigenvalues.

Because the truncated $\mathbf{U}_m$ matrix is not orthogonal, both the eigenvalues and the eigenvectors of $\mathbf{H}^{[1]}$ differ from those of $\mathbf{H}$. The accuracy of the approximate matrix $\mathbf{H}^{[1]}$ depends on the expansion length m, longer expansions being more accurate generally than shorter expansions. Matrix–vector products $(\mathbf{U}_m\mathbf{x})$ are computed recursively using the factored representation of Eq. (4.34), the effort for which scales modestly as $O(mN)$.

Table XII shows the convergence summaries for four different sets of calculations. In all cases, $N=1000, \alpha=0.1$, and the final convergence criteria are set to guarantee that $Sin(\psi_k) \leq 10^{-8}$ according to the bound Eq. (A15). Because the exact eigenvalue gaps are known for this model problem, they were used to set the convergence criteria. Up to two levels of approximation are used in these calculations: $\mathbf{H}^{[1]}$ is constructed from a $\mathbf{U}_{16}$ truncation, and $\mathbf{H}^{[2]}$ is constructed from a $\mathbf{U}_{12}$ truncation. Other lower-order expansions were also examined, but these approximations resulted either in impractically slow SPAM convergence, or they were not sufficiently accurate to improve convergence over the reference Davidson method. The exact matrix $\mathbf{H}$ is denoted as $m=\infty$ in Table XII. In all cases, the expansion vectors are constructed using diagonal preconditioned residuals. The initial vectors in all cases are the appropriate columns of the unit matrix. The four sets of calculations differ by the choice of Δ.

The first set of calculations corresponds to $\Delta=1.01$. The condition number for this matrix is $2.1 \cdot 10^4$, which corresponds to a fairly well-conditioned matrix. Convergence summaries for the lowest five eigenpairs are given in the first columns, and the convergence summaries for convergence of the highest five eigenpairs are given in the last columns. For the lowest eigenpair calculations, the number of exact matrix–vector products required is reduced from 19, for the straight Davidson method, to 10 for the SPAM method. For the highest eigenvalues, the product count is reduced from 13 to only 5—only a single exact matrix–vector product is required to achieve convergence for each of the higher eigenpairs. The individual eigenvalues are more widely separated at the high end of the spectrum, and this results in the superior convergence rate. The gaps for λ_1 and λ_N are shown in Table XII.

Both the Davidson and the SPAM convergence are improved because of the larger gaps. The relative errors e_k in the eigenvalues are also included in Table XII. The relative error basically indicates the number of correct significant digits in the computed eigenvalue. The maximum relative error for the five computed eigenvalues is given in the table, but in all cases, the errors were comparable for all of the individual eigenvalues in the set. As seen in Table XII, the relative error is somewhat larger for the small end of the spectrum than for the large end of the spectrum. The lowest computed eigenvalues are two or three significant digits less accurate than the highest computed eigenvalues, which in turn are correct to almost machine precision. This is a result of the condition number of the matrix as shown in Eq. (4.26). Loosely speaking, the relative error when a small eigenvalue is contaminated by a large eigenvalue is larger than the relative error when a large eigenvalue is contaminated by a small eigenvalue. The maximum subspace dimension is also given in Table XII, and it is seen that it changes very little for this matrix for the two levels of SPAM. The most significant improvement for the SPAM method is the reduction of the number of exact matrix–vector products that are required to achieve convergence.

The second set of calculations corresponds to $\Delta = (1.01)^{-1}$. It may be verified that this matrix is the inverse of the first matrix, so the condition number is the same. However, the eigenvectors corresponding to the small eigenvalues of the first matrix correspond to those of the large eigenvalues of the second matrix. The eigenvalues of the first matrix are the inverse of the eigenvalues of the second matrix. Consequently, the eigenvalue gaps of the second matrix are smaller than those of the first matrix. Because of this difference in the gaps, the convergence rates are slower for the second matrix than for the corresponding eigenpairs of the first matrix for the lower eigenpairs. This slower convergence is observed both for the Davidson iterations and for the SPAM iterations. Just as for the first matrix, the higher eigenpairs are converged with a single exact matrix–vector product each using the SPAM method. It is also seen that the relative errors are about the same for this second matrix as for the first matrix, and in particular, the lowest computed eigenvalues are less accurate than the highest computed eigenvalues by only two or three significant digits. The maximum subspace dimension is fairly constant for the convergence of the higher eigenpairs, but it becomes significantly larger for the two-level SPAM calculation for the lower eigenpairs due to the smaller eigenvalue gaps.

The third set of calculations corresponds to $\Delta = 1.05$. The condition number for this matrix is $1.5 \cdot 10^{21}$, a large value that corresponds to a very ill-conditioned matrix. Convergence could not be achieved for the lowest eigenpairs of this matrix. The relative errors are given for the partially converged eigenvalues, and it is clear that no progress toward convergence can be attained under any circumstances. Not only are there no significant digits that are correct in the eigenvalues, but, consistent with Eq. (4.26), the partially converged eigenvalues are incorrect by several orders of magnitude. Even if the procedure is started with eigenvectors that are exact to machine precision, the numerical errors involved in computing the matrix–vector product result in large residual norms and in incorrect computed eigenvalues. This is because of the extremely large condition number for this matrix. This demonstrates that it is not just the convergence of the iterative procedure that is problematic, it is the fundamental matrix–vector product operation itself that cannot be performed accurately. However, even with the poor condition number for this matrix, rapid convergence could be achieved for the highest eigenpairs, and furthermore, consistent with Eq. (4.26), the computed eigenvalues display small relative errors.

The fourth set of calculations corresponds to $\Delta = (1.05)^{-1}$. This results in the same poor condition number as for the third matrix, and just as for the third matrix, convergence

could not be achieved for the lowest eigenpairs. Rapid convergence could be achieved for the highest eigenpairs, and the computed eigenvalues show very small relative errors. It is interesting to note that the computed eigenvectors for the highest eigenvalues are exactly those that would have been computed (with exact arithmetic) for the lowest eigenpairs of the third matrix. Similarly, the computed eigenvectors corresponding to the highest eigenvalues of the third matrix correspond exactly to those that would have been computed (with exact arithmetic) for the lowest eigenpairs of the fourth matrix.

The need to compute eigenpairs of ill-conditioned eigenvalue equations or clustering of eigenvalues arises in a wide variety of applications. Among these are problems in computational chemistry (e.g., the cumulative reaction probability formulation of Miller [19] in chemical kinetics), two-dimensional disordered atomic systems [20, 21], and the solution of generalized eigenvalue problems arising in structural mechanics and other areas (e.g., ocean wave modeling) [22, 23]. The above examples show that the SPAM method may be applied to these equations in certain situations, and that significant improvements in efficiency can be achieved compared to the usual Davidson method. First, the problem should be expressed in such a way that eigenpairs at the high end of the spectrum are computed. This may involve the use of shift-and-invert transformations of the original problem in order to achieve this formulation. Secondly, appropriate, and sufficiently accurate, approximate matrices must be devised for this transformed problem in order to apply the SPAM procedure.

5. SUMMARY AND CONCLUSIONS

A new diagonalization method, SPAM, has been developed and applied to several matrix eigenproblems. This method is a modification of the Davidson subspace method. It uses an approximate matrix, or a sequence of approximate matrices, along with projection operators, in order to generate the basis vectors for the subspace expansion. The goal of the method is to reduce the number of exact matrix–vector products that are required, and, in this way, to reduce the overall effort required to achieve convergence. The method is applicable to the lowest eigenpair of the spectrum, the lowest few eigenpairs, the highest eigenpair, the highest few eigenpairs, or selected interior eigenpairs determined either with vector-following or root-homing approaches. A dynamical convergence criterion is developed that allows for efficient early termination of the intermediate iterations for single-level and multilevel SPAM. Contraction of the intermediate-converged eigenvectors in order to construct the expansion subspace for multiroot calculations is achieved with singular value decomposition.

The method is applied to banded matrices, perturbed-tensor-product matrices, MRSDCI Hamiltonian matrices, a set of ill-conditioned matrices, and the eigensystem that results from rational-function direct-SCF wave function optimization. In these applications, approximate matrices are generated by deletion of small matrix elements, deletion of off-diagonal blocks of matrix elements, tensor-product approximations, operator approximation, and by truncation of series expansion. With sufficiently accurate approximations, the SPAM method improves the convergence efficiency in all of these applications, in some cases only modestly, and in some cases dramatically. Several examples that involve “one vector at a time” convergence of multiple eigenpairs show extraordinary improvements over the reference Davidson procedure. The expansion vectors are generated using the usual preconditioned residual vector and the IIGD/GJD procedure, with the latter displaying superior convergence with suitably accurate preconditioners, and both procedures are observed to display convergence superior to the Krylov/Lanczos approach.

Many eigenvalue problems lend themselves naturally to formal approximation. The solution of the approximate problems leads to conceptual insight in addition to approximate numerical solutions to the original problem. In some cases, there exists a sequence of successively simpler approximations, each requiring less effort than its predecessor. The multilevel SPAM method provides a framework within which each of these approximations can be used to improve the efficiency of the original eigenproblem.

A standard Fortran 90/95 subroutine has been written that implements the multiroot multilevel SPAM method described in this work. This subroutine, along with documentation and test examples, is available from the *anonymous ftp* server `ftp.tcg.anl.gov`.

There are several directions for future extensions of this method. The first is to the generalized symmetric eigenvalue problem $(\mathbf{H} - \lambda_j \mathbf{S})\mathbf{v}_j = \mathbf{0}$, in which the metric matrix $\mathbf{S}$ is symmetric and positive definite. The iterative subspace solution of this equation has been analyzed in detail by Sleijpen *et al.* [27, 28], and we believe that this analysis applies in a straightforward way to the SPAM method. A second possible extension is to the general nonsymmetric eigenvalue problem. This extension is somewhat more problematic [3]. We are also examining the use of the SPAM in the solution of other linear and nonlinear matrix equations.

APPENDIX A: BOUNDS AND ESTIMATES

In this appendix, an analysis of the SPAM procedure is presented. This includes various bounds and error estimates of the eigenvalues and eigenvectors. Suppose a selected eigenvector $\mathbf{v}_j^{(0)}$ and eigenvalue $\lambda_j^{(0)}$ of a symmetric matrix $\mathbf{H} \equiv \mathbf{H}^{(0)}$ are desired, and an approximate matrix $\mathbf{H}^{(1)}$ is chosen, constructed, or made available with the corresponding eigenpair $\mathbf{v}_j^{(1)}$ and $\lambda_j^{(1)}$. In general, the eigenvalues and corresponding eigenvectors of the two matrices should be "close" in some sense, and in particular this should be true for the eigenpair of interest. The rigorous bounds [1]

$$\left|\lambda_j^{(0)} - \lambda_j^{(1)}\right| \le \left\|\mathbf{H}^{(1)} - \mathbf{H}^{(0)}\right\| \tag{A1}$$

$$\left|\mathrm{Sin}\left(\angle\left(\mathbf{v}_j^{(1)}, \mathbf{v}_j^{(0)}\right)\right)\right| \le \frac{\left\|\mathbf{H}^{(1)} - \mathbf{H}^{(0)}\right\|}{Gap\left(\lambda_j^{(1)}, j, \mathbf{H}^{(\gamma)}\right)} \tag{A2}$$

(with $\gamma = 0$ or 1) apply to all of the eigenvectors and eigenvalues of the two matrices. The matrix norm used in this discussion is the spectral norm defined as

$$\|\mathbf{A}\| = Max\{|\lambda_j(\mathbf{A})| : j = 1 \ldots N\}, \tag{A3}$$

where $\lambda_j(\mathbf{A})$ is the jth eigenvalue of the matrix $\mathbf{A}$ in which the eigenvalues are ordered from smallest to largest. For the matrix norm $\|\mathbf{H}^{(1)} - \mathbf{H}^{(0)}\|$ in particular, the eigenvalues of the matrix $(\mathbf{H}^{(1)} - \mathbf{H}^{(0)})$ will be, generally, both positive and negative, but they should all be "small" in magnitude in a qualitative sense for these bounds to be useful. The gap function in Eq. (A2) is defined as

$$Gap(\alpha, j, \mathbf{A}) = Min\{|\alpha - \lambda_k(\mathbf{A})| : k = 1 \ldots N; k \ne j\}. \tag{A4}$$

In words, it is the smallest gap between the scalar argument α and the nearest eigenvalues that surround the jth eigenvalue of the matrix $\mathbf{A}$. Equation (A2) suggests that, for an isolated eigenvalue (i.e., a large gap), the corresponding eigenvector may be approximated

well from the approximate matrix, but for closely spaced eigenvalues (with small gaps separating them), it is only the vector subspace spanned by the entire set of nearby vectors that is approximated well. This is discussed in more detail in [1]. Note that the gap of either matrix $\mathbf{H}^{(0)}$ or $\mathbf{H}^{(1)}$ may be used in Eq. (A2) as appropriate. In particularly bad situations of clustered eigenvalues, the corresponding eigenvectors may be very sensitive to the small differences in $(\mathbf{H}^{(1)} - \mathbf{H}^{(0)})$, whereas the eigenvalues themselves are relatively stable to these small differences. Another useful property of a matrix is the *Spread*($\mathbf{A}$), defined as

$$\mathit{Spread}(\mathrm{A}) \equiv \lambda_N(\mathbf{A}) - \lambda_1(\mathbf{A}), \tag{A5}$$

which is the numerical range of the eigenvalues of the matrix $\mathbf{A}$.

The angle $\psi = \angle(\mathbf{v}, \mathbf{w})$ between two arbitrary vectors $\mathbf{v}$ and $\mathbf{w}$ is defined in the usual way

$$\mathrm{Cos}(\psi) = \frac{\left(\mathbf{v}^T\mathbf{w}\right)}{|\mathbf{v}| \cdot |\mathbf{w}|}. \tag{A6}$$

It is also useful to decompose an arbitrary unit vector into orthonormal components, such as

$$\mathbf{w} = \mathrm{Cos}(\psi)\mathbf{v} + \mathit{Sin}(\psi)\mathbf{v}_\perp. \tag{A7}$$

This decomposition is consistent with the definition of ψ in Eq. (A6).

In the SPAM method, the selected eigenvector and eigenvalue are iterated to convergence, so the bounds in Eqs. (A1) and (A2) are not especially useful in determining the accuracy of this eigenpair. This is because the above general bounds must hold also for the eigenpairs that are not being improved during the iterative process. In order to refine the bounds of the selected eigenpair, the iterative procedure itself must be examined.

During the iterative process, there is some set of expansion vectors $\{\mathbf{x}_j : j = 1 \ldots n\}$, assumed herein to be orthonormal, that are collected into the matrix $\mathbf{X}^{[n]}$ and that define the projection operators $\mathbf{P}^{[n]} = \mathbf{X}^{[n]}\mathbf{X}^{[n]T}$ and $\mathbf{Q}^{[n]} = \mathbf{1} - \mathbf{P}^{[n]}$. These projectors, in turn, define the SPAM:

$$\bar{\mathbf{H}}^{[n]} \equiv \left(\mathbf{P}^{[n]}\mathbf{H}^{(0)}\mathbf{P}^{[n]} + \mathbf{P}^{[n]}\mathbf{H}^{(0)}\mathbf{Q}^{[n]} + \mathbf{Q}^{[n]}\mathbf{H}^{(0)}\mathbf{P}^{[n]}\right) + \mathbf{Q}^{[n]}\mathbf{H}^{(1)}\mathbf{Q}^{[n]} \tag{A8}$$

$$= \mathbf{H}^{(0)} + \mathbf{Q}^{[n]}\left(\mathbf{H}^{(1)} - \mathbf{H}^{(0)}\right)\mathbf{Q}^{[n]}. \tag{A9}$$

Note that the first form is used in the computation because the first three terms in parentheses may be constructed entirely from the stored vectors $\mathbf{X}^{[n]}$ and matrix–vector products $\mathbf{W}^{[n]} = \mathbf{H}^{(0)}\mathbf{X}^{[n]}$ and do not require an explicit computation of a matrix–vector product with the matrix $\mathbf{H}^{(0)}$. The second form is convenient for some of the formal analysis in this section. The eigenpair is determined from this approximate SPAM:

$$\left(\bar{\mathbf{H}}^{[n]} - \lambda_j^{[n]}\right)\mathbf{v}_j^{[n]} = \mathbf{0}. \tag{A10}$$

The normalized eigenvector $\mathbf{v}_j^{[n]}$ may be decomposed according to

$$\mathbf{v}_j^{[n]} = \mathbf{X}^{[n]}\mathbf{c}^{[n]} + \mathrm{Sin}\left(\psi^{[n]}\right)\mathbf{x}^{[n+1]}, \tag{A11}$$

in which the unit vector $\mathbf{x}^{[n+1]}$ is orthogonal to $\mathbf{X}^{[n]}$. The normalization is $\mathbf{c}^{[n]T}\mathbf{c}^{[n]} + \mathrm{Sin}^2(\psi^{[n]}) = 1$. This is a generalization of the decomposition of Eq. (A7). As the iterative SPAM procedure converges $|\mathbf{c}^{[n]}| \to 1$ and $\mathrm{Sin}(\psi^{[n]}) \to 0$.

Once an eigensolution of $\bar{\mathbf{H}}^{[n]}$ has been computed, the accuracy of the original $\mathbf{H}^{(0)}$ eigensolution may be assessed by computing the residual vector

$$\mathbf{r}_j = \left(\mathbf{H}^{(0)} - \rho_j\right)\mathbf{v}_j^{[n]}, \tag{A12}$$

with $\rho_j = \mathbf{v}_j^{[n]T}\mathbf{H}^{(0)}\mathbf{v}_j^{[n]}$ being the scalar that minimizes the residual norm. A conservative bound on an exact eigenvalue is given by [1]

$$\left|\rho_j - \lambda_j^{(0)}\right| \le |\mathbf{r}_j|. \tag{A13}$$

Another (ultimately tighter) bound is given by

$$\left|\rho_j - \lambda_j^{(0)}\right| \le \frac{|\mathbf{r}_j|^2}{Gap\left(\rho_j,\, j, \mathbf{H}^{(0)}\right)}, \tag{A14}$$

but this requires knowledge of the exact gap of $\mathbf{H}^{(0)}$, which is generally unknown. A useful lower bound on the gap may be computed sometimes from Eq. (A13), and that lower bound can be used in the RHS of Eq. (A14).

In principle, there is no lower bound on the residual norm magnitude $|\mathbf{r}_j|$. Consider, for example, the special case in which $\mathbf{H}^{(1)}$ and $\mathbf{H}^{(0)}$ share the same eigenvectors, but have different eigenvalues. As long as the vectors are ordered correctly, then $|\mathbf{r}_j| = 0$ and convergence would be achieved in a single iteration, regardless of the magnitude of $\|\mathbf{H}^{(1)} - \mathbf{H}^{(0)}\|$.

The accuracy of the vector $\mathbf{v}_j^{[n]}$ is determined by $\psi = \angle(\mathbf{v}_j^{[n]}, \mathbf{v}_j^{(0)})$, and this angle is bounded by [1]

$$\frac{|\mathbf{r}_j|}{Spread\left(\mathbf{H}^{(0)}\right)} \le |\mathrm{Sin}(\psi)| \le \frac{|\mathbf{r}_j|}{Gap\left(\rho_j,\, j, \mathbf{H}^{(0)}\right)}. \tag{A15}$$

The exact *Spread* and *Gap* of $\mathbf{H}^{(0)}$ are unknown, but useful upper and lower bounds, respectively, may sometimes be computed and used to bound the exact $Sin(\psi)$. In practical applications, any or all of the above bounds, on the eigenvalues, Eqs. (A13) and (A14), or the eigenvector error, Eq. (A15), may be used to terminate the iterative diagonalization procedure.

Substitution of Eq. (A9) into Eq. (A12) results in

$$\begin{aligned}\mathbf{r}_j &= \left(\bar{\mathbf{H}}^{[n]} - \mathbf{Q}^{[n]}\left(\mathbf{H}^{(1)} - \mathbf{H}^{(0)}\right)\mathbf{Q}^{[n]} - \rho_j\right)\mathbf{v}_j^{[n]} \\ &= \left(\lambda_j^{[n]} - \rho_j - \mathbf{Q}^{[n]}\left(\mathbf{H}^{(1)} - \mathbf{H}^{(0)}\right)\mathbf{Q}^{[n]}\right)\mathbf{v}_j^{[n]}.\end{aligned} \tag{A16}$$

Multiplying from the left by $\mathbf{v}_j^{[n]T}$ gives

$$\left(\lambda_j^{[n]} - \rho_j\right) = \mathbf{v}_j^{[n]T}\mathbf{Q}^{[n]}\left(\mathbf{H}^{(1)} - \mathbf{H}^{(0)}\right)\mathbf{Q}^{[n]}\mathbf{v}_j^{[n]} \tag{A17}$$

$$= \mathrm{Sin}^2\left(\psi^{[n]}\right)\mathbf{x}_j^{[n+1]T}\left(\mathbf{H}^{(1)} - \mathbf{H}^{(0)}\right)\mathbf{x}_j^{[n+1]} \tag{A18}$$

$$\left|\lambda_j^{[n]} - \rho_j\right| \le \mathrm{Sin}^2\left(\psi^{[n]}\right) \cdot \left\|\mathbf{H}^{(1)} - \mathbf{H}^{(0)}\right\|. \tag{A19}$$

The bound in Eq. (A19) follows from Eq. (A18) and the definition of the matrix norm Eq. (A3) This improves on the general eigenvalue bounds given directly by Eq. (A1). Substituting Eq. (A17) into Eq. (A16) gives

$$\begin{aligned}\mathbf{r}_j &= \mathbf{v}_j^{[n]}\mathbf{v}_j^{[n]T}\mathbf{Q}^{[n]}\big(\mathbf{H}^{(1)}-\mathbf{H}^{(0)}\big)\mathbf{Q}^{[n]}\mathbf{v}_j^{[n]} - \mathbf{Q}^{[n]}\big(\mathbf{H}^{(1)}-\mathbf{H}^{(0)}\big)\mathbf{Q}^{[n]}\mathbf{v}_j^{[n]} \\ &= -\big(1-\mathbf{v}_j^{[n]}\mathbf{v}_j^{[n]T}\big)\mathbf{Q}^{[n]}\big(\mathbf{H}^{(1)}-\mathbf{H}^{(0)}\big)\mathbf{Q}^{[n]}\mathbf{v}_j^{[n]} \\ &= -\mathrm{Sin}\big(\psi^{[n]}\big)\big(1-\mathbf{v}_j^{[n]}\mathbf{v}_j^{[n]T}\big)\mathbf{Q}^{[n]}\big(\mathbf{H}^{(1)}-\mathbf{H}^{(0)}\big)\mathbf{x}_j^{[n+1]} \qquad \text{(A20)}\end{aligned}$$

$$|\mathbf{r}_j| = \big|\mathrm{Sin}\big(\psi^{[n]}\big)\big|\cdot\big|\big(1-\mathbf{v}_j^{[n]}\mathbf{v}_j^{[n]T}\big)\mathbf{Q}^{[n]}\big(\mathbf{H}^{(1)}-\mathbf{H}^{(0)}\big)\mathbf{x}^{[n+1]}\big|. \qquad \text{(A21)}$$

This results in the bounds

$$|\mathbf{r}_j| \le \big|\mathrm{Sin}\big(\psi^{[n]}\big)\big|\cdot\big\|\mathbf{H}^{(1)}-\mathbf{H}^{(0)}\big\| \qquad \text{(A22)}$$

$$\begin{aligned}\big|\mathrm{Sin}\big(\psi\big)\big| &\le \big|\mathrm{Sin}\big(\psi^{[n]}\big)\big|\cdot\frac{\big|\big(1-\mathbf{v}_j^{[n]}\mathbf{v}_j^{[n]T}\big)\mathbf{Q}^{[n]}\big(\mathbf{H}^{(1)}-\mathbf{H}^{(0)}\big)\mathbf{x}^{[n+1]}\big|}{Gap\big(\rho_j,\,j,\mathbf{H}^{(0)}\big)} \\ &\le \big|\mathrm{Sin}\big(\psi^{[n]}\big)\big|\cdot\frac{\big\|\mathbf{H}^{(1)}-\mathbf{H}^{(0)}\big\|}{Gap\big(\rho_j,\,j,\mathbf{H}^{(0)}\big)} \qquad \text{(A23)}\end{aligned}$$

$$\begin{aligned}\big|\rho_j-\lambda_j^{(0)}\big| &\le |\mathbf{r}_j| = \big|\mathrm{Sin}\big(\psi^{[n]}\big)\big|\cdot\big|\big(1-\mathbf{v}_j^{[n]}\mathbf{v}_j^{[n]T}\big)\mathbf{Q}^{[n]}\big(\mathbf{H}^{(1)}-\mathbf{H}^{(0)}\big)\mathbf{x}^{[n+1]}\big| \\ &\le \big|\mathrm{Sin}\big(\psi^{[n]}\big)\big|\cdot\big\|\mathbf{H}^{(1)}-\mathbf{H}^{(0)}\big\| \qquad \text{(A24)}\end{aligned}$$

$$\big|\rho_j-\lambda_j^{(0)}\big| \le \mathrm{Sin}^2\big(\psi^{[n]}\big)\cdot\frac{\big|\big(1-\mathbf{v}_j^{[n]}\mathbf{v}_j^{[n]T}\big)\mathbf{Q}^{[n]}\big(\mathbf{H}^{(1)}-\mathbf{H}^{(0)}\big)\mathbf{x}^{[n+1]}\big|^2}{Gap\big(\rho_j,\,j,\mathbf{H}^{(0)}\big)} \qquad \text{(A25)}$$

$$\le \mathrm{Sin}^2\big(\psi^{[n]}\big)\cdot\frac{\big\|\mathbf{H}^{(1)}-\mathbf{H}^{(0)}\big\|^2}{Gap\big(\rho_j,\,j,\mathbf{H}^{(0)}\big)}. \qquad \text{(A26)}$$

On the first SPAM iteration, when the first vector is being computed to form the subspace $\mathbf{X}^{[1]}$, $Sin(\psi)=1$ and Eq. (A22) shows that the residual norm $|\mathbf{r}_j|$ is bounded from above by the matrix difference norm $\|\mathbf{H}^{(1)}-\mathbf{H}^{(0)}\|$. Similarly, the bound on the error angle $Sin(\psi)$ reduces to that given in Eq. (A2), and the eigenvalue error reduces to that given in Eqs. (A1) and (A14). It is only as vectors are added to the subspace $\mathbf{X}^{[n]}$ that the bounds improve. All of the bounds in Eqs. (A22)–(A26) improve upon the general bounds because the $Sin(\psi^{[n]})$ coefficient (which is a computable quantity) decreases toward zero as the procedure converges. Equation (A23) shows also that $Sin(\psi)$, the exact error in the eigenvector, and $Sin(\psi^{[n]})$ are of the same order, and both decrease together as convergence is achieved.

The accuracy from one SPAM iteration to the next is now examined. The eigenvector $\mathbf{v}_j^{[n]}$ of $\bar{\mathbf{H}}^{[n]}$ is decomposed according to Eq. (A11), and the vector $\mathbf{x}^{[n+1]}$ is appended to the $\mathbf{X}^{[n]}$ basis vectors to give the new projectors:

$$\mathbf{P}^{[n+1]} = \mathbf{X}^{[n+1]}\big(\mathbf{X}^{[n+1]}\big)^T = \mathbf{P}^{[n]} + \mathbf{x}^{[n+1]}\big(\mathbf{x}^{[n+1]}\big)^T \qquad \text{(A27)}$$

$$\mathbf{Q}^{[n+1]} = \mathbf{Q}^{[n]} - \mathbf{x}^{[n+1]}\big(\mathbf{x}^{[n+1]}\big)^T. \qquad \text{(A28)}$$

The next SPAM is then given by

$$\begin{aligned}\bar{\mathbf{H}}^{[n+1]} &= \mathbf{H}^{(0)} + \mathbf{Q}^{[n+1]}\big(\mathbf{H}^{(1)}-\mathbf{H}^{(0)}\big)\mathbf{Q}^{[n+1]} \qquad \text{(A29)} \\ &= \bar{\mathbf{H}}^{[n]} + \big(\bar{\mathbf{H}}^{[n+1]} - \bar{\mathbf{H}}^{[n]}\big) \qquad \text{(A30)} \\ &= \bar{\mathbf{H}}^{[n]} + \big(\mathbf{Q}^{[n+1]}\big(\mathbf{H}^{(1)}-\mathbf{H}^{(0)}\big)\mathbf{Q}^{[n+1]} - \mathbf{Q}^{[n]}\big(\mathbf{H}^{(1)}-\mathbf{H}^{(0)}\big)\mathbf{Q}^{[n]}\big) \qquad \text{(A31)} \\ &= \bar{\mathbf{H}}^{[n]} + \boldsymbol{\Delta}, \qquad \text{(A32)}\end{aligned}$$

with the obvious definition of the matrix $\boldsymbol{\Delta}$. The scalar expansion parameter β may be introduced in the $[n+1]$ eigenvalue equation as

$$\bar{\mathbf{H}}^{[n+1]} = \bar{\mathbf{H}}^{[n]} + \beta\boldsymbol{\Delta}, \tag{A33}$$

and the eigenvector (with intermediate normalization) and eigenvalue may be expanded in powers of this parameter as

$$\mathbf{0} = \left(\bar{\mathbf{H}}^{[n+1]} - \lambda_j^{[n+1]}\right)\mathbf{v}_j^{[n+1]} \tag{A34}$$

$$= \left(\left(\bar{\mathbf{H}}^{[n]} + \beta\boldsymbol{\Delta}\right) - \left(\lambda_j^{\{0\}} + \beta\lambda_j^{\{1\}} + \beta^2\lambda_j^{\{2\}} + \cdots\right)\right)\left(\mathbf{v}_j^{\{0\}} + \beta\mathbf{v}_j^{\{1\}} + \beta^2\mathbf{v}_j^{\{2\}} + \cdots\right). \tag{A35}$$

For notational simplicity, the $[n+1]$ superscript has been dropped in Eq. (A35). In the usual perturbation theory approach, it is the solution of the eigenvalue equation at $\beta = 1$ that is of interest, but only the low-order terms are kept to define various approximations to the desired eigenvalue $\lambda_j^{[n+1]}$ and eigenvector $\mathbf{v}_j^{[n+1]}$. Collecting the zeroth order terms together, the first-order terms together, and the second-order terms together gives

$$\mathbf{0} = \left(\bar{\mathbf{H}}^{[n]} - \lambda_j^{[n+1]\{0\}}\right)\mathbf{v}_j^{[n+1]\{0\}} \tag{A36}$$

$$\mathbf{0} = \left(\bar{\mathbf{H}}^{[n]} - \lambda_j^{[n+1]\{0\}}\right)\mathbf{v}_j^{[n+1]\{1\}} + \left(\boldsymbol{\Delta} - \lambda_j^{[n+1]\{1\}}\right)\mathbf{v}_j^{[n+1]\{0\}} \tag{A37}$$

$$\mathbf{0} = \left(\bar{\mathbf{H}}^{[n]} - \lambda_j^{[n+1]\{0\}}\right)\mathbf{v}_j^{[n+1]\{2\}} + \left(\boldsymbol{\Delta} - \lambda_j^{[n+1]\{1\}}\right)\mathbf{v}_j^{[n+1]\{1\}} + \lambda_j^{[n+1]\{2\}}\mathbf{v}_j^{[n+1]\{0\}}. \tag{A38}$$

Equation (A36) means that $\lambda_j^{[n+1]\{0\}} = \lambda_j^{[n]}$ and $\mathbf{v}_j^{[n+1]\{0\}} = \mathbf{v}_j^{[n]}$, the eigenpair from the previous SPAM $\bar{\mathbf{H}}^{[n]}$. Making these substitutions, multiplying Eq. (A37) from the left by $\mathbf{v}_j^{[n]T}$, and noting that $\mathbf{Q}^{[n+1]}\mathbf{v}_j^{[n]} = \mathbf{0}$, gives the first-order contribution and corresponding bound to the eigenvalue

$$\lambda_j^{[n+1]\{1\}} = \mathbf{v}_j^{[n]T}\boldsymbol{\Delta}\mathbf{v}_j^{[n]} = \mathbf{v}_j^{[n]T}\mathbf{Q}^{[n]}\left(\mathbf{H}^{(1)} - \mathbf{H}^{(0)}\right)\mathbf{Q}^{[n]}\mathbf{v}_j^{[n]} = \left(\rho_j - \lambda_j^{[n]}\right) \tag{A39}$$

$$\left|\lambda_j^{[n+1]\{1\}}\right| \le Sin^2\left(\psi^{[n]}\right)\cdot\left\|\mathbf{H}^{(1)} - \mathbf{H}^{(0)}\right\|. \tag{A40}$$

The first-order contribution to the eigenvector is

$$\mathbf{v}_j^{[n+1]\{1\}} = -\left(\bar{\mathbf{H}}^{[n]} - \lambda_j^{[n]}\right)^{-1^*}\left(\boldsymbol{\Delta} - \lambda_j^{[n+1]\{1\}}\right)\mathbf{v}_j^{[n]}, \tag{A41}$$

in which the pseudoinverse (denoted as -1^*) operates only within the subspace orthogonal to $\mathbf{v}_j^{[n]}$. Substitution of the matrix $\boldsymbol{\Delta}$ from Eq. (A31) results in

$$\mathbf{v}_j^{[n+1]\{1\}} = Sin\left(\psi^{[n]}\right)\left(\bar{\mathbf{H}}^{[n]} - \lambda_j^{[n]}\right)^{-1^*}\mathbf{Q}^{[n]}\left(\mathbf{H}^{(1)} - \mathbf{H}^{(0)}\right)\mathbf{x}^{[n+1]} \tag{A42}$$

$$\left|\mathbf{v}_j^{[n+1]\{1\}}\right| \le \left|Sin\left(\psi^{[n]}\right)\right|\cdot\frac{\left\|\left(\mathbf{H}^{(1)} - \mathbf{H}^{(0)}\right)\right\|}{Gap\left(\lambda_j^{[n]},\, j,\, \bar{\mathbf{H}}^{[n]}\right)}. \tag{A43}$$

Multiplying Eq. (A38) from the left by $\mathbf{v}_j^{[n]T}$, and noting that $\mathbf{Q}^{[n+1]}\mathbf{v}_j^{[n]} = \mathbf{0}$, gives the

second-order contribution and corresponding bound to the eigenvalue

$$\begin{aligned}\lambda_j^{[n+1]\{2\}} &= \mathbf{v}_j^{[n]T}\boldsymbol{\Delta}\mathbf{v}_j^{[n+1]\{1\}} \\ &= -\mathbf{v}_j^{[n]T}\mathbf{Q}^{[n]}\big(\mathbf{H}^{(1)}-\mathbf{H}^{(0)}\big)\mathbf{Q}^{[n]}\big(\bar{\mathbf{H}}^{[n]}-\lambda_j^{[n]}\big)^{-1^*} \\ &\quad\times\mathbf{Q}^{[n]}\big(\mathbf{H}^{(1)}-\mathbf{H}^{(0)}\big)\mathbf{Q}^{[n]}\mathbf{v}_j^{[n]} \\ &= -Sin^2\big(\psi^{[n]}\big)\mathbf{x}^{[n+1]T}\big(\mathbf{H}^{(1)}-\mathbf{H}^{(0)}\big)\mathbf{Q}^{[n]} \\ &\quad\times\big(\bar{\mathbf{H}}^{[n]}-\lambda_j^{[n]}\big)^{-1^*}\mathbf{Q}^{[n]}\big(\mathbf{H}^{(1)}-\mathbf{H}^{(0)}\big)\mathbf{x}^{[n+1]T} \qquad \text{(A44)}\end{aligned}$$

$$\big|\lambda_j^{[n+1]\{2\}}\big| \le Sin^2\big(\psi^{[n]}\big)\cdot\frac{\big\|\big(\mathbf{H}^{(1)}-\mathbf{H}^{(0)}\big)\big\|^2}{Gap\big(\lambda_j^{[n]},\, j,\, \bar{\mathbf{H}}^{[n]}\big)}. \qquad \text{(A45)}$$

Alternatively, Eq. (A9) may be used to define a perturbation theory for the eigenpair of the exact matrix. The scalar expansion parameter β may be introduced as

$$\mathbf{H}^{(0)} = \bar{\mathbf{H}}^{[n]} - \beta\mathbf{Q}^{[n]}\big(\mathbf{H}^{(1)}-\mathbf{H}^{(0)}\big)\mathbf{Q}^{[n]}. \qquad \text{(A46)}$$

Expanding the eigenvector and eigenvalue of $\mathbf{H}^{(0)}$ in powers of β and collecting the zeroth order terms

$$\mathbf{0} = \big(\bar{\mathbf{H}}^{[n]} - \lambda_j^{(0)\{0\}}\big)\mathbf{v}_j^{\{0\}}. \qquad \text{(A47)}$$

This means that $\lambda_j^{(0)\{0\}} = \lambda_j^{[n+1]\{0\}} = \lambda_j^{[n]}$ and $\mathbf{v}_j^{(0)\{0\}} = \mathbf{v}_j^{[n+1]\{0\}} = \mathbf{v}_j^{[n]}$. Collecting the first-order terms in β gives

$$\lambda_j^{(0)\{1\}} = \lambda_j^{[n+1]\{1\}} = \big(\rho_j - \lambda_j^{[n]}\big) \qquad \text{(A48)}$$

$$\mathbf{v}_j^{(0)\{1\}} = \mathbf{v}_j^{[n+1]\{1\}} = Sin\big(\psi^{[n]}\big)\big(\bar{\mathbf{H}}^{[n]}-\lambda_j^{[n]}\big)^{-1^*}\mathbf{Q}^{[n]}\big(\mathbf{H}^{(1)}-\mathbf{H}^{(0)}\big)\mathbf{x}^{[n+1]}. \qquad \text{(A49)}$$

Collecting the second-order terms in β gives

$$\lambda_j^{(0)\{2\}} = \lambda_j^{[n+1]\{2\}}. \qquad \text{(A50)}$$

It may be verified that the second- and higher-order terms in the eigenvector corrections, and the third- and higher-order terms in the eigenvalue corrections, are different in these two perturbation expansions. However, through first-order for the eigenvector and through second-order for the eigenvalue, Eqs. (A47)–(A50) demonstrate that the low-order corrections to the SPAM $\mathbf{H}^{[n+1]}$ and to the exact matrix $\mathbf{H}^{(0)}$ are identical. That is, these equations show that the same $Sin(\psi^{[n]})$ factor appears in the low-order corrections to the eigenvectors and eigenvalues of $\bar{\mathbf{H}}^{[n]}$ and $\mathbf{H}^{(0)}$. This is the basis of the improved efficiency with the SPAM method. The effort required to solve the $\bar{\mathbf{H}}^{[n]}$ eigensolution involves only matrix–vector products with the $\mathbf{H}^{(1)}$ matrix. Once found, a correction of approximately the same accuracy is incorporated (with the effort of only a single $\mathbf{H}^{(0)}$ matrix–vector product) into the desired eigenpair. The factor $Sin(\psi^{[n]})$ is a computable quantity, and it converges toward zero as the procedure converges toward the selected eigenpair.

APPENDIX B: GENERATION OF NEW EXPANSION VECTORS

There are several ways of generating expansion vectors that have been considered in the past with the Davidson diagonalization method. These are discussed briefly here and compared to the SPAM method. These correction vectors may be derived from perturbation theory, relaxation, minimization of the residual norm, stablization of the Rayleigh quotient, or, in a heuristic manner, by approximation [3, 4]. The latter approach is taken here. This facilitates comparisons, but it provides a rather narrow view of each of the methods; the reader should consult the original references for additional details. For a matrix $\mathbf{H}$, the desired eigenvector and eigenvalue satisfy the equation

$$(\mathbf{H} - \lambda)\mathbf{v} = \mathbf{0}. \tag{B1}$$

The exact eigenvector $\mathbf{v}$ may be written as a sum of a unit trial vector $\mathbf{x}$ and an orthogonal correction vector $\boldsymbol{\delta}$, as $\mathbf{v} = \mathbf{x} + \boldsymbol{\delta}$. Furthermore, the eigenvalue may be written as $\lambda = (\rho + \varepsilon)$ where $\rho = \mathbf{x}^T\mathbf{H}\mathbf{x}$ is the Rayleigh quotient. For this decomposition to be useful $\angle(\mathbf{v}, \mathbf{x})$ should be small, $|\boldsymbol{\delta}|$ should be small, and ε should be small. For practical reasons, it will be useful to introduce an approximate matrix $\mathbf{D}$. This allows the eigenvalue equation to be written in the various forms

$$(\mathbf{H} - \rho - \varepsilon)\boldsymbol{\delta} = -(\mathbf{H} - \rho - \varepsilon)\mathbf{x} \tag{B2}$$

$$= -\mathbf{r} + \varepsilon\mathbf{x} \tag{B3}$$

$$(\mathbf{D} - \rho + (\mathbf{H} - \mathbf{D} - \varepsilon))\boldsymbol{\delta} = -\mathbf{r} + \varepsilon\mathbf{x}. \tag{B4}$$

Note that the matrix $(\mathbf{H} - \lambda)$ is singular, so this expression is a statement about how the exact vector $\mathbf{v}$ is annihilated from the RHS of the Eqs. (B2)–(B4). All of the methods discussed in this appendix will be expressed as approximations to these exact equations.

The original Davidson [2–4] method follows from two separate approximations. The first is that the terms $(\mathbf{H} - \mathbf{D} - \varepsilon)$ are deleted from the LHS, and the ε term is deleted from the RHS of Eq. (B4). This results in the equation

$$(\mathbf{D} - \rho)\boldsymbol{\delta}^D = -\mathbf{r}. \tag{B5}$$

The second approximation in the Davidson method is that $\mathbf{D}$ is usually taken to be a diagonal matrix. Other choices have been used also [3–6], but the diagonal approximation makes the linear equation in Eq. (B5) trivial, and it is the most common choice. The residual vector $\mathbf{r}$ is proportional to the gradient of the Rayleigh quotient with respect to variations in the trial vector $\mathbf{x}$, and consequently $\boldsymbol{\delta}^D$ from Eq. (B5) may be regarded as a preconditioned gradient. This has been discussed by van Lenthe and Pulay [24] and by Davidson *et al.* [25]. The correction vector $\boldsymbol{\delta}^D$ from Eq. (B5) is not orthogonal to $\mathbf{x}$. This traditional Davidson method is denoted the diagonal-preconditioned-residual (DPR) method and is used as the reference for comparisons in this work.

Many methods are based on Rayleigh quotient inverse iteration (RQII). This is usually regarded as a single-vector method in which the trial vector is replaced, during each iteration, with the solution of the linear equation

$$(\mathbf{H} - \rho)\mathbf{x}^{new} = \mathbf{x}. \tag{B6}$$

This method displays asymptotic cubic convergence [1], which means that, when the reference vector $\mathbf{x}$ is sufficiently close, the error in the eigenvector of each iteration is proportional

to the cube of the error of the previous iteration. Of course, this cubic convergence cannot be exploited practically for matrices of large dimension (except for special or simple forms of the matrix **H**) because of the need for the linear equation solution. For a subspace method, it is not the new vector that is of interest, it is the component of the new vector that is orthogonal to the previous vector that is of primary importance. Writing $\mathbf{x}^{new} = (\mathbf{x} + \boldsymbol{\delta}^{RQII})/\varepsilon$ and rearranging the expression [26] gives

$$(\mathbf{H} - \rho)\boldsymbol{\delta}^{RQII} = -\mathbf{r} + \varepsilon\mathbf{x}. \tag{B7}$$

This shows that the subspace expansion vector generated from RQII is the approximation to the exact equation that results from deleting the ε term from the LHS of Eq. (B3). The scalar ε may be determined by operating on the left by $\mathbf{x}^T(\mathbf{H} - \rho\mathbf{1})^{-1}$. This gives

$$\varepsilon = \frac{1}{\mathbf{x}^T(H - \rho\mathbf{1})^{-1}\mathbf{x}}. \tag{B8}$$

This suggests that two linear equation solutions are required during each RQII iteration when the vector $\boldsymbol{\delta}^{RQII}$ is computed, one to determine ε, which then allows the RHS of Eq. (B7) to be evaluated, and the other linear equation solution to determine $\boldsymbol{\delta}^{RQII}$. Sleijpen *et al.* [27, 28] and van Dam *et al.* [29] suggest two alternatives in their Generalized Jacobi–Davidson (GJD) method. Operating on the left of Eq. (B7) by the projector $(1 - \mathbf{x}\mathbf{x}^T)$ gives the equation

$$(1 - \mathbf{x}\mathbf{x}^T)(\mathbf{H} - \rho)\boldsymbol{\delta}^{GJD} = -\mathbf{r}. \tag{B9}$$

This eliminates the parameter ε from appearing explicitly in the solution of the linear equation for $\boldsymbol{\delta}^{GJD}$. During the solution of this linear equation, care should be taken to ensure that the matrix operates only in the subspace that is complementary to **x**. Once $\boldsymbol{\delta}^{GJD}$ has been determined, ε may be computed, if desired, as $\varepsilon = \mathbf{r}^T\boldsymbol{\delta}^{GJD}$. Sleijpen *et al.* also suggest that the inverse iteration equation for the expansion vector may be solved in the augmented form

$$\begin{pmatrix} \mathbf{H} - \rho & -\mathbf{x} \\ -\mathbf{x}^T & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{\delta}^{GJD} \\ \varepsilon \end{pmatrix} = \begin{pmatrix} \mathbf{r} \\ 0 \end{pmatrix}, \tag{B10}$$

in which both unknowns, ε and $\boldsymbol{\delta}^{GJD}$, are determined together. Equations (B9) and (B10) both show that the vector $\boldsymbol{\delta}^{GJD}$ may be solved with a single linear equation. Sleijpen *et al.* [27] proposed that iterative solutions of the linear equations should be terminated early during the initial iterations in order to improve efficiency, and van Dam *et al.* [29] suggested the use of block-diagonal approximations to **H**.

Olsen *et al.* [26] have proposed the inverse-iteration generalized davidson (IIGD) method. The terms $(\mathbf{H} - \mathbf{D} - \varepsilon)$ are deleted from the LHS of Eq. (B4), resulting in the linear equation

$$(\mathbf{D} - \rho)\boldsymbol{\delta}^{IIGD} = -\mathbf{r} + \varepsilon\mathbf{x}. \tag{B11}$$

This is equivalent to replacing **H** by **D** in the preconditioner in the RQII equation (B7). The scalar ε is determined by operating on Eq. (B11) from the left by $\mathbf{x}^T(\mathbf{D} - \rho\mathbf{1})^{-1}$ and enforcing the orthogonality relation $\mathbf{x}^T\boldsymbol{\delta}^{IIGD} = 0$:

$$\varepsilon = \frac{\mathbf{r}^T(\mathbf{D} - \rho\mathbf{1})^{-1}\mathbf{x}}{\mathbf{x}^T(\mathbf{D} - \rho\mathbf{1})^{-1}\mathbf{x}}. \tag{B12}$$

In principle, the correction vector and parameter ε could also be determined using Eqs. (B9) and (B10), but for a diagonal $\mathbf{D}$, there is little practical advantage. Olsen *et al.* [26] pointed out that in the limit $\mathbf{D} \to \mathbf{H}$, the DPR correction $\boldsymbol{\delta}^D$ becomes exactly linearly dependent with the current trial vector $\mathbf{x}$ (which means it makes no progress toward convergence), whereas the IIGD step $\boldsymbol{\delta}^{IIGD}$ becomes equivalent to Rayleigh quotient inverse iteration (compare to Eq. (B7)), which not only converges, but converges cubically.

The other popular subspace generation approximation consists of deleting the ε term from the RHS of Eq. (B3) and approximating the entire $(\mathbf{H}-\lambda)$ matrix as a unit matrix (or a scalar multiple thereof). This results in the Lanczos expansion vector

$$\boldsymbol{\delta}^L = -\mathbf{r}. \tag{B13}$$

This requires the least amount of effort of any of the methods discussed in this appendix to generate the expansion vector, but it suffers from the slowest convergence properties. Because $\mathbf{r}$ is proportional to the gradient of the Rayleigh quotient, the Lanczos method may be considered a gradient search method. The slow convergence is because the sequence of expansion vectors corresponds to an orthogonalized Krylov sequence, which does not selectively converge to the desired eigenpair of interest. Its main advantage is the fact that the subspace matrix $\langle \mathbf{H} \rangle^{[n]}$ generated by this sequence of vectors is tridiagonal, which means that not only is the subspace eigenvalue equation relatively easy to solve, but also only the two most recent vectors must be saved. In contrast, all of the other preconditioned expansion vector methods discussed in this appendix result in a dense subspace matrix and require the storage of both $\mathbf{X}^{[n]}$ and $\mathbf{W}^{[n]}$. It is easily verified that $\mathbf{X}^{[n]T}\mathbf{r} = \mathbf{0}$, which means that $\boldsymbol{\delta}^L$ is orthogonal not only to the reference vector $\mathbf{x}$ but also to the entire expansion space $\mathbf{X}^{[n]}$.

The SPAM method may now be compared to these other expansion vector methods. In general, the SPAM equation (A10) may be rewritten using the splitting of the matrix, the eigenvalue, and the eigenvector given above. In particular, let $\mathbf{H}^{(1)} = \mathbf{D}$, $\mathbf{x} = \mathbf{X}^{[n]}\mathbf{c}^{[n]}$, $\lambda^{[n]} = (\rho - \varepsilon)$, and $\boldsymbol{\delta}^{SPAM} = Sin(\psi^{[n]})\mathbf{x}^{[n+1]}$ from Eq. (A11):

$$\left(\bar{\mathbf{H}}^{[n]} - \lambda^{[n]}\right)\mathbf{v}^{[n]} = \left(\mathbf{H} + \mathbf{Q}^{[n]}(\mathbf{D} - \mathbf{H})\mathbf{Q}^{[n]} - \rho - \varepsilon\right)(\mathbf{x} + \boldsymbol{\delta}^{SPAM}) = \mathbf{0}. \tag{B14}$$

Rearranging into the form of Eq. (B3) and noting that $\mathbf{Q}^{[n]}\mathbf{x} = \mathbf{0}$, this equation may be rewritten as

$$\left(\mathbf{H} - \mathbf{Q}^{[n]}(\mathbf{D} - \mathbf{H})\mathbf{Q}^{[n]} - \rho - \varepsilon\right)\boldsymbol{\delta}^{SPAM} = -\mathbf{r} + \varepsilon\mathbf{x}. \tag{B15}$$

It is clear that in the limit $\mathbf{D} \to \mathbf{H}$, the SPAM expansion vector approaches the exact correction vector, and convergence would be achieved in a single SPAM iteration. This is in contrast to all of the other expansion vector methods discussed in this appendix ($\boldsymbol{\delta}^D$, $\boldsymbol{\delta}^{RQII}$, $\boldsymbol{\delta}^{GJD}$, $\boldsymbol{\delta}^{IIGD}$, $\boldsymbol{\delta}^L$), none of which converge in a single iteration in this limit. This has some formal appeal in favor of SPAM regarding the potential accuracy, but it has little practical value in most situations because $\mathbf{D}$ is usually too coarse of an approximation to $\mathbf{H}$ for this formal difference to be significant. On the other hand, because SPAM requires the iterative solution of this eigenvalue equation, it would generally be expected to require more effort than either IIGD or the DPR methods. It also should be mentioned that in the limit $\mathbf{D} \to \mathbf{H}$, the RQII expansion vector, the GJD expansion vector, and the IIGD expansion vector are all the same—they are all slightly different implementations of RQII.

In the other limit, with a diagonal $\mathbf{D}$ approximating the matrix $\mathbf{H}$, the GJD and the IIGD expansion vector are still equivalent—they are slightly different implementations of the same approximate inverse iteration. Both of these expansion vectors are orthogonal, by design and by construction, to the reference vector $\mathbf{x}$, in contrast to the DPR update vector, which is not orthogonal and must be explicitly orthogonalized before being added to the expansion vector subspace. Multiplying Eq. (B15) from the left by $(\mathbf{D}-\rho\mathbf{1})^{-1}$ allows the SPAM expansion vector to be written as

$$\left(\mathbf{1}+(\mathbf{D}-\rho\mathbf{1})^{-1}\left((\mathbf{D}-\mathbf{H})-\mathbf{Q}^{[n]}(\mathbf{D}-\mathbf{H})\,\mathbf{Q}^{[n]}-\varepsilon\right)\right)\boldsymbol{\delta}^{SPAM}=\boldsymbol{\delta}^{IIGD}. \tag{B16}$$

In the first SPAM iteration $\mathbf{Q}^{[0]}=\mathbf{1}$ and the only difference between $\boldsymbol{\delta}^{SPAM}$ and $\boldsymbol{\delta}^{IIGD}$ is the ε term on the LHS of Eq. (B16). On subsequent iterations, there is also the $(\mathbf{D}-\mathbf{H})$ term that contributes. In general, the SPAM expansion vector is different from the IIGD expansion vector, the DPR expansion vector, and the Lanczos expansion vector. As the iterations proceed, the SPAM describes the eigenpair of interest more and more accurately. By contrast, the preconditioner used in the IIGD method, and in the DPR method, remain fixed in form and varies only because of ρ.

In addition to the differences in the form of Eq. (B16), another significant difference is in the definition of the reference vector $\mathbf{x}$. In all of the other methods, $\mathbf{x}$ is taken to be the current approximate eigenvector within the subspace $\mathbf{X}^{[n]}$. But in the SPAM method, it is defined as $\mathbf{x}=\mathbf{X}^{[n]}\mathbf{c}^{[n]}$, where $\mathbf{c}$ is the set of coefficients of the level-0 vectors from the diagonalization within the $[n_0, n_1]$ subspace. In the other methods described above, the expansion vector coefficients are "frozen" as the new expansion vector is computed, whereas in the SPAM method, these coefficients are "relaxed" to their optimal value as the new expansion vector is computed. The last significant difference is that the update vector is orthogonal to the reference vector $\mathbf{x}$ in the GJD and IIGD methods, whereas the $\boldsymbol{\delta}^{SPAM}$ update vector, like the Lanczos expansion vector $\boldsymbol{\delta}^{L}$, is orthogonal to the entire $\mathbf{X}^{[n]}$ subspace.

ACKNOWLEDGMENTS

This work was supported by the U.S. Department of Energy by the Office of Basic Energy Sciences, Division of Chemical Sciences, and by the Office of Advanced Scientific Computing Research, Mathematical, Information, and Computational Science Division, under contract W-31-109-ENG-38.

REFERENCES

1. B. N. Parlett, *The Symmetric Eigenvalue Problem* (Soc. for Industr. of Appl. Math./Prentice-Hall, Englewood Cliffs, NJ, 1998). In particular, see Chapters 10 and 11.
2. E. R. Davidson, The Iterative Calculation of a Few of the Lowest Eigenvalues and Corresponding Eigenvectors of Large Real Symmetric Matrices, *J. Comput. Phys.* **17**, 87 (1975).
3. E. R. Davidson, Super-Matrix Methods, *Comput. Phys. Comm.* **53**, 49 (1989).
4. E. R. Davidson, Monster Matrices: Their Eigenvalues and Eigenvectors, *Comput. in Phys.* **7**, 519 (1993).
5. M. Crouzeix, B. Philippe, and M. Sadkane, The Davidson Method, *J. Sci. Comput.* **15**, 62 (1994).
6. R. B. Morgan and D. S. Scott, Generalizations of Davidson's Method for Computing Eigenvalues of Sparse Symmetric Matrices, *J. Sci. Stat. Comput.* **7**, 817 (1986).
7. I. Shavitt, C. F. Bender, and A. Pipano, The Iterative Calculation of Several of the Lowest or Highest Eigenvalues and Corresponding Eigenvectors of Very Large Symmetric Matrices, *J. Comput. Phys.* **11**, 90 (1973).
8. J. H. Wilkinson, *The Algebraic Eigenvalue Problem* (Oxford Univ. Press, New York, 1965).

9. B. Liu, in *Numerical Algorithms in Chemistry: Algebraic Methods*, edited by C. Moler and I. Shavitt (Lawrence Berkeley Laboratory, Berkeley, CA, 1978).
10. For example, see F. L. Pilar, in *Elementary Quantum Chemistry* (McGraw-Hill, New York, 1968); or A. Szabo and N. S. Ostlund, in *Modern Quantum Chemistry* (McGraw-Hill, New York, 1989/Dover, Mineola, NY, 1996).
11. R. Shepard, I. Shavitt, R. M. Pitzer, D. C. Comeau, M. Pepper, H. Lischka, P. G. Szalay, R. Ahlrichs, F. B. Brown, and J.-G. Zhao, A Progress Report on the Status of the COLUMBUS MRCI Program System, *Int. J. Quantum Chem.* **S22**, 149 (1988).
12. H. Lischka, R. Shepard, R. M. Pitzer, I. Shavitt, M. Dallos, T. Muller, P. G. Szalay, M. Seth, G. S. Kedziora, S. Yabushita, and Z. Zhang, High-Level Multireference Methods in the Quantum-Chemistry Program System COLUMBUS: Analytic MR-CISD and MR-AQCC Gradients and MR-AQCC-LRT for Excited States, GUGA Spin-Orbit CI, and Parallel CI, *Phys. Chem. Chem. Phys.* **3**, 664 (2001).
13. R. Shepard, J. L. Tilson, A. F. Wagner, and M. Minkoff, presented at the Workshop "Large-Scale Matrix Diagonalization Methods in Chemistry," Argonne National Laboratory, May, 1996 (`ftp://info.mcs.anl.gov/pub/tech_reports/reports/TM219.ps.z`); R. Shepard, J. L. Tilson, A. F. Wagner, and M. Minkoff, presented at the American Conference on Theoretical Chemistry, Park City, Utah, July, 1996.
14. R. Shepard, Elimination of the Diagonalization Bottleneck in Parallel Direct-SCF Calculations, *Theoretica Chimica Acta* **84**, 343 (1993).
15. J. L. Tilson and R. Shepard, in preparation.
16. J. L. Tilson, M. Minkoff, A. F. Wagner, R. Shepard, P. Sutton, R. J. Harrison, R. A. Kendall, and A. T. Wong, High-Performance Computational Chemistry: Hartree-Fock Electronic Structure Calculations on Massively Parallel Processors, *Int. J. High Performance Computing Applications* **13**, 291 (1999).
17. J. L. Tilson and R. Shepard, presented at High performance computational chemistry workshop, Pleasanton, California, August 1995; J. L. Tilson and R. Shepard, presented at Frontiers of Electronic Structure Theory, American Chemical Society National Meeting, Physical Chemistry Symposium, San Francisco, California, April 1997.
18. A. T. Wong and R. J. Harrison, Approaches to Large-Scale Parallel Self-Consistent Field Calculations, *J. Comput. Chem.* **16**, 1291 (1995).
19. W. H. Miller, Semiclassical Limit of Quantum Mechanical Transition State Theory for Nonseparable Systems, *J. Chem. Phys.* **62**, 1899 (1975); U. Manthe and W. H. Miller, The Cumulative Reaction Probability as Eigenvalue Problem, *J. Chem. Phys.* **99**, 3411 (1993); U. Manthe, T. Seideman, and W. H. Miller, Quantum Mechanical Calculations of the Rate Constant for the $H_2 + OH \rightarrow H + H_2O$ Reaction: Full-Dimensional Results and Comparison to Reduced Dimensionality Models, *J. Chem. Phys.* **101**, 4759 (1994).
20. J. Cullum and R. A. Willoughby, Computing Eigenvalues of Very Large Symmetric Matrices—An Implementation of a Lanczos Algorithm with No Reorthogonalization, *J. Comput. Phys.* **44**, 329 (1981).
21. S. Kirkpatrick, IBM Research, Yorktown Heights, N.Y., private communcation, 1978, noted in Reference 20.
22. R. G. Grimes, J. G. Lewis, and H. D. Simon, A Shifted Block Lanczos Algorithm for Solving Sparse Symmetric Generalized Eigenproblems, *SIAM J. Matrix Anal. Appl.* **15**, 228 (1994).
23. I. Duff, R. G. Grimes, and J. G. Lewis. *Users' Guide for the Harwell-Boeing Sparse Matrix Collection (Release I)*, Technical Report TR/PA/92/86 (CERFACS, October 1992).
24. J. H. van Lenthe and P. Pulay, A Space-Saving Modification of Davidson's Eigenvector Algorithm, *J. Comput. Chem.* **11**, 1164 (1990).
25. C. W. Murray, S. C. Racine, and E. R. Davidson, Improved Algorithms for the Lowest Few Eigenvalues and Associated Eigenvectors of Large Matrices, *J. Comput. Physics.* **103**, 382 (1992).
26. J. Olsen, P. Jørgensen, and J. Simons, Passing the One-Billion Limit in Full Configuration-Interaction (FCI) Calculations, *Chem. Phys. Letters* **169**, 463 (1990).
27. G. L. G. Sleijpen, A. G. L. Booten, D. R. Fokkema, and H. A. van der Vorst, Jacobi-Davidson Type Methods for Generalized Eigenproblems and Polynomial Eigenproblems, *BIT* **36**, 595 (1996).
28. G. L. G. Sleijpen and H. A. Van der Vorst, A Jacobi-Davidson Iteration Method for Linear Eigenvalue Problems, *J. Matrix Anal. Appl.* **17**, 401 (1996).
29. H. J. J. van Dam, J. H. van Lenthe, G. L. G. Sleijpen, and H. A. van der Vorst, An Improvement of Davidson's Iteration Method: Applications to MRCI and MRCEPA Calculations, *J. Comput. Chem.* **17**, 267 (1996).