ANALYSIS OF SUBSPACE ITERATION FOR EIGENVALUE PROBLEMS WITH EVOLVING MATRICES [*]

YOUSEF SAAD [†]

Abstract. The subspace iteration algorithm, a block generalization of the classical power iteration, is known for its excellent robustness properties. Specifically, the algorithm is resilient to variations in the original matrix, and for this reason it has played an important role in applications ranging from Density Functional Theory in Electronic Structure calculations to matrix completion problems in machine learning, and subspace tracking in signal processing applications. This note explores its convergence properties in the presence of perturbations. The specific question addressed is the following. If we apply the subspace iteration algorithm to a certain matrix and this matrix is perturbed at each step, under what conditions will the algorithm converge?

Keywords: Subspace iteration, convergence theory, eigenvalue problems, subspace tracking, Density Functional Theory.

1. Introduction.

The subspace iteration algorithm, a rather straightforward generalization of the classical single-vector power iteration, used to be the method of choice for computing eigenspaces of large matrices. Beginning in the 1980s it lost ground to techniques based on Krylov subspaces, which turned out to be more efficient once practical difficulties related to such issues as loss of orthogonality are resolved. However, the method has recently regained some of the ground it lost, as it is increasingly regarded as a robust alternative to Krylov subspace methods, especially in the common situation where the coefficient matrix changes in the course of some outer iteration. For example, in the 'nonlinear eigenvector problem' that is at the heart of Density Functional Theory (DFT), see, e.g., [20], one seeks the lowest eigenmodes of a Hamiltonian that depends (nonlinearly) on its eigenvectors. The problem is solved by an iterative procedure referred to as the Self-Consistent Field (SCF) iteration, in which a set of eigenvectors of a matrix that changes at each iteration, are to be computed. In this situation a Krylov subspace approach is not appealing because it cannot easily take advantage of the calculations performed in previous SCF steps. In contrast, the subspace iteration algorithm is ideally suited for the problem, since it can use the approximate eigen-basis computed in the previous step as a starting block of vectors for the new iteration. In fact, it can be used in a *nonlinear form* in the sense that the matrix can be updated during the subspace iteration algorithm.

Before discussing a few examples of this scenario we provide a little background on the subspace iteration algorithm. A basic subspace iteration algorithm to compute the dominant eigenspace of a Hermitian (or real symmetric) matrix A is given below, where $\texttt{qf}\,(X)$ denotes the Q factor in the QR factorization $X = QR$

ALGORITHM 1.1. *Standard subspace iteration*

```
0. Start: select an initial $n \times p$ matrix $U$.
1. For $k = 1 : max\_iter$ Do
2.     $U := AU$
3.     If (condition) perform a Rayleigh-Ritz projection:
4.         $Q = \texttt{qf}\,(U)$
5.         $C = Q^T AQ$
6.         $C = W\Lambda W^T$   (Diagonalize $C$)
7.         $U := QW$
8.         Check convergence
```

[*] Work supported partly by the Scientific Discovery through Advanced Computing (SciDAC) program funded by U.S. Department of Energy, Office of Science, Advanced Scientific Computing Research and Basic Energy Sciences DE-SC0008877. This work was also partly supported by NSF under grant NSF/CCF-1318597

[†] Address: Computer Science & Engineering, University of Minnesota, Twin Cities. saad@cs.umn.edu

```
9.        EndIf
10. EndDo
```

The Rayleigh-Ritz (RR) step (lines 4 − 8) is performed once in a while and the test in Line 3 checks if such a projection procedure is required at this iteration. This is dictated mainly by a desire to keep the conditioning of the working basis U under a manageable limit. If l steps are taken, the conditioning of the basis evolves as the l-th power of a certain ratio. This ratio can be estimated from current approximations of the eigenvalues, thus enabling the user to select the next number l of iterations to perform before calling the Rayleigh-Ritz projection. Note that in exact arithmetic, such a projection would not be needed until the very last step when some hypothetical test would tell us that the subspace has converged.

There are other practical reasons why RR steps are needed. For example, step 2 of the algorithm is actually often replaced by a Chebyshev iteration of the form $U := C_l(A)U$ where C_l is a shifted and scaled Chebyshev polynomial of degree l, see, e.g., [19, 15, 22, 17, 18, 23, 24]. In this situation the parameters needed for defining the Chebyshev polynomial are to be computed from eigenvalue approximations and these can only be extracted from a RR step. In addition, these approximations change and therefore the optimal parameters need to also change, as a poor choice of the polynomial can lead to poor convergence, or even divergence.

In many practical situations the matrix A is not actually available in advance but is instead progressively obtained in a nonlinear, or even noisy process, sometimes using information, e.g., eigenvectors of a current instance of the matrix to compute its next instance. Thus, we would compute a new A, call it $\widetilde{A}$, and the above algorithm is repeated with A replaced by $\widetilde{A}$. We next consider a few applications that lead to this general framework.

Density Functional Theory. Density Functional Theory (DFT) studies the electronic structure of materials by solving a simplified version of the Schrödinger equation known as the Kohn-Sham equation :

$$\left[-\frac{\nabla^2}{2} + V_{ion}(r) + V_H(\rho(r), r) + V_{XC}(\rho(r), r)\right] \Psi_i(r) = E_i \Psi_i(r), \tag{1.1}$$

where $\Psi_i(r)$ is a wave function and E_i is a Kohn-Sham eigenvalue. The ionic potential V_{ion} reflects contributions from the core and depends on the position r only. Both the Hartree and the Exchange-Correlation potentials depend on the charge density:

$$\rho(r) = 2 \sum_{i=1}^{n_{occ}} |\Psi_i(r)|^2, \tag{1.2}$$

where n_{occ} is the number of occupied states (half the number of valence electrons in the system). Since the total potential $V_{total} = V_{ion} + V_H + V_{XC}$ depends on $\rho(r)$ which itself depends on eigenfunctions of the Hamiltonian, Equation (1.1) can be viewed as a nonlinear eigenvalue problem, or rather a *nonlinear eigenvector problem.* The Hatree potential V_H is obtained from ρ by solving the Poisson equation $\nabla^2 V_H(r) = -4\pi\rho(r)$ with appropriate boundary conditions. The Exchange-Correlation term V_{XC} is the key to the DFT approach and it captures the effects of reducing the problem from many particles to a one-electron problem, i.e., from replacing wavefunctions with many coordinates into ones that depend solely on space location r. It typically depends explicitly on ρ and r.

Equation (1.1) can be discretized, using, e.g., finite differences, and this leads to an eigenvalue problem $Au = \lambda u$ where one must compute the lowest n_{occ} eigenpairs of A. Starting with a given ρ we can calculate the total potential V_{total} and solve (1.1). However, the resulting (output) ρ as calculated from (1.2), will be different ('inconsistent') from the input ρ utilized to solve (1.1). Self-consistent iterations for solving the Kohn-Sham equation start with an initial guess of the charge density $\rho(r)$, from which a guess for V_{total} is computed. Then (1.1) is solved for $\Psi_i(r)$'s and a new $\rho(r)$ is obtained from (1.2) and the potentials are updated. Then (1.1) is solved again for a new ρ obtained from the new $\Psi_i(r)$'s, and the process is repeated until the difference between two consecutive ρ's falls below a certain tolerance, i.e. until the Hamiltonian matrix A converges.

Using a Chebyshev accelerated form of subspace iteration to compute the eigenpairs of the discretized Hamiltonian, the overall SCF algorithm would be as follows, where C_l is a shifted and scaled Chebyshev polynomial of degree l [23, 24].

Algorithm 1.2. *Standard SCF with Chebyshev-subspace iteration*

```
0. Start: select initial $A_0$ and initial $n \times p$ basis $U_0$. Set $k := 0$.
1. While {$A_k$ has not converged} Do:          % (SCF loop)
2.         Perform subspace iteration with matrix $C_l(A_k)$, starting with $U_k$. Output=$U_{k+1}$
3.         Use $U_{k+1}$ to update potentials and compute $A_{k+1}$. Set $k := k + 1$.
4. End
```

It was observed in [23, 24] that the outer SCF loop and the subspace iteration loop can be merged and the resulting algorithm would still converge with a few simple safeguards, such as starting from a good initial basis U_0. This combination, which leads to enormous savings in computational time, is sketched next.

Algorithm 1.3. *Nonlinear subspace iteration*

```
0. Start: select initial $A_0$ and initial $n \times p$ basis $U_0$. Set $k := 0$
1. While {$A_k$ has not converged} Do:      % (Combined SCF + Subspace iteration loop)
2.         Compute $U_k := C_l(A_k)U_k$
3.         Perform a Rayleigh-Ritz step with the $U_k$ just computed. Output=$U_{k+1}$
4.         Use $U_{k+1}$ to update potentials and compute $A_{k+1}$. Set $k := k + 1$.
5. End
```

The Rayleigh-Ritz step in Line 3 consists of Lines 4–7 of Algorithm 1.1 applied to A_k starting with U_k and yielding U_{k+1} in Line 7. Algorithm 1.3 works well in practice. Although it usually leads to a modest increase in the number of outer SCF iterations when compared to Algorithm 1.2, it can be substantially faster. Indeed, Step 2 of Algorithm 1.2 involves another loop of (standard) subspace iteration to obtain accurate eigenpairs and Algorithm 1.3 replaces this step by a simple application of $C_l(A_k)$ to the current basis U_k. As a result Algorithm 1.3 if often at least an order of magnitude faster than Algorithm 1.2, see [23, 24] for details. The second algorithm can be viewed as a nonlinear form of subspace iteration, whereby a subspace iteration algorithm is being applied to an evolving matrix A_k. The convergence of this algorithm is rather difficult to analyze.

Subspace Tracking. Possibly the best known use of subspace iteration for evolving matrices, is in the context of 'subspace tracking' in signal processing applications, see, e.g., [6, 21, 2, 7]. An excellent survey of relatively recent work on the topic can be found in the introduction of [7]. The problem here is to track the 'signal subspace', which is the eigenspace associated with the largest eigenvalues of a covariance matrix associated with a sequence of signals, in the form of vectors $x(t), t \in \mathbb{Z}$ that are being received sequentially. Variants of the so-called 'power iteration method' play a major role in this application see, e.g., [2, 7, 9]. These track the dominant subspace by essentially performing one step of the orthogonal iteration (subspace iteration without the Rayleigh Ritz projection, see [8]) with the evolving covariance matrix. In its simplest form this covariance matrix is computed as

$$C(t) = \beta C(t-1) + x(t)x(t)^T,$$

where $0 < \beta \leq 1$ is a damping parameter (termed a 'forgetting factor' in [21]). One of the issues is to keep the cost of the algorithm low enough for real time applications.

Matrix Completion. Another notable application where subspace iteration with evolving matrices plays a major role [14] is that of the *matrix completion problem.* In this problem, a rectangular matrix $B \in \mathbb{R}^{m\times n}$ is partially known from its entries in some locations $(i, j) \in \Omega$, and we wish to fully recover it. The underlying assumption is that B has a small rank, i.e., its rank r is such that $r \ll m, n$.

This problem has attracted much attention in recent years because of its occurrence in several important applications that include for example, recommender systems, multitask learning, and structure from motion (see e.g. [16, 13, 5, 1, 12, 4, 11]). Let $\Omega = \{(i,j) | B_{ij} \text{ is observed}\}$ and define $P_\Omega(X) \in \mathbb{R}^{m \times n}$ to be the projection of X onto the observed entries Ω. This means that $P_\Omega(X)$ is a matrix such that $P_\Omega(X)_{ij} = X_{ij}$ if $(i,j) \in \Omega$ and $P_\Omega(X)_{ij} = 0$ otherwise. Ideally, we wish to find *the matrix of smallest rank whose entries in the observed locations are given*, i.e., such that $P_\Omega(X)$ equals $P_\Omega(B)$. This, however, is a hard problem. Instead, a common workable alternative is to seek a matrix of a given (small) rank, say k, that deviates the least from B in the observed entries:

$$\text{Find matrix } X \text{ that minimizes } \|P_\Omega(X) - P_\Omega(B)\|_F \text{ subject to } \operatorname{rank}(X) = k. \tag{1.3}$$

An intuitive algorithm to solve the above problem, is to exploit the subspace iteration for computing the SVD of B. Since B is not known, we use its latest approximation to perform one step of the algorithm and then update the approximation based on the outcome of the step. An initial approximation is first constructed from the truncated rank-k SVD approximation of $P_\Omega(B)$, $X = U_0 S_0 V_0^T = svd_k(P_\Omega(B))$. Here $svd_k(X)$ denotes the rank-k SVD of X obtained by keeping only the first k columns of U, V and the $k \times k$ leading part of Σ in the full SVD of X, $X = U \Sigma V^T$. The next step is to 'correct' this matrix by replacing the entries indexed by Ω by the corresponding known values of B, i.e., we would define the new intermediate approximation of B as $\hat{X} = X + P_\Omega(B - X)$. Then we take a subspace iteration step from $\hat{X}$ to improve the current subspaces spanned by U and V, respectively. This will yield a new approximation X in the form of a partial, rank-k SVD factorization and the process is repeated until convergence. The actual algorithm is sketched below.

```
ALGORITHM 1.4. Subspace Iteration for incomplete matrices.
1. Initialize: $[U_0, S_0, V_0] = svd_k(P_\Omega(B))$,  $X_0 = U_0 S_0 V_0^T$;
2. For i = 0,1,2,..., Do:
3.      $X_{i+1} = X_i + t_i E_i$ where $E_i = P_\Omega(B - X_i)$ and $t_i$ is a damping parameter
4.      $U_{i+1} = \texttt{qf}(X_{i+1} V_i)$;   $V_{i+1} = \texttt{qf}(X_{i+1}^T U_{i+1})$        % Subspace iteration step
5.      $S_{i+1} = U_{i+1}^T X_{i+1} V_{i+1}$;   $X_{i+1} := U_{i+1} S_{i+1} V_{i+1}^T$.     % with matrix $X_{i+1}$.
6. EndFor
```

The algorithm can be stopped when $\|E_i\|_F$ is small enough. Consider the subspace spanned by the sequence of V_i's alone to see what happens in Line 4 of the algorithm. From the QR factorizations, and using appropriate $k \times k$ upper triangular matrices $R_{V,i}$ and $R_{U,i}$, in Line 4 we get :

$$U_{i+1} R_{U,i+1} = X_{i+1} V_i, \quad V_{i+1} R_{V,i} = X_{i+1}^T U_{i+1}, \quad \to \quad V_{i+1} R_{V,i} R_{U,i+1} = X_{i+1}^T X_{i+1} V_i.$$

In other words, $V_{i+1} = \texttt{qf}(X_{i+1}^T X_{i+1} V_i)$. This allows to interpret the algorithm as the standard subspace iteration applied to a varying matrix $X_{i+1}^T X_{i+1}$, obtained from a certain perturbation of the unknown matrix $A = B^T B$.

In this paper we consider the general situation where the subspace iteration is applied to a sequence of evolving matrices A_k, i.e., matrices that vary at each step of the algorithm. The subspace onto which the Rayleigh-Ritz projection is performed at each step is $\mathcal{L}_k = A_k \mathcal{L}_{k-1}$. We will be particularly interested in the situation where A_k converges to a matrix A. Then the question is: under what conditions can we guarantee that the subspace $\mathcal{L}_k$ will converge to the dominant invariant subspace of the limit matrix A? We do not address the convergence of the sequence of matrices A_k, when these matrices are defined from the eigenvectors computed dynamically by the algorithm. This is specific to the given application and is a much more complex issue.

2. Analysis of standard subspace iteration.

We first consider the standard subspace iteration in the unperturbed case. It suffices to examine a single iteration of the algorithm. For this, let $\mathcal{L} = \text{Span}\{V\}$ of dimension m and $\widetilde{\mathcal{L}} = \text{Span}\{AV\}$. Let $u_i, i = 1, \cdots, n$ be the eigenvectors of A associated with the eigenvalues

$$|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_m| > |\lambda_{m+1}| \geq \cdots \geq |\lambda_n|.$$

The case of interest to us is that of real symmetric matrices. We will call P the spectral projector associated with the first m eigenvalues. In this case

$$P = \sum_{i=1}^{m} u_i u_i^T.$$

Finally, we will call S the eigenspace associated with these eigenvalues, i.e., $S = Ran(P)$.

The analysis of subspace iteration is remarkably simple. It rests on a result which shows the existence of a vector in the subspace that is close to u_i. The following result is a simplification of Theorem 5.2 in [19, p. 199].

THEOREM 2.1. *Assume that $dim(P\mathcal{L}) = m$. Then for each vector $u_i, i \leq m$ there is a (unique) vector s_i in $\mathcal{L}$ such that $Ps_i = u_i$. In addition, the vector $\widetilde{s}_i = \frac{1}{\lambda_i} As_i$ of $\widetilde{\mathcal{L}}$ is also such that $P\widetilde{s}_i = u_i$, and it satisfies:*

$$\|\widetilde{s}_i - u_i\| \leq \frac{|\lambda_{m+1}|}{|\lambda_i|} \|s_i - u_i\| \ . \tag{2.1}$$

Proof. Let $Pz_1, Pz_2, \cdots, Pz_m$ be a basis for $P\mathcal{L}$. Since $P\mathcal{L}$ has dimension m, then $P\mathcal{L} = Ran(P)$ and because u_i belongs to this subspace, then we can write (in a unique way):

$$u_i = \sum_{j=1}^{m} \eta_j P z_j = P \left(\sum_{j=1}^{m} \eta_j z_j \right) \equiv P s_i.$$

This shows the first part with $s_i = \sum_{j=1}^{m} \eta_j z_j$. Observe that $s_i - u_i = (I - P)s_i \equiv w_i$ has no components in the first m eigenvectors. In other words the relation $Ps_i = u_i$ is equivalent to:

$$s_i = u_i + w_i; \quad Pw_i = 0. \tag{2.2}$$

Consider now $\widetilde{s}_i$ defined above:

$$\widetilde{s}_i = \frac{1}{\lambda_i} As_i = \frac{1}{\lambda_i} A[u_i + w_i] = u_i + \frac{1}{\lambda_i} Aw_i. \tag{2.3}$$

What is interesting is that $Aw_i = A(I - P)w_i = (I - P)Aw_i$. So the vector $\widetilde{w}_i = \frac{1}{\lambda_i} Aw_i$ satisfies the same properties as w_i, i.e., we have

$$\widetilde{s}_i = u_i + \widetilde{w}; \quad P\widetilde{w}_i = 0.$$

Consider now $\|\widetilde{s}_i - u_i\| = \|\frac{1}{\lambda_i} Aw_i\|$:

$$\|\widetilde{s}_i - u_i\| = \|\frac{1}{\lambda_i} Aw_i\| = \|\frac{1}{\lambda_i}(I - P)A(I - P)w_i\| \leq \|\frac{1}{\lambda_i}(I - P)A(I - P)\| \ \|w_i\|.$$

To show the desired inequality (2.1) note that $w_i = s_i - u_i$, and that the spectral radius of $(I - P)A(I - P)$ is $|\lambda_{m+1}|$. □

What the theorem shows is that if at a given step the assumptions are satisfied, there is a vector

s_i in $\mathcal{L}$ such that $Ps_i = u_i$, and in the following step, there will be an approximate eigenvector in the subspace of approximants which will satisfy the same requirement $P\widetilde{s}_i = u_i$ and which will get closer to u_i by at least $|\lambda_{m+1}/\lambda_i|$. We now need to show that the assumptions remain valid at each step if they are valid at the first step.

LEMMA 2.2. *Let $\{x_i\}_{i=1,\cdots,m}$ a basis of $\mathcal{L}$ such that $\{Px_i\}_{i=1,\cdots,m}$ is also a basis of $P\mathcal{L}$. Then for any $l \geq 1$ the vectors $\{PA^l x_i\}_{i=1,\cdots,m}$ form a basis of $PA^l\mathcal{L}$.*

Proof. The proof is based on the fact that the matrix A^l is *non singular when restricted to S.* This is because the eigenvalues of $A^l_{|S}$ are λ_i^l with $i \geq m$ and they are all nonzero. If $\{Px_i\}_{i=1,\cdots,m}$ are linearly independent then $\{A^l Px_i\}_{i=1,\cdots,m}$ are also linearly independent. Since $A^l P = PA^l$ the result follows. □

A consequence of the lemma is that the dimension of the subspace $PA^l\mathcal{L}$ will always be equal to m if this is true for the initial subspace, i.e., when $l = 0$. When subspace iteration is applied l times, the result is simply a projection method onto the subspace $A^l\mathcal{L}$. Thus, we obtain the following corollary:

COROLLARY 2.3. *Assume that $dim(P\mathcal{L}) = m$. Then for each vector $u_i, i \leq m$ there is a vector s_i in $\mathcal{L}$ such that $Ps_i = u_i$ and a vector $\widetilde{s}_i$ in $A^l\mathcal{L}$ such that $P\widetilde{s}_i = u_i$ and*

$$\|\widetilde{s}_i - u_i\| \leq \left(\frac{|\lambda_{m+1}|}{|\lambda_i|}\right)^l \|s_i - u_i\| \ . \tag{2.4}$$

A consequence of the corollary is that the distance between u_i and the subspace $A^l\mathcal{L}$ is bounded as follows

$$\min_{x\ \in A^l\mathcal{L}} \|x - u_i\| \leq \left(\frac{|\lambda_{m+1}|}{|\lambda_i|}\right)^l \|s_i - u_i\| \ .$$

This allows to prove some results on the accuracy of the projection process onto the subspace $A^l\mathcal{L}$, see [19] for details.

In practice Algorithm 1.1 is often replaced by a filtered-subspace iteration whereby the subspace $A^l\mathcal{L}$ is replaced by $\phi_l(A)\mathcal{L}$, where ϕ_l can be a certain polynomial, e.g., shifted and scaled Chebyshev polynomial, or a rational function such as $(A - \sigma I)^{-l}$ for example. The analysis just discussed can easily be adapted to these cases.

3. Analysis of the perturbed subspace iteration.

We now consider a single step of subspace iteration in which the matrix A is replaced by a matrix of the form $\widetilde{A} = A + E$, where E is some small (real symmetric) perturbation. Thus, the projection process will consist of applying a Rayleigh-Ritz procedure with the subspace $\widetilde{\mathcal{L}} = \widetilde{A}\mathcal{L}$. That is, $\widetilde{\lambda}, \widetilde{u}$ are such that:

$$\begin{cases} \widetilde{\lambda} \in \mathbb{R}, & \widetilde{u} \ \in \ \widetilde{\mathcal{L}}, \\ (\widetilde{A} - \widetilde{\lambda} I)\widetilde{u} & \perp \ \widetilde{\mathcal{L}}. \end{cases}$$

Note that in the next iteration of the algorithm, the subspace $\widetilde{\mathcal{L}}$ is again multiplied by a new perturbed matrix different from $\widetilde{A}$ and the process repeated. In order to simplify notation, we will refrain from using indices until later when then become necessary but we want to stress for example that $A, \mathcal{L}$ should be in reality $A_{k-1}, \mathcal{L}_{k-1}$, where k is some iteration number, and $\widetilde{A}$ and $\widetilde{\mathcal{L}}$ will then stand for A_k and $\mathcal{L}_k = A_k\mathcal{L}_{k-1}$ respectively.

3.1. A useful projector.

We would like to prove a sort of analogue to Theorem 2.1. Under the same assumptions there exists s_i such that $Ps_i = u_i$. However, when applying $\widetilde{A}$ to $\mathcal{L}$, the vector $\widetilde{s}_i = (1/\lambda_i)\widetilde{A}s_i$ which has been instrumental in Theorem 2.1, no longer satisfies the property

$P\widetilde{s}_i = u_i$. Recall that the key property that was exploited in the theorem is that $As_i = \lambda_i u_i + r$ where $r \perp S$. Instead of applying $\widetilde{A}$ to this $s_i \in \mathcal{L}$ we will seek another vector that satisfies this property. For this we need to introduce a projector that will play a key role.

Consider the projector Q that projects *onto* $\widetilde{A}\mathcal{L}$*, orthogonally to* $S = Ran(P)$. Recall that for such a projector, Qz is uniquely defined by the two requirements

$$\begin{cases} Qz & \in & \widetilde{A}\mathcal{L}; \\ z - Qz & \perp & S. \end{cases} \tag{3.1}$$

An illustration is shown in Figure 3.1. These requirements define the range $\widetilde{A}\mathcal{L}$ of Q and its null space $S^\perp$. This projector is well defined when no vector of $\widetilde{A}\mathcal{L}$ is orthogonal to S.

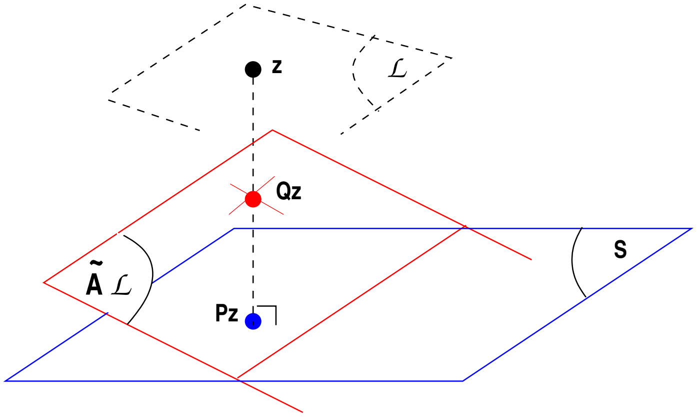

Fig. 3.1. *Illustration of the projector Q*

Lemma 3.1. *Assume that dim*$(P\mathcal{L}) = m$*. Then for each vector* $u_i, i \le m$ *there is a (unique) vector* s_i *in* $\mathcal{L}$ *such that* $Ps_i = u_i$*. In addition, there is a vector* $z_i \in \mathcal{L}$ *such that*

$$\widetilde{A}z_i = \lambda_i u_i + r_i \quad \textit{where:} \tag{3.2}$$

$$r_i = A(s_i - u_i) + (I - Q)Es_i \in S^\perp. \tag{3.3}$$

Proof. The proof of the existence of s_i is the same as for Theorem 2.1. For the second part we will seek z_i in the form $z_i = s_i - f$. Then, defining

$$y = (\widetilde{A} - A)s_i = Es_i, \tag{3.4}$$

we obtain

$$\begin{aligned} \widetilde{A}(s_i - f) &= \widetilde{A}s_i - \widetilde{A}f = As_i + y - \widetilde{A}f \\ &= \lambda_i u_i + A(s_i - u_i) + (y - \widetilde{A}f). \end{aligned} \tag{3.5}$$

Note that $s_i - u_i$ belongs to $S^\perp$, because $Ps_i = u_i$, and so does $A(s_i - u_i)$. Therefore, it remains to select f so that $y - \widetilde{A}f$ also belongs to $S^\perp$, the null space of P. We seek a vector $\widetilde{A}f$ that belongs to $\widetilde{A}\mathcal{L}$ and such that $y - \widetilde{A}f$ is perpendicular to S, i.e., such that $P(y - \widetilde{A}f) = 0$. With the projector Q defined above, clearly the unique vector f such that $\widetilde{A}f = Qy$, is the desired vector. We can then rewrite (3.5) as

$$\widetilde{A}(s_i - f) = \lambda_i u_i + A(s_i - u_i) + (I - Q)Es_i. \tag{3.6}$$

□

In results that will be established later we will need to define the vector:

$$\widetilde{s}_i = \frac{1}{\lambda_i}\widetilde{A}z_i, \tag{3.7}$$

where z_i was defined in the lemma. Thus, (3.6) translates to

$$\lambda_i[\widetilde{s}_i - u_i] = A(s_i - u_i) + (I - Q)Es_i. \tag{3.8}$$

Note that the right-hand side belongs to $S^\perp$.

3.2. A few properties of the projector Q. We saw above that the null space of Q is $S^\perp$. Therefore, $Qw_i = 0$ for any vector $w_i \in S^\perp$. This implies that

$$Q(I - P) = 0 \quad \text{or} \quad Q = QP. \tag{3.9}$$

At the same time, the condition $x - Qx \perp S$ implies that

$$P(I - Q) = 0 \quad \text{or} \quad P = PQ. \tag{3.10}$$

The above two relations show two distinct ways of expressing $Q - P$:

$$Q - P = (I - P)Q = -(I - Q)P. \tag{3.11}$$

Another equality is based on the fact that Q projects onto $\text{Span}(\widetilde{A}\mathcal{L})$ so we have

$$(I - Q)\widetilde{A}x = 0 \quad \text{for any} \quad x \in \mathcal{L}. \tag{3.12}$$

Finally, for the vector $\widetilde{s}_i$ defined in (3.7) we have

$$Qu_i = \widetilde{s}_i \quad \text{and} \quad P\widetilde{s}_i = u_i. \tag{3.13}$$

The 2nd relation is obtained by multiplying (3.8) by P on both sides and noting that the right-hand side of (3.8) is in $S^\perp$, so $P(\widetilde{s}_i - u_i) = 0$, which yields the result. For the first, we proceed in the same way, this time multiplying (3.8) by Q. The right-hand side vanishes again by (3.9), and $Q\widetilde{s}_i = \widetilde{s}_i$ by (3.12). The result follows.

Incidentally, the relations (3.13) show that

$$PQu_i = u_i \quad \text{and} \quad QP\widetilde{s}_i = \widetilde{s}_i, \tag{3.14}$$

and it can also be easily shown from (3.9) and (3.10) that $QPQ = Q$ and $PQP = P$. Note that all these properties can also be visualized on the illustration in Figure 3.1. The relations (3.14) show that P and Q act as inverses of each other in restricted spaces. For example, for any $x \in \tilde{\mathcal{L}}$ we have $QPx = x$ by (3.9) (because $QPx = Qx = x$ if $x \in \tilde{\mathcal{L}}$), which indicates that Q acts as an inverse of $P_{|\tilde{\mathcal{L}}}$ the restriction of P to $\tilde{\mathcal{L}}$. Similarly, by (3.10) $PQx = x$ for any $x \in S$ and so P can be viewed as the inverse of $Q_{|S}$.

It is interesting to observe that in the extreme case achieved at the limit, when $A = \widetilde{A}$, and $\mathcal{L} = S$ then we do have $Q = P$. Indeed, $Px \in A\mathcal{L} = AS$ and $(I - P)x \perp S$ and these are the defining equations for Q for this particular situation. This can also be proved from a matrix formulation by resorting to specific bases, as will be done below.

Later we will need to use the set of vectors z_i found in Lemma 3.1. As it turns out, these are linearly independent under the assumption that the λ_i's under consideration are all nonzero.

Lemma 3.2. *Let the assumptions of Lemma 3.1 hold and assume also that* $|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_m| > 0$. *Then the vectors* $[z_1, z_2, \cdots, z_m]$ *form a basis of* $\mathcal{L}$.

Proof. From equations (3.2) and (3.3) of Lemma 3.1 we have

$$P\tilde{A}z_i = \lambda u_i, i = 1, \cdots, m.$$

Since the λ_i's are nonzero and the u_i form an orthonormal set, it is clear that the z_i's are linearly independent (and that the restriction of $P\tilde{A}$ to $\mathcal{L}$ is invertible). □

Defining the matrix Λ_m as the diagonal matrix, with diagonal entries the eigenvalues $\lambda_1, \cdots, \lambda_m$, and setting $Z = [z_1, \cdots, z_m]$ and $R = [r_1, \cdots, r_m]$, we can rewrite (3.2) as

$$\widetilde{A}Z = U\Lambda_m + R, \tag{3.15}$$

where the columns of R, which we will call 'residuals', are orthogonal to S, i.e., $U^T R = 0$.

It is useful to express the projector Q in matrix form, by using the bases Z of $\mathcal{L}$ and U of S. Since the range of Q is $\tilde{A}Z$ and its null space is $S^\perp$, Q must have the form $Q = (\tilde{A}Z)GU^T$, for a certain $m \times m$ matrix G. To find G, we express the fact that $U^T(I - Q) = 0$ which yields $G = (U^T\widetilde{A}Z)^{-1} = \Lambda_m^{-1}$. Thus, $Q = (\tilde{A}Z)\Lambda_m^{-1}U^T$ and from (3.15) we get:

$$Q = (\tilde{A}Z)\Lambda_m^{-1}U^T = (U\Lambda_m + R)(\Lambda_m^{-1}U^T) = UU^T + R\Lambda_m^{-1}U^T = P + R\Lambda_m^{-1}U^T. \tag{3.16}$$

This shows that at the limit when R goes to zero, the projector Q will converge to the projector P. The next section considers generalizations of Theorem 2.1.

3.3. Vanishing perturbation case. This section considers the general situation when we apply a small perturbation E to the matrix A and this perturbation diminishes in size with each step of the subspace iteration. Such a perturbation could be, for example, the result of an algorithm that builds a sequences of matrices A_k converging to a certain matrix A.

Theorem 3.3. *Assume that dim*$(P\mathcal{L}) = m$. *Then for each eigenvector* $u_i, i \leq m$ *there is a (unique) vector* s_i *in* $\mathcal{L}$ *such that* $Ps_i = u_i$. *In addition, there is a unique vector* $\widetilde{s}_i$ *belonging to* $\widetilde{\mathcal{L}} = \widetilde{A}\mathcal{L}$ *that is also such that* $P\widetilde{s}_i = u_i$, *and that satisfies:*

$$\|\widetilde{s}_i - u_i\| \leq \left[\frac{|\lambda_{m+1}| + \|(I-Q)E\|}{|\lambda_i|}\right]\|s_i - u_i\| + \frac{\|(I-Q)Eu_i\|}{|\lambda_i|}. \tag{3.17}$$

Proof. The existence of $\tilde{s}_i$ was established with the help of the projector Q at the end of Section 3.1, see equations (3.7), (3.8). The uniqueness of $\tilde{s}_i$ follows from the assumption that $P\mathcal{L}$ has dimension m, i.e., the rank of P restricted to $\mathcal{L}$ is m. This assumption implies that when restricted to $\mathcal{L}$, the mapping P is one to one.

Returning to (3.8) we have

$$\widetilde{s}_i - u_i = \frac{1}{\lambda_i}\left[A(s_i - u_i) + (I - Q)Es_i\right]. \tag{3.18}$$

Recalling that $Es_i = (\widetilde{A} - A)s_i$, and setting $w_i \equiv s_i - u_i$, this yields

$$\lambda_i[\widetilde{s}_i - u_i] = Aw_i + (I-Q)Es_i = Aw_i + (I-Q)E(s_i - u_i) + (I-Q)Eu_i \tag{3.19}$$

and so,

$$\widetilde{s}_i - u_i = \frac{1}{\lambda_i}\left[(A + (I-Q)E)(s_i - u_i) + (I-Q)Eu_i\right]. \tag{3.20}$$

The proof ends by taking norms and noting that $\|Aw_i\| \leq |\lambda_{m+1}| \|w_i\|$ since $w_i \in S^\perp$. □
It is interesting to note that the vector Eu_i is equal to $(\widetilde{A} - A)u_i = (\widetilde{A} - \lambda_i)u_i$, which is *the residual of the exact eigenpair* (λ_i, u_i) *with respect to the perturbed matrix* $\widetilde{A}$.

We now re-introduce the subspace iteration index k. So $A, \widetilde{A}$, and Q become A_{k-1}, A_k and Q_k respectively and E and $\mathcal{L}$ are indexed into E_k, and $\mathcal{L}_k$, respectively. Recall that

$$\mathcal{L}_k = A_k \mathcal{L}_{k-1}. \tag{3.21}$$

Superscripts are also needed for the vectors s_i so that s_i becomes $s_i^{(k-1)}$ and $\widetilde{s}_i$ becomes $s_i^{(k)}$. If we assume that $P\mathcal{L}_k = P(A_k\mathcal{L}_{k-1})$, is again of dimension m, then the vector $\tilde{s}_i$, i.e., $s_i^{(k)}$ is unique. This means that provided $dim(P\mathcal{L}_k)$ remains equal to m at each step k we will have a sequence of vectors $s_i^{(k)}$, $k = 0, 1, \cdots$, all satisfying the condition $Ps_i^{(k)} = u_i$ and the relation (3.20) with $\tilde{s}_i$ replaced by $s_i^{(k)}$ and s_i replaced by $s_i^{(k-1)}$.

The above results lead to the following conclusion: If we have a sequence of E_k's that converge to zero and if we can ensure that the projector $I - Q_k$ remains bounded, then there is a sequence of vectors $s_i^{(k)}$ in the subspace that converges to u_i. To state this more formally we first rewrite the inequality (3.17) in the slightly weaker form:

$$\|s_i^{(k)} - u_i\| \leq \left[\frac{|\lambda_{m+1}| + \|(I - Q_k)E_k\|}{|\lambda_i|} \right] \|s_i^{(k-1)} - u_i\| + \frac{\|(I - Q_k)E_k\|}{|\lambda_i|}. \tag{3.22}$$

The above relation, can be written as:

$$d^{(k)} \leq (\alpha + \beta_k) d^{(k-1)} + \beta_k, \quad k \geq 1, \tag{3.23}$$

in which $d^{(k)} = \|s_i^{(k)} - u_i\|$, $\beta_k = \|(I - Q_k)E_k\|/|\lambda_i|$, and $\alpha = |\lambda_{m+1}/\lambda_i|$. The following lemma states a sufficient condition under which $d^{(k)}$ converges to zero.

Lemma 3.4. *Let a sequence* $d^{(k)}$ *of nonnegative numbers satisfy (3.23) where* $0 \leq \alpha < 1$ *and* $\beta_k \geq 0$. *Then a sufficient condition for this sequence to converge to zero is that:*

$$\lim_{k \to \infty} \beta_k = 0. \tag{3.24}$$

Proof. For convenience we rewrite relation (3.23) as $d^{(k)} \leq \gamma_k d^{(k-1)} + \beta_k$, where $\gamma_k = \alpha + \beta_k$. We also set $\beta_0 \equiv d^{(0)}$ and $d^{(-1)} \equiv 0$ so (3.23) is now valid for $k = 0$. By the assumption that β_k converges to zero, the scalar γ_k does not exceed a certain $\gamma < 1$ for k larger than a certain k_0. Without any loss of generality we can assume that this is in fact true for all k (otherwise consider only the sequence starting with index k_0). From this and the assumptions we can easily get:

$$0 \leq d^{(k)} \leq \gamma^k \beta_0 + \gamma^{k-1}\beta_1 + \cdots + \gamma\beta_{k-1} + \beta_k. \tag{3.25}$$

Define $\eta_k = \sum_{i=0}^k \beta_i \gamma^{k-i}$, and split this sum in two as follows

$$\eta_k = \sum_{i=0}^{k} \beta_i \gamma^{k-i} = \sum_{i=0}^{p} \beta_i \gamma^{k-i} + \sum_{i=p+1}^{k} \beta_i \gamma^{k-i}. \tag{3.26}$$

Since the sequence β_i converges to zero, it is bounded from above by a certain β so the first term above can be bounded as follows:

$$\sum_{i=0}^{p} \beta_i \gamma^{k-i} = \gamma^{k-p} \sum_{i=0}^{p} \beta_i \gamma^{p-i} \leq \gamma^{k-p} \beta \frac{1 - \gamma^{p+1}}{1 - \gamma} \leq \frac{\gamma^{k-p}\beta}{1 - \gamma}.$$

For large enough p all β_i's for $i \geq p$ are less than an arbitrarily small value ϵ. So for a large enough p the second term in the right-hand side of (3.26) is bounded from above by

$$\sum_{i=p+1}^{k} \beta_i \gamma^{k-i} \leq \epsilon \sum_{i=p+1}^{k} \gamma^{k-i} = \epsilon \frac{1-\gamma^{k-p}}{1-\gamma} \leq \frac{\epsilon}{1-\gamma}.$$

As can be seen, the two terms in which η_k has been split can be made arbitrarily small by taking a large enough p and a large enough k. As a result η_k converges to zero and so does $d^{(k)}$ as is clear from (3.25). □

3.4. The non-vanishing perturbation case. We can have a situation whereby the perturbation E_k does not go down to zero but remains small. We wish to adapt the result of the previous section for this case. The assumptions we make now are that the norms $\|(I - Q_k)E_k\|$ remain small enough throughout the iterations. The result we prove is a rather straightforward adaptation of Lemma 3.4. First, we define $d_i^{(k)} = \|s_i^{(k)} - u_i\|$ and use the same symbol β_k as before for $\|(I - Q_k)E_k\|$. Then, Equation (3.22) yields:

$$d_i^{(k)} \leq \left(\frac{|\lambda_{m+1}|}{|\lambda_i|} + \frac{\beta_k}{|\lambda_i|} \right) d_i^{(k-1)} + \frac{\beta_k}{|\lambda_i|}, \quad k \geq 1, \tag{3.27}$$

from which the following result can be derived.

Proposition 3.5. *Let i between 1 and m. Assume that for all $k \geq 1$ we have $\beta_k \leq \beta$ where β satisfies:*

$$\frac{|\lambda_{m+1}|}{|\lambda_i|} + \frac{\beta}{|\lambda_i|} \equiv \gamma_i < 1.$$

Then, the sequence $d_i^{(k)}$ is such that

$$d_i^{(k)} \leq \gamma_i^k d_i^{(0)} + \frac{\beta}{(|\lambda_i| - |\lambda_{m+1}|) - \beta}, \tag{3.28}$$

i.e., the vectors $s_i^{(k)}$ will remain relatively close to u_i if β is small.

Proof. We proceed similarly to the proof of Lemma 3.4. An induction argument shows that

$$d_i^{(k)} \leq \gamma_i^k d_i^{(0)} + \frac{\beta}{|\lambda_i|} \sum_{j=0}^{k-1} \gamma_i^{k-j}, \quad \text{for} \quad k \geq 1,$$

and using the fact that $\gamma_i < 1$ we obtain:

$$d_i^{(k)} \leq \gamma_i^k d_i^{(0)} + \frac{\beta}{|\lambda_i|} \times \frac{1}{1-\gamma_i} = \gamma_i^k d_i^{(0)} + \frac{\beta}{|\lambda_i| - |\lambda_{m+1}| - \beta}.$$

□

The first term in the right-hand side of (3.28) tends to zero as k tends to infinity while the second term remains of the order of β, the bound on the norm of the projected perturbation $\|(I-Q_k)E_k\|$. It is important to note that the assumptions of the proposition depend on the index i of the eigenpair. They may be satisfied for one i and not another, and so one may well have a situation in which some (typically the largest) eigenpairs will be well approximated but not the other ones.

3.5. Analysis of $(I - Q_k)E_k$. The results of Section 3.3 show that at each step of the subspace iteration for an evolving matrix we can find a good approximate eigenvector in the subspace of approximants provided that $(I - Q_k)E_k$ tends to zero as k tends to infinity. All of the convergence analysis is now captured by this term, which we would like to analyze.

The goal of this section is to show how we can bound $\|I - Q_k\|$. Recall that Q_k is a projector but that it is not orthogonal so $\|I - Q_k\|$ will be larger than one in general. Using (3.11) we can write $I - Q_k$ as

$$(I - Q_k) = (I - P) + (P - Q_k) = (I - P) + (P - Q_k)P. \tag{3.29}$$

Take the norm of the last term. Since $\|Px\| \leq \|x\|$, we get

$$\|(P - Q_k)P\| = \max_{x \neq 0} \frac{\|(P - Q_k)Px\|}{\|x\|} \leq \max_{Px \neq 0} \frac{\|(P - Q_k)Px\|}{\|Px\|}.$$

It is easy to show that the inequality in the above relations is actually an equality. This is because if the maximum in the last term is reached for a certain x_0 then clearly:

$$\max_{Px \neq 0} \frac{\|(P - Q_k)Px\|}{\|Px\|} = \frac{\|(P - Q_k)Px_0\|}{\|Px_0\|} = \frac{\|(P - Q_k)P(Px_0)\|}{\|Px_0\|} \leq \max_{x \neq 0} \frac{\|(P - Q_k)Px\|}{\|x\|}.$$

Therefore we have,

$$\|(P - Q_k)P\| = \max_{x \in S, Px \neq 0} \frac{\|(P - Q_k)Px\|}{\|Px\|} = \max_{z \in S, z \neq 0} \frac{\|(P - Q_k)z\|}{\|z\|} = \max_{z \in S, z \neq 0} \frac{\|z - Q_k z\|}{\|z\|}$$

As is illustrated in Figure 3.2, because $z - Q_k z$ is perpendicular to S, the above maximum is simply the tangent of the angle between the two subspaces, S and $\mathcal{L}_k = A_k \mathcal{L}_{k-1}$, i.e.,

$$\|(P - Q_k)P\| = \max_{z \in S, \|z\| = 1} \|z - Q_k z\| = \tan \angle(\mathcal{L}_k, S) \tag{3.30}$$

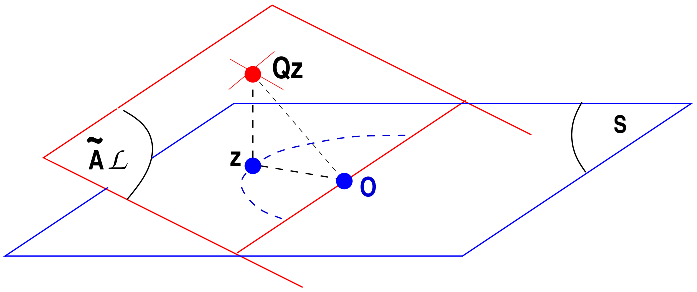

FIG. 3.2. *Tangent of the angle between the subspaces $A_k\mathcal{L}$ and S.*

In the end, putting (3.29) and (3.30) together we obtain,

$$\|I - Q_k\| \leq 1 + \tan \angle(\mathcal{L}_k, S), \tag{3.31}$$

which will allow us to state a convergence result as a corollary of Lemma 3.4. Looking at (3.22), if we assume that the angle between S and $\mathcal{L}_k$ stays bounded away from $\pi/2$ then $I - Q_k$ is bounded from above and so if E_k converges to zero we will also have $\lim_{k \to \infty} \|(I - Q_k)E_k\| = 0$. We are now in the situation of Lemma 3.4 and we can say that the sequence $s_i^{(k)}$ converges to u_i. The result is formally stated next.

COROLLARY 3.6. *Let $dim(P\mathcal{L}_0) = m$ and $|\lambda_m| > 0$ and assume that there is an angle θ such that at each step of subspace iteration we have $\angle(\mathcal{L}_k, S) \leq \theta < \pi/2$. Then for each k, we have $dim(P\mathcal{L}_k) = m$ and there is a sequence $s_i^{(k)} \in \mathcal{L}_k$ such that $Ps_i^{(k)} = u_i$. If, in addition, $\lim_{k\to\infty} \|E_k\| = 0$, the sequence $s_i^{(k)}$ converges to u_i.*

Proof. The existence of $s_i^{(k)}$ for each k was established at the end of Section 3.1, see equations (3.7) and (3.8), provided $dim(P\mathcal{L}_{k-1}) = m$. An induction argument and the result of Lemma 2.2 show that indeed for each k, $dim(\mathcal{L}_{k-1}) = m$ because we have $|\lambda_m| > 0$. The rest of the proof follows from the arguments given above which show that $lim_{k\to\infty}\|(I - Q_k)E_k\| = 0$. □

3.6. Subspace convergence analysis. The results of the previous two sections focus on one specific eigenvector and establish results by making a few assumptions on the subspace encountered at each step of the subspace iteration algorithm. Intuitively, one can expect that by considering the eigenvectors of interest, i.e., those associated with the dominant p eigenvalues, then we should be able to prove a sort of global result, which deals with the whole subspace rather than single eigenvectors. A result of this type, whose derivation is more complex, is discussed in this section.

We begin by splitting $(I-Q_k)E_k$ in two terms. Relation (3.9) implies that $(I-Q_k)(I-P) = I-P$ and so we have

$$(I - Q_k)E_k = (I - Q_k)PE_k + (I - Q_k)(I - P)E_k = (I - Q_k)PE_k + (I - P)E_k. \tag{3.32}$$

Next we will derive an expression for the first term of the right-hand side. Recall expressions (3.16) and (3.15) of Section 3.2, which use the Z basis of $\mathcal{L}$ and the canonical basis U of S to express Q_k. Exploiting (3.16) and (3.11), in which R is now replaced by R_k, we obtain:

$$(I - Q_k)PE_k = R_k\Lambda_m^{-1}U^TE_k. \tag{3.33}$$

Consider the matrix $R_k\Lambda_m^{-1}$. Each column of this matrix represents the residual $r_i = A_kz_i - \lambda_iu_i$ scaled by λ_i. We also know from the definition of $s_i^{(k)}$, see Equation (3.7), that $A_kz_i = \lambda_is_i^{(k)}$. Thus, the $i-th$ column of $R_k\Lambda_m^{-1}$ is equal to $s_i^{(k)} - u_i$ and

$$R_k\Lambda_m^{-1} = [s_1^{(k)} - u_1, s_2^{(k)} - u_2^{(k)}, \cdots, s_m^{(k)} - u_m]. \tag{3.34}$$

It will be convenient to utilize the 1-norm of this matrix. However, the analysis up to this point relied on the 2-norm. To overcome this issue, we will use component-wise analysis in the eigenspace. Note that all vectors $s_i^{(k-1)} - u_i$, as well as the vector $s_i^{(k)} - u_i$, belong to the space orthogonal to S. If we call W the basis of the subspace Span$\{u_{m+1}, \cdots, u_n\}$, which is the range of $I - P$, then we can write the components of the vectors $s_i^{(k-1)} - u_i$ in the W basis as $W^T(s_i^{(k-1)} - u_i)$. The starting point now is (3.20) which translates to

$$W^T(s_i^{(k)} - u_i) = \frac{1}{\lambda_i}\left[(W^TA + W^T(I - Q_k)E_k)w_iW^T(s_i - u_i) + W^T(I - Q_k)E_ku_i\right]. \tag{3.35}$$

Using the 1-norm we can state:

$$\|W^T(s_i^{(k)} - u_i)\|_1 \leq \left[\frac{\|W^TAw_i\|_1}{|\lambda_i|} + \frac{\|W^T(I - Q_k)E_kw_i\|_1}{|\lambda_i|}\right]\|W^T(s_i - u_i)\|_1 + \frac{\|W^T(I - Q_k)E_ku_i\|_1}{|\lambda_i|}. \tag{3.36}$$

The matrix W^TAW is diagonal with diagonal entries $\lambda_{m+1}, \cdots, \lambda_n$ and so,

$$\|W^TAw_i\|_1 = |\lambda_{m+1}|. \tag{3.37}$$

Let:

$$\eta_i^{(k)} = \|W^T(s_i^{(k)} - u_i)\|_1, \qquad \eta^{(k)} = \max_i \ \eta_i^{(k)} \tag{3.38}$$

$$\mu^{(k)} = \|W^T(I - Q_k)E_k\|_1, \qquad \epsilon^{(k)} = \|E_k\|_1, \tag{3.39}$$

$$\alpha = \|U\|_1, \qquad \varphi = \|w_i\|_1 \tag{3.40}$$

$$\delta = \|U^T\|_1, \qquad \nu = \|W^T\|_1. \tag{3.41}$$

With these definitions, inequality (3.36) and relation (3.37) lead to

$$\eta_i^{(k)} \leq \left[\frac{|\lambda_{m+1}|}{|\lambda_i|} + \frac{\varphi}{|\lambda_i|}\mu^{(k)}\right]\eta_i^{(k-1)} + \frac{\alpha}{|\lambda_i|}\mu^{(k)}, \tag{3.42}$$

and our goal is to show that $\eta^{(k)}$ converges to zero under some conditions.

Proposition 3.7. *If there is a step k_0 such that:*

$$\eta^{(k_0)} \leq 1 \quad \textit{and} \quad \epsilon^{(k)} \leq \frac{|\lambda_m| - |\lambda_{m+1}|}{2\max\{\nu, \delta\}(\varphi + \alpha)} \quad \forall k \geq k_0,$$

then for all subsequent steps $k \geq k_0$, $\eta^{(k)}$ will remain ≤ 1. If in addition $\lim_{k\to\infty} \epsilon^{(k)} = 0$, then $\lim_{k\to\infty} \eta^{(k)} = 0$, establishing the convergence of the subspace iteration scheme in this situation.

Proof. Combining (3.32), (3.33) and (3.34) yields

$$\begin{aligned}\|W^T(I - Q_k)E_k\|_1 &\leq \|W^T(I - Q_k)PE_k\|_1 + \|W^T(I - P)E_k\|_1 \\ &\leq \max_i \|W^T(s_i^{(k-1)} - u_i)\|_1\|U^TE_k\|_1 + \|W^T(I - P)E_k\|_1 \end{aligned} \tag{3.43}$$

Note that $\|W^T(I - P)E_k\|_1 = \|W^T(WW^T)E_k\|_1 = \|W^TE_k\|_1 \leq \nu\epsilon^{(k)}$. Then from (3.38–3.41),

$$\mu^{(k)} \leq \left[\delta \times \max_i \eta_i^{(k)} + \nu\right]\epsilon^{(k)} = \left[\delta \times \eta^{(k)} + \nu\right]\epsilon^{(k)}. \tag{3.44}$$

Equations (3.42) and (3.44) can now be combined:

$$\eta^{(k)} \leq \frac{|\lambda_{m+1}|}{|\lambda_m|}\eta^{(k-1)} + \frac{\epsilon^{(k)}}{|\lambda_m|}\left[\delta\eta^{(k)} + \nu\right]\left[\varphi\eta^{(k-1)} + \alpha\right] \rightarrow$$

$$\eta^{(k)}\left[1 - \frac{\delta}{|\lambda_m|}(\varphi\eta^{(k-1)} + \alpha)\epsilon^{(k)}\right] \leq \frac{|\lambda_{m+1}|}{|\lambda_m|}\eta^{(k-1)} + \frac{\nu}{|\lambda_m|}(\varphi\eta^{(k-1)} + \alpha)\epsilon^{(k)}.$$

We end up with:

$$\eta^{(k)} \leq \frac{|\lambda_{m+1}|\eta^{(k-1)} + \nu(\varphi\eta^{(k-1)} + \alpha)\epsilon^{(k)}}{|\lambda_m| - \delta(\varphi\eta^{(k-1)} + \alpha)\epsilon^{(k)}}. \tag{3.45}$$

With this, the first part of the proposition follows from a straightforward induction argument using (3.45). If in addition, $\lim_{k\to\infty} \epsilon^{(k)} = 0$, considering only those k's that are larger than k_0, then $\eta^{(k)} \leq 1$ and (3.44) shows that $\mu^{(k)} \to 0$. Now, inequality (3.42) brings us to the same situation as that treated by Lemma 3.4 and this proves convergence of the sequence $\eta^{(k)}$ to zero. □

4. Numerical Illustrations. The goal of this section is to provide a few illustrations of the convergence analysis discussed in this paper. The focus is on bounds (3.17) and its weaker version (3.22) since these are at the basis of all the analysis in Section 3.

4.1. Matrix completion. In a first example we build a small rectangular matrix of size $m \times n$ which is exactly of rank p, then we replace a number of its entries by zeros and try to recover the original matrix. The matrix is built from the following `Matlab` commands where we have set $m = 128, n = 16, p = 4$. In the script `had(m,p)` extract the first p columns of a Hadamard matrix of dimension m. These columns are orthogonal with entries ± 1 and the scaling by $\sqrt{m}$ makes the resulting columns orthonormal. The second line sets the singular values of X to $1/p, 2/p, \cdots, 1$. The 3rd and 4th lines apply a kind of cosine transform to obtain an $m \times n$ matrix of rank p.

```
X = had(m,p)/sqrt(m);       % % First p columns of Hadamard matrix (X'*X==I)
X = X*diag([1:p]/p);        % % Resets singular values
u = [1:n]/(2*n+1);  v = [1:p];
B = X*cos(2*pi*v'*u);       % % Matrix B is now 128 × 16 of rank p = 4
```

All entries of the matrix B are nonzero. Next we will hide a big portion of the matrix by replacing some entries by zeroes. Specifically, we hide every 6-th element of the matrix by considering it as a long vector of length mn. This is achieved with the help of a 'mask' matrix C (the negative of the boolean matrix called Ω in the introduction) obtained with the following matlab commands, where we have set $l = 6, istart = 1, mn = m * n$,

```
C = ones(mn,1);   C(istart:l:mn) = 0;   C = reshape(C,m,n);
```

Our incomplete matrix X results from the pointwise product command `X = B .* C`, whose effect is to replace by zeros all those entries b_{ij} for which $C_{ij} = 0$ while leaving the others unchanged. The result of this is that 342 entries out of the total 2048 are hidden (replaced by zeros). Our task is to recover the original B by algorithm 1.4 exactly as it is described with $t_i \equiv 1$ for all i. We compare the matrix $A = B^T B$ ('exact' matrix) and its successive approximations $A_j = X_j^T X_j$ produced by the algorithm. The algorithm converges steadily after an initial irregular phase (steps 1 to 12), so we show only the first 25 steps. Note that here the limit matrix is known in advance since it is the original matrix $A = B^T B$. Results have been shown in the literature that guarantee that recovery is feasible under some conditions [3, 10].

In addition to $\|E_j\| = \|A - A_j\|$ we illustrate how the first approximate eigenvector obtained by the procedure converges to the exact eigenvector u_1. Thus, Figure 4.1 shows the norms $\|s_1^{(j)} - u_1\|$ (dash line) and its upper bound (continous line) as given from (3.17). Specifically, what is plotted for (3.17) is the predicted value of $\|s_1^{(j+1)} - u_1\|$ from the right-hand sides of (3.17) using the current value of $\|s_1^{(j)} - u_1\|$. In other words, this is a 'one-step' prediction in which the bound is applied once at each step and uses the exact value of $\|s_1^{(j)} - u_1\|$ for the prediction, not one that is itself predicted as we would do in a standard subspace iteration where the matrix is constant. The term $\|(I - Q_j)E_j u_1\|$ of the bound is also shown.

As can be seen, after an initial stage (10 steps or so), the curve of the exact value of $\|s_1^{(j)} - u_1\|$ (dash line) becomes basically indistiguishible from its predicted value from (3.17) (solid line). The weaker bound (3.22) is not shown here for clarity. It follows a similar pattern, although it takes a few more steps for the two curves to become indistiguishable on the plot (about 15 steps or so). Finally, we show the actual error $\|u_1 - u_1^{(j)}\|$ between the exact and approximate eigenvectors obtained from the algorithm. Note that here $\lambda_{m+1} = 0$ since the limit matrix A is of rank exactly $p = 4$ and we use subspace iteration with $m = 4$. The situation illustrated here is idealistic because in practice the matrix X represents an incomplete version of a noisy matrix B which, therefore, does not typically have a small rank p. The target matrix B may have a small 'approximate' rank p but that rank is not known in advance. However, this example does illustrate the power of this approach which is capable of recovering a matrix that lost 16.7% of its entries. More powerful algorithms have been developed for this problem, see, e.g., [14] for references.

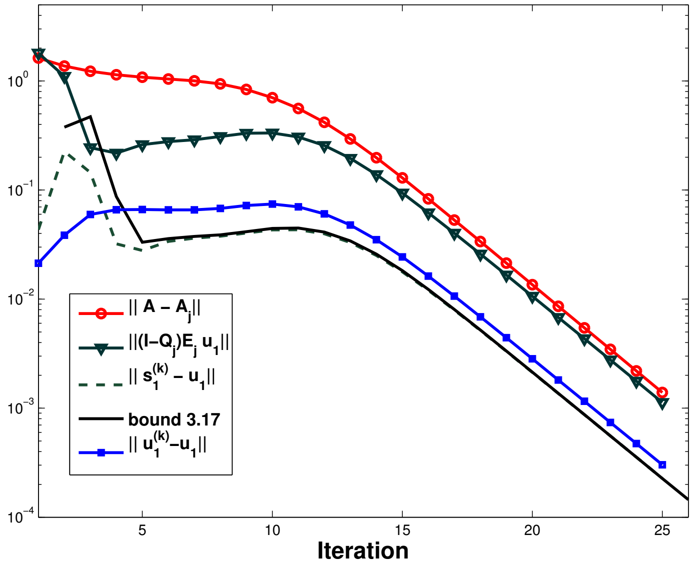

Fig. 4.1. *Illustration of bounds for a simple matrix completion example Q*

4.2. An evolving matrix from a Schrödinger-type equation. We construct a finite difference approximation to solve the following Schrödinger-type equation set in a two-dimensional domain:

$$(-\Delta + V)u = \lambda u. \tag{4.1}$$

with Dirichlet boundary conditions on the square $\Omega = (0,\ 1)^2$. Using centered finite differences with $n_x = n_y = 27$ discretization points in each direction results in an eigenvalue problem of size $n = 25^2 = 625$. The potential $V(x, y)$ is a function that has a value of zero everywhere except in a disk or radius $r = 0.2$ centered at the center of the domain, where it has a constant value of $V = 0.5$. The final matrix resulting from discretizing the Laplacean alone is scaled by h^2 where $h = 1/n_x$, resulting in a standard so-called 5-point matrix, with diagonal entries equal to 4, and off-diagonal entries a_{ij} equal to -1, when i, j are labels of vertices of an edge on the mesh. We call this matrix A_0. The term V is added to A_0 so the diagonal entries of the resulting matrix are either 4.0 or 4.5. We will refer to the discretized V as V_h (a diagonal matrix). Let A be the resulting discretized matrix $A_0 + V_h$. We will use a sort of homotopy approach and illustrate the bounds (3.17) and (3.22) in the process. Since we are interested in the lowest eigenvalues, we will assume in what follows that that eigenvalues are always labeled in ascending order.

We select a number of steps nit (here $nit = 30$) and then apply one step of subspace iteration with the matrices

$$A_k = A_0 + \frac{k}{nit} V_h, \qquad k = 1, 2, \cdots, nit,$$

in sequence. Thus, the matrix evolves from A_1 to $A_{nit} \equiv A$ and the subspace iteration algorithm 1.1 is applied where in Line 2, A is replaced by A_k, or, to be more accurate, by a shifted version $A_k - \sigma I$ as is explained below. The subspace dimension is now $p = 12$ and we again watch the convergence of the first eigenvector (lowest). Since we seek the p smallest eigenvalues we need to apply a shift to the matrix. In practice the most appropriate shift to use at a given iteration is computed dynamically using information gathered along the process. However, for simplicity we fix the shift to be equal to the optimal shift at the limit which is equal to $\sigma = (\lambda_{p+1} + \lambda_n)/2 = 4.3596..$ where the λ_i's are the eigenvalues of the limit matrix A.

We now show both of the bounds (3.17) and (3.22). Similarly to the previous example, what is plotted for (3.17) and (3.22) are the predicted values of $\|s_1^{(k+1)} - u_1\|$ from the right-hand sides of (3.17) and (3.22) respectively using exact values of $\|s_1^{(k)} - u_1\|$ at step k. Once more, the bounds (3.17) and (3.22) are quite close to each other. As the iteration number increases toward 30, the matrix becomes close to the final matrix A and therefore E_k becomes close to zero and the two bounds become identical when $E_k = 0$ at the end-point. As we approach the end point these bounds also yield a remarkably good prediction of the actual $\|s_1^{(k+1)} - u_1\|$, reaching a near-linear process similar to the one achieved by the standard subspace iteration, which improves the error by $|\lambda_{m+1}/\lambda_1|$ each time. In contrast, this experiment revealed that the initial part of the process is far from linear. When E is large, the term $\|(I - Q)E\|/|\lambda_i|$ can be larger than one. In this case the bounds (3.17), (3.22) will be pessimistic as is illustrated in the figure for the first few iterations of the process.

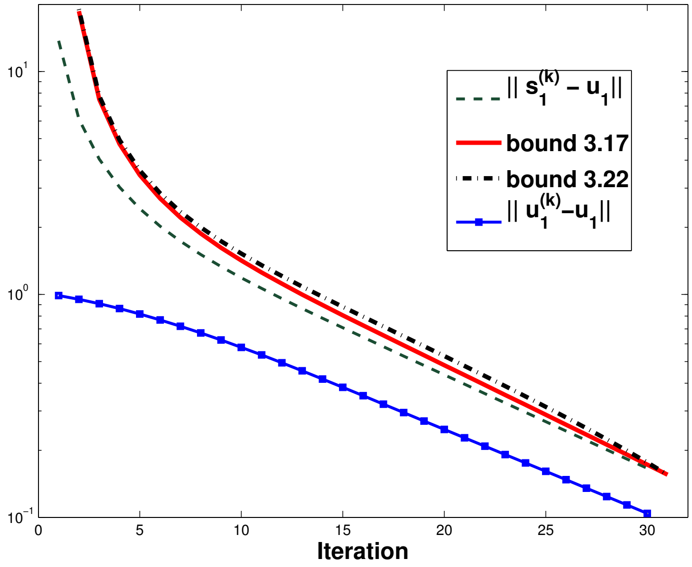

FIG. 4.2. *Illustration of bounds (3.17) and (3.22) for a Laplacean matrix plus an evolving potential.*

5. Concluding remarks. The analysis of the subspace iteration algorithm for evolving matrices shows that we can easily obtain convergence when the perturbation vanishes as the algorithm progresses. This may seem intuitively obvious except that it is not clear without the analysis whether a sufficiently fast decay of the perturbations is required to ensure convergence, nor is it known how fast is the resulting algorithm when it converges. The convergence analysis of Section 3 shows that all that is needed, in addition to the convergence of the evolving matrix, is that the angle between the subspace on which the Rayleigh-Ritz projection is performed, and the desired eigenspace, remains bounded away from the right angle. In the non-converging case, the best we can hope for is to show that there is a sequence of approximations that remain close to the exact eigenvector when the perturbations remain small.

A difficult question not addressed here is to study the convergence of the matrices A_k themselves in the situation when A_k depends on the eigenvectors of the previous matrix A_{k-1}, as is the case in DFT. Such an analysis would require tools that are common when dealing with systems of nonlinear equations or nonlinear optimization problems. It is hoped that in the future the analysis of this paper can help advance the study of these nonlinear forms of the subspace iteration algorithm.

REFERENCES

[1] Y. Amit, M. Fink, N. Srebro, and S. Ullman, *Uncovering shared structures in multiclass classification*, in Proceedings of the 24th international conference on Machine learning, ICML '07, 2007, pp. 17–24.

[2] R. Badeau, B. David, and G. Richard, *Fast approximated power iteration subspace tracking*, IEEE Transactions on Signal Processing, 53 (2005), pp. 2931–2941.

[3] E. Candes and T. Tao, *The power of convex relaxation: Near-optimal matrix completion*, Information Theory, IEEE Transactions on, 56 (2010), pp. 2053–2080.

[4] P. Chen and D. Suter, *Recovering the Missing Components in a Large Noisy Low-Rank Matrix: Application to SFM*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 26 (2004), pp. 1051–1063.

[5] P. L. Combettes and V. R. Wajs, *Signal recovery by proximal forward-backward splitting*, Multiscale Modeling and Simulation, 4 (2005), pp. 1168–1200.

[6] P. Comon and G. Golub, *Tracking a few extreme singular values and vectors in signal processing*, Proceedings of the IEEE, 78 (1990), pp. 1327–1343.

[7] X. Doukopoulos and G. Moustakides, *Fast and stable subspace tracking*, IEEE Transactions on Signal Processing, 56 (2008), pp. 1452–1465.

[8] G. H. Golub and C. F. V. Loan, *Matrix Computations, 4th edition*, Johns Hopkins University Press, Baltimore, MD, 4th ed., 2013.

[9] Y. Hua, Y. Xiang, T. Chen, K. Abed-Meraim, and Y. Miao, *A new look at the power method for fast subspace tracking*, Digital Signal Processing, 9 (1999), pp. 297 – 314.

[10] R. H. Keshavan, S. Oh, and A. Montanari, *Matrix completion from a few entries*, CoRR, abs/0901.3150 (2009).

[11] Y. Koren, R. Bell, and C. Volinsky, *Matrix factorization techniques for recommender systems*, Computer, 42 (2009), pp. 30–37.

[12] M. Mardani, G. Mateos, and G. B. Giannakis, *Decentralized sparsity-regularized rank minimization: Algorithms and applications*, IEEE Transaction on Signal Processing, 61 (2013).

[13] B. Marlin, *Collaborative filtering: A machine learning perspective*, 2004.

[14] T. Ngo and Y. Saad, *Scaled gradients on Grassmann manifolds for matrix completion*, in Advances in Neural Information Processing Systems 25, P. Bartlett, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, eds., NIPS, 2012, pp. 1421–1429. (Spotlight), paper available in http://books.nips.cc/nips25.html.

[15] B. N. Parlett, *The Symmetric Eigenvalue Problem*, no. 20 in Classics in Applied Mathematics, SIAM, Philadelphia, 1998.

[16] J. D. M. Rennie and N. Srebro, *Fast maximum margin matrix factorization for collaborative prediction*, in Proceedings of the 22nd International Conference on Machine Learning (ICML, ACM, 2005, pp. 713–719.

[17] H. Rutishauser, *Computational aspects of F. L. Bauer's simultaneous iteration method*, Numerische Mathematik, 13 (1969), pp. 4–13.

[18] ———, *Simultaneous iteration for symmetric matrices*, in Handbook for automatic computations (linear algebra), J. Wilkinson and C. Reinsch, eds., New York, 1971, Springer Verlag, pp. 202–211.

[19] Y. Saad, *Numerical Methods for Large Eigenvalue Problems- classics edition*, SIAM, Philadelpha, PA, 2011.

[20] Y. Saad, J. Chelikowsky, and S. Shontz, *Numerical methods for electronic structure calculations of materials*, SIAM review, 52 (2009), pp. 3–54.

[21] G. Stewart, *An updating algorithm for subspace tracking*, IEEE Transactions on Signal Processing, 40 (1992), pp. 1535–1541.

[22] G. W. Stewart, *Matrix Algorithms II: Eigensystems*, SIAM, Philadelphia, 2001.

[23] Y. Zhou, Y. Saad, M. L. Tiago, and J. R. Chelikowsky, *Parallel self-consistent-field calculations via Chebyshev-filtered subspace acceleration*, Phy. rev. E, 74 (2006), p. 066704.

[24] Y. Zhou, Y. Saad, M. L. Tiago, and J. R. Chelikowsky, *Self-consistent-field calculation using Chebyshev-filtered subspace iteration*, J. Comp. Phys., 219 (2006), pp. 172–184.