Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 16, pp. 99-110, March 2007

ELA
http://math.technion.ac.il/iic/ela

USING LEAST-SQUARES TO FIND AN APPROXIMATE EIGENVECTOR*

DAVID HECKER† AND DEBORAH LURIE†

Abstract. The least-squares method can be used to approximate an eigenvector for a matrix when only an approximation is known for the corresponding eigenvalue. In this paper, this technique is analyzed and error estimates are established proving that if the error in the eigenvalue is sufficiently small, then the error in the approximate eigenvector produced by the least-squares method is also small. Also reported are some empirical results based on using the algorithm.

Key words. Least squares, Approximate eigenvector.

AMS subject classifications. 65F15.

1. Notation. We use upper case, bold letters to represent complex matrices, and lower case bold letters to represent vectors in $\mathbb{C}^k$. We consider a vector $\mathbf{v}$ to be a column, and so its adjoint $\mathbf{v}^*$ is a row vector. Hence $\mathbf{v}_1^*\mathbf{v}_2$ yields the complex dot product $\mathbf{v}_2 \cdot \mathbf{v}_1$. The vector $\mathbf{e}_i$ is the vector having 1 in its ith coordinate and 0 elsewhere, and $\mathbf{I}_n$ is the $n \times n$ identity matrix. We use $\|\mathbf{v}\|$ to represent the 2-norm on vectors; that is $\|\mathbf{v}\|^2 = \mathbf{v}^*\mathbf{v}$. Also, $|||\mathbf{F}|||$ represents the spectral matrix norm of a square matrix $\mathbf{F}$, and so $\|\mathbf{Fv}\| \leq |||\mathbf{F}|||\ \|\mathbf{v}\|$ for every vector $\mathbf{v}$. Finally, for an $n \times n$ Hermitian matrix $\mathbf{F}$, we will write each of the n (not necessarily distinct) real eigenvalues for $\mathbf{F}$ as $\lambda_i(\mathbf{F})$, where $\lambda_1(\mathbf{F}) \leq \lambda_2(\mathbf{F}) \leq \cdots \leq \lambda_n(\mathbf{F})$.

2. The Method and Our Goal. Suppose $\mathbf{M}$ is an arbitrary $n\times n$ matrix having λ as an eigenvalue, and let $\mathbf{A} = \lambda\mathbf{I}_n - \mathbf{M}$. Generally, one can find an eigenvector for $\mathbf{M}$ corresponding to λ by solving the homogeneous system $\mathbf{Ax} = \mathbf{0}$. However, the computation of an eigenvalue does not always result in an exact answer, either because a numerical technique was used for its computation, or due to roundoff error. Suppose λ' is the approximate, known value for the actual eigenvalue λ. If $\lambda \neq \lambda'$, then the known matrix $\mathbf{K} = \lambda'\mathbf{I}_n - \mathbf{M}$ is most likely nonsingular, and so the homogeneous system $\mathbf{Kx} = \mathbf{0}$ has only the trivial solution. This situation occurs frequently when attempting to solve small eigenvalue problems on calculators.

Let $\epsilon = \lambda' - \lambda$. Then $\mathbf{K} = \mathbf{A} + \epsilon\mathbf{I}_n$. Our goal is to approximate a vector in the kernel of $\mathbf{A}$ when only the matrix $\mathbf{K}$ is known. We assume that $\mathbf{M}$ has no other eigenvalues within $|\epsilon|$ units of λ, so that $\mathbf{K}$ is nonsingular, and thus has trivial kernel. Let $\mathbf{u}$ be a unit vector in $\ker(\mathbf{A})$. Although we know that $\mathbf{u}$ exists, $\mathbf{u}$ is unknown. Let $\mathbf{v}$ be an arbitrarily chosen unit vector in $\mathbb{C}^n$ such that $w = \mathbf{v}^*\mathbf{u} \neq 0$. In practice, when choosing $\mathbf{v}$, the value of w is unknown, but if $\mathbf{v}$ is chosen at random, the probability that $w = 0$ is zero. Let $\mathbf{B}$ be the $(n+1) \times n$ matrix formed by appending the row $\mathbf{v}^*$

*Received by the editors 21 March 2005. Accepted for publication 19 March 2007. Handling Editor: Michael Neumann.

†Department of Mathematics and Computer Science, Saint Joseph's University, 5600 City Avenue, Philadelphia, Pa. 19131 (dhecker@sju.edu, lurie@sju.edu).

Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 16, pp. 99-110, March 2007

ELA
http://math.technion.ac.il/iic/ela

to $\mathbf{K}$. Written in block form, $\mathbf{B} = \begin{bmatrix} \mathbf{K} \\ \mathbf{v}^* \end{bmatrix}$. The system $\mathbf{Bx} = \mathbf{e}_{n+1}$ is inconsistent, since any solution must satisfy both $\mathbf{Kx} = \mathbf{0}$ and $\mathbf{v}^*\mathbf{x} = 1$, which is impossible since $\mathbf{Kx} = \mathbf{0}$ has only the trivial solution. We apply the least-squares method to $\mathbf{Bx} = \mathbf{e}_{n+1}$ obtaining a vector $\mathbf{y}$ such that $\mathbf{By}$ is as close to $\mathbf{e}_{n+1}$ as possible. Then, we normalize $\mathbf{y}$ producing $\mathbf{s} = \mathbf{y}/\left\|\mathbf{y}\right\|$, an approximation for a unit vector in $\ker(\mathbf{A})$. This is essentially the technique for approximating an eigenvector described, but not verified, in [1]. Our goal in this paper is to find constants N_1 and N_2, each independent of ϵ, and a unit vector $\mathbf{u}' \in \ker(\mathbf{A})$, such that

$$\|\mathbf{As}\| \leq N_1|\epsilon| \text{ and} \tag{2.1}$$
$$\|\mathbf{u}' - \mathbf{s}\| \leq N_2|\epsilon|. \tag{2.2}$$

Note that the unit vector $\mathbf{u}'$ might depend upon $\mathbf{s}$, which is dependent on ϵ. These inequalities will show that as $\epsilon \to 0$, $\mathbf{As} \to \mathbf{0}$ and $\mathbf{s}$ gets close to $\ker(\mathbf{A})$. Although (2.1) clearly follows from (2.2) by continuity, we need to prove (2.1) first since that inequality is used in the proof of (2.2).

3. Proving Estimate (2.1). The method of least-squares is based upon the following well-known theorem [1]:

THEOREM 3.1. *Let* $\mathbf{F}$ *be an* $m \times n$ *matrix, let* $\mathbf{q} \in \mathbb{C}^m$*, and let* $\mathcal{W}$ *be the subspace* $\{\mathbf{Fx} \mid \mathbf{x} \in \mathbb{C}^n\}$. *Then the following three conditions on a vector* $\mathbf{y}$ *are equivalent:*

(i) $\mathbf{Fy} = \mathbf{proj}_{\mathcal{W}}\mathbf{q}$
(ii) $\|\mathbf{Fy} - \mathbf{q}\| \leq \|\mathbf{Fz} - \mathbf{q}\|$ *for all* $\mathbf{z} \in \mathbb{C}^n$
(iii) $(\mathbf{F}^*\mathbf{F})\mathbf{y} = \mathbf{F}^*\mathbf{q}$

So, given the inconsistent system $\mathbf{Bx} = \mathbf{e}_{n+1}$, parts (i) and (iii) of Theorem 3.1 show that the system $\mathbf{B}^*\mathbf{Bx} = \mathbf{B}^*\mathbf{e}_{n+1}$ is consistent. Let $\mathbf{y}$ be a solution to this system. With $\mathbf{s} = \mathbf{y}/\left\|\mathbf{y}\right\|$, we use the properties of $\mathbf{y}$ from part (ii) to prove inequalities (2.1) and (2.2).[1]

First, using block notation,

$$\begin{aligned}
\left\|\mathbf{B}\left(\frac{\mathbf{u}}{w}\right) - \mathbf{e}_{n+1}\right\| &= \left\|\begin{bmatrix} \frac{1}{w}\mathbf{Ku} \\ \frac{1}{w}\mathbf{v}^*\mathbf{u} - 1 \end{bmatrix}\right\| = \sqrt{\frac{1}{|w|^2}\left\|\mathbf{Ku}\right\|^2 + \left|\frac{1}{w}w - 1\right|^2} \\
&= \frac{1}{|w|}\left\|\mathbf{Ku}\right\| = \frac{1}{|w|}\left\|(\mathbf{A} + \epsilon\mathbf{I}_n)\mathbf{u}\right\| \\
&= \frac{1}{|w|}\left\|\mathbf{Au} + \epsilon\mathbf{u}\right\| = \frac{1}{|w|}\left\|\epsilon\mathbf{u}\right\| = \frac{|\epsilon|}{|w|},
\end{aligned}$$

[1] A quick computation shows that $\mathbf{B}^*\mathbf{e}_{n+1} = \mathbf{v}$, and so $\mathbf{y}$ is the solution to a nonhomogeneous system. Therefore $\mathbf{y} \neq \mathbf{0}$, and $\mathbf{y}$ can be normalized.

Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 16, pp. 99-110, March 2007

since $\mathbf{u}$ is a unit vector in $\ker(\mathbf{A})$. Hence, by part (ii) of Theorem 3.1 with $\mathbf{z} = \frac{\mathbf{u}}{w}$,

$$\begin{aligned}\frac{|\epsilon|}{|w|} &\geq \|\mathbf{B}\mathbf{y} - \mathbf{e}_{n+1}\| = \left\| \begin{bmatrix} \mathbf{K}\mathbf{y} \\ \mathbf{v}^*\mathbf{y} - 1 \end{bmatrix} \right\| = \sqrt{\|\mathbf{K}\mathbf{y}\|^2 + |\mathbf{v}^*\mathbf{y} - 1|^2} \\ &\geq \|\mathbf{K}\mathbf{y}\| = \|(\mathbf{A} + \epsilon \mathbf{I}_n)\mathbf{y}\| \\ &\geq \Big| \|\mathbf{A}\mathbf{y}\| - |\epsilon|\, \|\mathbf{y}\| \Big|, \text{ by the Reverse Triangle Inequality.}\end{aligned}$$

Now, if $\|\mathbf{A}\mathbf{y}\| - |\epsilon|\, \|\mathbf{y}\| < 0$, then

$$\|\mathbf{A}\mathbf{s}\| = \frac{\|\mathbf{A}\mathbf{y}\|}{\|\mathbf{y}\|} < |\epsilon|,$$

and we have inequality (2.1) with $N_1 = 1$.

But, instead, if $\|\mathbf{A}\mathbf{y}\| - |\epsilon|\, \|\mathbf{y}\| \geq 0$, then

$$\begin{gathered}\frac{|\epsilon|}{|w|} \geq \|\mathbf{A}\mathbf{y}\| - |\epsilon|\, \|\mathbf{y}\|, \text{ implying} \\ \|\mathbf{A}\mathbf{s}\| = \frac{\|\mathbf{A}\mathbf{y}\|}{\|\mathbf{y}\|} \leq |\epsilon| \left(1 + \frac{1}{|w|\, \|\mathbf{y}\|}\right).\end{gathered}$$

In this case, we would like to set N_1 equal to $1 + \frac{1}{|w|\|\mathbf{y}\|}$. However, $\|\mathbf{y}\|$ depends upon ϵ. Our next goal is to bound $1 + \frac{1}{|w|\|\mathbf{y}\|}$ independent of ϵ.

Now, since $\mathbf{K}$ is nonsingular, $\operatorname{rank}(\mathbf{K}) = n$. Therefore $\operatorname{rank}(\mathbf{B}) = n$. This implies that $\operatorname{rank}(\mathbf{B}^*\mathbf{B}) = n$, and so $\mathbf{B}^*\mathbf{B}$ is nonsingular. Since $\mathbf{B}^*\mathbf{B}$ is Hermitian, there is a unitary matrix $\mathbf{P}$ and a diagonal matrix $\mathbf{D}$ such that $\mathbf{B}^*\mathbf{B} = \mathbf{P}^*\mathbf{D}\mathbf{P}$, where the eigenvalues of $\mathbf{B}^*\mathbf{B}$, which must be real and nonnegative, appear on the main diagonal of $\mathbf{D}$ in increasing order. Also, none of these eigenvalues are zero since $\mathbf{B}^*\mathbf{B}$ is nonsingular.

As noted above, $\mathbf{B}^*\mathbf{e}_{n+1} = \mathbf{v}$. The vector $\mathbf{y}$ is thus defined by the equation $\mathbf{B}^*\mathbf{B}\mathbf{y} = \mathbf{B}^*\mathbf{e}_{n+1} = \mathbf{v}$. Hence, $\mathbf{P}^*\mathbf{D}\mathbf{P}\mathbf{y} = \mathbf{v}$, or $\mathbf{P}\mathbf{y} = \mathbf{D}^{-1}\mathbf{P}\mathbf{v}$. Therefore,

$$\begin{aligned}\|\mathbf{y}\| = \|\mathbf{P}\mathbf{y}\| &= \left\|\mathbf{D}^{-1}\mathbf{P}\mathbf{v}\right\| \geq \min_{\|\mathbf{x}\|=1} \left\|\mathbf{D}^{-1}\mathbf{P}\mathbf{x}\right\| \\ &= \min_{\|\mathbf{x}\|=1} \left\|\mathbf{D}^{-1}\mathbf{x}\right\| = \left\|\mathbf{D}^{-1}\mathbf{e}_n\right\| = \frac{1}{\lambda_n(\mathbf{B}^*\mathbf{B})}.\end{aligned}$$

And so,

$$\begin{gathered}\frac{1}{\|\mathbf{y}\|} \leq \lambda_n(\mathbf{B}^*\mathbf{B}), \text{ implying} \\ 1 + \frac{1}{|w|\, \|\mathbf{y}\|} \leq 1 + \frac{\lambda_n(\mathbf{B}^*\mathbf{B})}{|w|}.\end{gathered}$$

Our next step is to relate $\lambda_n(\mathbf{B}^*\mathbf{B})$ to $\lambda_n(\mathbf{K}^*\mathbf{K})$. Now,

$$\mathbf{B}^*\mathbf{B} = \begin{bmatrix} \mathbf{K}^* & \mathbf{v} \end{bmatrix} \begin{bmatrix} \mathbf{K} \\ \mathbf{v}^* \end{bmatrix} = \mathbf{K}^*\mathbf{K} + \mathbf{v}\mathbf{v}^*.$$

Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 16, pp. 99-110, March 2007

ELA
http://math.technion.ac.il/iic/ela

Because $\mathbf{v}$ is a unit vector, $\mathbf{vv}^*$ is the matrix for the orthogonal projection onto the subspace spanned by $\mathbf{v}$. Hence, $\lambda_i(\mathbf{vv}^*) = 0$ for $i < n$, and $\lambda_n(\mathbf{vv}^*) = 1$. Therefore, by Weyl's Theorem [2], $\lambda_n(\mathbf{B}^*\mathbf{B}) \le \lambda_n(\mathbf{K}^*\mathbf{K}) + \lambda_n(\mathbf{vv}^*) = \lambda_n(\mathbf{K}^*\mathbf{K}) + 1$.

Now $\mathbf{K} = \mathbf{A} + \epsilon\mathbf{I}_n$, and so $\mathbf{K}^*\mathbf{K} = (\mathbf{A}^* + \bar{\epsilon}\mathbf{I}_n)(\mathbf{A} + \epsilon\mathbf{I}_n) = \mathbf{A}^*\mathbf{A} + (\bar{\epsilon}\mathbf{A} + \epsilon\mathbf{A}^*) + |\epsilon|^2\mathbf{I}_n$. But $\mathbf{A}^*\mathbf{A}$, $(\bar{\epsilon}\mathbf{A} + \epsilon\mathbf{A}^*)$, and $|\epsilon|^2\mathbf{I}_n$ are all Hermitian matrices. Hence, Weyl's Theorem implies that $\lambda_n(\mathbf{K}^*\mathbf{K}) \le \lambda_n(\mathbf{A}^*\mathbf{A}) + \lambda_n(\bar{\epsilon}\mathbf{A} + \epsilon\mathbf{A}^*) + |\epsilon|^2$. Since we are only interested in small values of ϵ, we can assume that there is some bound C such that $|\epsilon| \le C$. Therefore, $\lambda_n(\mathbf{K}^*\mathbf{K}) \le \lambda_n(\mathbf{A}^*\mathbf{A}) + \lambda_n(\bar{\epsilon}\mathbf{A} + \epsilon\mathbf{A}^*) + C^2$.

Next, we need to compute a bound on $\lambda_n(\bar{\epsilon}\mathbf{A} + \epsilon\mathbf{A}^*)$ that is independent of ϵ. Suppose $p_\epsilon(z)$ is the characteristic polynomial of $\bar{\epsilon}\mathbf{A} + \epsilon\mathbf{A}^*$ and $a_i(\epsilon)$ is the coefficient of z^i in $p_\epsilon(z)$. The coefficients of the characteristic polynomial of a matrix are a sum of products of entries of the matrix, so $a_i(\epsilon)$ is a polynomial in ϵ and $\bar{\epsilon}$. Therefore, $|a_i(\epsilon)|$ attains its maximum on the compact set $|\epsilon| \le C$. Let m_i be this maximum value. Since $\lambda_n(\bar{\epsilon}\mathbf{A} + \epsilon\mathbf{A}^*)$ is a root of $p_\epsilon(z)$, Cauchy's bound [2] implies that

$$\begin{aligned} |\lambda_n(\bar{\epsilon}\mathbf{A} + \epsilon\mathbf{A}^*)| &\le 1 + \max\{|a_0(\epsilon)|, |a_1(\epsilon)|, \ldots, |a_{n-1}(\epsilon)|\} \\ &\le 1 + \max\{m_0, m_1, \ldots m_{m-1}\}. \end{aligned}$$

Hence, $\lambda_n(\mathbf{K}^*\mathbf{K}) \le \lambda_n(\mathbf{A}^*\mathbf{A}) + 1 + \max\{m_0, m_1, \ldots m_{m-1}\} + C^2$, which is independent of ϵ.

Finally, we let

$$N_1 = 1 + \frac{2 + \lambda_n(\mathbf{A}^*\mathbf{A}) + \max\{m_0, m_1, \ldots m_{m-1}\} + C^2}{|w|}.$$

Our argument so far shows that $\|\mathbf{As}\| \le N_1|\epsilon|$, completing the proof of (2.1).[2]

4. Proving Estimate (2.2). Next, we find $\mathbf{u}'$ and N_2 that satisfy inequality (2.2). Since $\mathbf{A}^*\mathbf{A}$ is Hermitian, there is a unitary matrix $\mathbf{Q}$ such that $\mathbf{A}^*\mathbf{A} = \mathbf{Q}^*\mathbf{HQ}$, where $\mathbf{H}$ is a diagonal matrix whose main diagonal entries $h_1, h_2, \ldots, h_n$ are the real, nonnegative eigenvalues of $\mathbf{A}^*\mathbf{A}$ in increasing order. Let $l = \dim(\ker(\mathbf{A}^*\mathbf{A})) = \dim(\ker(\mathbf{A})) > 0$. Thus, $h_1 = h_2 = \cdots = h_l = 0$, and $h_{l+1} > 0$. Let $\mathbf{t} = \mathbf{Qs}$, with coordinates $t_1, \ldots t_n$. Note that $\|\mathbf{t}\| = \|\mathbf{Qs}\| = \|\mathbf{s}\| = 1$. Let

$$\mathbf{t}_\alpha = \begin{bmatrix} t_1 \\ \vdots \\ t_l \end{bmatrix}, \quad \mathbf{t}_\beta = \begin{bmatrix} t_{l+1} \\ \vdots \\ t_n \end{bmatrix}, \text{ in which case } \mathbf{t} = \begin{bmatrix} \mathbf{t}_\alpha \\ \mathbf{t}_\beta \end{bmatrix},$$

where we have written $\mathbf{t}$ in block form. Using this notation, $1 = \|\mathbf{t}\|^2 = \|\mathbf{t}_\alpha\|^2 + \|\mathbf{t}_\beta\|^2$. Note that $\begin{bmatrix} \mathbf{t}_\alpha \\ \mathbf{0} \end{bmatrix}$ is essentially the projection of $\mathbf{s}$ onto $\ker(\mathbf{A}^*\mathbf{A}) = \ker(\mathbf{A})$, expressed in the coordinates that diagonalizes $\mathbf{A}^*\mathbf{A}$.

[2]Technically, there are two cases in the proof. In the first case, we obtained $N_1 = 1$. However, the expression for N_1 in Case 2 is always larger than 1, allowing us to use that expression for both cases.

Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 16, pp. 99-110, March 2007

ELA
http://math.technion.ac.il/iic/ela

First, we claim that for $|\epsilon|$ sufficiently small, $\|\mathbf{t}_\alpha\| \neq 0$. We prove this by showing that $\|\mathbf{t}_\beta\| < 1$.

Assume that $\mathbf{A} \neq \mathbf{O}$, so that $|||\mathbf{A}^*||| \neq 0$. (If $\mathbf{A} = \mathbf{O}$, the entire problem is trivial.) In addition, suppose that $|\epsilon| < \frac{h_{l+1}}{N_1\,|||\mathbf{A}^*|||}$. Then,

$$\begin{aligned}|||\mathbf{A}^*|||\,N_1\,|\epsilon| &\geq |||\mathbf{A}^*|||\,\|\mathbf{As}\| \geq \|\mathbf{A}^*\mathbf{As}\| = \|\mathbf{Q}^*\mathbf{HQs}\| = \|\mathbf{HQs}\| \\ &= \|\mathbf{Ht}\| = \sqrt{\sum_{i=1}^{n}|h_i t_i|^2} = \sqrt{\sum_{i=l+1}^{n}|h_i t_i|^2} \\ &\geq h_{l+1}\sqrt{\sum_{i=l+1}^{n}|t_i|^2} = h_{l+1}\,\|\mathbf{t}_\beta\|\,.\end{aligned}$$

Therefore,

$$\|\mathbf{t}_\beta\| \leq \frac{|||\mathbf{A}^*|||\,N_1|\epsilon|}{h_{l+1}} < \frac{|||\mathbf{A}^*|||\,N_1}{h_{l+1}}\frac{h_{l+1}}{N_1\,|||\mathbf{A}^*|||} = 1,$$

completing the proof that $\|\mathbf{t}_\alpha\| > 0$.

Next, we find $\mathbf{u}' \in \ker(\mathbf{A})$ that is close to $\mathbf{s}$. Since $\|\mathbf{t}_\alpha\| > 0$, we can define $\mathbf{z} = \frac{1}{\|\mathbf{t}_\alpha\|}\sum_{i=1}^{l} t_i\mathbf{e}_i = \begin{bmatrix}\frac{1}{\|\mathbf{t}_\alpha\|}\mathbf{t}_\alpha \\ \mathbf{0}\end{bmatrix}$, and let $\mathbf{u}' = \mathbf{Q}^*\mathbf{z}$. Note that $\|\mathbf{u}'\| = \|\mathbf{z}\| = 1$. Now,

$$\mathbf{A}^*\mathbf{Au}' = (\mathbf{Q}^*\mathbf{HQ})(\mathbf{Q}^*\mathbf{z}) = \mathbf{Q}^*\mathbf{Hz} = \mathbf{Q}^*\mathbf{0} = \mathbf{0},$$

and so $\mathbf{u}' \in \ker(\mathbf{A}^*\mathbf{A}) = \ker(\mathbf{A})$. But

$$\begin{aligned}\|\mathbf{u}'-\mathbf{s}\|^2 &= \|\mathbf{Q}\,(\mathbf{u}'-\mathbf{s})\|^2 = \left\|\mathbf{Qu}'-\mathbf{Qs}\right\|^2 = \|\mathbf{QQ}^*\mathbf{z}-\mathbf{t}\|^2 = \|\mathbf{z}-\mathbf{t}\|^2 \\ &= \left\|\begin{bmatrix}\frac{1}{\|\mathbf{t}_\alpha\|}\mathbf{t}_\alpha - \mathbf{t}_\alpha \\ -\mathbf{t}_\beta\end{bmatrix}\right\|^2 = \sum_{i=1}^{l}|t_i|^2\left(\frac{1}{\|\mathbf{t}_\alpha\|}-1\right)^2 + \sum_{i=l+1}^{n}|t_i|^2 \\ &= \sum_{i=1}^{l}|t_i|^2\left(\frac{1}{\|\mathbf{t}_\alpha\|^2}-\frac{2}{\|\mathbf{t}_\alpha\|}+1\right) - \sum_{i=1}^{l}|t_i|^2 + \sum_{i=1}^{n}|t_i|^2 \\ &= \left(\frac{1}{\|\mathbf{t}_\alpha\|^2}-\frac{2}{\|\mathbf{t}_\alpha\|}\right)\left(\sum_{i=1}^{l}|t_i|^2\right) + 1 \qquad \text{since } \sum_{i=1}^{n}|t_i|^2 = \|\mathbf{t}\|^2 = 1 \\ &= \left(\frac{1}{\|\mathbf{t}_\alpha\|^2}-\frac{2}{\|\mathbf{t}_\alpha\|}\right)\|\mathbf{t}_\alpha\|^2 + 1 = 2 - 2\,\|\mathbf{t}_\alpha\| \\ &\leq 2 - 2\,\|\mathbf{t}_\alpha\|^2 = 2(1-\|\mathbf{t}_\alpha\|^2) = 2\,\|\mathbf{t}_\beta\|^2\,.\end{aligned}$$

Hence, $\|\mathbf{u}'-\mathbf{s}\| \leq \sqrt{2}\,\|\mathbf{t}_\beta\|$. Next,

$$\|\mathbf{Ht}\| = \sqrt{\sum_{i=1}^{n}|h_i t_i|^2} = \sqrt{\sum_{i=l+1}^{n}{h_i}^2|t_i|^2} \;\geq\; h_{l+1}\sqrt{\sum_{i=l+1}^{n}|t_i|^2} = h_{l+1}\,\|\mathbf{t}_\beta\|\,.$$

Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 16, pp. 99-110, March 2007

ELA
http://math.technion.ac.il/iic/ela

But,

$$\|\mathbf{Ht}\| = \|\mathbf{Q}^*\mathbf{Ht}\| = \|\mathbf{Q}^*\mathbf{HQs}\| = \|\mathbf{A}^*\mathbf{As}\| \leq |||\mathbf{A}^*|||\ \|\mathbf{As}\| \\ \leq |||\mathbf{A}^*|||\ N_1|\epsilon|.$$

Putting this all together, we let

$$N_2 = \frac{\sqrt{2}\,|||\mathbf{A}^*|||\,N_1}{h_{l+1}}.$$

Then,

$$N_2|\epsilon| = \frac{\sqrt{2}\,|||\mathbf{A}^*|||\,N_1\,|\epsilon|}{h_{l+1}} \geq \frac{\sqrt{2}}{h_{l+1}}\,\|\mathbf{Ht}\| \geq \frac{\sqrt{2}}{h_{l+1}}\,(h_{l+1}\,\|\mathbf{t}_\beta\|) \\ = \sqrt{2}\,\|\mathbf{t}_\beta\| \geq \|\mathbf{u}' - \mathbf{s}\|\,,$$

thus proving inequality (2.2).

5. Further Observations. The formulas we present for N_1 and N_2 are derived from worst-case scenarios, not all of which should be expected to occur simultaneously. Also, the constants N_1 and N_2 depend upon various other fixed, but unknown, values, such as $w = \mathbf{v}^*\mathbf{u}$ and $|||\mathbf{A}^*|||$. However, they are still useful, since they demonstrate that, so long as $\mathbf{v}$ is chosen such that $\mathbf{v}^*\mathbf{u} \neq 0$, the least-squares technique works; that is, it will produce a vector close to an actual eigenvector (provided ϵ is sufficiently small). Of course, if one happens to choose $\mathbf{v}$ so that $\mathbf{v}^*\mathbf{u} = 0$ (which is highly unlikely) and the method fails, one could just try again with a new randomly chosen unit vector $\mathbf{v}$.

In a particular case, we might want good estimates for $\|\mathbf{As}\|$ and $\|\mathbf{u}' - \mathbf{s}\|$. Using $\mathbf{A} = \mathbf{K} - \epsilon\mathbf{I}_n$ and the triangle inequality produces

$$\|\mathbf{As}\| \leq \|\mathbf{Ks}\| + |\epsilon|\,.$$

Hence, $\|\mathbf{As}\|$ can be estimated just by computing $\|\mathbf{Ks}\|$ after $\mathbf{s}$ has been found. (One usually has a good idea of an upper bound on $|\epsilon|$ based on the method used to find the approximate eigenvalue.) Similarly, tracing through the proof of estimate (2.2), it can be seen that

$$\|\mathbf{u}' - \mathbf{s}\| \leq \frac{\sqrt{2}\ (|||\mathbf{K}||| + |\epsilon|)\ (\|\mathbf{Ks}\| + |\epsilon|)}{h_{l+1}}.$$

Applying Weyl's Theorem shows that

$$h_{l+1} \geq \lambda_{l+1}(\mathbf{K}^*\mathbf{K}) + \lambda_1\,(-\epsilon\mathbf{K}^* - \bar{\epsilon}\mathbf{K}) + |\epsilon|^2,$$

yielding an estimate for $\|\mathbf{u}' - \mathbf{s}\|$ in terms of (hopefully) computable quantities.

It should also be noted that this technique has the advantage that if there is no error in the eigenvalue, that is if $\epsilon = 0$, then the method produces an exact eigenvector, assuming that no further roundoff errors occur.

Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 16, pp. 99-110, March 2007

ELA
http://math.technion.ac.il/iic/ela

6. Empirical Results. We wrote a set of five computer simulation programs in Pascal to test the algorithm for several hundred thousand matrices. Our goal was to empirically answer several questions:

1. How well does the algorithm work?
2. What effect does the choice of the random vector $\mathbf{v}$ have on the algorithm?
3. What happens as the values of $\mathbf{v}^*\mathbf{u}$ approach zero?
4. In the general case, how often does the value of $\mathbf{v}^*\mathbf{u}$ get close to zero?
5. How does the size of the closest eigenvalue to λ affect the error?
6. How does the algorithm behave as the size of the matrix $\mathbf{M}$ increases?
7. Can the algorithm be used to find a second vector for a two-dimensional eigenspace?

Let us first describe the general method used to generate the matrices to be tested. Since we expect the algorithm to be most useful for small matrices, we restricted ourselves to considering $n \times n$ matrices with $3 \leq n \leq 9$. For simplicity, we used $\lambda = 0$. In four of the programs, the dimension of the eigenspace for $\lambda = 0$ was 1, in the fifth we specifically made it 2 in order to answer Question #7. In order to get a variety of types of matrices, the programs generated matrices with every possible set of patterns of Jordan block sizes for the Jordan canonical form of the matrix (for example the pattern $1, 1, 2, 2$ for a 6×6 matrix represents a 1×1 block for the eigenvalue 0, and then, for other eigenvalues, a 1×1 block, and two 2×2 blocks). The other eigenvalues were chosen using a pseudo-random number generator that created complex numbers whose real and imaginary parts each ranged from -10 to 10. The Jordan matrix was created, and then we conjugated it with a matrix of random complex numbers (generated by the same random number generator). In this way, the program knew all of the actual eigenvalues and corresponding eigenvectors. The number of different matrices generated for each possible block structure ranged from 100 to 5000, depending upon the particular program being run. Our method of generating matrices introduced an eighth question:

8. How does the Jordan block pattern of the matrix effect the error?

Next, in all cases, we used $\lambda' = 0.001$ as the estimate for the actual eigenvalue 0. Although in practice, the error in λ' will be much smaller since it is typically caused by round-off error in some other computation, we used this larger value for two reasons. First, we wanted to be sure that error caused by the error in λ' would dominate natural round-off error in our computer programs, and second, we wanted to allow for $\mathbf{v}^*\mathbf{u}$ to be significantly smaller than the error in λ'.[3]

The random vector $\mathbf{v}$ used in the algorithm was created by merely generating n random complex numbers for its entries, and then normalizing the result to obtain a unit vector.

We measured the error in the approximate unit eigenvector $\mathbf{s}$ by first projecting $\mathbf{s}$ onto the actual eigenspace for $\lambda = 0$, normalizing this result to get a unit eigenvector $\mathbf{u}'$, and then computing $\|\mathbf{u}' - \mathbf{s}\|$. If the magnitude of the error in each coordinate of

[3]We also tried several runs of the programs with smaller values of λ'. Although we did not do a detailed analysis of this data, the final error, for the same matrices, seemed to be proportional to the size of λ'.

Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 16, pp. 99-110, March 2007

$\mathbf{s}$ is about 0.001 (the error in λ), then the value of $\|\mathbf{u}' - \mathbf{s}\|$ would be about $0.001\sqrt{n}$. Thus, we used this level of error as our standard for considering the algorithm to have successfully found an approximate eigenvector. We deemed the error to be large if it exceeded $0.001\sqrt{n}$. We also recorded the absolute value of the eigenvalue closest to zero, in order to answer Question #5.

Let us consider the results of our simulations:

1. How well does the algorithm work? (Simulation 1)

When the algorithm was used on 325,000 matrices of varying sizes and Jordan block patterns with a randomly generated vector $\mathbf{v}$, the error exceeded $0.001\sqrt{n}$ in only 0.86% (2809/325000) of the matrices. The error had a strong negative correlation (-0.7) with the size of the next smallest eigenvalue, indicating a strong inverse relationship between the size of this eigenvalue and error in the algorithm. This is exactly what we should expect, considering that h_{l+1} appears in the denominator of N_2 in theoretical error estimate (2.2). The mean value of the size of the smallest eigenvalue when the error exceeded $0.001\sqrt{n}$ was less than 0.7, as compared to a mean of over 4 for the other trials. (This answers Question #5.) No correlation was found between the error and the absolute value of $\mathbf{v}^*\mathbf{u}$ for this general case. In this simulation, the smallest value $|\mathbf{v}^*\mathbf{u}|$ observed was 0.0133.

2. What effect does the choice of the random vector $\mathbf{v}$ have on the algorithm? (Simulation 2)

The goal of this simulation was to evaluate the variability in the error in $\mathbf{s}$ if the initial random vector $\mathbf{v}$ is varied. For each of 13,000 matrices, 200 different random vectors $\mathbf{v}$ were selected and the algorithm was executed, for a total of 2,600,000 trials. In general, for each of the 13,000 matrices, there was very little variation in the error in $\mathbf{s}$ over the 200 vectors for $\mathbf{v}$. The range of errors in a given matrix was as small as 9.0×10^{-12} and as large as 0.0225. The average standard deviation of error per matrix was at most 1.45×10^{-5}. Since simulation 2 had more trials than simulation 1, the minimum value of $|\mathbf{v}^*\mathbf{u}|$ in simulation 2 (0.00001) was smaller than in simulation 1 (reported above). However, the percent of trials in which the error exceeded $0.001\sqrt{n}$ was still only 0.90%. Again, the large errors were associated with low values for the size of the smallest eigenvalue.

3. What happens as the values of $\mathbf{v}^*\mathbf{u}$ approach zero? (Simulation 3)

In this simulation, for each of 6,500 matrices, 200 vectors $\mathbf{v}$ were specifically selected so that $|\mathbf{v}^*\mathbf{u}|$ approached zero. Hence, $\mathbf{v}$ was not truly random. (This yields a total of 1,300,000 trials.) The value of $|\mathbf{v}^*\mathbf{u}|$ ranged from 1.39×10^{-17} to 0.0969. Errors in $\mathbf{s}$ that exceeded $0.001\sqrt{n}$ were observed in 3.48% of the trials (as compared to 0.86% and 0.90% in simulations 1 and 2). The trials were grouped by value of $|\mathbf{v}^*\mathbf{u}|$ into 10 intervals, labeled 1 through 10 in Figure 6.1: $(0, 10^{-14})$, $(10^{-14}, 10^{-12})$, $(10^{-12}, 10^{-10})$, $(10^{-10}, 10^{-8})$, $(10^{-8}, 10^{-6})$, $(10^{-6}, 10^{-5})$, $(10^{-5}, 10^{-4})$, $(10^{-4}, 10^{-3})$, $(10^{-3}, 10^{-2})$, $(10^{-2}, 1.0)$. The percent of large errors is approximately 1% for $|\mathbf{v}^*\mathbf{u}|$ between 0.001 and 1. The rate increases to 2.3% for values between .0001 and .001 and then to 4% for values between .0001 and .00001. The rate ranges between 4.18% and 4.25% as $|\mathbf{v}^*\mathbf{u}|$ decreases toward 0.

Although the error rate increased in this simulation, the errors are not excessive. The error exceeded $0.01\sqrt{n}$ (ten times higher) 0.23% of the time. The error rates

Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 16, pp. 99-110, March 2007

ELA
http://math.technion.ac.il/iic/ela

per interval of values for $|\mathbf{v}^*\mathbf{u}|$ are graphed in Figure 6.1.

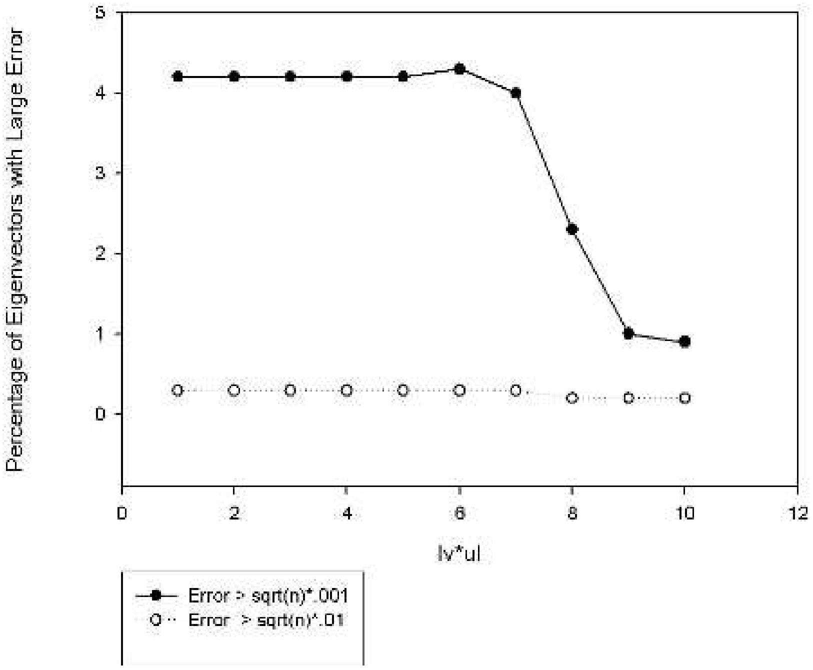

FIG. 6.1. *A comparison of the error rates by* $|\mathbf{v}^*\mathbf{u}|$

4. In the general case, how often does the value of $\mathbf{v}^*\mathbf{u}$ get close to zero?

A value for $|\mathbf{v}^*\mathbf{u}|$ less than 0.001 (Categories 9 and 10, above) never occurred in simulation 1 and was observed in only 20 cases (out of 2,600,000) in simulation 2. Thus, we conclude that if the vector $\mathbf{v}$ is truly chosen at random, as in simulations 1 and 2, the value of $|\mathbf{v}^*\mathbf{u}|$ is rarely an issue. The effect of a small value of $|\mathbf{v}^*\mathbf{u}|$ causing large error was really only seen in simulation 3, in which $\mathbf{v}$ was not chosen randomly, but rather $|\mathbf{v}^*\mathbf{u}|$ was forced to approach zero.

5. How does the size of the closest eigenvalue to λ effect the error?

This question was answered by simulation 1, in which we found a strong inverse relationship between the size of this eigenvalue and error in the algorithm, as was expected.

6. How does the algorithm behave as the size of the matrix $\mathbf{M}$ increases?

As the size of the matrix increases, there are typically more eigenvalues corresponding to each matrix. Therefore, since these eigenvalues were chosen randomly, the likelihood of getting an alternate eigenvalue that is close to λ increases. For example, in simulation 1 we saw the minimum value for the size of the smallest eigenvalue decrease slightly as the size of the matrix increased. Correspondingly, we also saw a

Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 16, pp. 99-110, March 2007

ELA
http://math.technion.ac.il/iic/ela

slight increase in the percentage of cases with large error as n increases. Because of the strong inverse relationship between error and the size of the closest eigenvalue, we suspect that this small increase in error is mostly an effect of the random generation method used for choosing the matrices.

7. Can the algorithm be used to find a second vector for a two-dimensional eigenspace? (Simulation 4)

In this simulation, the matrices ranged from 4×4 to 9×9 in size, and we generated 5000 matrices for each possible Jordan block pattern having at least two Jordan blocks, for a total of 895,000 matrices. Two of the blocks in each matrix were assigned to the eigenvalue $\lambda = 0$. Thus, in some cases, zero was the only actual eigenvalue. Eigenvalues corresponding to the other blocks were assigned at random. For each matrix we computed the following:

- The size of the smallest (nonzero) eigenvalue, if it existed.
- An approximate unit eigenvector $\mathbf{u}_1$ and the error from its projection onto the actual eigenspace for $\lambda = 0$.
- The norm of $\mathbf{Au}_1$, to help test theoretical Estimate (2.1).
- A second approximate unit eigenvector, $\mathbf{u}_2$, found by choosing a second random vector $\mathbf{v}$ in the algorithm.
- The error for $\mathbf{u}_2$ and the value of $\|\mathbf{Au}_2\|$.
- The length of the projection of $\mathbf{u}_2$ onto $\mathbf{u}_1$, to measure how different the two approximate eigenvectors were; that is, how close is $\mathbf{u}_2$ to being a scalar multiple of $\mathbf{u}_1$? We considered a value > 0.9 to be unacceptable, and between 0.8 and 0.9 to be moderately acceptable, but not desirable.
- A unit vector $\mathbf{u}_3$ found by normalizing the component of $\mathbf{u}_2$ orthogonal to $\mathbf{u}_1$. (So $\mathbf{u}_1$ and $\mathbf{u}_3$ would form an orthonormal basis (approximately) for the two-dimensional eigenspace.)
- The error for $\mathbf{u}_3$ and the value of $\|\mathbf{Au}_3\|$.
- Another approximate unit eigenvector, $\mathbf{u}_4$, found by choosing the random vector $\mathbf{v}$ in the algorithm to be orthogonal to the vector $\mathbf{u}_1$, thus making it less likely for $\mathbf{u}_4$ to be a scalar multiple of $\mathbf{u}_1$.
- The error for $\mathbf{u}_4$ and the value of $\|\mathbf{Au}_4\|$.
- The length of the projection of $\mathbf{u}_4$ onto $\mathbf{u}_1$.
- A unit vector $\mathbf{u}_5$ found by normalizing the component of $\mathbf{u}_4$ orthogonal to $\mathbf{u}_1$.
- The error for $\mathbf{u}_5$ and the value of $\|\mathbf{Au}_5\|$.

These were our results:

- In assessing $\mathbf{u}_1$, the error exceeded $0.001\sqrt{n}$ in only 0.2% (1800) of the 895,000 matrices. The value of $\|\mathbf{Au}_1\|$ was always within the acceptable range. Similarly, for $\mathbf{u}_2$, the error exceeded $0.001\sqrt{n}$ in 0.104% (933) of the matrices. The value of $\|\mathbf{Au}_2\|$ was also always within the acceptable range. Since there is no real difference in the methods used to generate $\mathbf{u}_1$ and $\mathbf{u}_2$, this amounts to 1,790,000 trials of the algorithm with only 2733 cases of large error. This is comparable to the general results for the algorithm found in simulation 1, in which the eigenspace was one-dimensional.
- The absolute value of the projection of $\mathbf{u}_2$ onto $\mathbf{u}_1$ was greater than 0.8 in

Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 16, pp. 99-110, March 2007

ELA
http://math.technion.ac.il/iic/ela

87.6% of the cases, and greater than 0.9 in 81.0% of the cases. Hence, $\mathbf{u}_2$ was frequently close to being a scalar multiple of $\mathbf{u}_1$.

- The error in $\mathbf{u}_3$ exceeded $0.001\sqrt{n}$ in only 0.4% of the matrices (3597/895000). The value of $\|\mathbf{Au}_3\|$ was always within the acceptable range. Thus, while this is still a relatively small percentage of cases, the percentage of large errors almost quadrupled from $\mathbf{u}_2$ to $\mathbf{u}_3$.
- The error in $\mathbf{u}_4$ exceeded $0.001\sqrt{n}$ in 0.24% of the matrices (2111/895000), with $\|\mathbf{Au}_4\|$ always being within the acceptable range. The length of the projection of $\mathbf{u}_4$ onto $\mathbf{u}_1$ exceeded 0.8 in 39.0% of the cases, and exceeded 0.9 in 24.5% of the cases, a large improvement over $\mathbf{u}_2$.
- The vector $\mathbf{u}_5$, orthogonal to $\mathbf{u}_1$, had large error in 0.47% of the cases (4203/895000), and had small error in $\|\mathbf{Au}_5\|$ in all cases.
- We computed the Spearman rank-order correlation between the size of the smallest eigenvalue and the error of each vector. Surprisingly, for matrices 5×5 and larger, there was no correlation found – the correlation coefficients ranged from -0.052 to 0.052. For the 4×4 matrices, there was a low negative relationship. Correlation coefficients with each of the five error terms ranged from -0.314 to -0.231. However, further analysis of the error in $\mathbf{u}_4$ found that for those matrices in which the error in the estimated eigenvector was greater than $0.001\sqrt{n}$, the size of the smallest eigenvalue was less than 1 in 74.5% of the matrices, greater than 1 in 14.7% of the matrices, while 10.8% of these large errors were from matrices in which 0 was the only eigenvalue.

In running this simulation there were 13 cases in which, while computing either $\mathbf{u}_1$, $\mathbf{u}_2$, or $\mathbf{u}_4$, the program reported that the matrix $\mathbf{B}^*\mathbf{B}$ was singular, and so the desired vector could not be computed. (The program actually arbitrarily changed a 0 in a pivot position to a 1 and continued on anyway, with remarkably good results.) Twelve of these 13 cases occurred with matrices in which there were only two Jordan blocks, and so $\lambda = 0$ was the only eigenvalue. The remaining case was a 5×5 matrix with a 1×1 and a 3×3 Jordan block for $\lambda = 0$, and a single 1×1 block for some other eigenvalue (having absolute value about 4.96).

From simulation 4, we came to the conclusion that the algorithm works equally well for cases in which the eigenspace is two-dimensional, and that the preferred method for finding a second eigenvector not parallel to the first is to choose a random vector $\mathbf{v}$ for the algorithm that is orthogonal to the first approximate eigenvector found, as done in the computation of $\mathbf{u}_4$.

8. How does the Jordan block pattern of the matrix effect the error? (Simulation 5)

In this simulation, we allowed the size of the single Jordan block for the eigenvalue $\lambda = 0$ to range from 1 to $(n-1)$. Now, a generic matrix should be diagonalizable, and so the Jordan block pattern will have all 1×1 blocks. We compared the typical error in this generic block pattern with the error observed in other non-generic block patterns. Our results indicated that there was no change in error if non-generic block patterns are used rather than the generic block pattern.

Our conclusion is that these empirical results support the use of the least squares algorithm to find an approximate eigenvector when only an approximation for the

Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 16, pp. 99-110, March 2007

ELA
http://math.technion.ac.il/iic/ela

corresponding eigenvalue is known.

Acknowledgments. We would like to thank William Semus for his helpful work at the beginning of this project computing many examples by hand.

REFERENCES

[1] S. Andrilli and D. Hecker. *Elementary Linear Algebra*, third edition. Elsevier Academic Press, Burlington, 2003.
[2] R. Horn and C.R. Johnson. *Matrix Analysis.* Cambridge University Press, Cambridge, 1985.