2005

TR-2005007: Null Spaces, Eigensystems, and Small-Rank Modifications

Victor Y. Pan

Brian Murphy

Rhys Eric Rosholt

Null Spaces, Eigensystems, and Small-Rank Modifications

Victor Y. Pan, Brian Murphy, Rhys Eric Rosholt,
Department of Mathematics and Computer Science
Lehman College of the City University of New York
Bronx, NY 10468, USA
vpan,bmurphy,rosholt@lehman.cuny.edu
Yuqing Tang and Xiaodong Yan
Ph.D. Programs in Computer Science
The City University of New York
New York, NY 10036 USA
ytang,xyan@gc.cuny.edu

Abstract

We combine the techniques of the perturbation and error analysis, randomization and small-rank modification of the input matrices to accelerate the known numerical algorithms for computing a vector from or a basis for the null space of a matrix and for solving the algebraic eigenproblem, particularly where the input matrix is sparse and/or structured.

Key words: Null spaces, Eigenspaces, Small-rank modifications.

1 Introduction

The inverse power iteration is a classical tool for approximating the solution to the algebraic eigenproblem [W65], [P80], [GL96], [I97], [S98], [BDDRvV00]. The iteration has a more recent block version, called the inverse orthogonal iteration [GL96, page 339] and the inverse Rayleigh–Ritz subspace iteration [S98, Section 6.1]. Hereafter we use the respective abbreviations IPI and IR–RI.

Somewhat counter-intuitively, the IPI produces accurate eigenvectors as the solutions of ill-conditioned linear systems of equations. Ill-conditioning, however, complicates the application of some highly effective algorithms for sparse and/or structured linear systems such as the conjugate gradient algorithm. In [BDDRvV00] the exposition of the inverse power iteration is concluded with the following sentence: "... inverse iteration does require a factorization of the matrix $A - \delta I$, making it less attractive when this factorization is expensive."

Similar comments can be applied to computing a vector from and a basis for the null space of a sparse matrix, which are fundamental computational problems of independent interest.

Small-rank modifications of a matrix are a natural remedy against its ill-conditioning. Their efficiency is obvious for some special, e.g., triangular matrices, but we prove that in combination with randomization and proper scaling they are likely to help for general matrices as well. (The known applications of small-rank modifications to the eigenproblem [G73], [GL96], [S98] are distinct from ours and are limited to the Hermitian or tridiagonal input.)

We also prove theoretically and confirm experimentally that such a remedy implies no side effect of slowing down the convergence of the IPI and the IR–RI. Furthermore, since the modifications enable us to keep the input matrix nonsingular even when it is shifted by the eigenvalue, the perturbation/error analysis and the proof of local quadratic convergence can be simplified.

We organize our paper as follows. After the definitions in Section 2, we apply small-rank modifications to computing the null space vectors in Sections 3–5, to the inverse iteration for the eigenproblems in Section 6, and to expressing the characteristic equations in Section 8. We estimate the perturbations and errors for our modified versions of the inverse iteration in Section 7 and report the results of our numerical tests in Section 9. The tests have been designed by the first author and performed by the other authors. Otherwise the paper (with all missed typos and other errors) is due to the first author.

2 Basic definitions

$A = (a_{i,j})_{i,j=1}^{n}$ is a real or complex $n \times n$ matrix, $\mathbf{v} = (v_i)_{i=1}^{n}$ is a column vector of dimension n. A^T and $\mathbf{v}^T$ are their transposes, A^H and $\mathbf{v}^H$ are their Hermitian (that is, complex conjugate) transposes, $A^H = A^T$ and $\mathbf{v}^H = \mathbf{v}^T$ where A and $\mathbf{v}$ are real. A^{-H} is $(A^{-1})^H = (A^H)^{-1}$. I_k is the $k \times k$ identity matrix, $I = I_n$. $\mathbf{e}_i$ is its i-th column vector. 0 is the null matrix of a proper size. 0_k is the $k \times k$ null matrix. $\mathrm{diag}(A, B)$ is the block diagonal matrix with the diagonal blocks A and B. (A, B) is the 1×2 block matrix with the blocks A and B.

For a matrix A, $\det A$ is its determinant, $\mathrm{rank}\, A$ its rank, and $\mathrm{range}\, A$ its range, that is, the linear space generated by its columns. $LN(A) = \{\mathbf{x} : \mathbf{x}^T A = \mathbf{0}^T\}$ and $RN(A) = \{\mathbf{x} : A\mathbf{x} = \mathbf{0}\}$ are the left and right null spaces of a matrix A, respectively. $\mathrm{Null}\, A = n - \mathrm{rank}\, A$, the nullity of A, is their common dimension.

We extend all the above definitions to the matrix polynomials $A(\lambda) = \sum_{i=0}^{m} A_i \lambda^i$ where $A_0, \ldots, A_m$ are matrices of the same size. The eigenvalues of such a matrix polynomial $A(\lambda)$ of a positive degree m are the roots of the characteristic polynomial $c_A(\lambda) = \det A(\lambda)$. The eigenvalues of a scalar matrix M are the eigenvalues of the linear matrix polynomial $A(\lambda) = \lambda I - M$. The (algebraic) multiplicity $m(\mu)$ of an eigenvalue μ of $A(\lambda)$ is defined as the multiplicity of the root μ of $c_A(\lambda)$. An eigenvalue μ of $A(\lambda)$ is associated with its left and right eigenspaces $LN(A(\mu))$ and $RN(A(\mu))$ and has geometric multiplicity $g.m._A(\mu) = \mathrm{Null}\, A(\mu) \le m(\mu)$. To a fixed set $\{\lambda_1, \ldots, \lambda_h\}$ of the eigenvalues of

$A(\lambda)$ we associate the left and right invariant eigenspaces $LN(A)$ and $RN(A)$ of the matrix $A = \prod_{i=1}^{h} A(\lambda_i)$.

"ops" stands for "arithmetic operations", "IPI" for "inverse power iteration", and "IR–RI" for "inverse Rayleigh–Ritz iteration".

3 Small rank modifications and the null spaces

Theorem 3.1. *Suppose that $A \in \mathbb{C}^{n\times n}$; U, $V \in \mathbb{C}^{n\times k}$, $\operatorname{rank} A = r < n$, and the matrix $S = A + UV^H$ is nonsingular. Then*

$$LN(A) \subseteq \operatorname{range}(S^{-H}V), \tag{3.1}$$

$$RN(A) \subseteq \operatorname{range}(S^{-1}U). \tag{3.2}$$

If $k = n - r$, then

$$LN(A) = \operatorname{range}(S^{-H}V), \tag{3.3}$$

$$RN(A) = \operatorname{range}(S^{-1}U), \tag{3.4}$$

$$V^H S^{-1} U = I_k. \tag{3.5}$$

Proof. If $\mathbf{x} \in LN(A)$, then $\mathbf{x}^H S = \mathbf{x}^H(A + UV^H) = \mathbf{x}^H UV^H$, and therefore

$$\mathbf{x}^H = (\mathbf{x}^H U)V^H S^{-1}. \tag{3.6}$$

This proves (3.1).

Now let $k = n-r = \operatorname{Null} A$. To deduce equation (3.3) from (3.1), observe that $\operatorname{rank} V = \operatorname{rank}(S^{-H}V) = n - r$. Indeed $k \geq \operatorname{rank} V \geq \operatorname{rank} S - \operatorname{rank} A = n - r$.

To deduce equations (3.2) and (3.4), apply (3.1) and (3.3) to the matrices A^H and S^H instead of the matrices A and S and recall that $RN(A) = LN(A^H)$.

To prove (3.5) (assuming $k = n-r$), post-multiply equation (3.6) by U, apply equation (3.3), and deduce that $V^H S^{-1} U(V^H S^{-1} U - I_k) = 0$. (3.5) follows unless the matrix $V^H S^{-1} U$ is singular, but if it is, then $\mathbf{z}^H V^H S^{-1} U = \mathbf{0}^H$ for some vector $\mathbf{z} \neq \mathbf{0}$. Let us show a contradiction to complete the proof. Write $\mathbf{x}^H = \mathbf{z}^H V^H S^{-1}$ and deduce that $\mathbf{x}^H U = \mathbf{0}^H$ and $\mathbf{x} \in \operatorname{range}(S^{-H}V)$. It follows from (3.3) that $\mathbf{x}^H A = \mathbf{0}^H$. Therefore $\mathbf{x}^H S = \mathbf{x}^H A + \mathbf{x}^H UV^H = \mathbf{0}^H$. Consequently, $\mathbf{x} = \mathbf{0}$ because the matrix S is nonsingular. It follows that $\mathbf{z} = \mathbf{0}$ because the matrix $V^H S^{-1}$ has full rank. □

Remark 3.1. *The relations $S^H * L(A) \subseteq \operatorname{range} V$, $S * RN(A) \subseteq \operatorname{range} U$ hold even where the matrix S is singular.*

Let us perturb the matrices A and S and estimate how this affects equations (3.3) and (3.4).

Theorem 3.2. *Under the assumptions of Theorem 3.1, suppose that $E \in \mathbb{C}^{n\times n}$, $\mathbf{u} \in \operatorname{range} U$, $\mathbf{v} \in \operatorname{range} V$, the matrix $S + E$ is nonsingular, and equations (3.3) and (3.4) hold. Then we have $\mathbf{v}^H(S + E)^{-1} = \mathbf{x}^H(I - E(S + E)^{-1})$, $(S+E)^{-1}\mathbf{u} = (I-(S+E)^{-1}E)\mathbf{y}$ for $\mathbf{x} = S^{-H}\mathbf{v} \in LN(A)$, $\mathbf{y} = S^{-1}\mathbf{u} \in RN(A)$.*

Proof. Theorem 3.2 follows from the equations

$$S^{-1} - (S+E)^{-1} = (S+E)^{-1}ES^{-1} = S^{-1}E(S+E)^{-1}.$$

□

4 The null space computation

Given an $n \times n$ matrix A of a rank $r < n$, we can compute a vector from its null space $LN(A)$ or $RN(A)$ based on QR or LU factorizations of A with pivoting. As an alternative to these customary computations, we can reduce the problem to solving one of the two linear systems

$$S^H\mathbf{x} = \mathbf{v},\ S\mathbf{y} = \mathbf{u} \tag{4.1}$$

where $\mathbf{u} \in \text{range}\, U$, $\mathbf{v} \in \text{range}\, V$, U and V are $n \times k$ matrices, $k = n - r$, and the matrix $S = A + UV^H$ is nonsingular. In virtue of Theorem 3.1 we have $\mathbf{x} \in LN(A)$, $\mathbf{y} \in RN(A)$, and Theorem 3.2 bounds the approximation errors where the input matrix A is perturbed. If the matrix A is sparse and/or structured, we can choose the matrices U and V for which these features of A are preserved or enhanced, and then we can apply the known effective algorithms in [BBCD93], [DDSV98], [DER86], [CN96], [CN99], [PVWC04], [PKRKa] to solve systems (4.1).

We can extend this approach to computing a pair of $n \times k$ unitary matrices U and V whose columns form the bases for the null spaces $LN(A)$ and $RN(A)$, respectively, and consequently to computing the rank and numerical rank of an $n \times n$ matrix A.

5 Small-rank modification, singularity and conditioning

Let us show that the matrix S is not likely to be singular if the matrices U and V are random.

Theorem 5.1. *For a finite set Σ of cardinality $|\Sigma|$ in a ring $\mathbb{R}$ and for four matrices $A \in \mathbb{R}^{n\times n}$ of rank r, U and $V \in \Sigma^{n\times k}$, and $S = A + UV^T$, we have*

a) $\text{rank}\, S \le r + k$,

b) $\text{rank}\, S = r + k$ *with a probability of at least* $1 - \frac{2k}{|\Sigma|}$ *provided* $k \le n - r$ *and the entries of the matrices U and V have been randomly sampled from the set Σ,*

c) $\text{rank}\, S = r + k$ *with a probability of at least* $1 - \frac{k}{|\Sigma|}$ *if $k \le n - r$, the matrix U (resp. V) has full rank k, and the entries of the matrix V (resp. U) have been randomly sampled from the set Σ.*

Here random sampling of the matrix entries from Σ means their random choice from Σ, independently of each other and under the uniform probability distribution on Σ; furthermore we replace the matrices U^T and V^T with U^H and V^H if the ring $\mathbb{R}$ is the field $\mathbb{C}$ of complex numbers.

Proof. Part a) is verified immediately. To prove part b), project the matrices A and $S = A+UV^T$ into their $(r+k)\times(r+k)$ submatrices A_{r+k} and $S_{r+k} = A_{r+k}+(UV^T)_{r+k}$, respectively, such that $\operatorname{rank} A_{r+k} = r$. Then $\operatorname{rank} S_{r+k} = r+k$ if the entries of the matrices U and V are indeterminates. Since $\det S_{r+k}$ is a polynomial of total degree of at most $2k$ in these entries, part b) follows from a celebrated result in [DL78] (also in [Z79], [S80]). Part c) is proved similarly to part b). □

To estimate the ratio $(\operatorname{cond}_2 S)/(\operatorname{cond}_2 A)$ we first factorize the matrix S.

Theorem 5.2. *Assume that $S = A + UV^H$, S and A are $n \times n$ matrices, U and V are $n \times k$ matrices,* $\operatorname{rank} A = n-k$, $\operatorname{rank} S = n$, *and $A = G_A D F_A^H$ denotes the SVD of the matrix A, so that the matrices G_A and F_A are unitary, $D = \operatorname{diag}(D_A, 0_k)$, and D_A is the diagonal matrix of the singular values of the matrix A. Write $r = n-k$,*

$$G_A^H U = \begin{pmatrix} U_r \\ U_k \end{pmatrix},\ F_A^H V = \begin{pmatrix} V_r \\ V_k \end{pmatrix},\ T_U = \begin{pmatrix} I_r & U_r \\ 0 & U_k \end{pmatrix},\ T_V = \begin{pmatrix} I_r & V_r \\ 0 & V_k \end{pmatrix},$$

where the matrices U_k and V_k have size $k \times k$ (and have $\operatorname{rank} k$ *because* $\operatorname{rank} S = n$*). Then we have $S = G_A T_U \operatorname{diag}(D_A, I_k) T_V^H F_A^H$.*

Proof. We have $S = A + UV^H = G_A D F_A^H + G_A G_A^H U V^H F_A F_A^H = G_A \tilde{S} F_A^H$, $\tilde{S} = D + G_A^H U V^H F_A$, $T_U D T_V^H = D$, $G_A^H U = T_U \begin{pmatrix} 0 \\ I_k \end{pmatrix}$, $F_A^H V = T_V \begin{pmatrix} 0 \\ I_k \end{pmatrix}$. Deduce that $\tilde{S} = T_U D T_V^H + T_U \operatorname{diag}(0, I_k) T_V^H = T_U \operatorname{diag}(D_A, I_k) T_V^H$. Substitute this expression into the equation $S = G_A \tilde{S} F_A^H$. □

Corollary 5.1. *Under the assumptions of Theorem 5.2 we have*

$$\operatorname{cond}_2 S \le (\operatorname{cond}_2 T_U)(\operatorname{cond}_2 \operatorname{diag}(D_A, I_k)) \operatorname{cond}_2 T_V^H,$$

$$\operatorname{cond}_2 T_U = ||T_U||_2 ||T_U^{-1}||_2,\ \operatorname{cond}_2 T_V^H = \operatorname{cond}_2 T_V = ||T_V||_2 ||T_V^{-1}||_2,$$

$$T_U^{-1} = \begin{pmatrix} I_r & -U_r U_k^{-1} \\ 0 & U_k^{-1} \end{pmatrix},\ T_V^{-1} = \begin{pmatrix} I_r & -V_r V_k^{-1} \\ 0 & V_k^{-1} \end{pmatrix}.$$

Let us specify the latter estimate for $\operatorname{cond}_2 S$ provided the matrices U and V are unitary and the matrix A is scaled properly.

Theorem 5.3. *If the matrices U and V are unitary, then we have*

$$||T_U||_2 \le \sqrt{2},\ ||T_V||_2 \le \sqrt{2},$$

$$||T_U^{-1}||_2 \le 1 + \sqrt{2}||U_k^{-1}||_2,\ ||T_V^{-H}||_2 \le 1 + \sqrt{2}||V_k^{-1}||_2.$$

Proof. The theorem follows because

$$T_U = \mathrm{diag}(I_r, 0) + (0, G_A^H U)\,\mathrm{diag}(0, I_k),$$

$$T_V = \mathrm{diag}(I_r, 0) + (0, F_A^H V)\,\mathrm{diag}(0, I_k),$$

$$T_U^{-1} = \mathrm{diag}(I_r, 0) + \begin{pmatrix} 0 & -U_r \\ 0 & I_k \end{pmatrix} \mathrm{diag}(0, U_k^{-1}),$$

$$T_V^{-H} = \mathrm{diag}(I_r, 0) + \begin{pmatrix} 0 & -V_r \\ 0 & I_k \end{pmatrix} \mathrm{diag}(0, V_k^{-1}),$$

the 2-norms of the matrices F_A, G_A, U and V are equal to one, and $||(E,F)||_2 = ||(E,F)^H||_2 \le \sqrt{2}$ for a pair of unitary matrices E and F. □

Corollary 5.2. *Under the assumptions and definitions of Theorem 5.2, if the matrices U and V are unitary and $||A^{-1}||_2 \ge 1$, $||A||_2 \ge 1$, then* $\mathrm{cond}_2 S \le 2(1+\sqrt{2}||U_k^{-1}||_2)(1+\sqrt{2}||V_k^{-1}||_2)\,\mathrm{cond}_2 A$.

Proof. Combine Corolary 5.1 and Theorem 5.3 and note that

$$\mathrm{cond}_2\,\mathrm{diag}(D_A, I_k) = \mathrm{cond}_2 A \quad \text{if} \quad ||A^{-1}||_2 \ge 1, \quad ||A||_2 \ge 1.$$

□

Based on Corollary 5.2, we can control the growth of the condition number in the transition from the matrix A to the matrix S. The recipe is to choose $S = \frac{A}{\sigma} + UV^H$ or equivalently $S = A + \sigma UV^H$ where U and V are unitary matrices and $||A^{-1}||_2^{-1} \le \sigma \le ||A||_2$. If $\sigma/||A||_2 = \theta$ or $\sigma||A^{-1}||_2 = 1/\theta$, $\theta > 1$, the condition number should increase by roughly the factor of θ apart from the effect of the resulting dynamics in the factors of $||U_k^{-1}||_2$ and $||V_k^{-1}||$. The results of our extensive tests with random $n \times n$ matrices A of rank $n-1$ and random vectors $\mathbf{u}$ and $\mathbf{v}$ of dimension n have confirmed a substantial impact of scaling on the ratio $\frac{\mathrm{cond}_2 S}{\mathrm{cond}_2 A}$ (see Section 9.1).

6 Inverse iterations with small-rank modifications

For a matrix polynomial $A(\lambda) = \sum_{i=0}^{m} A_i \lambda^i$, we define its eigenpairs (λ, Y) such that λ is a scalar, Y is a unitary matrix (or a normalized vector), $\det A(\lambda) = 0$, and $\mathrm{range}\, Y = RN(A(\lambda))$.

Given a vector $\mathbf{u}$ and an approximation $\tilde{\lambda}$ to a geometrically simple eigenvalue λ of $A(\lambda)$, the IPI first computes the vector $\mathbf{y} = A^{-1}(\tilde{\lambda})\mathbf{u}/||A^{-1}(\tilde{\lambda})\mathbf{u}||$, approximating the associated eigenvectors, and then recursively alternates updating the scalar $\tilde{\lambda}$ by using the vector $\mathbf{y}$ and updating the vector $\mathbf{y}$ by using the scalar $\tilde{\lambda}$.

For $\tilde{\lambda} \approx \lambda$ the matrix $A(\tilde{\lambda})$ is ill-conditioned, but we can solve a distinct linear system $S(\tilde{\lambda})\mathbf{y} = \mathbf{u}$ with the coefficient matrix $S(\tilde{\lambda}) = A(\tilde{\lambda}) + \mathbf{u}\mathbf{v}^H$ defined by a

pair of fixed or random normalized vectors $\mathbf{u}$ and $\mathbf{v}$. This matrix is expected to be better-conditioned than $A(\lambda)$ according to our study in the previous section, whereas the vector $S^{-1}(\tilde{\lambda})\mathbf{u}$ is close to the vector $S(\lambda)^{-1}\mathbf{y}$, and the latter vector is proportional to the vector $A^{-1}(\lambda)\mathbf{u}$ because $S(\lambda)\mathbf{y} = b\mathbf{u},\ b = \mathbf{v}^H\mathbf{y}$.

Describing the resulting algorithm below, we write $||\cdot||_\nu$ for $\nu = 2$ or $\nu = F$ to denote the 2-norm or the Frobenius norm of a matrix, write $\mathbf{y}$ instead of $\mathbf{u}$, and recursively update the vector $\mathbf{y}$ by over-writing it with the vector $S^{-1}(\tilde{\lambda})\mathbf{y}$ where $S(\tilde{\lambda}) = A(\tilde{\lambda})$ if the matrix $A(\tilde{\lambda})$ is well-conditioned and $S(\tilde{\lambda}) = A(\tilde{\lambda}) + \mathbf{y}\mathbf{v}^H$ otherwise.

Algorithm 6.1. Inverse iteration with rank-one modifications.

INPUT: *a matrix polynomial $A(\lambda)$, an approximation $\tilde{\lambda}$ to its eigenvalue λ, two positive values t and κ, $\nu = 2$ or $\nu = F$, and a black box subroutine LIN·SOLVE for solving a linear system of equations.*

OUTPUT: *either FAILURE, or PROBABLY G·MULTIPLE, or an approximation $(\lambda_{final}, \mathbf{y}_{final})$ to an eigenpair of $A(\lambda)$ such that $||A(\lambda_{final})\mathbf{y}_{final}||_2 \le t||A(\lambda_{final})||_\nu$.*

INITIALIZATION: *Set $COUNTER \longleftarrow 0$, $\sigma \longleftarrow ||A(\tilde{\lambda})||_\nu$, and $(\mathbf{v}, \mathbf{y}) \longleftarrow$ a pair of fixed or random vectors satisfying $||\mathbf{v}||_2 = ||\mathbf{y}||_2 = 1$. (The vector $\mathbf{v}$ is needed only if the matrix $A(\tilde{\lambda})$ is ill-conditioned for the input or updated value of $\tilde{\lambda}$.)*

COMPUTATIONS:

1. *If $COUNTER > \kappa$, output FAILURE and stop. Otherwise set $S(\tilde{\lambda}) \longleftarrow A(\tilde{\lambda})$ and apply LIN·SOLVE to compute the vector $S^{-1}(\tilde{\lambda})\mathbf{y}$.*
2. *If this application fails (that is, if the matrix $S(\tilde{\lambda})$ is singular or ill-conditioned), then $S(\tilde{\lambda}) \longleftarrow \frac{1}{\sigma}S(\tilde{\lambda}) + \mathbf{y}\mathbf{v}^H$. Apply LIN·SOLVE to compute the vector $S^{-1}(\tilde{\lambda})\mathbf{y}$. If this application fails (that is, if the matrix $S(\tilde{\lambda})$ is still singular or ill-conditioned), then output PROBABLY G·MULTIPLE and stop.*
3. *Otherwise $\mathbf{y} \longleftarrow \frac{S^{-1}\mathbf{y}}{||S^{-1}\mathbf{y}||_2}$ (compute or update a normalized approximate eigenvector).*
4. *$\tilde{\lambda} \longleftarrow$ a root $\tilde{\lambda}$ of the equation $\mathbf{y}^H A(\tilde{\lambda})\mathbf{y} = 0$ which minimizes the residual norm $||A(\tilde{\lambda})\mathbf{y}||_2$. ($\tilde{\lambda} = \frac{\mathbf{y}^H M\mathbf{y}}{\mathbf{y}^H N\mathbf{y}}$ if $A = \lambda N - M$.) Update the matrix $A(\tilde{\lambda})$ for the updated value of $\tilde{\lambda}$.*
5. *$\sigma \longleftarrow ||A(\tilde{\lambda})||_\nu$. ($\sigma \longleftarrow \sigma + \tilde{\lambda}_{new} - \tilde{\lambda}_{old}$ if $\nu = 2$ and $A = \lambda I - M$.) If $||A(\tilde{\lambda})\mathbf{y}||_2 \le \sigma t$ (that is, if the residual norm is small enough), output $\lambda_{final} = \tilde{\lambda}$, $\mathbf{y}_{final} = \mathbf{y}$ and stop. Otherwise set $COUNTER \longleftarrow COUNTER + 1$ and go to Stage 1.*

Algorithm 6.1 outputs PROBABLY G·MULTIPLE if LIN·SOLVE fails for both coefficient matrices $A(\tilde{\lambda})$ and $S(\tilde{\lambda})$. According to our study in the previous

section, this can occur either because the vectors $\mathbf{v}$ and/or $\mathbf{y}$ lie in or near the ranges of the matrices $A(\tilde{\lambda})^H$ and/or $A(\tilde{\lambda})$, respectively (although such a case is unlikely for random vectors $\mathbf{v}$ and $\mathbf{y}$ according to Section 5), or because λ is a geometrically multiple eigenvalue of the matrix $A(\lambda)$ or lies near its another eigenvalue. We can modify Algorithm 6.1 to approximate such an eigenvalue λ as well. We just need to keep adding the outer products $\mathbf{y}\mathbf{v}^H$ of pairs of random vectors $\mathbf{y}$ and $\mathbf{v}^H$ to the matrix $S(\tilde{\lambda})$ until it becomes well-conditioned. The resulting algorithm can be viewed as the IPI/IR–RI complemented with small-rank modifications.

Algorithm 6.2. Inverse iteration with small-rank modification.

INPUT: *as in Algorithm 6.1.*

OUTPUT: *either FAILURE or an approximation* $(\lambda_{final}, Y_{final})$ *to an eigenpair of* $A(\lambda)$ *such that* $||A(\lambda_{final})Y_{final}||_2 \le t||A(\lambda_{final})||_\nu$.

INITIALIZATION: $COUNTER \longleftarrow 0$, $i \longleftarrow 0$, $\sigma \longleftarrow ||A(\tilde{\lambda})||_\nu$, $V_0 \longleftarrow ()$, $Y_0 \longleftarrow ()$ *where* $()$ *denotes the* $n \times 0$ *empty matrix,* $Y_1 \longleftarrow$ *an* $n \times 1$ *random unitary matrix.*

COMPUTATIONS:

1. *If* $COUNTER > \kappa$, *output FAILURE and stop. Otherwise* $S(\tilde{\lambda}) \longleftarrow \frac{1}{\sigma}A(\tilde{\lambda}) + Y_iV_i^H$, *apply LIN·SOLVE to compute the matrix or vector* $S^{-1}(\tilde{\lambda})Y_{\mu(i)}$ *where* $\mu(i) = 1$ *if* $i = 0$, $\mu(i) = i$ *otherwise.*
2. *If this application fails (that is, if the matrix* $S(\tilde{\lambda})$ *is singular or ill-conditioned), then set* $(\mathbf{v}, \mathbf{y}) \longleftarrow$ *a pair of random or fixed vectors such that* $\mathbf{v}^HV_i = \mathbf{0}$ *if* $i > 0$ *and* $||\mathbf{v}||_2 = ||\mathbf{y}||_2 = 1$, $V_{i+1} \longleftarrow (V_i, \mathbf{v})$, $Y_{i+1} \longleftarrow (Y_i, \mathbf{y})$ *(so that* $Y_{i+1}V_{i+1}^H = Y_iV_i^H + \mathbf{y}\mathbf{v}^H$*),* $i \longleftarrow i + 1$, $COUNTER \longleftarrow COUNTER + 1$, *and go to Stage 1.*
3. $Y \longleftarrow$ *the Q-factor in QR factorization of the matrix* $S^{-1}(\tilde{\lambda})Y_{\mu(i)}$ *(or, numerically, a properly truncated Q-factor in the rank revealing QR factorization of the matrix* $S^{-1}(\tilde{\lambda})Y_{\mu(i)}$*).*
4. $\tilde{\lambda} \longleftarrow$ *a root* $\tilde{\lambda}$ *of the equation* $\mathrm{trace}(Y^HA(\tilde{\lambda})Y) = 0$ *which minimizes the residual norm* $||A(\tilde{\lambda})Y||_2$. *(*$\tilde{\lambda} = \mathrm{trace}\,\frac{Y^HMY}{Y^HNY}$ *if* $A = \lambda N - M$*.) Update the matrix* $A(\tilde{\lambda})$ *for the updated value of* $\tilde{\lambda}$.
5. $\sigma \longleftarrow ||A(\tilde{\lambda})||_\nu$. *(*$\sigma \longleftarrow \sigma + \tilde{\lambda}_{new} - \tilde{\lambda}_{old}$ *if* $\nu = 2$ *and* $A = \lambda I - M$*.) If* $||A(\tilde{\lambda})Y||_2 \le \sigma t$ *(that is, if the residual norm is small enough), output* $\lambda_{final} = \tilde{\lambda}$, $Y_{final} = Y$ *and stop. Otherwise set* $COUNTER \longleftarrow COUNTER + 1$, $Y_{\mu(i)} \longleftarrow Y$, *and go to Stage 1.*

Remark 6.1. *Application of Algorithm 6.1 (resp. 6.2) to the matrix* $A^H(\lambda)$ *amounts to swapping the roles of the vectors* $\mathbf{v}$ *and* $\mathbf{y}$ *(resp. the matrices* V_i *and* Y_i*), so that the vectors* $\mathbf{v}$ *(resp. the vectors in the range of the matrices* V_i*) become directed towards the left eigenspace* $LN(A(\lambda))$. *One can update the*

pairs of the vectors $\mathbf{v}$ *and* $\mathbf{y}$ *(resp. matrices* V_i *and* Y_i*) simultaneously to yield convergence to both left and right eigenspaces* $LN(A(\lambda))$ *and* $RN(A(\lambda))$.

Remark 6.2. *In some cases we can simpify the computations by properly selecting the vectors* $\mathbf{v}$ *in Algorithms 6.1 and 6.2. For example, if* $A(\lambda) = R(\lambda) + \mathbf{w}\mathbf{z}^H$ *for a triangular matrix polynomial* $R(\lambda)$ *and two vectors* $\mathbf{w}$ *and* $\mathbf{z}$*, then* $\mathbf{v} = \mathbf{z}$ *is a natural simplifying selection. If we seek no such benefits, we can choose random vectors* $\mathbf{v}$ *or let* $\mathbf{v} = \mathbf{y}$.

Remark 6.3. *For some bad choices of the vectors* $\mathbf{v}$ *and* $\mathbf{y}$ *in Algorithm 6.2, the parameter* i *can exceed the rank of the matrix* $Y_i V_i^H$*. For random vectors* $\mathbf{v}$ *and* $\mathbf{y}$*, this degeneration occurs rarely; moreover, the QR factorization at Stage 3 can fix it. As a more costly additional fix, at Stage 1 we can apply LIN·SOLVE not to the matrix* $\frac{1}{\sigma}A(\tilde{\lambda}) + Y_i V_i^H$ *but recursively to the matrices* $\frac{1}{\sigma}A(\tilde{\lambda}) + Y_j V_j^H$ *for* $j = 0, 1, \ldots$ *where the matrices* Y_j *and* V_j *are made up of the first* j *columns of the matrices* Y_i *and* V_i*, respectively; here we increment* j *as long as LIN·SOLVE fails and* $j < i$*. By applying the binary search, we can do with at most* $\lceil \log_2 i \rceil$ *applications of LIN·SOLVE.*

We can extend Algorithm 6.2 to simultaneous approximation of more than one eigenvalue of $A(\lambda)$ (e.g., a cluster of h eigenvalues or a pair of complex conjugate eigenvalues of a real matrix M) by modifying its Stage 4 as follows:

> *4.* $(\tilde{\lambda}_i, \tilde{Y}_i) \longleftarrow$ *the eigenpairs of the* $k \times k$ *matrix polynomial* $B(\lambda) = Y^H A(\tilde{\lambda}) Y$*;* $Y_i \longleftarrow Y\tilde{Y}_i$*,* $i = 1, \ldots, h$.

In this case we should continue the computations with h applications of Algorithm 6.2 initialized with approximate eigenpairs $(\tilde{\lambda}_i, Y_i)$.

Especially, suppose we have a pair of the initial complex conjugate approximations $\tilde{\lambda}_1 \approx \lambda_1$ and $\tilde{\lambda}_2 \approx \lambda_2$ to a pair (λ_1, λ_2) of complex conjugate eigenvalues of a matrix $\lambda I - M$ where M is a real matrix. In this case the right invariant subspace of M associated with λ_1 and λ_2 equals $RN(A_{1,2})$, we have $RN(\tilde{A}_{1,2}) \approx RN(A_{1,2})$ where $A_{1,2} = (\lambda_1 I - M)(\lambda_2 I - M)$ and $\tilde{A}_{1,2} = (\tilde{\lambda}_1 I - M)(\tilde{\lambda}_2 I - M)$ are real matrices, $\tilde{A}_{1,2} \approx A_{1,2}$, and the real matrix $\tilde{S}_{1,2} = \frac{1}{\sigma}\tilde{A}_{1,2} + Y_i V_i^H$ replaces the matrix $S(\tilde{\lambda})$ in Algorithm 6.2.

Similarly we can refine approximations $\tilde{\lambda}_1, \ldots, \tilde{\lambda}_h$ to complex eigenvalues $\lambda_1, \ldots, \lambda_h$ of $A(\lambda)$ where the matrices $A_h = \prod_{i=1}^{h}(\lambda_i I - M)$ and $\tilde{A}_h = \prod_{i=1}^{h}(\tilde{\lambda}_i I - M)$ play the roles of the matrices $A_{1,2}$ and $\tilde{A}_{1,2}$, respectively.

7 Perturbations and errors

Eor $\lambda \approx \lambda$ Algorithms 6.1 and 6.2, compute nearly the same approximations to the eigenspaces of $A(\lambda)$ as the IPI and the IR–RI do, and so we can extend the extensive analysis of the latter iterations from [W65], [P80], [GL96], [I97], [S98], [BDDRvV00] and the bibliography therein. Moreover, we can simplify this analysis because we can involve the matrix $S^{-1}(\lambda)$ even where λ is an eigenvalue

of the matrix polynomial $A(\lambda)$. Let us show some simple error/perturbation estimates which imply local quadratic convergence of Algorithms 6.1 and 6.2.

Theorem 7.1. *Let $\tilde{Y}$ and Y be $n \times k$ matrices and write $\Delta = \tilde{Y} - Y$. Then for an $n \times n$ matrix A we have $\tilde{Y}^H A \tilde{Y} - Y^H A Y = \Delta^H A Y + Y^H A \Delta + \Delta^H A \Delta$.*

Theorem 7.2. *Let* range $Y = RN(A)$ *for an $n \times n$ matrix A and a unitary $n \times k$ matrix Y. Let a pair of matrices $\tilde{A}$, $\tilde{Y}$ approximate A, Y. Write $S = A + \tilde{Y} V^H$, $\tilde{S} = \tilde{A} + \tilde{Y} V^H$, $\delta = \tilde{S} - S = \tilde{A} - A$, $\Delta = \tilde{Y} - Y$ for an $n \times k$ matrix V such that the matrices $B = V^H Y$ and $\tilde{S}$ are nonsingular, $B = I_k$ if $\tilde{Y}$ is a unitary matrix and $V = \tilde{Y}$. Then we have*

a) $\tilde{S}^{-1} \tilde{Y} = Y B^{-1} - \tilde{S}^{-1} \delta Y B^{-1}$.

b) Furthermore, $\tilde{S}^{-1} \tilde{Y} = Y B^{-1} (I - \tilde{\gamma}) - \tilde{S}^{-1} Y \tilde{\gamma}^2 - \tilde{S}^{-1} \Delta \tilde{\gamma}$ *provided* range$(\delta Y) \subseteq$ range $Y = RN(A)$, *$Y \gamma = \delta Y$, and $\tilde{\gamma} = \gamma B^{-1}$.*

Proof. First assume that the matrix S is nonsingular.

Observe that $\tilde{S}^{-1} = (I - \tilde{S}^{-1} \delta) S^{-1}$. Recall that $AY = 0$, and so $SY = (A + \tilde{Y} V^H) Y = \tilde{Y} (V^H Y) = \tilde{Y} B$, $S^{-1} \tilde{Y} = Y B^{-1}$. Therefore,

$$\tilde{S}^{-1} \tilde{Y} = (I - \tilde{S}^{-1} \delta) S^{-1} \tilde{Y} = Y B^{-1} - \tilde{S}^{-1} \delta Y B^{-1}.$$

This proves part a).

Substitute the equation $\delta Y B^{-1} = Y \tilde{\gamma}$ into the equation of part a) and obtain that $\tilde{S}^{-1} \tilde{Y} = Y B^{-1} - \tilde{S}^{-1} Y \tilde{\gamma}$.

Substitute

$$\tilde{S}^{-1} Y = \tilde{S}^{-1} \tilde{Y} - \tilde{S}^{-1} \Delta = Y B^{-1} - \tilde{S}^{-1} Y \tilde{\gamma} - \tilde{S}^{-1} \Delta$$

on the right-hand side and obtain that

$$\tilde{S}^{-1} \tilde{Y} = Y B^{-1} (I - \tilde{\gamma}) - \tilde{S}^{-1} Y \tilde{\gamma}^2 - \tilde{S}^{-1} \Delta \tilde{\gamma}.$$

This proves part b).

Relax the assumption that the matrix S is nonsingular by applying infinitesimal perturbations of the matrix A. □

Corollary 7.1. *Under the assumptions of Theorem 7.2, we have $||S^{-1} \tilde{Y} - Y B^{-1}|| = O((||\Delta|| + ||\tilde{\gamma}||) ||\tilde{\gamma}||)$ provided* range$(\delta Y) \subseteq$ range $Y = RN(A)$.

Theorem 7.1 and Corollary 7.1 together imply local quadratic convergence of Algorithms 6.1 and 6.2 provided range$(\delta Y) \subseteq$ range $Y = RN(A)$. We easily observe the latter provision where $A = \lambda I - M$.

Theorem 7.3. *Under the assumptions of Theorem 7.2, let $A = \lambda I - M$ and $\tilde{A} = A(\tilde{\lambda})$. Then* range$(\delta Y) \subseteq$ range $Y = RN(A)$.

Proof. If $A = A(\lambda) = \lambda I - M$, and $\tilde{A} = A(\tilde{\lambda})$, then $\delta = \tilde{S} - S = A(\tilde{\lambda}) - A(\lambda) = (\tilde{\lambda} - \lambda) I$, $\delta Y = (\tilde{\lambda} - \lambda) Y$, and therefore, range$(\delta Y) =$ range Y unless $\delta Y = 0 \in$ range Y. □

Our next theorem implies the inclusion $\text{range}(\delta Y) = \text{range}((\tilde{A} - A)Y) \subseteq \text{range} Y = RN(A)$ and therefore implies quadratic convergence of Algorithms 6.1 and 6.2, for a little more general class of matrix polynomials $A(\lambda)$ discussed at the end of the previous section.

Theorem 7.4. *Suppose that $\lambda_1, \ldots, \lambda_h$ are the eigenvalues of a matrix polynomial $A(\lambda) = \lambda I - M$. Let Y be a unitary $n \times k$ matrix such that* $\text{range} Y = RN(A_h(\lambda))$ *for $A = A_h(\lambda) = \prod_{i=1}^{h}(\lambda_i I - M)$. Then*

$$\text{range}(\delta Y) = \text{range}((\tilde{A} - A)Y) =$$

$$\text{range}(A_h(\tilde{\lambda})Y - A_h(\lambda)Y) = \text{range}(A_h(\tilde{\lambda})Y) \subseteq \text{range} Y$$

for $\tilde{A} = A_h(\tilde{\lambda}) = \prod_{i=1}^{h}(\tilde{\lambda}_i I - M)$ and any set of scalars $\tilde{\lambda}_1, \ldots, \tilde{\lambda}_h$.

Proof. Let $\mathbf{y}$ be a vector from $\text{range} Y = RN(A_h(\lambda)) = RN(\prod_{i=1}^{h}(\lambda_i I - M))$. Represent $\mathbf{y}$ as $\sum_{i=1}^{h} \mathbf{y}_i$ where $(\lambda_i I - M)\mathbf{y}_i = \mathbf{0}$, $i = 1, \ldots, h$, so that

$$(\tilde{\lambda}_j I - M)\mathbf{y}_i = (\tilde{\lambda}_j - \lambda_i)\mathbf{y}_i, \ i = 1, \ldots, h.$$

Therefore,

$$\prod_{j=1}^{h}(\tilde{\lambda}_j I - M)\mathbf{y}_i = \prod_{j=1}^{h}(\tilde{\lambda}_j - \lambda_i)\mathbf{y}_i \in \text{range} Y, \ i = 1, \ldots, h,$$

$$A_h(\tilde{\lambda})\mathbf{y} = \prod_{j=1}^{h}(\tilde{\lambda}_j I - M)\mathbf{y} = \prod_{j=1}^{h}(\tilde{\lambda}_j I - M)\sum_{i=1}^{h}\mathbf{y}_i = \sum_{i=1}^{h}\prod_{j=1}^{h}(\tilde{\lambda}_j - \lambda_i)\mathbf{y}_i \in \text{range} Y.$$

□

Remark 7.1. *Assume a linear generalized eigenproblem with the input matrix polynomial $\lambda N - M$ where the matrix N is nonsingular. Then we can perform Algorithms 6.1 and 6.2 for the matrix polynomials $A(\lambda) = \lambda I - N^{-1}M$ and $S(\lambda) = \lambda I - N^{-1}M + YV^H$; this enables us to apply Theorem 7.3. Furthermore, we can relax the assumption that the matrix N is nonsingular because we can compute the matrix $S(\lambda)^{-1}Y = (A(\lambda) + NYV^H)^{-1}NY$ for a given λ without explicitly involving the matrix N^{-1}.*

8 Small-rank modifications and rational equations for the eigenvalues

Let us deduce some rational equations for the eigenvalues of a matrix polynomial $A(\lambda)$ by applying its small-rank modifications.

For $A = A(\lambda)$ and a scalar λ, let $g.m._A(\lambda) = k$. Then matrix equation (3.5) turns into the system of k^2 rational equations

$$F(\lambda) = I_k - V^H S^{-1} U = 0_k \tag{8.1}$$

satisfied by the eigenvalues λ with $g.m._A(\lambda) = k$. Here U and V are $n \times k$ matrix polynomials in λ and $S = A + UV^H$. By pre- and post-multiplying matrix equation (8.1) by vectors $\mathbf{s}^H$ and $\mathbf{t}$ of dimension k, respectively, we obtain a single scalar equation in λ,

$$f(\lambda) = \mathbf{s}^H F(\lambda)\mathbf{t} = \mathbf{s}^H\mathbf{t} - \mathbf{s}^H V^H S^{-1} U\mathbf{t} = 0.$$

Let us deduce a similar expression independent of the multiplicity $g.m._A(\lambda)$.

Theorem 8.1. *Let $A = S - UV^H$ for four matrices A, U, V and S of the sizes $n \times n$, $n \times k$, $n \times k$, and $n \times n$, respectively. Let the matrix S be nonsingular. Then $\det A = (\det S)\det(I_k - V^H S^{-1} U)$. If A, U, V and S are matrix polynomials in λ, then this equation can be rewritten as follows,*

$$c_A(\lambda) = c_S(\lambda) r(\lambda),\ r(\lambda) = \det(I_k - V^H S^{-1} U). \tag{8.2}$$

Proof. We have $A = S(I - S^{-1}UV^H)$. Recall that $\det(I_k - BC) = \det(I_k - CB)$ [H64, Exercise 1.14]. Apply this equation for $C = S^{-1}U$, $B = V^H$ and deduce that $\det A = (\det S)\det(I_k - S^{-1}UV^H) = (\det S)\det(I_k - V^H S^{-1}U)$. □

Remark 8.1. *For $k = 1$, $U = \mathbf{u}$, and $V = \mathbf{v}$, equation (8.2) is simplified as follows,*

$$c_A(\lambda) = c_S(\lambda) r(\lambda),\ r(\lambda) = r_{A,S}(\lambda) = 1 - \mathbf{v}^H S^{-1}\mathbf{u}.$$

For $A = \lambda I - M$, this turns into the classical result in [H64] (cf. also [G73]). The expressions $B - V^H(\lambda I - A)^{-1}U$ are called the realizations of rational matrix functions and are extensively used in linear systems and control [K80].

Recalling that $(S^{-1})' = -S^{-1}S'S^{-1}$, one can obtain expressions for the derivatives and higher order derivatives of the functions $F(\lambda)$, $f(\lambda)$, and $r(\lambda)$.

9 Numerical tests

In our tests we used the following CPU and memory configuration, operating system, mathematical application software, and random number generator:

CPU	AMD Athlon XP 2800+ 2.09GHZ
Memory	512MB
OS	Microsoft Windows XP Professional Version 2002 Service Pack 2
Platform	Matlab Version 7.0.0.19920(R14)
Random Number Generator	Matlab Statistics Toolbox's Uniform Distribution

9.1 Scaled small-rank perturbations and their impact on condition numbers

In every entry of Table 9.1 we show the average (mean), maximum and minimum values as well as the standard deviation of the ratios $\frac{\mathrm{cond}_2\, S_k(\rho)}{\mathrm{cond}_2\, A}$. Here A and $S_k(\rho)$ are 100×100 matrices, $A = (\mathbf{a}_j)_{j=1}^{100}$; the entries of the vectors $\mathbf{a}_j$ of dimension 100 for $j = 1, \ldots, 99$ have been randomly generated in the range $[-1, 1]$; $\mathbf{a}_{100} = \sum_{j=1}^{99} c_j \mathbf{a}_j$, c_j are random numbers in the range $[-1, 1]$; $S_k(\rho) = \rho A + \sum_{i=1}^{k} \mathbf{u}_i \mathbf{v}_i^H$ where $k = 1, 2, 3$; $\rho = 10^{-10}$, 10^{-3}, $\frac{1}{||A||_2}$ (cf. Corollary 5.2), 10^3, 10^{10}; $\mathbf{u}_i = \frac{\mathbf{y}_i}{||\mathbf{y}_i||_2}$, $\mathbf{v}_i = \frac{\mathbf{x}_i}{||\mathbf{x}_i||_2}$; $\mathbf{x}_i$ and $\mathbf{y}_i$ for $i = 1, 2, 3$ are vectors of dimension 100 with random coordinates in the range $[-1, 1]$.

For every choice of k and ρ we have generated 100 pairs of the matrices A and $S_k(\rho)$ in this way.

In Table 9.2 we cover almost the same tests except that we let $\mathbf{y}_i = \mathbf{x}_i$ and consequently $\mathbf{u}_i = \mathbf{v}_i$ for random vectors $\mathbf{x}_i$, $i = 1, 2, 3$.

In Tables 9.3 and 9.4 we dispay the results of the tests similar to the tests covered in Tables 9.1 and 9.2, respectively, except that for every input pair of A and ρ, we have generated three sets of the vectors $\mathbf{x}_i$, $\mathbf{y}_i$, $\mathbf{u}_i$, $\mathbf{v}_i$, $i = 1, 2, 3$, and computed the triple of the respective ratios $\frac{\mathrm{cond}_2\, S_k(\rho)}{\mathrm{cond}_2\, A}$ but included the data only for the minimum in the triple.

For random vectors $\mathbf{x}_i$ and $\mathbf{y}_i$, we have occasionally observed very large values of the computed ratios $\frac{\mathrm{cond}_2\, S_k(\rho)}{\mathrm{cond}_2\, A}$, and this has made the test results more chaotic. Overall, the statistics of our tests, however, is still in sufficiently good accordance with our study in Section 5.

9.2 Iteration count for the IPI and Algorithm 6.1

In Tables 9.5 and 9.6 we show the numbers of iterations required for convergence of the IPI and Algorithm 6.1. We display the average (mean) values and standard deviations in 200 tests with $n \times n$ matrices $M = S^{-1} T S$ for $n = 64$ and $n = 100$, for S being either a random matrix or the Q-factor in QR factorization of a random matrix, and for T from one of the following four matrix classes.

1. $T = D_r$ is a real diagonal matrix with random entries in the range $[0, 10]$.
2. $T = D_c$ is a complex diagonal matrix whose entries have random absolute values in the range $[0, 10]$ and random arguments in the range $[0, 2\pi)$.
3. $T = D_r + \mathbf{e}_1 \mathbf{v}^T + \mathbf{u}\mathbf{e}_n^T$ is an arrow-head matrix, D_r is a matrix of class 1; $\mathbf{u}$, $\mathbf{v}$ have random entries in the range $[0, 10]$.
4. $T = D_r + \mathbf{u}\mathbf{v}^T$, D_r and $\mathbf{v}$ are as in matrix class 3, the vector $\mathbf{u}$ has random coordinates in the range $[0, 1]$.

The results of our extensive tests reported in Tables 9.5 and 9.6 confirm that the IPI and Algorithm 6.1 converge with about the same rate.

Table 9.1: The ratios $\frac{\mathrm{cond}_2 S_k(\rho)}{\mathrm{cond}_2 A}$ in 100 tests with random vectors $\mathbf{x}_i$ and $\mathbf{y}_i$ for $i = 1, 2, 3$

ρ \ k		1	2	3
10^{-10}	Mean	$9.24E+09$	$3.92E+09$	$2.28E+09$
	Std. Dev.	$5.27E+10$	$3.20E+10$	$6.05E+09$
	Max. Ratio	$1.16E+12$	$9.60E+11$	$1.03E+11$
	Min. Ratio	$2.57E+08$	$4.78E+07$	$3.71E+07$
10^{-3}	Mean	$8.20E+02$	$3.49E+02$	$5.76E+02$
	Std. Dev.	$4.59E+03$	$2.46E+03$	$7.38E+03$
	Max. Ratio	$9.24E+04$	$6.93E+04$	$2.29E+05$
	Min. Ratio	$2.28E+01$	$3.28E+00$	$2.29E+00$
$1/\|\|A\|\|_2$	Mean	$2.61E+01$	$1.11E+01$	$6.44E+00$
	Std. Dev.	$7.42E+01$	$3.39E+01$	$1.11E+01$
	Max. Ratio	$4.63E+02$	$3.19E+02$	$6.98E+01$
	Min. Ratio	$1.06E+00$	$2.52E-01$	$1.64E-01$
10^{3}	Mean	$3.73E+05$	$5.21E+04$	$4.25E+04$
	Std. Dev.	$2.14E+06$	$1.53E+05$	$1.96E+05$
	Max. Ratio	$2.06E+07$	$1.14E+06$	$1.92E+06$
	Min. Ratio	$3.66E+02$	$6.39E+02$	$6.25E+02$
10^{10}	Mean	$8.23E+11$	$4.58E+11$	$3.33E+11$
	Std. Dev.	$2.09E+12$	$1.93E+12$	$1.09E+12$
	Max. Ratio	$1.56E+13$	$1.86E+13$	$1.00E+13$
	Min. Ratio	$1.22E+10$	$2.62E+09$	$2.82E+09$

Table 9.2: The ratios $\frac{\mathrm{cond}_2\, S_k(\rho)}{\mathrm{cond}_2\, A}$ in 100 tests with random vectors $\mathbf{x}_i$ and with $\mathbf{y}_i = \mathbf{x}_i$, $i = 1, 2, 3$

k / rho		1	2	3
10^{-10}	Mean	$1.32E+10$	$3.49E+09$	$1.05E+10$
	Std. Dev.	$5.15E+10$	$1.00E+10$	$8.44E+10$
	Max. Ratio	$4.15E+11$	$8.83E+10$	$8.45E+11$
	Min. Ratio	$2.86E+08$	$1.25E+08$	$1.73E+07$
10^{-3}	Mean	$6.80E+02$	$3.14E+02$	$2.70E+02$
	Std. Dev.	$2.21E+03$	$1.12E+03$	$5.97E+02$
	Max. Ratio	$1.99E+04$	$8.56E+03$	$4.91E+03$
	Min. Ratio	$3.02E+01$	$8.10E+00$	$8.41E+00$
$1/\|\|A\|\|_2$	Mean	$2.29E+01$	$1.43E+01$	$8.64E+00$
	Std. Dev.	$6.65E+01$	$3.65E+01$	$1.94E+01$
	Max. Ratio	$5.57E+02$	$2.43E+02$	$1.12E+02$
	Min. Ratio	$1.08E+00$	$2.10E-01$	$1.34E-01$
10^{3}	Mean	$2.70E+05$	$3.76E+04$	$2.07E+04$
	Std. Dev.	$1.67E+06$	$1.36E+05$	$4.91E+04$
	Max. Ratio	$1.66E+07$	$1.24E+06$	$3.39E+05$
	Min. Ratio	$8.74E+02$	$3.76E+02$	$1.47E+02$
10^{10}	Mean	$2.47E+12$	$1.38E+12$	$4.32E+11$
	Std. Dev.	$1.07E+13$	$9.96E+12$	$2.84E+12$
	Max. Ratio	$8.35E+13$	$9.93E+13$	$2.85E+13$
	Min. Ratio	$1.03E+10$	$2.21E+09$	$1.60E+09$

Table 9.3: The ratios $\frac{\text{cond}_2 S_k(\rho)}{\text{cond}_2 A}$ in 100 tests minimized in the triples of random pairs of vectors $\mathbf{x}_i$ and $\mathbf{y}_i$, $i = 1, 2, 3$

ρ \ k		1	2	3
10^{-10}	Mean	$7.53E+08$	$6.15E+08$	$5.78E+08$
	Std. Dev.	$7.19E+08$	$5.30E+08$	$4.98E+08$
	Max. Ratio	$7.43E+09$	$5.15E+09$	$4.62E+09$
	Min. Ratio	$2.54E+08$	$1.76E+07$	$9.78E+06$
10^{-3}	Mean	$7.67E+01$	$5.84E+01$	$5.84E+01$
	Std. Dev.	$6.40E+01$	$5.70E+01$	$6.50E+01$
	Max. Ratio	$5.42E+02$	$1.01E+03$	$1.01E+03$
	Min. Ratio	$2.50E+01$	$6.84E-01$	$2.53E+00$
$1/\|\|A\|\|_2$	Mean	$2.99E+00$	$2.08E+00$	$2.10E+00$
	Std. Dev.	$2.49E+00$	$1.38E+00$	$2.74E+00$
	Max. Ratio	$1.53E+01$	$7.70E+00$	$2.42E+01$
	Min. Ratio	$1.08E+00$	$2.89E-01$	$7.85E-02$
10^3	Mean	$9.93E+03$	$6.96E+03$	$4.58E+03$
	Std. Dev.	$1.09E+04$	$8.41E+03$	$5.08E+03$
	Max. Ratio	$6.30E+04$	$5.08E+04$	$3.66E+04$
	Min. Ratio	$9.62E+02$	$2.11E+02$	$4.34E+02$
10^{10}	Mean	$1.12E+11$	$5.95E+10$	$4.77E+10$
	Std. Dev.	$1.43E+11$	$5.54E+10$	$5.15E+10$
	Max. Ratio	$8.54E+11$	$3.42E+11$	$3.02E+11$
	Min. Ratio	$6.44E+09$	$6.39E+09$	$1.46E+09$

Table 9.4: The ratios $\frac{\mathrm{cond}_2 S_k(\rho)}{\mathrm{cond}_2 A}$ in 100 tests minimized in the triples of the pairs $(\mathbf{x}_i, \mathbf{y}_i)$ where the vectors $\mathbf{x}_i$ are random and $\mathbf{y}_i = \mathbf{x}_i$, $i = 1, 2, 3$

ρ \ k		1	2	3
10^{-10}	Mean	$8.19E+08$	$6.02E+08$	$4.99E+08$
	Std. Dev.	$8.27E+08$	$3.48E+08$	$3.74E+08$
	Max. Ratio	$4.76E+09$	$2.00E+09$	$1.70E+09$
	Min. Ratio	$2.82E+08$	$8.53E+07$	$1.14E+07$
10^{-3}	Mean	$7.32E+01$	$6.60E+01$	$4.77E+01$
	Std. Dev.	$6.72E+01$	$9.13E+01$	$3.16E+01$
	Max. Ratio	$4.69E+02$	$8.10E+02$	$2.28E+02$
	Min. Ratio	$2.71E+01$	$3.31E-01$	$5.51E+00$
$1/\|\|A\|\|_2$	Mean	$2.99E+00$	$2.08E+00$	$2.30E+00$
	Std. Dev.	$2.64E+00$	$1.40E+00$	$2.05E+00$
	Max. Ratio	$1.61E+01$	$9.57E+00$	$1.50E+01$
	Min. Ratio	$1.08E+00$	$2.12E-02$	$2.67E-01$
10^{3}	Mean	$1.14E+04$	$5.79E+03$	$4.32E+03$
	Std. Dev.	$2.60E+04$	$4.97E+03$	$3.63E+03$
	Max. Ratio	$2.41E+05$	$2.78E+04$	$1.96E+04$
	Min. Ratio	$3.95E+02$	$2.19E+02$	$2.04E+02$
10^{10}	Mean	$1.10E+11$	$6.59E+10$	$4.42E+10$
	Std. Dev.	$2.08E+11$	$6.41E+10$	$4.24E+10$
	Max. Ratio	$1.61E+12$	$2.87E+11$	$2.95E+11$
	Min. Ratio	$5.29E+09$	$2.09E+09$	$3.14E+09$

Table 9.5: Iteration count for IPI and Algorithm 6.1 with unitary matrix S

Matrix		Algorithm 6.1		IPI	
Classes	n	iter	std dev	iter	std dev
$T = D_r$	64	4.74	1.145	4.93	1.242
	100	4.71	1.277	4.88	1.299
$T = D_c$	64	5.67	1.415	5.61	1.396
	100	5.67	1.461	5.62	1.321
$T = D_r + \mathbf{e}_1\mathbf{v}^T + \mathbf{u}\mathbf{e}_n^T$	64	4.94	1.230	5.01	1.341
	100	4.75	1.176	4.75	1.260
$T = D_r + \mathbf{u}\mathbf{v}^T$	64	5.77	1.668	5.95	1.808
	100	5.54	1.445	5.67	1.553

Table 9.6: Iteration count for IPI and Algorithm 6.1 with random matrix S

Matrix		Algorithm 6.1		IPI	
Classes	n	iter	std dev	iter	std dev
$T = D_r$	64	5.36	2.532	5.36	2.520
	100	4.88	2.509	4.86	2.452
$T = D_c$	64	5.76	1.716	5.71	1.516
	100	5.59	1.401	5.64	1.497
$T = D_r + \mathbf{e}_1\mathbf{v}^T + \mathbf{u}\mathbf{e}_n^T$	64	5.09	1.621	5.03	1.605
	100	4.72	1.473	4.67	1.467
$T = D_r + \mathbf{u}\mathbf{v}^T$	64	5.550	1.907	5.550	1.872
	100	5.660	2.118	5.555	1.992

References

[BBCD93] R. Barrett, M. W. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, H. Van Der Vorst, *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*, SIAM, Philadelphia, 1993.

[BDDRvV00] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, H. van der Vorst, editors, *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*, SIAM, Philadelphia, 2000.

[BGTa] D. A. Bini, L. Gemignani, F. Tisseur, The Ehrlich-Aberth Method for the Nonsymmetric Tridiagonal Eigenvalue Problem, Numerical Analysis Report 428, *Dept. of Math., University of Manchester*, England, June 2003 (revised April 2004).

[CN96] R. H. Chan, M. K. Ng, Conjugate Gradient Methods for Toeplitz Systems, *SIAM Review*, **38**, 427–482, 1996.

[CN99] R. H. Chan, M. K. Ng, Iterative Methods for Linear Systems with Matrix Structures, in *Fast Reliable Algorithms for Matrices with Structure* (T. Kailath and A. H. Sayed, editors), 117–152, SIAM, Philadelpha, 1999.

[DDSV98] I. S. Duff, A. M. Erisman, J. K. Reid, *Direct Methods for Sparse Matrices*, Clarendon Press, Oxford, England, 1986.

[DER86] J. J. Dongarra, I. S. Duff, D. C. Sorensen, H. A. Van Der Vorst, *Numerical Linear Algebra for High-Performance Computers*, SIAM, Philadelphia, 1998.

[DL78] R. A. Demillo, R. J. Lipton, A Probabilistic Remark on Algebraic Program Testing, *Information Processing Letters*, **7**, **4**, 193–195, 1978.

[G73] G. H. Golub, Some Modified Matrix Eigenvalue Problems, *SIAM Review*, **15**, 318–334, 1973.

[GL96] G. H. Golub, C. F. Van Loan, *Matrix Computations*, 3rd edition, The Johns Hopkins University Press, Baltimore, Maryland, 1996.

[H64] A. Householder, *The Theory of Matrices in Numerical Analysis*, Dover, New York, 1964.

[I97] I. C. F. Ipsen, Computing an Eigenvector with Inverse Iteration, *SIAM Review*, **39**, 354–391, 1998.

[K80] T. Kailath, *Linear Systems*, Prentice-Hall, Englewood Cliffs, New Jersey, 1980.

[P80] B. N. Parlett, *The Symmetric Eigenvalue Problem*, Prentice-Hall, Englewood Cliffs, New Jersey, 1980, and SIAM, Philadelphia, 1998.

[PKRKa] V. Y. Pan, M. Kunin, R. E. Rosholt, H. Kodal, Homotopic Residual Correction Processes, *Math. of Computation*, in press.

[PVWC04] V. Y. Pan, M. Van Barel, X. Wang, G. Codevico, Iterative Inversion of Structured Matrices, *Theoretical Computer Science*, Special Issue on Algebraic and Numerical Algorithms (I. Z. Emiris, B. Mourrain, V. Y. Pan editors), **315**, **2–3**, 581–592, 2004.

[S80] J. T. Schwartz, Fast Probabilistic Algorithms for Verification of Polynomial Identities, *Journal of ACM*, **27**, **4**, 701–717, 1980.

[S98] G. W. Stewart, *Matrix Algorithms, Vol II: Eigensystems*, SIAM, Philadelphia, 1998.

[TM01] F. Tisseur, K. Meerbergen, The Quadratic Eigenvalue Problem, *SIAM Review*, **43**, **2**, 235–286, 2001.

[W65] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, 1965.

[Z79] R. E. Zippel, Probabilistic Algorithms for Sparse Polynomials, *Proceedings of EUROSAM'79, Lecture Notes in Computer Science*, **72**, 216–226, Springer, Berlin, 1979.