2005

TR-2005009: Additive Preconditioning in Matrix Computations

Victor Y. Pan

Brian Murphy

Rhys Eric Rosholt

Additive Preconditioning in Matrix Computations

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 propose additive preconditioning in matrix computations as a complement to the popular techniques of multiplicative preconditioning. We specify our approach for computing a vector in the null space of a matrix and solving a linear system of equations and show further applications to the algebraic eigenproblem. We prove theoretically and confirm experimentally the efficiency of the resulting algorithms.

Key words: Small-rank modifications, Null spaces, Linear systems of equations, Eigenspaces.

1 Introduction

1.1 Additive preconditioners

Matrix preconditioning is a popular tool for the acceleration of the solution of a linear system of equations (see [C05] and the bibliography therein). The basic idea is in replacing a system $M\mathbf{x} = \mathbf{b}$ with one of the equivalent systems $PM\mathbf{x} = P\mathbf{b}$, $MS\mathbf{y} = \mathbf{b}$, or $PMS\mathbf{y} = P\mathbf{b}$ where $\mathbf{x} = S\mathbf{y}$ and the multiplication with the preconditioners P and/or S accelerates the convergence of the known iterative solution algorithms.

As a complementary or alternative approach to the same goal, we propose adding rank-one matrices recursively to the input matrix M until we yield a desired simplification of the subsequent solution. We prove formally and verify

experimentally that typically we arrive at a well-conditioned matrix in a small number of such random and properly scaled rank-one modifications.

Shifting to a well-conditioned linear system generally allows more accurate numerical solution in less time [C05]. Here are some specifications: the convergence of various iterative algorithms is accelerated (cf. [BBCD93], [SvV00], [vV03], [P01, Chapter 6], [PKRKa]), whereas adding a small rank matrix little affects their basic operation of multiplication by a vector; we can safely apply symmetrization because squaring the condition number can be repaired with our preconditioning, and the power of the ILU and some other multiplicative preconditioners is enhanced [TB97, Lecture 40], [C05].

Nonrandom additive preconditioners also support simplifications. In some cases, they can improve the structure of the input matrix (e. g., its Hermitian, positive definite and/or displacement structures), and they can support the divide-and-conquer solution of banded linear systems or, more generally, of the systems associated with the graphs that have small separator families [LRT79], [DER86], [GH90], [GS92], [PR93], [DDSV98]. In yet another instance, these preconditioners enable us to increase the absolute values of small pivot entries in the Gaussian elimination and Cyclic Reduction algorithms, and similarly we can improve the conditioning of the pivot blocks of a smaller size whenever we need to invert them in block matrix algorithms such as block Gaussian elimination and block Cyclic Reduction.

All this may seem to be too good to be real, and indeed it is not immediately clear whether our preconditioning still leaves for us any chance for retrieving the solution of the original linear system. The Sherman–Morrison–Woodbury formula supports the desired retrival in a small number of flops in the important case where the matrix M has a small displacement rank [P01], but this way is prone to the numerical stability problems.

Surprisingly, there exists a simple way out, that is, back to the solution of the original linear system. We first reduce the original problem to computing a vector from the null space of an extended coefficient matrix. Then we apply additive preconditioning of this matrix and shift the computation back to solving a linear system, but with the preconditioned matrix.

We elaborate upon some basic issues of this approach to computing a vector in the null space of a matrix and to solving a linear system of equations. We also incorporate the resulting improved algorithms as blocks of the inverse iteration for the algebraic eigenproblem and study the resulting impact both theoretically and experimentally.

Further extensions of additive preconditioning and its further applications to matrix computations are the subjects of our current and future study.

1.2 The preceding and future study

Small-rank modifications are a known tool in the divide-and-conquer eigensolvers [G73], [GL96], [S98] but other than that have been rarely used in matrix computations. In the discussions that followed the presentations of this work by the first author at the Internatonal Conferences on the Matrix Methods and

Operator Equations in Moscow, Russia, in June 20-25, 2005, and on the Foundations of Computational Mathematics (FoCM'2005) in Santander, Spain, June 30-July 9, 2005, only a few other relevant citations have emerged. G. H. Golub cited the paper by Paige, Styan, and Wachter in the Journal of Statistical Comp. Simulation, 1975, and pointed out to some old works by P. Lancaster, whereas V. B. Khazanov recalled a small-rank modification exploited in [PW86]. Such earlier applications, although important, seem to be rather sporadic, and we are not aware of any attempt of studying small-rank modifications systematically as additive preconditioners for solving linear systems of equations.

1.3 Organization of the paper

We organize our paper as follows. After the definitions in Section 2, we briefly examine additive preconditioning based on the Sherman–Morrison–Woodbury formula in Section 3, apply small-rank modifications to computing the null space vectors and solving a linear system of equations in Sections 4 and 5, examine their affect on the singularity, the conditioning and the eigensystem of an input matrix in Section 6, and extend their applications to the inverse iteration for the eigenproblem in Section 7. We estimate the perturbations and errors for our modified versions of the inverse iteration in Section 8 and report the results of our numerical tests in Section 9. Our numerical 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.

Acknowledgements: Victor Pan is grateful to the Institute of Numerical Analysis of the Russian Academy of Sciences (particularly to E. E. Tyrtyshnikov, S. A. Goreinov, and N. L. Zamarashkin), for providing him with the access to the computer and library facilities during his visit to the Institute in 2005, and to the participants of the cited Conferences in Moscow and Santander (particularly to J. W. Demmel, G. H. Golub, V. B. Khazanov, L. Reichel, and M. Van Barel), for helpful comments.

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 $g.m._A(\mu) = \text{Null } A(\mu) \le m(\mu)$. 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 $A = \prod_{i=1}^{h} A(\lambda_i)$. $\text{cond}_2 A$ for a possibly nonsquare matrix A of a rank r is the ratio $\sigma_1(A)/\sigma_r(A)$ of its largest and smallest singular values. $\text{cond}_2 A = ||A||_2 / ||A^{-1}||_2$ for a nonsingular matrix A.

"ops" stands for "arithmetic operations", "CG" for "Conjugate Gradient", "CGN" for "Conjugate Gradient Normal", "IPI" for "inverse power iteration", and "IR–RI" for "inverse Rayleigh–Ritz iteration".

Random sampling of a subset Σ_0 from a finite set Σ is the selection of the elements of Σ_0 from Σ at random, independently from each other and under the uniform probability distribution on Σ.

We need the following lemma in [DL78] (also in [Z79], [S80]).

Lemma 2.1. *Suppose that a multivariate polynomial of degree d does not vanish identically. Let the values of its variables be randomly sampled from a fixed finite set Σ. Then the polynomial vanishes with a probability of at most $\frac{d}{|\Sigma|}$.*

3 Additive preconditioning based on the Sherman–Morrison–Woodbury formula

Matrix inversion is not as omnipresent in computational practice as solving linear systems of equations but is important, e. g., in block matrix computations and computations in statistics. Furthermore if a the matrix A has either displacement structure or rank structure, also called quasi-separable structure (see [P01], [EG03], [V04] and the bibliography therein), then the inverse A^{-1} is expressed via its generator made up of the solution vectors to a small number of linear systems of equations with the coefficient matrix A. In such cases one can first compute a generator for the inverse $S^{-1} = (A + UV^H)^{-1}$ of the auxiliary matrix S and then apply the Sherman–Morrison–Woodbury formula to obtain a generator for the matrix $A^{-1} = S^{-1} + S^{-1}U(I_k - V^H S^{-1} U)^{-1} V^H S^{-1}$. The modification with the matrix UV^H of a smaller rank can simplify the inversion of the matrix S; the numerical stability problems, however, can be serious in the subsequent transition to the matrix A^{-1}. Thus we propose a distinct approach in the next sections.

4 The null spaces and small rank modifications

Theorem 4.1. *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*

$$LN(A) \subseteq \operatorname{range}(S^{-H}V), \tag{4.1}$$

$$RN(A) \subseteq \operatorname{range}(S^{-1}U). \tag{4.2}$$

If $k = n - r$, *then*

$$LN(A) = \operatorname{range}(S^{-H}V), \tag{4.3}$$

$$RN(A) = \operatorname{range}(S^{-1}U), \tag{4.4}$$

$$V^H S^{-1} U = I_k. \tag{4.5}$$

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

$$\mathbf{x}^H = (\mathbf{x}^H U)V^H S^{-1}. \tag{4.6}$$

This proves (4.1).

Now let $k = n-r = \operatorname{Null} A$. To deduce equation (4.3) from (4.1), 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 (4.2) and (4.4), apply (4.1) and (4.3) 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 (4.5) (assuming $k = n-r$), post-multiply equation (4.6) by U, apply equation (4.3), and deduce that $V^H S^{-1} U(V^H S^{-1} U - I_k) = 0$. (4.5) 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 (4.3) 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 4.1. *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 (4.3) and (4.4).

Theorem 4.2. *Under the assumptions of Theorem 4.1, 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 (4.3) and (4.4) 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 4.2 follows from the equations

$$S^{-1} - (S + E)^{-1} = (S + E)^{-1}ES^{-1} = S^{-1}E(S + E)^{-1}. \tag{4.7}$$

□

5 Computations in the null space and solving ill-conditioned linear systems of equations

5.1 Computations in the null space

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

$$S^H \mathbf{x} = \mathbf{v}, \ S\mathbf{y} = \mathbf{u} \tag{5.1}$$

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 4.1 we have $\mathbf{x} \in LN(A)$, $\mathbf{y} \in RN(A)$, and Theorem 4.2 bounds the approximation errors where the input matrix A is perturbed.

Unless the matrix A is sparse and/or has structure, we choose

$$S = A + \sigma UV^H \tag{5.2}$$

for random $n \times k$ unitary matrices U and V, k equal to the nullity of A, and a scalar parameter σ specified in the next section.

If the matrix A is sparse and/or has structure, we are challenged to choose the matrices U and V such that the sparsity and/or structure are preserved or enhanced in the transition to the matrix $S = A + UV^H$, so that we can apply the effective algorithms in [BBCD93], [DDSV98], [DER86], [CN96], [CN99], [PVWC04], [PKRKa] to solve systems (5.1). For example, we choose random matrix U and write $V = U$ if $A = A^H$ is a Hermitian matrix; if A is an $n \times n$ Toeplitz matrix, we let $UV^H = F^{-1}\,\text{diag}(D_k, 0_{n-k})F$ where F is the $n \times n$ matrix of the discrete Fourier transform and D_k is the $k \times k$ random diagonal matrix.

If the rank r of the matrix A is not known, we can recursively add the outer products $\mathbf{u}_i \mathbf{v}_i^H$ to the matrix A for $i = 1, \dots, k$ until the resulting sum defines a nonsingular matrix.

We can extend this approach to computing a pair of $n \times k$ unitary matrices X and Y 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.

If the matrix S is nonsingular but ill-conditioned, we keep adding the outer products $\mathbf{u}_i \mathbf{v}_i^H$ for $i = k+1, \dots, q$ to the matrix S until it becomes well-conditioned (see Remark 5.1). Then we solve q linear systems of equations with the resulting matrix S and finally recover a desired vector in the null space in $O(nq^2)$ ops based on relations (4.1) and (4.2).

5.2 Solving ill-conditioned linear systems of equations

Let us extend our approach to the solution of an ill-conditioned linear system of equations

$$M\mathbf{z} = \mathbf{b} \tag{5.3}$$

where with no loss of generality we assume that $||\mathbf{b}||_2 \approx 1, ||M||_2 \approx 1$. We first reduce the solution to computing the vector $\mathbf{w}^H = (-1, \mathbf{z}^H)$, which is a normalized vector in the right null space $RN(A)$ for the matrix

$$A = \mathrm{diag}(0, M) + \tilde{\mathbf{b}}\mathbf{e}_1^H \tag{5.4}$$

where

$$\tilde{\mathbf{b}}^H = (0, \mathbf{b}^H) \tag{5.5}$$

and $\mathbf{e}_1$ is the first coordinate vector of the dimension $n+1$ representing the first column of the matrix I_{n+1}. This computation is further reduced to solving k linear systems of the form $S\mathbf{y} = \mathbf{u}$ in (5.1) for $S = A + UV^H$ in (5.2), $\sigma = 1$, n replaced with $n+1$,

$$U = (\mathbf{e}_1 - \tilde{\mathbf{b}}, \tilde{W}),\ V = (\mathbf{e}_1, \tilde{Z}),\ \tilde{W}^H = (\mathbf{0}, W)^H,\ \tilde{Z}^H = (\mathbf{0}, Z)^H, \tag{5.6}$$

(random) $n \times k$ unitary matrices W and Z, a proper vector $\mathbf{u} \in$ range U, and k being the minimum positive integer for which the matrix S is well-conditioned (see Remark 5.1).

If $M = M^H$ is a Hermitian matrix, then we choose S in (5.2) for $\sigma = 1$, n replaced with $n+1$,

$$U = (\mathbf{e}_1, T),\ V = (\tilde{\mathbf{b}} + g\mathbf{e}_1, T),\ \tilde{\mathbf{b}}^H = (0, \mathbf{b}^H), \tag{5.7}$$

(random) $(n+1) \times k$ unitary matrix T, and a real parameter g, so that S is also a Hermitian matrix.

Theorem 5.1. *For a positive definite matrix M and a positive scalar g, the above matrix S is positive definite.*

Proof. Express the matrix S as follows: $S = \mathrm{diag}(g, M) + TT^H + B$ where

$$B = \begin{pmatrix} 0 & \mathbf{b} \\ \mathbf{b}^H & 0 \end{pmatrix}. \tag{5.8}$$

We have $\det(\lambda I - B) = \lambda^n - \mathbf{b}^H\mathbf{b}\lambda^{n-1}$, and so the matrix B is nonnegative definite. Clearly, so is the term TT^H as well. The theorem follows because the matrix $\mathrm{diag}(M, g)$ is positive definite. □

Remark 5.1. *To complete our definintion of the integer parameter k, we can specify a suitable iterative algorithm (such as the CG or CGN algorithms [BBCD93], [SvV00], [vV03]) and define that a linear system is well-conditioned if the algorithm converges to its solution within h steps for a fixed h.*

6 How small-rank modifications of a matrix affect its singularity, conditioning, and eigensystem

6.1 The affect on the singularity

Let us show that the matrix S in (5.2) is not likely to be singular if the matrices U and V are random.

Theorem 6.1. *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) $\operatorname{rank} S \le r + k$,

b) $\operatorname{rank} S = r + k$ *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) $\operatorname{rank} S = r + k$ *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 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 Lemma 2.1. Part c) is proved similarly to part b). □

6.2 The affect on the conditioning of computations in a null space

Suppose that the matrix A is singular but well-conditioned. In virtue of Theorem 6.1, the matrix S is unlikely to be singular for a random choice of the matrices U and V, but is it likely to be well-conditioned? To estimate the ratio $(\operatorname{cond}_2 S)/(\operatorname{cond}_2 A)$ from above, we first factorize the matrix S.

Theorem 6.2. *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 6.1. *Under the assumptions of Theorem 6.2 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 6.3. *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

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

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

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

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

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 6.2. *Under the assumptions and definitions of Theorem 6.2, if the matrices U and V are unitary and $||A^{-1}||_2 \ge 1$, $||A||_2 \ge 1$, then* $\operatorname{cond}_2 S \le 2(1 + \sqrt{2}||U_k^{-1}||_2)(1 + \sqrt{2}||V_k^{-1}||_2) \operatorname{cond}_2 A$.

Proof. Combine Corollary 6.1 and Theorem 6.3 and note that

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

□

Corollary 6.2 guides us in choosing the scaling parameter σ to 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 indeed showed such an impact of scaling on the ratio $r = \frac{\text{cond}_2 S}{\text{cond}_2 A}$ (see Section 9.1).

In the above analysis we have assumed the $n \times k$ matrices U and V where k is the nullity of A. To yield well-conditioned matrices S, required in particular in Subsection 5.2, we generally need to choose k equal to the nullity of a well-conditioned matrix $A - E$ of a smaller rank; there must exist such a matrix near the matrix A [GL96]. Thus we can extend our analysis and deduce that the matrix $S - E = (A - E) + UV^H$ is likely to be well-conditioned as well. If $\delta = ||E||$ is small, the matrix S is also well-conditioned due to the bound $||S||_2 \le \delta + ||S - E||_2$ and equation (4.7).

6.3 The affect on the conditioning of a linear system

Our study in the two previous subsections can be extended to the solution of a linear system of equations in Subsection 5.2 based on the modification of the input matrix with random matrices T, W and Z. The matrices $S - \tilde{W}\tilde{Z}^H$ and (in the Hermitian case) $S - TT^H$ are not random, however, and so we should estimate the ratios $r_1 = \frac{\text{cond}_2(S - \tilde{W}\tilde{Z}^H)}{\text{cond}_2 M}$ and (in the Hermitian case) $r_2 = \frac{\text{cond}_2(S - TT^H)}{\text{cond}_2 M}$. We immediately observe that $S - WZ^H = \text{diag}(1, M)$ and therefore $r_1 = 1$ if $||M||_2 = 1$. To estimate r_2, recall that $S - TT^H = \text{diag}(g, M) + B$ for the Hermitian and nonnegative definite matrix B in (5.8). Therefore,

$$||S - TT^H||_2 \le ||\,\text{diag}(g, M)||_2 + ||B||_2, ||(S - TT^H)^{-1}||_2 \le ||\,\text{diag}(g^{-1}, M^{-1})||_2.$$

Recall that $\det(xI - B) = x^{n+1} - \mathbf{b}^H \mathbf{b} x^n$, and so $||B||_2 = ||[\mathbf{b}]||_{\mathbf{2}}$. We have $\text{cond}_2 \,\text{diag}(g, M) = \text{cond}_2 M$ if $1/||M^{-1}||_2 \le g \le ||M||_2$. Therefore if we choose the positive parameter g satisfying the latter inequalities, then $r_2 \le 1 + ||\mathbf{b}||_2/||M||_2$.

6.4 The affect on the eigenspaces

By definition, the eigenspaces of a matrix A associated with its eigenvalue λ are just the null spaces $LN(\lambda I - A)$ and $RN(\lambda I - A)$. Therefore, Theorems 4.1 and 4.2 describe the affect of small-rank modifications on the eigenspaces of a matrix. Let us specify the case of a rank-one modification of a diagonalizable matrix A.

Corollary 6.3. *Suppose* $A = G_A D_A^{-1} G_A$, $S = A + \mathbf{u}\mathbf{v}^H = G_S D_S^{-1}$ *where* G_A, D_A, G_S, D_S *are* $n \times n$ *matrices, the matrices* G_A *and* G_S *are nonsingular,* $D_A = \mathrm{diag}(\lambda_i(A))_{i=1}^n$, $D_S = \mathrm{diag}(\lambda_i(S))_{i=1}^n$, *and both matrices* A *and* S *have only simple eigenvalues. Then*

$$G_A \mathbf{e}_i = c_i(\lambda_i(A)I - S)^{-1}\mathbf{u}, \quad G_S \mathbf{e}_i = d_i(\lambda_i(S)I - A)^{-1}\mathbf{u},$$
$$\mathbf{e}_i^H G_A = f_i \mathbf{v}^H(\lambda_i(A)I - S)^{-1}, \quad \mathbf{e}_i^H G_S = g_i \mathbf{v}^H(\lambda_i(S)I - A)^{-1}$$

where

$$c_i = -\mathbf{v}^H G_A \mathbf{e}_i, \ d_i = \mathbf{v}^H G_S \mathbf{e}_i,$$
$$f_i = -\mathbf{e}_i^H G_A^{-1}\mathbf{u}, \ g_i = \mathbf{e}_i^H G_S^{-1}\mathbf{u},$$

$i = 1, \ldots, k$.

Proof. The corollary follows from Theorem 4.1 (cf. equation (4.6)) because both matrices A and S are rank-one modifications of one another. □

6.5 The affect on the eigenvalues

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

For $A = A(\lambda)$ and a scalar λ, let $g.m._A(\lambda) = k$. Then matrix equation (4.5) turns into the system of k^2 rational equations

$$F(\lambda) = I_k - V^H S^{-1} U = 0_k \tag{6.1}$$

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 (6.1) 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 6.4. *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,*

$$c_A(\lambda) = c_S(\lambda) r(\lambda), \ r(\lambda) = \det(I_k - V^H S^{-1} U). \tag{6.2}$$

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 6.1. *For $k = 1$, $U = \mathbf{u}$, and $V = \mathbf{v}$, equation (6.2) 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].

Recall that $\frac{d(S^{-1})}{d\lambda} = -S^{-1}\frac{dS}{d\lambda}S^{-1}$ and obtain expressions for the derivatives and higher order derivatives of the functions $F(\lambda), f(\lambda)$, and $r(\lambda)$.

Finally, let us specify the impact of small-rank perturbations of a matrix on the geometric multiplicity of its eigenvalues.

Theorem 6.5. *Let $g.m._A(\lambda) = h$ for an $n \times n$ matrix polynomial $A(\lambda)$ and a scalar λ. Let $S(\lambda) = A(\lambda) + UV^H$ for a pair of $n \times k$ matrices U and V. Then*

a) $g.m._S(\lambda) \geq h - k$ if $h > k$, whereas

b) λ is an eigenvalue of $S(\lambda)$ with a probability of at most $\frac{2k}{|\Sigma|}$ provided $h \leq k$ and the entries of the matrices U and V have been randomly sampled from a set Σ of cardinality $|\Sigma|$.

Proof. Theorem 4.1 implies part a). It also implies that λ is not an eigenvalue of $S(\lambda)$, that is, $\det S(\lambda) \neq 0$, provided $h \leq k$ and the matrices U and V are generic. Since $\det S(\lambda) = \det(A(\lambda) + UV^H)$ is a polynomial of degree of at most $2k$ in the entries of the matrices U and V, part b) follows from Lemma 2.1. □

7 Inverse iteration with small-rank modifications

7.1 Inverse iteration and our modifications

As a characteristic and important example of the applications of additive preconditioning to eigen-solving, we recall the inverse power iteration, which is a classical tool for the refinement of a crude 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 an accurate eigenvector as the solution of an ill-conditioned linear systems of equations. This is not completely painless, however. 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." Our next goal is to counter this deficiency by applying additive preconditioning.

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 $\text{range}\, Y = RN(A(\lambda))$.

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.

7.2 Inverse iteration with rank-one modifications

Algorithm 7.1. 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* $||A(\tilde{\lambda})\mathbf{y}||_2 \leq \sigma t$ *(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.*

7.3 Inverse iteration with small-rank modifications

Algorithm 7.1 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 $\mathbf{y}$), or because λ is a geometrically multiple eigenvalue of the matrix $A(\lambda)$ or lies near its another eigenvalue. We can modify Algorithm 7.1 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 7.2. Inverse iteration with small-rank modifications.

INPUT: *as in Algorithm 7.1.*

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

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* $S(\tilde{\lambda}) \longleftarrow \frac{1}{\sigma}A(\tilde{\lambda}) + Y_i V_i^H$, *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* $\mathbf{v}^H V_i = \mathbf{0}$ *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* $Y_{i+1}V_{i+1}^H = Y_i V_i^H + \mathbf{y}\mathbf{v}^H$*),* $i \longleftarrow i + 1$, $COUNTER \longleftarrow COUNTER + 1$, *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* $\mathrm{trace}(Y^H A(\tilde{\lambda})Y) = 0$ *which minimizes the residual norm* $||A(\tilde{\lambda})Y||_2$. *(*$\tilde{\lambda} = \mathrm{trace}\,\frac{Y^H MY}{Y^H NY}$ *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 7.1. *Application of Algorithm 7.1 (resp. 7.2) 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 7.2. *In some cases we can simpify the computations by properly selecting the vectors* $\mathbf{v}$ *in Algorithms 7.1 and 7.2. 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* $\mathbf{v} = \mathbf{y}$. *The latter choice is motivated in Remark 8.1.*

Remark 7.3. *For some bad choices of the vectors* $\mathbf{v}$ *and* $\mathbf{y}$ *in Algorithm 7.2, 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.*

7.4 Further extensions

We can extend Algorithm 7.2 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 7.2 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 7.2.

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.

Finally, we can extend the application of additive preconditioning within the IPI/IR–RI, by combining small-rank modifications with any eigen-solver which involves the solution of an auxiliary ill-conditioned linear system of equations. This includes the Jacobi–Davidson algorithm, the shift-and-invert enhancements of the Arnoldi and Lanczos algorithms [S98], and the deflation stage of the QR algorithm.

8 Perturbations and errors

For $\lambda \approx \lambda$ Algorithms 7.1 and 7.2, 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 7.1 and 7.2.

Theorem 8.1. *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 8.2. *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) $\tilde{S}^{-1}\tilde{Y} = YB^{-1} - \tilde{S}^{-1}\delta Y B^{-1}$.

b) Furthermore, $\tilde{S}^{-1}\tilde{Y} = YB^{-1}(I - \gamma) + \tilde{S}^{-1}Y\gamma^2 + \tilde{S}^{-1}\Delta\gamma$ *provided* $Y\gamma = \delta Y B^{-1}$.

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 $SY = (A + \tilde{Y}V^H)Y = \tilde{Y}(V^H Y) = \tilde{Y}B$, $S^{-1}\tilde{Y} = YB^{-1}$. Therefore,

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

This proves part a).

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

Substitute

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

on the right-hand side and obtain that

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

This proves part b).

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

Corollary 8.1. *Under the assumptions of Theorem 8.2, we have* $||S^{-1}\tilde{Y} - YB^{-1}|| = O((||\Delta|| + ||\gamma||)||\gamma||)$ *provided* $\text{range}(\delta Y) \subseteq \text{range}\, Y = RN(A)$.

Theorem 8.1 and Corollary 8.1 together imply local quadratic convergence of Algorithms 7.1 and 7.2.

Remark 8.1. *With the choice of unitary matrix* $V = \tilde{Y}$, *we have* $B = I_k$, *which enables better control over the error bound in Theorem 8.2.*

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 the 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 $\frac{\text{cond}_2\, S_k(\rho)}{\text{cond}_2\, A}$. 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 6.2), 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 choose 100×100 symmetric and nonnegative definite matrices A of rank 99 and let $\mathbf{y}_i = \mathbf{x}_i$ and consequently $\mathbf{u}_i = \mathbf{v}_i$ for random vectors $\mathbf{x}_i$, $i = 1, 2, 3$. To generate the matrices A we followed the recipe of case 1 in Subsection 9.2 for $n = 100$ and $S = Q$, $Q^TQ = I$, except that we replace with zero the leading principal (north-eastern) entry of the matrix D.

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 7.1

In Tables 9.5 and 9.6 we show the numbers of iterations required for the convergence of the IPI and Algorithm 7.1. We display the average (mean) values and standard deviations in 200 tests with $n \times n$ matrices $M = G^{-1}TG$ for $n = 64$ and $n = 100$, G being either a random matrix or the Q-factor in QR factorization of a random matrix, and 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 7.1 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 $\mathbf{y}_i = \mathbf{x}_i$, $i = 1, 2, 3$

rho \ k		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.

[C05] K. Chen, *Matrix Preconditioning Techniques and Applications*, Cambridge University Press, Cambridge, England, 2005.

[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.

[EG03] Y. Eidelman, I. Gohberg, Fast Inversion Algorithms for a Class of Structured Operator Matrices, *Linear Algebra and Its Applications*, **371**, 153-190, 2003.

[GH90] J. R. Gilbert, H. Hafsteinsson, Parallel Symbolic Factorization of Sparse Linear Systems, *Parallel Computing*, **14**, 151–162, 1990.

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

[GS92] J. R. Gilbert, R. Schreiber, Highly Parallel Sparse Cholesky Factorization, *SIAM J. on Scientific Computing*, **13**, 1151–1172, 1992.

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

[H64] A. S. 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.

[LRT79] R. J. Lipton, D. Rose, R. E. Tarjan, Generalized Nested Dissection, *SIAM J. on Numerical Analysis*, **16**, **2**, 346–358, 1979.

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

[P01] V. Y. Pan, *Structured Matrices and Polynomials: Unified Superfast Algorithms*, Birkhäuser/Springer, Boston/New York, 2001.

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

[PR93] V. Y. Pan, J. Reif, Fast and Efficient Parallel Solution of Sparse Linear Systems, *SIAM J. on Computing*, **22**, **6**, 1227–1250, 1993.

[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.

[PW86] G. Peters, J. H. Wilkinson, Computing the Generalized Singular Value Decompositions, *SIAM J. on Sientific and Statistical Computing*, **4**, 1112–1146, 1986.

[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.

[SS86] Y. Saad, M. N. Schultz, GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems, *SIAM J. on Sientific and Statistical Computing*, **7**, 856–869, 1986.

[SvV00] Y. Saad, H. A. van der Vorst, Iterative Solution of Linear Systems in the 20th Century, *J. of Computational and Applied Math.*, **123**, 1–33, 2000.

[TB97] L. N. Trefethen, D. Bau, III, *Numerical Linear Algebra*, SIAM, Philadelphia, 1997.

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

[V04] R. Vandebril, Semiseparable Matrices and the Symmetric Eigenvalue Problem, PhD Thesis, *Katholieke Universiteit Leuven, Departement Computerwetenschappen*, Leuven, Belgium, 2004.

[vV03] H. A. van der Vorst, *Iterative Krylov Methods for Large Linear Systems,* Cambridge University Press, Cambridge, England, 2003.

[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.