2014

TR-2014006: New Algorithms in the Frobenius Matrix Algebra for Polynomial Root-Finding

Victor Y. Pan

Ai-Long Zheng

New Algoirthms in the Frobenius Matrix Algebra for Polynomial Root-finding *

Victor Y. Pan[1,2],[a] and Ai-Long Zheng[2],[b]
Supported by NSF Grant CCF-1116736 and PSC CUNY Award 64512–0042

[1] Department of Mathematics and Computer Science
Lehman College of the City University of New York
Bronx, NY 10468 USA
[2] Ph.D. Programs in Mathematics and Computer Science
The Graduate Center of the City University of New York
New York, NY 10036 USA
[a] victor.pan@lehman.cuny.edu
http://comet.lehman.cuny.edu/vpan/
[b] azheng-1999@yahoo.com

Abstract

In 1996 Cardinal applied fast algorithms in Frobenius matrix algebra to complex root-finding for univariate polynomials, but he resorted to some numerically unsafe techniques of symbolic manipulation with polynomials at the final stages of his algorithms. We extend his work to complete the computations by operating with matrices at the final stage as well and also to adjust them to real polynomial root-finding. Our analysis and experiments show efficiency of the resulting algorithms.

2000 Math. Subject Classification: 65H05, 65F15, 30C15, 26C10, 12Y05

KEYWORDS: Polynomial, Root-finding, Eigen-solving, Companion matrix, Dominant eigenspaces, Real roots.

1 Introduction

Polynomial root-finding is the oldest subject of mathematics and computational mathematics and is still an area of intensive research worldwide. The list of hundreds if not thousands algorithms known for this task still grows every year (see the books and articles [2], [3], [12], [23], [24], [19], [20], [21], [28], and the bibliography therein). Many algorithms are directed to computing a single, e.g., absolutely largest root of a polynomial or a subset of all its n roots, e.g., all r its real roots. In some applications, e.g., to algebraic geometric optimization, only the real roots are of interest, and they can be much less numerous than all n complex roots. Nevertheless the best numerical subroutines such as MPSolve approximate all these r real roots about as fast and as slow as all n complex roots.

Root-finding for a polynomial $p(x)$ via eigen-solving for the associated companion matrix C_p is a classical approach recently revived, with the incorporation of the well developed numerical matrix

*Some results of this paper have been presented at the 14th Annual Conference on Computer Algebra in Scientific Computing (CASC 2012), September 2012, Maribor, Slovenia, and at the 18th Conference of the International Linear Algebra Society (ILAS'2013), Providence, RI, June 2013

methods (see [5], [15], [31], [35], and the bibliography therein). The QR algorithm, adopted for polynomial root-finding by Matlab, avoids numerical problems, faced by many other companion matrix methods [15, Section 7.4.6], but is not readily amenable to exploiting the rich structure of the companion matrix. Extensive research toward such exploitation by using QR- and LR-based root-finders has been initiated in the papers [8], [9] and [6] and still goes on (see [4], [33], [37], [1], and the references therein). The QR algorithm is celebrated for its fast empirical convergence, and it has been modified to exploit the structure of the companion matrix to approximate all its eigenvalues in quadratic (versus the classical cubic) arithmetic time.

We decrease the arithmetic cost to nearly linear by extending the approach of Cardinal [11] (cf. also [10] and [26]). As in the algorithms of [11], [10] and [26] we apply repeated squaring, approximate the matrix sign function, and perform fast computations in the Frobenius algebra, generated by the $n \times n$ companion matrix C_p of the input polynomial. Unlike these papers, however, we keep the arithmetic cost nearly linear without the transition to numerically unsafe computations with polynomials at the final stage of the root approximation. Instead of the computation of polynomial GCDs, used in [11], we approximate a dominant eigenspace of appropriate functions of the companion matrix, recall that this eigenspace is shared with the latter matrix, and then approximate its associated eigenvalues. We also adjust the matrix sign iterations to approximate the real roots of the input polynomial.

We organize our presentation as follows. The next section is devoted to definitions and preliminary results. In Section 3 we reduce the eigenvalue problem to the approximation of the dominant eigenspaces of appropriate functions of the input matrix. In Sections 4 and 5 we compute such matrix functions. In Section 4 we repeatedly square the companion matrix to achieve the domination of the images of its absolutely largest eigenvalues. In Section 5 we explore the matrix sign iteration to achieve the domination of the images of real eigenvalues. Section 6 covers our numerical tests, which are the contribution of the second author. The test results are in reasonably good accordance with our formal analysis. Our concluding Section 7 lists some directions for further study.

2 Definitions and preliminaries

Hereafter "flop" stands for "arithmetic operation", "is expected" and "is likely" mean "with a probability near 1", and "small", "large", "close", and "near" are meant in the context. We assume computations in the fields of complex and real numbers $\mathbb{C}$ and $\mathbb{R}$, respectively. For $\rho' > \rho > 0$ and a complex c, define the circle $\mathcal{C}_\rho(c) = \{\lambda : \ |\lambda - c| = \rho\}$ and the disc $\mathcal{D}_\rho(c) = \{\lambda : \ |\lambda - c| \le \rho\}$. A scalar λ is *nearly real* (within $\epsilon > 0$) if $|\Im(\lambda)| \le \epsilon|\lambda|$.

Matrix computations: fundamentals [15], [30], [34]. M^T is the transpose of a matrix M. $\mathcal{R}(M)$ is the range of a matrix M, that is the linear space generated by its columns. $\mathcal{N}(M) = \{\mathbf{v} : \ M\mathbf{v} = \mathbf{0}\}$ is its null space. $\mathrm{rank}(M) = \dim(\mathcal{R}(A))$. A matrix of full column rank is a *matrix basis* of its range. $I = I_n = (\mathbf{e}_1 \mid \mathbf{e}_2 \mid \ldots \mid \mathbf{e}_n)$ is the $n \times n$ identity matrix with columns $\mathbf{e}_1, \ \mathbf{e}_2, \ldots, \mathbf{e}_n$. $J = J_n = (\mathbf{e}_n \mid \mathbf{e}_{n-1} \mid \ldots \mid \mathbf{e}_1)$ is the $n \times n$ reflection matrix, $J^2 = I$. $O_{k,l}$ is the $k \times l$ matrix filled with zeros. A matrix Q is called *orthogonal* if $Q^TQ = I$ or $QQ^T = I$.

We use the matrix norms $||\cdot||_h$ for $h = 1, 2, \infty$ [15, Section 2.3] and write $||\cdot|| = ||\cdot||_2$. We write $a \approx 0$ and $A \approx O$ if the values $|a|$ and $||A||$ are small in context. We write $a \approx b$ for $b \neq 0$ and $A \approx B$ and $B \neq O$ if the ratios $|a|/|b|$ and $||A||/||B||$ are small.

M^+ is the Moore–Penrose pseudo inverse of M [15, Section 5.5.4]. An $n \times m$ matrix $X = M^{(I)}$ is a left (resp. right) inverse of an $m \times n$ matrix M if $XM = I_n$ (resp. if $MY = I_m$). M^+ is an $M^{(I)}$ for a matrix M of full rank. $M^{(I)} = M^{-1}$ for a nonsingular matrix M.

Matrix computations: eigenspaces [15], [31], [34], [35], [5]. $\mathcal{S}$ is an *invariant subspace* or *eigenspace* of a square matrix M if $M\mathcal{S} = \{M\mathbf{v} : \mathbf{v} \in \mathcal{S}\} \subseteq \mathcal{S}$.

Theorem 2.1. *[31, Theorem 4.1.2], [34, Section 6.1], [35, Section 2.1]. Let $U \in \mathbb{C}^{n \times r}$ be a matrix basis for an eigenspace $\mathcal{U}$ of a matrix $M \in \mathbb{C}^{n \times n}$. Then the matrix $L = U^{(I)}MU$ is unique (that is independent of the choice of the left inverse $U^{(I)}$) and satisfies $MU = UL$.*

The above pair $\{L,\mathcal{U}\}$ is an eigenpair of a matrix M, L is its eigenblock, and $\mathcal{U}$ is the associated eigenspace of the matrix L [31]. If $L = \lambda I_n$, then also $\{\lambda,\mathcal{U}\}$ is called an eigenpair of a matrix M. In this case $\det(\lambda I - M) = 0$, whereas $\mathcal{N}(M - \lambda I)$ is the eigenspace associated with the eigenvalue λ, made up of its eigenvectors. $\Lambda(M)$ is the set of all eigenvalues of M, called its spectrum. $\rho(M) = \max_{\lambda\in\Lambda(M)} |\lambda|$ is the spectral radius of M. Theorem 2.1 implies that $\Lambda(L) \subseteq \Lambda(M)$. For an eigenpair $\{\lambda,\mathcal{U}\}$ write $\psi = \min\ |\lambda/\mu|$ over $\lambda \in \Lambda(L)$ and $\mu \in \Lambda(M) - \Lambda(L)$. Call the eigenspace $\mathcal{U}$ dominant if $\psi > 1$, dominated if $\psi < 1$, strongly dominant if $1/\psi \approx 0$, and strongly dominated if $\psi \approx 0$.

We readily verify the following results.

Theorem 2.2. *Suppose M is a square matrix, $\mathcal{U}$ is its eigenspace, and a rational function $f(M)$ is defined on the spectrum of the matrix M. Then $\Lambda(f(M)) = f(\Lambda(M))$, and $\mathcal{U}$ is an eigenspace of the matrix $f(M)$.*

Polynomials and companion matrices. Write

$$p(x) = \sum_{i=0}^{n} p_i x^i = p_n \prod_{j=1}^{n} (x - \lambda_j), \tag{2.1}$$

$$p_{\rm rev}(x) = x^n p(1/x) = \sum_{i=0}^{n} p_i x^{n-i} = p_n \prod_{j=1}^{n} (1 - x\lambda_j), \tag{2.2}$$

$$C_p = \begin{pmatrix} 0 & & & & -p_0/p_n \\ 1 & \ddots & & & -p_1/p_n \\ & \ddots & \ddots & & \vdots \\ & & \ddots & 0 & -p_{n-2}/p_n \\ & & & 1 & -p_{n-1}/p_n \end{pmatrix}$$

and $C_{p_{\rm rev}} = JC_pJ$ are the $n \times n$ companion matrices of the polynomials $p(x) = \det(xI_n - C_p)$ and $p_{\rm rev}(x) = \det(xI_n - C_{p_{\rm rev}})$, respectively.

Fact 2.1. *(See [11] or [26].) The companion matrix $C_p \in \mathbb{C}^{n\times n}$ of a polynomial $p(x)$ of (2.1) generates a matrix algebra $\mathcal{A}_p$. One needs $O(n)$ flops for addition, $O(n \log n)$ flops for multiplication, and $O(n \log^2 n)$ flops for inversion in this algebra.*

3 Approximation of the dominant eigenvalues

Algorithm 3.1. Dominant Eigen-solving with a Random Multiplier.

INPUT: *an $n \times n$ matrix M, its rational function $\phi(M)$ having a dominant eigenspace $\mathcal{U}$ (shared with the matrix M, but is not assumed to be dominant for it), an upper bound r_+ on an unknown dimension r of the eigenspace $\mathcal{U}$, and a positive integer h.*

OUTPUT: *Approximations to the eigenvalues of the matrix M associated with the eigenspace $\mathcal{U}$.*

COMPUTATIONS:

1. *Generate a standard Gaussian random $n \times r_+$ matrix G and compute the $n \times r_+$ matrix $(\phi(M))^h G$.*
2. *Compute the Q factor of the rank revealing QR factorization of the matrix $\phi(M)G$.*
3. *Compute the left inverse $\widehat{U}^{(I)}$ of the matrix $\widehat{U}$.*
4. *Compute the matrix $\widehat{L} = \widehat{U}^{(I)}\phi(M)\widehat{U}$.*
5. *Apply the QR algorithm to the matrix $\widehat{L}$ and output its eigenvalues.*

Stage 2 amounts to h steps of the Power Method [15], [31]. By extending the analysis in [17] we readily deduce that even for $h = 1$ the matrix Q computed at Stage 2 is expected to approximate an orthogonal matrix basis $\widehat{U}$ for the dominant eigenspace of the matrix $\phi(M)$ provided that the matrix $\phi(M)$ has a strongly dominant eigenspace. Now correctness of our randomized algorithm follows from Theorem 2.2. At Stage 1 we generate nr_+ i.i.d. Gaussian random values, but instead we can employ r_+ uniform i.i.d. choices among the n coordinate vectors by using SRFT multipliers, at the price of only a minor deterioration of the probability of obtaining a desired approximation of the matrix basis (see the definitions and results on using SRFT matrices in [17, Section 11]). By virtue of Fact 2.1 we only need $O(hnr_+ \log n)$ flops for computing the product $\phi(M)G$ at Stage 1 provided that $\phi(M) \in \mathcal{A}_p$. $O(nr_+^2)$ flops are used at its Stages 2, 3 and 4, and $O(r_+^3)$ flops at Stage 5. Overall we use $O(((r_+ + h \log n)n + r_+^2)r_+)$ flops at Stages 1–5, not counting the cost of generating the random parameters.

Remark 3.1. *We can approximate the dominated eigenvalues of a matrix $\phi(M)$ or a cluster of its eigenvalues isolated about a complex point μ by applying the above algorithms to the matrix $(\phi(M))^{-1}$ or $(\mu I - (\phi(M)))^{-1}$, respectively.*

Remark 3.2. *Seeking a single eigenvalue of a matrix M we can apply the Power (resp. Inverse Power) Method (cf. [15, Sections 7.3.1 and 7.6.1]) to approximate an eigenvector $\mathbf{v}$ of the matrix $\phi(M)$ in its dominant (resp. dominated) eigenspace $\mathcal{U}$. This eigenvector is shared with the matrix M by virtue of Theorem 2.2, and we can approximate the associated eigenvalue by the Rayleigh quotient $\mathbf{v}^T M\mathbf{v}/\mathbf{v}^T\mathbf{v}$ or a simple quotient $\mathbf{v}^T M\mathbf{e}_j/\mathbf{v}^T\mathbf{e}_j$ for a fixed or random integer j, $1 \le j \le n$. We can employ deflation or reapply our algorithm for other initial approximations to approximate the other eigenvalues of the matrix M (cf. Section 5.5).*

Remark 3.3. *The matrix $\widehat{L}$ has the size $r_+ \times r_+$ and shares r desired and $r_+ - r$ extraneous eigenvalues with the matrix M. For example, in numerical real eigen-solving the spectrum of the matrix $\widehat{L}$ consists of all real and nearly real eigenvalues of the matrix M. Having them approximated we can readily select among them the r real eigenvalues.*

4 Repeated squaring in the Frobenius algebra with simplified recovery of the eigenvalues

Clearly for sufficiently large integers k, the matrices M^k have dominant eigenspace $\mathcal{U}$ associated with the set of the absolutely largest eigenvalues of M. For a fixed or random real or complex shift s write $M_0 = M - sI$ and compute $M_0^{2^h}$ in h squarings,

$$M_{h+1} = a_h M_h^2,\ a_h \approx 1/||M_h||^2 \text{ for } h = 0, 1, \dots \tag{4.1}$$

Then for a sufficiently large integer h the matrix M_h has a dominant eigenspace, and Algorithm 3.1 is expected to approximate the dominant eigenvalues provided that we know an upper bound r_+ on the dimension of their eigenspace.

For $M = C_p$ we can follow [11] and apply the FFT-based algorithms that support Fact 2.1 to perform every squaring and every multiplication in $O(n \log n)$ flops. The bottleneck of that paper and its amelioration in [26] is the recovery of the roots of $p(x)$ at the end of the squaring process where $|\lambda_j| \approx |\lambda_k|$ for $j \ne k$. The paper [26] relieves some difficulties of [11] by employing approximations to the roots of $p'(x)$, $p''(x)$, etc., but these techniques are still too close to the symbolic recovery methods of the paper [11], which operates with polynomials and does not employ numerical linear algebra. In contrast Algorithm 3.1 reduces the computation of the r eigenvalues of a selected subset of the spectrum $\Lambda(M)$ to eigen-solving for the $r_+ \times r_+$ matrix $\widehat{L}$, and this computation is simple unless we know only a large upper bound on the number of the absolutely largest eigenvalues.

Now replace M_0 in (4.1) by $M_0 = (M - \sigma I)^{-1}$ for a fixed complex σ. Then Algorithm 3.1 approximates the dominant eigenspace of the matrix M_h for a large integer h and the associated set of the eigenvalues of M, which are the nearest to the point σ. E.g., this is the set of the absolutely

smallest eigenvalues where $\sigma = 0$. For $M = C_p$ we can alternatively write $M_0 = C_{p_{\rm rev}(x-\sigma)}$ in (4.1) to replace the inversion of the shifted companion matrix with the shift by σ of the variable x of the polynomial $p(x)$ and the reversion of the order of its coefficients.

5 The generation and computation of the dominant eigenspaces by approximating the matrix sign function

5.1 The matrix sign function: definition and basic properties

Definition 5.1. *For two real numbers $x \neq 0$ and y, the function sign $(x + y\sqrt{-1})$ is equal to 1 if $x > 0$ and is equal to -1 if $x < 0$.*

Definition 5.2. *(See [16].) Let $A = ZJZ^{-1}$ be a Jordan canonical decomposition of an $n \times n$ matrix A where $J = \mathrm{diag}(J_-, J_+)$, J_- is a $p \times p$ matrix and all its p diagonal entries have negative real parts, whereas J_+ is a $q \times q$ matrix and all its q diagonal entries have positive real parts. Then sign$(A) = Z\,\mathrm{diag}(-I_p, I_q)Z^{-1}$. Equivalently sign$(A) = A(A^2)^{-1/2}$ or sign$(A) = \frac{2}{\pi}A\int_0^\infty (t^2 I_n + A^2)^{-1}dt$.*

Definition 5.3. *Assume the matrices $A = ZJZ^{-1}$, J_- and J_+ above, except that $n = p + q + r$ and $J = \mathrm{diag}(J_-, J_0, J_+)$ for a $r \times r$ matrix J_0 whose all r diagonal entries have real parts 0. Then fix some $r \times r$ real diagonal matrix D_r, e.g., $D_r = O_{r,r}$, and define a* generalized matrix sign function *sign(A) by writing sign$(A) = Z\,\mathrm{diag}(-I_p, D_r\sqrt{-1}, I_q)Z^{-1}$.*

We have the following simple but basic results.

Theorem 5.1. *Assume the generalized matrix sign function sign(A) defined for an $n \times n$ matrix $A = ZJZ^{-1}$. Then for some real $r \times r$ diagonal matrix D_r we have*

$$I_n - \mathrm{sign}(A) = Z^{-1}\,\mathrm{diag}(2I_p, I_r - D_r\sqrt{-1}, O_{q,q})Z,$$
$$I_n + \mathrm{sign}(A) = Z^{-1}\,\mathrm{diag}(O_{p,p}, I_r + D_r\sqrt{-1}, 2I_q)Z,$$
$$I_n - \mathrm{sign}(A)^2 = Z^{-1}\,\mathrm{diag}(O_{p,p}, I_r + D_r^2, O_{q,q})Z.$$

Corollary 5.1. *Under the assumptions of Theorem 5.1 the matrix $I_n - \mathrm{sign}(A)^2$ has dominant eigenspace of dimension r associated with the eigenvalues of the matrix A that lie on the imaginary axis $\mathcal{IA} = \{\lambda : \Re(\lambda) = 0\}$, whereas the matrices $I_n - \mathrm{sign}(A)$ (resp. $I_n + \mathrm{sign}(A)$) have dominant eigenspaces associated with the eigenvalues of A that either lie on the left (resp. right) of the axis $\mathcal{IA}$ or lie on this axis and have nonzero images in $I_n - \mathrm{sign}(A)$ (resp. $I_n + \mathrm{sign}(A)$).*

5.2 Eigen-solving by applying matrix sign approximation and Quad Tree construction

Having the matrices A and $\phi(A) = I_n - \mathrm{sign}(A)$ (resp. $\phi(A) = I_n + \mathrm{sign}(A)$) available, we can apply Algorithm 3.1 to approximate all eigenvalues of the matrix A that lie either on the axis $\mathcal{IA}$ or on the left (resp. right) from it. The algorithm would compute square matrices $\widehat{L}$ of dimensions p_+ and q_+, respectively, where $p \le p_+ \le p + r$ and $q \le q_+ \le q + r$. For $M = C_p$ this is associated with splitting out a factor of the polynomial $p(x)$ having degree p_+ or q_+. If this degree is large, then typically we observe dramatic growth of the coefficients in the transition from the polynomial to the factor. (Consider, e.g., splitting the polynomial $x^n + 1$ into the product of two factors of about the same degree and such that all roots of one of them have positive real parts.) We are not led to such problem, however, if we work with matrices and if we approximate their eigenspaces associated with the eigenvalues lying in a bounded region. In the rest of this section we will briefly comment on this approach.

The subdivision techniques (cf. [25]) enable us to deal with matrices whose sizes are decreased recursively, and we can stop when their eigenvalues are the roots of the small degree factors of the polynomial $p(x)$, and so the coefficients of these factors are of the same order of magnitude as their roots. The approach relies on the following simple fact.

Fact 5.1. *Suppose $\mathcal{U}$ and $\mathcal{V}$ are two eigenspaces of A and $\Lambda(\mathcal{U})$ and $\Lambda(\mathcal{V})$ are the sets of the associated eigenvalues. Then $\Lambda(\mathcal{U})\cap\Lambda(\mathcal{V})$ is the set of the eigenvalues of A associated with the eigenspace $\mathcal{U}\cap\mathcal{V}$.*

By computing the matrix sign function of the matrices $\alpha A-\sigma I$ for various selected pairs of complex scalars α and σ, we can define the eigenspace of the matrix A associated with the eigenvalues lying in a selected region on the complex plane bounded by straight lines, e.g., in any rectangle. In particular this supports the search policy widely known as *Quad Tree Construction*, proposed by H. Weyl in 1924 for polynomial root-finding. Strengthened by some modern techniques of numerical computing, Weyl's algorithm is practically promising and supports the record Boolean complexity estimates for approximating a single root of a univariate polynomial [25]. By including matrix inversions into these computations, we define the eigenvalue regions bounded by straight lines, their segments, circles and their arcs.

5.3 Iterative algorithms for computing the matrix sign function and their convergence

[16, equations (6.17)–(6.20)] define effective iterative algorithms for approximating the square root function $B^{1/2}$. One can readily extend them to approximating the matrix sign function $\mathrm{sign}(A)=A(A^2)^{-1/2}$. [16, Chapter 5] presents a number of effective iterative algorithms devised directly for the matrix sign function. Among them we recall Newton's iteration

$$N_0=A,\ N_{i+1}=0.5(N_i+\alpha_i\ N_i^{-1}),\ i=0,1,\dots, \tag{5.1}$$

based on the Möbius transform $x\to(x+1/x)/2$, and the [2/0] Padé iteration

$$N_0=A,\ N_{i+1}=(15I_n-10N_i^2+3N_i^4)N_i/8,\ i=0,1,\dots \tag{5.2}$$

Theorem 2.2 implies the following simple corollary.

Corollary 5.2. *Assume iterations (5.1) and (5.2) where neither of the matrices N_i is singular. Let $\lambda=\lambda^{(0)}$ denote an eigenvalue of the matrix N_0 and define*

$$\lambda^{(i+1)}=(\lambda^{(i)}+(\lambda^{(i)})^{-1})/2 \text{ for } i=0,1,\dots, \tag{5.3}$$

$$\lambda^{(i+1)}=\lambda^{(i)}(15-10(\lambda^{(i)})^2+3(\lambda^{(i)})^4)/8,\ i=0,1,\dots \tag{5.4}$$

Then $\lambda^{(i)}\in\Lambda(N_i)$ for $i=1,2,\dots$ provided the pairs $\{N_i,\lambda^{(i)}\}$ are defined by the pairs of equations (5.1), (5.3) or (5.2), (5.4), respectively.

Corollary 5.3. *In iterations (5.3) and (5.4) the images $\lambda^{(i)}$ of an eigenvalue λ of the matrix N_0 for all i lie on the imaginary axis $\mathcal{IA}$ if so does λ.*

By virtue of the two following theorems, the sequences $\{\lambda^{(0)},\lambda^{(1)},\dots\}$ defined by equations (5.3) and (5.4) converge to ±1 exponentially fast right from the start. The convergence is quadratic for sequence (5.3) where $\Re(\lambda)\neq0$ and cubic for sequence (5.4) where $|\lambda-\mathrm{sign}(\lambda)|\le1/2$.

Theorem 5.2. *(See [16], [10, page 500].) Write $\lambda=\lambda^{(0)}$, $\delta=\mathrm{sign}(\lambda)$ and $\gamma=|\frac{\lambda-\delta}{\lambda+\delta}|$. Assume (5.3) and $\Re(\lambda)\neq0$. Then $|\lambda^{(i)}-\delta|\le\frac{2\gamma^{2^i}}{1-\gamma^{2^i}}$ for $i=0,1,\dots$.*

Theorem 5.3. *Write $\gamma_i=|\lambda^{(i)}-\mathrm{sign}(\lambda^{(i)}|$ for $i=0,1,\dots$. Assume (5.4) and $\gamma_0\le1/2$. Then $\gamma_i\le\frac{32}{113}(\frac{113}{128})^{3^i}$ for $i=1,2,\dots$*

Proof. Complete the proof of [10, Proposition 4.1] by using the bound $\gamma_0\le1/2$. First verify that $\gamma_{i+1}=\gamma_i^3|3(\lambda^{(i)})^2+9\lambda^{(i)}+8|/8$ and therefore $\gamma_{i+1}\le\frac{113}{32}\gamma_i^3$ for $i=0,1,\dots$. Now the claimed bounds follow by induction on i for $\gamma_0\le1/2$. □

5.4 Real versions of Newton's and Padé's iterations

Having the matrix $F(A) = I_n - \text{sign}(A)^2$ available, we can apply Algorithm 3.1 to approximate the eigenvalues of the matrix A that lie on the axis $\mathcal{IA}$, and we can devise real eigen-solvers for a real $n \times n$ matrix M, based on applying these techniques to the matrix $A = M\sqrt{-1}$. Next we modify this approach slightly, to avoid involving nonreal values. We substitute $N_0 = M$ in lieu of $N_0 = A$ into matrix sign iterations (5.1) and (5.2) and equivalently rewrite them as follows,

$$N_0 = M,\ N_{i+1} = 0.5(N_i - N_i^{-1}) \text{ for } i = 0, 1, \dots, \tag{5.5}$$

$$N_0 = M,\ N_{i+1} = -(3N_i^5 + 10N_i^3 + 15N_i)/8 \text{ for } i = 0, 1, \dots. \tag{5.6}$$

The matrices N_i and the images $\lambda^{(i)}$ of every real eigenvalue λ of M are real for all i, whereas the results of Theorems 5.2 and 5.3 are immediately extended. The images of every nonreal point λ converge to the complex point $\text{sign}(\Im(\lambda))\sqrt{-1}$ with quadratic rate under (5.5) if $\Re(\lambda) \neq 0$ and with cubic rate under (5.6) if $\lambda \in \mathcal{D}_{1/2}(\text{sign}(\Im(\lambda))\sqrt{-1})$. Under the maps $M \to I_n + N_i^2$ for the matrices N_i of the above iterations, the images $1 + (\lambda^{(i)})^2$ of nonreal eigenvalues λ of the matrix M converge to 0 as long as the iteration is initiated in its basin of convergence, whereas the images of a real point λ are real and are at least 1 for all i. Thus for sufficiently large integers i we yield strong domination of the eigenspace of the matrix N_i associated with the images of the real eigenvalues of the matrix M.

5.5 Newton's iteration with shifts for real matrix sign function

Iteration (5.5) fails where for some integer i the matrix N_i is singular or nearly singular, that is has an eigenvalue equal to 0 or lying near 0, but then we can approximate this eigenvalue by applying the Rayleigh Quotient Iteration [15, Section 8.2.3], [7] or the Inverse Orthogonal Iteration [15, page 339].

If we seek other real eigenvalues as well, we can deflate the matrix M and apply Algorithm 3.1 to the resulting matrix of a smaller size. Alternatively we can apply it to the matrix $N_i + \rho_i I_n$ for a shift ρ_i randomly generated in the range $-r \le \rho_i \le r$ for a positive r. We choose the value r reasonably small and then can expect to avoid degeneracy and, by virtue of Theorems 5.2 and 5.3, to have the images of all nonreal eigenvalues of M still rapidly converging to a small neighborhood of the points $\pm\sqrt{-1}$, thus ensuring their isolation from the images of the real eigenvalues.

5.6 Controlling the norms in the Padé's [2/0] iterations

We have no singularity problem with iteration (5.6), but have numerical problems where the norms $||N_i||$ grow large. If the nonreal eigenvalues of the matrix N_0 lie in the union of the two discs $\mathcal{D}_{1/2}(\pm\sqrt{-1})$, then their images also stay there by virtue of a simple extension of Theorem 5.3, and then the norms $||N_i||$ can be large only where some real eigenvalues of the matrices N_i are absolutely large.

Now suppose the nonreal eigenvalues of the matrix M have been mapped into the union of the two discs $\mathcal{D}_{y_i}(\pm\sqrt{-1})$ for $0 < y_i < 0.1$. (One or two steps (5.6) move every $\mu \in \mathcal{D}_{1/2}(\pm\sqrt{-1})$ into the discs $\mathcal{D}_{y_i}(\pm\sqrt{-1})$, cf. Theorem 5.3.) Then the transformation $N_i \to N_i(N_i^2 + 2I_n)^{-1}$ confronts excessive norm growth by mapping all real eigenvalues of N_i into the range $[-\frac{1}{4}\sqrt{2}, \frac{1}{4}\sqrt{2}]$ and mapping all nonreal eigenvalues of N_i into the discs $\mathcal{D}_{w_i}(\pm\sqrt{-1})$ for $w_i \le \frac{1+y_i}{1-2y_i-y_i^2}$. E.g., $w_i < 0.4$ for $y_i = 0.1$, whereas $w_i < 0.17$ for $y_i = 0.05$, and then single step (5.6) would more than compensate for such a minor dilation of the discs $\mathcal{D}_{y_i}(\pm\sqrt{-1})$ (see Theorem 5.3).

6 Numerical tests

We performed a series of numerical tests in the Graduate Center of the City University of New York using a Dell server with a dual core 1.86 GHz Xeon processor and 2G memory running Windows Server 2003 R2. The test Fortran code was compiled with the GNU gfortran compiler within the

Cygwin environment. We generated random numbers with the *random_number()* intrinsic Fortran function assuming the uniform probability distribution over the range $\{x:\ 0 \le x < 1\}$. To shift to the range $\{y:\ b \le y \le a+b\}$ for fixed real a and b, we applied the linear transform $x \to y = ax+b$.

We tested our algorithms for the approximation of the eigenvalues of $n \times n$ companion matrix C_p and of the shifted matrix $C_p - sI_n$ defined by polynomials $p(x)$ with random real coefficients for $n = 64, 128, 256$ and by random real s. For each class of matrices, each input size and each iterative algorithm we generated 100 input instances and run 100 tests. Our tables show the minimum, maximum, and average (mean) numbers of iteration loops in these runs (until convergence) as well as the standard deviations in the columns marked by "**min**", "**max**", "**mean**", and "**std**", respectively. We applied repeated squaring of Section 4 to the matrix $C_p - sI$, where we used shifts s because polynomials $p(x)$ with random real coefficients tend to have all roots near the circle $\mathcal{C}_1(0)$ and consequently repeated squaring of C_p advances towards eigen-solving very slowly. We applied real Newton's iteration (5.5) to approximate the matrix sign function for the matrix C_p using no shifts. Then we applied Algorithm 3.1 to approximate real eigenvalues.

In both groups of the tests we output roots with at least four correct decimals. In our next group of tests we output roots with at least three correct decimals. In these tests we applied real Padé iteration (5.6) without stabilization to the matrices produced by five Newton's steps (5.5). Table 6.1 displays the results of our tests of repeated squaring of Section 4. The first three lines show the dimension of the output subspace and the matrix L. The next three lines show the number of squarings performed until convergence. Table 6.2 displays the number of Newton's steps (5.5) performed until convergence.

Table 6.4 covers the tests where we first performed five Newton's steps (5.5) followed by sufficiently many Padé steps (5.6) required for convergence. The first three lines of the table show the number of the Padé steps. The next three lines display the percent of the real roots of the polynomials $p(x)$ that the algorithm computed with at least three correct decimals (compared to the overall number of the real eigenvalues of L). The next three lines show the increased percent of computed roots when we refined the crude approximations by means of Rayleigh Quotient iteration. The iteration rapidly converged from all these initial approximations but in many cases to the same roots from distinct initial points.

Table 6.1: Repeated Squaring

n	**dimension/squarings**	**min**	**max**	**mean**	**std**
64	dimension	1	10	5.31	2.79
128	dimension	1	10	3.69	2.51
256	dimension	1	10	4.25	2.67
64	squarings	6	10	7.33	0.83
128	squarings	5	10	7.37	1.16
256	squarings	5	11	7.13	1.17

Table 6.2: Newton's iteration (5.5).

n	**min**	**max**	**mean**	**std**
64	7	11	8.25	0.89
128	8	11	9.30	0.98
256	9	13	10.22	0.88

Table 6.3: 5 N-steps (5.5) + P-steps (5.6)

n	**P-steps or %**	**min**	**max**	**mean**	**std**
64	P-steps	1	4	2.17	0.67
128	P-steps	1	4	2.05	0.63
256	P-steps	1	3	1.99	0.58
64	% w/o RQ steps	0	100	64	28
128	% w/o RQ steps	0	100	39	24
256	% w/o RQ steps	0	100	35	20
64	% w/RQ steps	0	100	89	19
128	% w/RQ steps	0	100	74	26
256	% w/RQ steps	0	100	75	24

Table 6.4: 5 N-steps (5.5) + P-steps (5.6)

n	**P-steps or %**	**min**	**max**	**mean**	**std**
64	P-steps	2	8	4.26	1.19
128	P-steps	2	10	4.20	1.23
256	P-steps	2	6	4.24	1.22
64	% w/o RQ steps	0	100	67	26
128	% w/o RQ steps	0	100	43	24
256	% w/o RQ steps	0	100	33	23
64	% w/RQ steps	0	100	87	21.3
128	% w/RQ steps	0	100	87	20.5
256	% w/RQ steps	0	100	88	21.5

7 Discussion

While presenting a number of promising approaches we have only partly developed them to demonstrate their power and to motivate further research efforts. In some cases we skipped even some known efficient modifications. For example, one can dramatically accelerate initial convergence of Newton's iteration (5.1) for computing matrix sign function and can make it more robust by means of the following scaling,

$$N_0 = A,\ N_{i+1} = 0.5(N_i + \alpha_i\ N_i^{-1}),\ \alpha_i^2 = ||N_i^{-1}||/||N_i||,\ i = 0, 1, \dots, \tag{7.1}$$

which for our variation can be rewritten as follows,

$$N_0 = M,\ N_{i+1} = 0.5(N_i - \alpha_i N_i^{-1}) \text{ for } \alpha_i^2 = ||N_i^{-1}||/||N_i|| \text{ and } i = 0, 1, \dots. \tag{7.2}$$

One can expect to see new advances of our approaches, e.g., based on more intricate maps of the complex plane and employing appropriate shifts and scaling of the matrices involved into our iterative algorithms. Another potential resource of further progress is the combination with other matrix eigen-solvers and polynomial root-finders, for example, a variant of the Lanczos algorithm for real eigen-solving, the Rayleigh Quotient iteration, and the subdivision and continued fraction methods of polynomial root-finding (see [13], [14], [18], [22], [32], [36], and the bibliography therein). Various symbolic techniques can supply auxiliary information for our computations (e.g., the number of real roots and their bounds) and can handle the inputs that are hard for numerical treatment.

Acknowledgements: Our research has been supported by NSF Grant CCF–1116736 and PSC CUNY Awards 64512–0042 and 65792–0043.

References

[1] J. L. Aurentz, R. Vandebril, and D. S. Watkins, Fast computation of the zeros of a polynomial via factorization of the companion matrix, *SIAM J. Sci. Comput.*, **35, 1**, A255–A269, 2013.

[2] E. T. Bell, *The Development of Mathematics*, McGraw-Hill, New York, 1940.

[3] C. A. Boyer, *A History of Mathematics*, Wiley, New York, 1968.

[4] D. A. Bini, P. Boito, Y. Eidelman, L. Gemignani, I. Gohberg, A Fast Implicit QR Algorithm for Companion Matrices, *Linear Algebra and Its Applications*, **432**, 2006–2031, 2010.

[5] 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.

[6] D. A. Bini, F. Daddi, and L. Gemignani, On the Shifted QR Iteration Applied to Companion Matrices, *Electronic Transactions on Numerical Analysis (ETNA)*, **18**, 137–152, 2004.

[7] D. A. Bini, L. Gemignani, V. Y. Pan, Inverse Power and Durand/Kerner Iteration for Univariate Polynomial Root-finding, *Computers and Mathematics (with Applications)*, **47**, **2/3**, 447–459, 2004.

[8] D. A. Bini, L. Gemignani, V. Y. Pan, Algorithms for Generalized Companion Matrices and Secular Equation, *Numerische Math.* **3**, 373–408, 2005. Also Technical Report 1470, *Department of Math., University of Pisa*, Pisa, Italy (July 2003).

[9] D. A. Bini, L. Gemignani, V. Y. Pan, Improved Initialization of the Accelerated and Robust QR-like Polynomial Root-finding, *Electronic Transactions on Numerical Analysis* **17**, 195–205, 2004. Proc. version in CASC'2004.

[10] D. Bini, V. Y. Pan, Graeffe's, Chebyshev, and Cardinal's Processes for Splitting a Polynomial into Factors, *J. Complexity*, **12**, 492–511, 1996.

[11] J. P. Cardinal, On Two Iterative Methods for Approximating the Roots of a Polynomial, *Lectures in Applied Mathematics*, **32** (*Proceedings of AMS-SIAM Summer Seminar: Mathematics of Numerical Analysis: Real Number Algorithms* (J. Renegar, M. Shub, and S. Smale, editors), Park City, Utah, 1995), 165–188, American Mathematical Society, Providence, Rhode Island, 1996.

[12] F. Cajori, *A History of Mathematics, 5/E*, AMS Chelsea Publ., Providence, Rhode Island, 1999.

[13] A. Eigenwillig, V. Sharma, C. K. Yap, Almost Tight Recursion Tree Bounds for the Descartes Method, *Proc. Int. Symp. on Symbolic and Algebraic Computation (ISSAC 2006)*, 71–78, ACM Press, New York, 2006.

[14] I. Z. Emiris, B. Mourrain, E. Tsigaridas, Real Algebraic Numbers: Complexity Analysis and Experimentation, in *RELIABLE IMPLEMENTATIONS OF REAL NUMBER ALGORITHMS: THEORY AND PRACTICE, LNCS*, **5045**, 57–82, Springer, 2008 (also available in www.inria.fr/rrrt/rr-5897.html).

[15] G. H. Golub, C. F. Van Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, Maryland, 1996 (third edition).

[16] N. J. Higham, *Functions of Matrices: Theory and Computations*, SIAM, Philadelphia, 2008.

[17] N. Halko, P. G. Martinsson, J. A. Tropp, Finding Structure with Randomness: Probabilistic Algorithms for Constructing Approximate Matrix Decompositions, *SIAM Review*, **53, 2**, 217–288, 2011.

[18] M. Hemmer, E.P. Tsigaridas, Z. Zafeirakopoulos, I. Z. Emiris, M. I. Karavelas, B. Mourrain, Experimental Evaluation and Cross-Benchmarking of Univariate Real Solvers, in *Proc. International Symposium on Symbolic-Numerical Computations*, (Kyoto, Japan, August 2009), (edited by Hiroshi Kai and Hiroshi Sekigawa), pp.105–113, ACM Press, New York, 2009.

[19] J.M. McNamee, A 2002 Update of the Supplementary Bibliography on Roots of Polynomials, *J. of Computational and Applied Math.* **142**, 433-434, 2002; also at web-site `www.yorku.ca/~mcnamee/`

[20] J.M. McNamee, *Numerical Methods for Roots of Polynomials (Part 1)*, Elsevier, Amsterdam, 2007.

[21] J.M. McNamee and V.Y. Pan, *Numerical Methods for Roots of Polynomials, Part II*, Elsevier, 2013.

[22] K. Mehlhorn, M. Sagraloff, A Deterministic Algorithm for Isolating Real Roots of a Real Polynomial, *J. of Symbolic Computation* **46, 1**, 70–90, 2011.

[23] V. Y. Pan, Solving a Polynomial Equation: Some History and Recent Progress, *SIAM Review*, **39, 2**, 187–220, 1997.

[24] V. Y. Pan, Solving Polynomials with Computers, *American Scientist*, **86**, January–February 1998. Available via

http://comet.lehman.cuny.edu/vpan/research/publications

[25] V. Y. Pan, Approximating Complex Polynomial Zeros: Modified Quadtree (Weyl's) Construction and Improved Newton's Iteration, *J. of Complexity*, **16**, **1**, 213–264, 2000.

[26] V. Y. Pan, Amended DSeSC Power Method for Polynomial Root-finding, *Computers and Math. (with Applications)*, **49**, **9**–**10**, 1515–1524, 2005.

[27] V. Y. Pan, G. Qian, A. Zheng, Z. Chen, Matrix Computations and Polynomial Root-finding with Preprocessing, *Linear Algebra and Its Applications*, **434**, 854–879, 2011.

[28] V. Y. Pan, E. P. Tsigaridas, On the Boolean Complexity of the Real Root Refinement, preprint 2013, Proc. version in in *Proc. Intern. Symposium on Symbolic and Algebraic Computation (ISSAC 2013)*, (M. Kauers ed.), 299–306, Boston, MA, June 2013, ACM Press, New York, 2013.

[29] V. Y. Pan, A. Zheng, New Progress in Real and Complex Polynomial Root-Finding, *Computers and Math. (with Applications)* **61**, 1305–1334, 2011. Proceedings version: Real and Complex Polynomial Root-Finding with Eigen-Solving and Preprocessing, in *Proc. International Symp. on Symbolic and Algebraic Computation (ISSAC 2010)*, pages 219–226, ACM Press, New York, 2010.

[30] G. W. Stewart, *Matrix Algorithms, Vol I: Basic Decompositions*, SIAM, Philadelphia, 1998.

[31] G. W. Stewart, *Matrix Algorithms, Vol II: Eigensystems*, SIAM, Philadelphia, 2001 (second edition).

[32] E. P. Tsigaridas, I. Z. Emiris, Univariate Polynomial Real Root Isolation: Continued Fractions Revisited, *ESA'06 Proceedings of the 14th Conference on Annual European Symposium*, Zurich, 2006, *LNCS*, **4168**, 817–828, Springer, London, 2006.

[33] M. Van Barel, R. Vandebril, P. Van Dooren, K. Frederix, Implicit Double Shift QR-algorithm for Companion Matrices, *Numerische Mathematik* **116**, **2**, 177–212, 2010.

[34] D. S. Watkins, *Fundamentals of Matrix Computations*, Wiley, New York, 2002 (second edition).

[35] D. S. Watkins, *The Matrix Eigenvalue Problem: GR and Krylov Subspace Methods*, SIAM, Philadelphia, PA, 2007.

[36] C. Yap, M. Sagraloff, A Simple but Exact and Efficient Algorithm for Complex Root Isolation, *Proc. of International Symp. on Symbolic and Algebraic Computation (ISSAC '11)*, San Jose, California, June 2011 (edited by A. Leykin), 353–360, ACM Press, New York, 2011.

[37] P. Zhlobich, Differential qd Algorithm with Shifts for Rank-Structured Matrices. *SIAM J. on Matrix Analysis and Applications*, 33(4), 1153–1171, 2012.