A Schur-Newton Method for the Matrix p'th Root and its Inverse

Guo, Chun-Hua and Higham, Nicholas J.

2006

MIMS EPrint: **2005.9**

Manchester Institute for Mathematical Sciences
School of Mathematics

The University of Manchester

Reports available from: http://eprints.maths.manchester.ac.uk/
And by contacting: The MIMS Secretary
School of Mathematics
The University of Manchester
Manchester, M13 9PL, UK

ISSN 1749-9097

A SCHUR–NEWTON METHOD FOR THE MATRIX PTH ROOT AND ITS INVERSE*

CHUN-HUA GUO† AND NICHOLAS J. HIGHAM‡

Abstract. Newton's method for the inverse matrix pth root, $A^{-1/p}$, has the attraction that it involves only matrix multiplication. We show that if the starting matrix is $c^{-1}I$ for $c \in \mathbb{R}^+$ then the iteration converges quadratically to $A^{-1/p}$ if the eigenvalues of A lie in a wedge-shaped convex set containing the disc $\{ z : |z-c^p| < c^p \}$. We derive an optimal choice of c for the case where A has real, positive eigenvalues. An application is described to roots of transition matrices from Markov models, in which for certain problems the convergence condition is satisfied with $c = 1$. Although the basic Newton iteration is numerically unstable, a coupled version is stable and a simple modification of it provides a new coupled iteration for the matrix pth root. For general matrices we develop a hybrid algorithm that computes a Schur decomposition, takes square roots of the upper (quasi)triangular factor, and applies the coupled Newton iteration to a matrix for which fast convergence is guaranteed. The new algorithm can be used to compute either $A^{1/p}$ or $A^{-1/p}$, and for large p that are not highly composite it is more efficient than the method of Smith based entirely on the Schur decomposition.

Key words. Matrix pth root, principal pth root, matrix logarithm, inverse, Newton's method, preprocessing, Schur decomposition, numerical stability, convergence, Markov model, transition matrix

AMS subject classifications. 65F30, 15A18, 15A51

1. Introduction.

Newton methods for computing the principal matrix square root have been studied for almost fifty years and are now well understood. Since Laasonen proved convergence but observed numerical instability [25], several Newton variants have been derived and proved numerically stable, for example by Higham [13], [15], Iannazzo [18], and Meini [27]. For matrix pth roots, with p an integer greater than 2, Newton methods were until recently little used, for two reasons: their convergence in the presence of complex eigenvalues was not well understood and the iterations were found to have poor numerical stability. The subtlety of the question of convergence is clear from the scalar case, since the starting values for which Newton's method for $z^p-1 = 0$ converges to some pth root of unity form fractal Julia sets in the complex plane for $p > 2$ [28], [30], [33]. Nevertheless, Iannazzo [19] has recently proved a new convergence result for the scalar Newton iteration and has thereby shown how to build a practical algorithm for the matrix pth root.

Throughout this work we assume that $A \in \mathbb{C}^{n\times n}$ has no eigenvalues on $\mathbb{R}^-$, the closed negative real axis. The particular pth root of interest is the principal pth root (and its inverse), denoted by $A^{1/p}$ ($A^{-1/p}$), which is the unique matrix X such that $X^p = A$ ($X^{-p} = A$) and the eigenvalues of X lie in the segment $\{ z : -\pi/p < \arg(z) < \pi/p \}$. We are interested in methods both for computing $A^{1/p}$ and for computing $A^{-1/p}$.

We briefly summarize Iannazzo's contribution, which concerns Newton's method for $X^p - A = 0$, and then turn to the inverse Newton iteration. Newton's method

*Version of February 15, 2006. This work was supported by a Royal Society-Wolfson Research Merit Award to the second author.

†Department of Mathematics and Statistics, University of Regina, Regina, SK S4S 0A2, Canada (chguo@math.uregina.ca, http://www.math.uregina.ca/~chguo/). This work was supported in part by a grant from the Natural Sciences and Engineering Research Council of Canada.

‡School of Mathematics, The University of Manchester, Sackville Street, Manchester, M60 1QD, UK (higham@ma.man.ac.uk, http://www.ma.man.ac.uk/~higham/).

takes the form

$$X_{k+1} = \frac{1}{p}\big[(p-1)X_k + X_k^{1-p}A\big], \qquad X_0A = AX_0. \tag{1.1}$$

Iannazzo [19] shows that $X_k \to A^{1/p}$ quadratically if $X_0 = I$ and each eigenvalue of A belongs to the set

$$S = \{\, z \in \mathbb{C} : \operatorname{Re} z > 0 \text{ and } |z| \le 1 \,\} \cup \mathbb{R}^+, \tag{1.2}$$

where $\mathbb{R}^+$ denotes the open positive real axis. Based on this result, he obtains the following algorithm for computing the principal pth root.

ALGORITHM 1.1 (Matrix pth root via Newton iteration [19]). *Given $A \in \mathbb{C}^{n\times n}$ with no eigenvalues on $\mathbb{R}^-$ this algorithm computes $X = A^{1/p}$ using the Newton iteration.*

1 $B = A^{1/2}$
2 $C = B/\|B\|$ (any norm)
3 Use the iteration (1.3) to compute $X = C^{2/p}$ (p even) or $X = \big(C^{1/p}\big)^2$ (p odd).
4 $X \leftarrow \|B\|^{2/p} X$

The iteration used in the algorithm is a rewritten version of (1.1):

$$\begin{aligned} X_{k+1} &= X_k\left(\frac{(p-1)I + M_k}{p}\right), && X_0 = I, \\ M_{k+1} &= \left(\frac{(p-1)I + M_k}{p}\right)^{-p} M_k, && M_0 = A, \end{aligned} \tag{1.3}$$

where $M_k \equiv X_k^{-p}A$. Iannazzo shows that, unlike (1.1), this coupled form is numerically stable.

Newton's method can also be applied to $X^{-p} - A = 0$, for which it takes the form

$$X_{k+1} = \frac{1}{p}\big[(p+1)X_k - X_k^{p+1}A\big], \qquad X_0A = AX_0. \tag{1.4}$$

The iteration has been studied by several authors. R. A. Smith [34] uses infinite product expansions to show that X_k converges to an inverse pth root of A if the initial matrix X_0 satisfies $\rho(I - X_0^pA) < 1$, where ρ denotes the spectral radius. Lakić [26] reaches the same conclusion, under the assumption that A is diagonalizable, for a family of iterations that includes (1.4). Bini[1], Higham, and Meini take $X_0 = I$ and prove convergence of the residuals $I - X_k^pA$ to zero when $\rho(I - A) < 1$ (see Lemma 2.1 below) as well as convergence of X_k to $A^{-1/p}$ if A has real, positive eigenvalues and $\rho(A) < p + 1$ [4]. They also show that (1.4) has poor numerical stability properties. In none of these papers is it proved to which inverse pth root the iteration converges when $\rho(I - X_0^pA) < 1$. The purpose of our work is to determine a larger region of convergence to $A^{-1/p}$ for (1.4) and to build a numerically stable algorithm applicable to arbitrary A having no eigenvalues on $\mathbb{R}^-$.

In Section 2 we present convergence analysis to show that if the spectrum of A is contained in a certain wedge-shaped convex region depending on a parameter $c \in \mathbb{R}^+$ then quadratic convergence of the inverse Newton method with $X_0 = c^{-1}I$ to $A^{-1/p}$

[1]The authors of [4] were unaware of the papers of Lakić [26] and M. I. Smith [34], and Lakić appears to have been unaware of Smith's paper.

is guaranteed—with no restrictions on the Jordan structure of A. In Section 3 we consider the practicalities of choosing c and implementing the inverse Newton iteration. We derive an optimal choice of c for the case where A has real, positive eigenvalues, and we prove a finite termination property for a matrix with just one distinct eigenvalue. A stable coupled version of (1.4) is noted, and by a simple modification a new iteration is obtained for $A^{1/p}$. For general A we propose a hybrid algorithm for computing $A^{-1/p}$ or $A^{1/p}$ that precedes application of the Newton iteration with a preprocessing step, in which a Schur reduction to triangular form is followed by the computation of a sequence of square roots. An interesting and relatively unexplored application of pth roots is to Markov models; in Section 4 we discuss this application and show that convergence of the inverse Newton iteration is ensured with $c = 1$ in certain cases. Numerical experiments are presented in Section 5, wherein we derive a particular scaling of the residual that is appropriate for testing numerical stability. Section 6 presents our conclusions.

Finally, we mention some other reasons for our interest in computing the inverse matrix pth root. The pth root arises in the computation of the matrix logarithm by the inverse scaling and squaring method. This method uses the relation $\log(A) = p \log A^{1/p}$, where p is typically a power of 2, and approximates $\log A^{1/p}$ using a Padé approximant [6], [22, App. A]. Since $\log(A) = -p \log A^{-1/p}$, the inverse pth root can equally well be employed. The inverse pth root also appears in the matrix sector function, defined by $\operatorname{sect}_p(A) = A(A^p)^{-1/p}$ (of which the matrix sign function is the special case with $p = 2$) [23], [31], and in the expression $A(A^*A)^{-1/2}$ for the unitary polar factor of a matrix [12], [29]. For scalars $a \in \mathbb{R}$ the inverse Newton iteration is employed in floating point hardware to compute the square root $a^{1/2}$ via $a^{-1/2} \times a$, since the whole computation can be done using only multiplications [7], [21]. The inverse Newton iteration is also used to compute $a^{1/p}$ in arbitrarily high precision in the MPFUN and ARPREC packages [1], [2], [3]. Our work will be useful for computing *matrix* pth roots in high precision—a capability currently lacking in MATLAB's Symbolic Math Toolbox (Release 14, Service Pack 3).

2. Convergence to the inverse principal pth root. We begin by recalling a result of Bini, Higham, and Meini [4, Prop. 6.1].

LEMMA 2.1. *The residuals $R_k = I - X_k^p A$ from (1.4) satisfy*

$$R_{k+1} = \sum_{i=2}^{p+1} a_i R_k^i, \tag{2.1}$$

where the a_i are all positive and $\sum_{i=2}^{p+1} a_i = 1$. Hence if $0 < \|R_0\| < 1$ for some consistent matrix norm then $\|R_k\|$ decreases monotonically to 0 as $k \to \infty$, with $\|R_{k+1}\| < \|R_k\|^2$.

In the scalar case, Lemma 2.1 implies the convergence of (1.4) to an inverse pth root when $\|R_0\| < 1$, and we will use this fact below; the limit is not necessarily the inverse principal pth root, however. R. A. Smith [34] shows likewise that $\|R_0\| < 1$ implies convergence to an inverse pth root for matrices. Note that the convergence of X_k in the matrix case does not follow immediately from the convergence of R_k in Lemma 2.1. Indeed, when $\|R_0\| < 1$, the sequence of pth powers, $\{X_k^p\}$, is bounded since $X_k^p = (I - R_k)A^{-1}$, but the boundedness of $\{X_k\}$ itself does not follow when $n > 1$.

Our aim in this section is to show that for an appropriate range of X_0 the Newton iterates X_k converge to $A^{-1/p}$. We begin with the scalar case. Thus, for a given

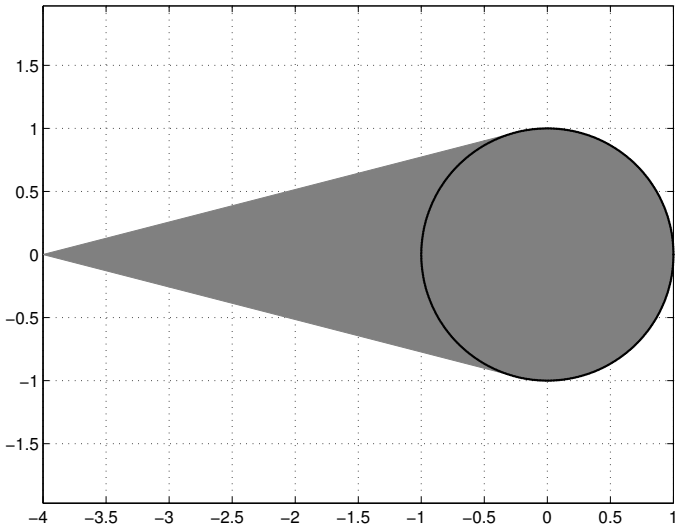

FIG. 2.1. *The region E for $p = 4$. The solid line marks the disk of radius 1, center 0, whose interior is D.*

$\lambda \in \mathbb{C} \setminus \mathbb{R}^-$ we wish to determine for which $x_0 \in \mathbb{C}$ the iteration

$$x_{k+1} = \frac{1}{p}\big[(p+1)x_k - x_k^{p+1}\lambda\big] \tag{2.2}$$

yields $\lambda^{-1/p}$, the principal inverse pth root of λ, which we know lies in the segment

$$\{\, z : -\pi/p < \arg(z) < \pi/p \,\}. \tag{2.3}$$

We denote by $D = \{\, z : |z| < 1 \,\}$ the open unit disc and by $\overline{D}$ its closure. Let

$$E = \mathrm{conv}\{\overline{D}, -p\} \setminus \{-p, 1\},$$

where conv denotes the convex hull. Figure 2.1 depicts E for $p = 4$. The next result is a restatement of [34, Thm. 4].

LEMMA 2.2. *For iteration* (2.2)*, if $1 - x_0^p\lambda \in E$ then $1 - x_1^p\lambda \in D$.*

The following result generalizes the scalar version of [4, Prop. 6.2] from $x_0 = 1$ to $x_0 > 0$ and the proof is essentially the same.

LEMMA 2.3. *Let $\lambda \in \mathbb{R}^+$. If $x_0 \in \mathbb{R}^+$ and $1 - x_0^p\lambda \in (-p, 1)$ then the sequence $\{x_k\}$ defined by* (2.2) *converges quadratically to $\lambda^{-1/p}$.*

We will also need the following complex mean value theorem from [9]. We denote by $\mathrm{Re}(z)$ and $\mathrm{Im}(z)$ the real and imaginary parts of $z \in \mathbb{C}$ and define the line

$$L(a, b) = \{\, a + t(b - a) : t \in (0, 1) \,\}.$$

LEMMA 2.4. *Let Ω be an open convex set in $\mathbb{C}$. If $f : \Omega \mapsto \mathbb{C}$ is an analytic function and a, b are distinct points in Ω then there exist points u, v on $L(a, b)$ such that*

$$\mathrm{Re}\left(\frac{f(b) - f(a)}{b - a}\right) = \mathrm{Re}(f'(u)), \quad \mathrm{Im}\left(\frac{f(b) - f(a)}{b - a}\right) = \mathrm{Im}(f'(v)).$$

The next result improves Lemma 2.3 by extending the region of allowed $1 - x_0^p\lambda$ from the interval $(-p, 1)$ to the convex set E in the complex plane.

LEMMA 2.5. *Let $\lambda \in \mathbb{C} \setminus \mathbb{R}^-$ and let $x_0 \in \mathbb{R}^+$ be such that $1 - x_0^p\lambda \in E$. Then the iterates x_k from (2.2) converge quadratically to $\lambda^{-1/p}$.*

Proof. By Lemma 2.2 we have $1 - x_1^p\lambda \in D$. It then follows from the scalar version of Lemma 2.1 that x_k converges quadratically to $x(\lambda)$, an inverse pth root of λ (see the discussion after Lemma 2.1). We need to show that $x(\lambda) = \lambda^{-1/p}$. There is nothing to prove for $p = 1$, so we assume $p \geq 2$.

For any $\lambda \in \mathbb{R}^+$ with $1 - x_0^p\lambda \in (-p, 1)$ we know from Lemma 2.3 that $x(\lambda) = \lambda^{-1/p}$. Intuition suggests that $x(\lambda)$ is a continuous function of λ. Since the principal segment (2.3) is disjoint from the other $p-1$ segments it then follows that for each λ with $1-x_0^p\lambda \in E$, $x(\lambda)$ must be the inverse of the principal pth root. We now provide a rigorous proof of $x(\lambda) = \lambda^{-1/p}$. (Once this is proved, the continuity of $x(\lambda)$ as a function of λ follows.)

We write $x_0 = 1/c$. Then $1 - x_0^p\lambda \in (-p, 1)$ becomes $\lambda \in (0, (p+1)c^p)$, and $1 - x_0^p\lambda \in E$ is the same as $\lambda \in E_c$, where

$$E_c = \operatorname{conv}\big\{ \{ z : |z - c^p| \leq c^p \}, (p+1)c^p \big\} \setminus \{ 0, (p+1)c^p \}.$$

We rewrite E_c in polar form

$$E_c = \{ (r, \theta) : 0 < r < (p+1)c^p,\ -\theta_r \leq \theta \leq \theta_r \},$$

where the exact expression for $\theta_r \equiv \theta(r)$ is unimportant. We fix $\delta \in (0, 1)$ and define the compact set

$$E_{c,\delta} = \{ (r, \theta) : \delta c^p \leq r \leq (p+1-\delta)c^p,\ -\theta_r \leq \theta \leq \theta_r \}.$$

We will prove that $x(\lambda)$ is in the segment (2.3) for each $\lambda \in E_{c,\delta}$. This will yield $x(\lambda) = \lambda^{-1/p}$ for $\lambda \in E_c$, since δ can be arbitrarily small. More precisely, for each fixed $r \in [\delta c^p, (p+1-\delta)c^p]$, we will show that $x(\lambda)$ is in the same segment for each λ on the arc given in polar form by

$$\Gamma_r = \{ (r, \theta) : -\theta_r \leq \theta \leq \theta_r \}.$$

This will complete the proof, since we already know that $x(\lambda)$ is in the segment (2.3) when $\theta = 0$. Thus we only need to show that there exists $\epsilon > 0$ such that for all $a, b \in \Gamma_r$ with $|a - b| < \epsilon$, $x(a)$ and $x(b)$ are in the same segment. To do so, we suppose that for all $\epsilon > 0$ there exist $a, b \in \Gamma_r$ with $|a - b| < \epsilon$ such that $x(a)$ is in segment i and $x(b)$ is in segment $j \neq i$, and we will obtain a contradiction.

Let a and b be any such pair for a suitably small ϵ to be chosen below. Let $\widetilde{x}(b)$ be the inverse pth root of b in segment i. Then $|x(b) - \widetilde{x}(b)|$ is at least the distance between two neighboring inverse pth roots of b, i.e.,

$$|x(b) - \widetilde{x}(b)| \geq 2r^{-1/p} \sin\frac{\pi}{p} =: 4\eta.$$

Also, we have, by Lemma 2.4,

$$|x(a) - \widetilde{x}(b)| \leq \sqrt{2} \sup_{\xi \in L(a,b)} \left| -\frac{1}{p}\xi^{-1/p-1} \right| |a - b| \leq \frac{\sqrt{2}}{p} \left(\frac{r}{2}\right)^{-1/p-1} |a - b|$$

when $|a-b| \le \sqrt{3}r$. Therefore

$$|x(a) - \widetilde{x}(b)| \le \eta$$

when $|a-b| \le \min\{\sqrt{3}r, \frac{p}{\sqrt{2}}(\frac{r}{2})^{1/p+1}\eta\} =: \epsilon_1$.

For every $\lambda \in E_{c,\delta} \subset E_c$, we have $1 - x_0^p\lambda \in E$. Thus $1 - x_1^p\lambda \in D$ by Lemma 2.2. Since $E_{c,\delta}$ is compact, the set $\{\, 1-x_1^p\lambda : \lambda \in E_{c,\delta} \,\}$ is a compact subset of D. Therefore there is constant $\delta_1 \in (0,1)$, independent of λ, such that $|1 - x_1^p\lambda| \le 1 - \delta_1$.

Now, for the iteration (2.2) with $\lambda \in \Gamma_r$, Lemma 2.1 implies

$$|1 - x_k^p\lambda| \le |1 - x_1^p\lambda|^{2^{k-1}} \le (1-\delta_1)^{2^{k-1}}$$

for $k \ge 1$. So

$$|(x_k - r_1)(x_k - r_2)\cdots(x_k - r_p)| = |x_k^p - \lambda^{-1}| \le \frac{1}{r}(1-\delta_1)^{2^{k-1}},$$

where $r_1, r_2, \ldots, r_p$ are the pth roots of λ^{-1}. Let

$$|x_k - r_s| = \min_{1\le j\le p} |x_k - x_j|.$$

Then

$$|x_k - r_s| \le r^{-1/p}(1-\delta_1)^{2^{k-1}/p} =: \eta_1.$$

The iteration (2.2) is given by $x_{k+1} = g(x_k)$, where

$$g(x) = \frac{1}{p}\left[(p+1)x - x^{p+1}\lambda\right].$$

Note that for all x with $|x - r_s| \le \eta_1$,

$$|x - r_j| \le |r_s| + |r_j| + \eta_1 = 2r^{-1/p} + \eta_1, \qquad j \ne s,$$

and

$$\begin{aligned} |g'(x)| &= \frac{p+1}{p}|1 - x^p\lambda| = \frac{p+1}{p}r|(x-r_1)(x-r_2)\cdots(x-r_p)| \\ &\le \frac{p+1}{p}r\,\eta_1(2r^{-1/p} + \eta_1)^{p-1}. \end{aligned}$$

We now take a sufficiently large k, independent of λ, such that $\eta_1 \le \eta$ and $\frac{p+1}{p}r\eta_1(2r^{-1/p} + \eta_1)^{p-1} \le \frac{1}{2}$. Then, by Lemma 2.4,

$$|x_{k+1} - r_s| = |g(x_k) - g(r_s)| \le \frac{\sqrt{2}}{2}|x_k - r_s|$$

and hence $|x_{k+m} - r_s| \le \left(\frac{\sqrt{2}}{2}\right)^m |x_k - r_s|$ for all $m \ge 0$. Thus $x_i \to r_s$ as $i \to \infty$ and $|x_k - r_s| \le \eta_1 \le \eta$. It follows that $r_s = x(\lambda)$ and $|x_k(\lambda) - x(\lambda)| \le \eta$, where we write $x_k(\lambda)$ for x_k to indicate its dependence on λ. In particular, we have

$$|x_k(a) - x(a)| \le \eta, \quad |x_k(b) - x(b)| \le \eta.$$

Now

$$\begin{aligned}|x_k(a) - x_k(b)| &= |(x_k(a) - x(a)) + (x(a) - \widetilde{x}(b)) + (\widetilde{x}(b) - x(b)) + (x(b) - x_k(b))| \\ &\geq |\widetilde{x}(b) - x(b)| - |x_k(a) - x(a)| - |x(a) - \widetilde{x}(b)| - |x(b) - x_k(b)| \\ &\geq 4\eta - \eta - \eta - \eta = \eta.\end{aligned}$$

On the other hand, for the chosen k, $x_k(\lambda)$ is a continuous function of λ on the compact set Γ_r and is therefore uniformly continuous on Γ_r. Thus there exists $\epsilon \in (0, \epsilon_1)$ such that for all $a, b \in \Gamma_r$ with $|a - b| < \epsilon$, $|x_k(a) - x_k(b)| < \eta$. This is a contradiction since we have just shown that for any $\epsilon \in (0, \epsilon_1)$, $|x_k(a) - x_k(b)| \geq \eta$ for some $a, b \in \Gamma_r$ with $|a - b| < \epsilon$. Our earlier assumption is therefore false, and the proof is complete. □

We are now ready to prove the convergence of (1.4) in the matrix case. The iterations (1.4) and (2.2) have the form $X_{k+1} = g(X_k, A)$ and $x_{k+1} = g(x_k, \lambda)$, respectively, where $g(x, t)$ is a polynomial in two variables. We will need the following special case of Theorem 4.16 in [11].

LEMMA 2.6. *Let $g(x, t)$ be a rational function of two variables. Let the scalar sequence generated by $x_{k+1} = g(x_k, \lambda)$ converge superlinearly to $f(\lambda)$ for a given λ and x_0. Then the matrix sequence generated by $X_{k+1} = g(X_k, J(\lambda))$ with $X_0 = x_0 I$, where $J(\lambda)$ is a Jordan block, converges to a matrix X_* with* $\operatorname{diag}(X_*) = \operatorname{diag}(f(J(\lambda)))$.

We now apply Lemmas 2.5 and 2.6 with $x_0 = 1/c$ and $f(\lambda) = \lambda^{-1/p}$, where $c > 0$ is a constant.

THEOREM 2.7. *Let $A \in \mathbb{C}^{n\times n}$ have no eigenvalues on $\mathbb{R}^-$. For all $p \geq 1$, the iterates X_k from* (1.4) *with $X_0 = \frac{1}{c} I$ and $c \in \mathbb{R}^+$ converge quadratically to $A^{-1/p}$ if all the eigenvalues of A are in the set*

$$E(c, p) = \operatorname{conv}\big\{ \{ z : |z - c^p| \leq c^p \}, (p+1)c^p \big\} \setminus \{ 0, (p+1)c^p \}.$$

Proof. Since X_0 is a multiple of I the X_k are all rational functions of A. The Jordan canonical form of A therefore enables us to reduce the proof to the case of Jordan blocks $J(\lambda)$, where $\lambda \in E(c, p)$. Using Lemmas 2.5 and 2.6 we deduce that X_k has a limit X_* that satisfies $X_*^{-p} = A$ and has the same eigenvalues as $A^{-1/p}$. Since $A^{-1/p}$ is the only inverse pth root having these eigenvalues, $X_* = A^{-1/p}$. Now

$$\begin{aligned}X_{k+1} - A^{-1/p} &= \frac{1}{p}\left[(p+1)X_k(A^{-1/p})^p - p(A^{-1/p})^{p+1} - X_k^{p+1}\right] A \\ &= \frac{1}{p}\left[-(X_k - A^{-1/p})^2 \sum_{i=1}^{p} i X_k^{p-i}(A^{-1/p})^{i-1}\right] A,\end{aligned}$$

and hence we have

$$\|X_{k+1} - A^{-1/p}\| \leq \|X_k - A^{-1/p}\|^2 \cdot p^{-1}\|A\| \sum_{i=1}^{p} i \|X_k^{p-i}\| \, \|A^{(1-i)/p}\|,$$

which implies that the convergence is quadratic. □

Recall that the convergence results summarized in Section 1 require $\rho(I - X_0^p A) < 1$ and do not specify to which root the iteration converges. When $X_0 = c^{-1} I$ this condition is $\max_i |\lambda_i - c^p| < c^p$, where $\Lambda(A) = \{\lambda_1, \ldots, \lambda_n\}$ is the spectrum of A. Theorem 2.7 guarantees convergence to the inverse principal pth root for $\Lambda(A)$ lying in the much larger region $E(c, p)$. The actual convergence region, determined experimentally, is shown together with $E(c, p)$ in Figure 2.2 for $c = 1$ and several values of p.

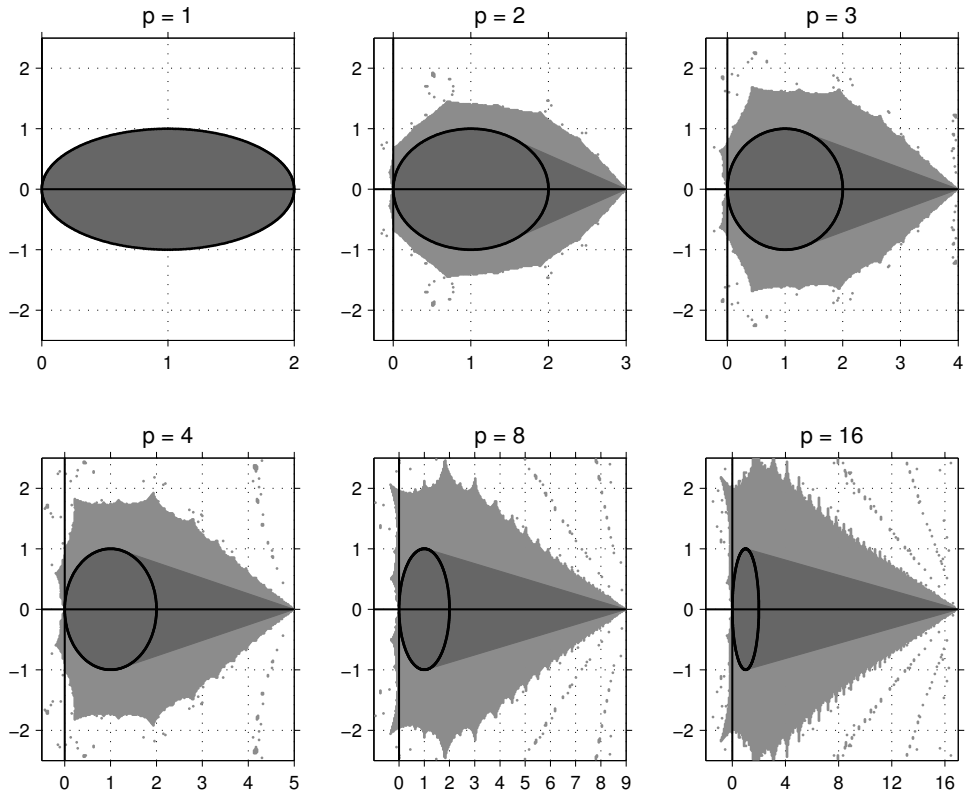

FIG. 2.2. *Regions of $\lambda \in \mathbb{C}$ for which the inverse Newton iteration* (2.2) *with $x_0 = 1$ converges to $\lambda^{-1/p}$. The dark shaded region is $E(1,p)$. The union of that region with the lighter shaded points is the experimentally determined region of convergence. The solid line marks the disk of radius* 1*, center* 1*. Note the differing x-axis limits.*

3. Practical algorithms. Armed with the convergence result in Theorem 2.7, we now build two practical algorithms applicable to arbitrary $A \in \mathbb{C}^{n\times n}$ having no eigenvalues on $\mathbb{R}^-$. Both preprocess A by computing square roots before applying the Newton iteration, one by computing a Schur decomposition and thereby working with (quasi)triangular matrices.

We take $X_0 = c^{-1}I$, where the parameter $c \in \mathbb{R}^+$ is at our disposal. Thus, to recap, the iteration is

$$X_{k+1} = \frac{1}{p}\left[(p+1)X_k - X_k^{p+1}A\right], \qquad X_0 = \frac{1}{c}I. \tag{3.1}$$

Note that scaling X_0 through c is equivalent to fixing $X_0 = I$ and scaling A: if $X_k(X_0, A)$ denotes the dependence of X_k on X_0 and A then

$$X_k(c^{-1}I, A) = c^{-1}X_k(I, c^{-p}A).$$

We begin, in the next section, by considering numerical stability.

3.1. Coupled iterations. The Newton iteration (3.1) is usually numerically unstable. Indeed, the iteration can be guaranteed to be stable only if the eigenvalues

of A satisfy [4]

$$\frac{1}{p}\left| p - \sum_{r=1}^{p} \left(\frac{\lambda_i}{\lambda_j}\right)^{r/p} \right| \le 1, \quad i,j = 1\colon n.$$

This is a very restrictive condition on A. However, by introducing the matrix $M_k = X_k^p A$, the iteration can be rewritten in the coupled form

$$\begin{aligned} X_{k+1} &= X_k \left(\frac{(p+1)I - M_k}{p}\right), & X_0 &= \frac{1}{c} I, \\ M_{k+1} &= \left(\frac{(p+1)I - M_k}{p}\right)^p M_k, & M_0 &= \frac{1}{c^p} A. \end{aligned} \tag{3.2}$$

When $X_k \to A^{-1/p}$ we have $M_k \to I$. This coupled iteration was suggested, and its unconditional stability noted, by Iannazzo [19]. In fact, (3.2) is a special case of a family of iterations of Lakić [26], and stability of the whole family is proved in [26].

Since the X_k in (3.2) are the same as those in the original iteration, their residuals R_k satisfy Lemma 2.1. Since $M_k = I - R_k$ and $M_k \to I$, the R_k are errors for the M_k.

Note that by setting $Y_k = X_k^{-1}$ we obtain from (3.2) a new coupled iteration for computing $A^{1/p}$:

$$\begin{aligned} Y_{k+1} &= \left(\frac{(p+1)I - M_k}{p}\right)^{-1} Y_k, & Y_0 &= cI, \\ M_{k+1} &= \left(\frac{(p+1)I - M_k}{p}\right)^p M_k, & M_0 &= \frac{1}{c^p} A. \end{aligned} \tag{3.3}$$

If $A^{1/p}$ is wanted without computing any inverses then $A^{1/p}$ can be computed from (3.2) and the formula $A^{1/p} = A(A^{-1/p})^{p-1}$ used (cf. (1.3)).

3.2. Algorithm not requiring eigenvalues. We now outline an algorithm that works directly on A and does not compute any spectral information. We begin by taking the square root twice by any iterative method [15]. This preprocessing step brings the spectrum into the sector $\arg z \in (-\pi/4, \pi/4)$. The nearest point to the origin that is both within this sector and on the boundary of $E(c,p)$ is at a distance $c^p\sqrt{2}$. Hence the inverse Newton iteration in the form (3.2) can be applied to $B = A^{1/4}$ with $c \ge (\rho(B)/\sqrt{2})^{1/p}$. If $\rho(B)$ is not known and cannot be estimated then we can replace it by the upper bound $\|B\|$, for some norm. This corresponds with the scaling used by Iannazzo in Algorithm 1.1 for $A^{1/p}$. A disadvantage of using the norm is that for nonnormal matrices $\rho(B) \ll \|B\|$ is possible, and this can lead to much slower convergence, as illustrated by the following example.

We use the inverse Newton iteration to compute $B^{-1/2}$, where $B = \left[\begin{smallmatrix} \epsilon & 1 \\ 0 & \epsilon \end{smallmatrix}\right]$ and $\epsilon \ll 1$. If we use $c = (\|B\|_1/\sqrt{2})^{1/2}$, the convergence will be very slow, since for the eigenvalue ϵ, $r_0(\epsilon) = 1 - x_0^2\epsilon \approx 1 - \sqrt{2}\epsilon$. If we use $c = (\rho(B)/\sqrt{2})^{1/2}$, then we have $r_0(\epsilon) = 1 - \sqrt{2}$ and the convergence will be fast (modulo the nonnormality). The best choice of c for this example, however, is $c = \epsilon^{1/2}$. For this c we have immediate convergence to the inverse square root: $X_1 = B^{-1/2}$. This finite convergence behavior is a special case of that described in the next result.

LEMMA 3.1. *Suppose that $A \in \mathbb{C}^{n\times n}$ has a positive eigenvalue λ of multiplicity n and that the largest Jordan block is of size q. Then for the iteration* (3.1) *with $c = \lambda^{1/p}$ we have $X_m = A^{-1/p}$ for the first m such that $2^m \ge q$.*

Proof. Let A have the Jordan from $A = ZJZ^{-1}$. Then $R_0 = I - X_0^p A = Z(I - \frac{1}{\lambda}J)Z^{-1}$. Thus $R_0^q = 0$. By Lemma 2.1, $R_m = (R_0)^{2^m} h(R_0)$, where $h(R_0)$ is a polynomial in R_0. Thus $R_m = 0$ if $2^m \geq q$. □

As for the complexity of iteration (3.2), the benchmark with which to compare is the Schur method for the pth root of M. I. Smith [33]. It computes a Schur decomposition and obtains the pth root of the triangular factor by a recurrence, with a total cost of $(28 + (p-1)/3)n^3$ flops. The cost of one iteration of (3.2) is about $2n^3(2 + \theta \log_2 p)$ flops, where $\theta \in [1, 2]$, assuming that the pth power in (3.2) is evaluated by binary powering [10, Alg. 11.2.2]. Since at least four iterations will typically be required, unless p is large ($p \geq 200$, say) it is difficult for (3.2) to be competitive in its operation count with the Schur method. However, the Newton iterations are rich in matrix multiplication and matrix inversion, and on a modern machine with a hierarchical memory these operations are much more efficient relative to a Schur decomposition than their flop counts suggest. For special matrices A, such as the strictly diagonally dominant stochastic matrices arising in the Markov model application in Section 4, we can apply (3.2) and (3.3) with $c = 1$ without any preprocessing, which makes this approach more efficient.

3.3. Schur–Newton algorithm. We now develop a more sophisticated algorithm that begins by computing a Schur decomposition $A = QRQ^*$ (Q unitary, R upper triangular). The Newton iteration is applied to a triangular matrix obtained from R, thereby greatly reducing the cost of each iteration. We begin by considering the choice of c, exploiting the fact that the spectrum of A is now available.

We consider first the case where the eigenvalues λ_i of A are all real and positive: $0 < \lambda_n \leq \cdots \leq \lambda_1$. Consider the residual $r_k(\lambda) = 1 - x_k^p \lambda$, and note that

$$r_{k+1}(\lambda) = 1 - \frac{1}{p^p}(1 - r_k(\lambda))(p + r_k(\lambda))^p. \tag{3.4}$$

Recall from Lemmas 2.1 and 2.2 that if $r_0 \in E$, or equivalently $\lambda \in E(c, p)$, then $|r_1| < 1$ and $|r_{i+1}| \leq |r_i|^2$ for $i \geq 1$. For c large enough, the spectrum of A lies in $E(c, p)$ and convergence is guaranteed. However, if c is too large, then $r_0(\lambda_n) = 1 - (\frac{1}{c})^p \lambda_n$ is extremely close to 1; $r_1(\lambda_n)$ is then also close to 1, by (3.4), and the convergence for the eigenvalue λ_n is very slow. On the other hand, if c is so small that $(\frac{1}{c})^p \lambda_1$ is close to (but still less than) $p + 1$, then $r_0(\lambda_1) = 1 - (\frac{1}{c})^p \lambda_1$ is close to $-p$, and, by (3.4), $r_1(\lambda_1)$ is very close to 1. Ideally we would like to choose c to minimize $\max_i |r_1(\lambda_i)|$.

LEMMA 3.2. *Let A have real, positive eigenvalues, $0 < \lambda_n \leq \cdots \leq \lambda_1$ and consider the residual $r_k(\lambda) = 1 - x_k^p \lambda$. For any $c \in \mathbb{R}^+$ such that*

$$-p < r_0(\lambda_1) \leq r_0(\lambda_2) \leq \cdots \leq r_0(\lambda_n) < 1, \tag{3.5}$$

we have $0 \leq r_j(\lambda_i) < 1$ for $j \geq 1$ and $i = 1\colon n$, and

$$\widehat{r}_j := \max_{1 \leq i \leq n} r_j(\lambda_i) = \max\bigl(r_j(\lambda_1), r_j(\lambda_n)\bigr).$$

Moreover, for all $j \geq 1$, $\widehat{r}_j$ is minimized when

$$c = \left(\frac{\alpha^{1/p}\lambda_1 - \lambda_n}{(\alpha^{1/p} - 1)(p + 1)}\right)^{1/p}, \qquad \alpha = \frac{\lambda_1}{\lambda_n}, \tag{3.6}$$

if $\lambda_1 > \lambda_n$. If $\lambda_1 = \lambda_n$ then $\widehat{r}_j = 0$ for all $j \geq 0$ for $c = \lambda_n^{1/p}$.

TABLE 3.1
Values of $f(\alpha,p)$ for some particular α and p.

α	2	5	10	50	100
$p=2$	0.0852	0.3674	0.5883	0.8877	0.9403
$p=5$	0.0690	0.3109	0.5190	0.8452	0.9125
$p=10$	0.0635	0.2902	0.4915	0.8247	0.8979
$p=1000$	0.0580	0.2688	0.4618	0.7999	0.8795

Proof. For each eigenvalue λ, we have, by (3.4), $r_{k+1}(\lambda) = f(r_k(\lambda))$ with $f(x) = 1 - \frac{1}{p^p}(1-x)(p+x)^p$. Since $f'(x) = \frac{p+1}{p^p}x(p+x)^{p-1}$, $f(x)$ is decreasing on $(-p,0]$ and increasing on $[0,1)$, and since $f(-p) = f(1) = 1$ and $f(0) = 0$ it follows that $0 \le f(x) < 1$ on $(-p,1)$. The first part of the result follows immediately. Since $f(x)$ is increasing on $[0,1)$, $\widehat{r}_j$ is minimized for all $j \ge 1$ if and only if $\widehat{r}_1$ is minimized. If $\lambda_1 > \lambda_n$ it is easily seen that $\widehat{r}_1$ is minimized when $r_1(\lambda_1) = r_1(\lambda_n)$, i.e.,

$$\lambda_1 \left(p + 1 - \lambda_1/c^p\right)^p = \lambda_n \left(p + 1 - \lambda_n/c^p\right)^p,$$

from which we find that c is given by (3.6). It is straightforward to verify that for this c, (3.5) holds. The formula (3.6) is not valid when $\lambda_1 = \lambda_n$. However, we have

$$\lim_{\lambda_1 \to \lambda_n} c = \lim_{\alpha \to 1} \left(\frac{\alpha^{1+1/p} - 1}{\alpha^{1/p} - 1} \frac{\lambda_n}{p+1} \right)^{1/p} = \lambda_n^{1/p}.$$

Note that when $\lambda_1 = \lambda_n$, $r_0(\lambda_1) = r_0(\lambda_n) = 0$ for $c = \lambda_n^{1/p}$. So $\widehat{r}_j = 0$ for all $j \ge 0$. ∎

When $\lambda_1 > \lambda_n$, a little computation shows that the minimum value of $\widehat{r}_1$, achieved for c in (3.6), is

$$f(\alpha,p) = 1 - \alpha \frac{(p+1)^{p+1}}{p^p} \frac{(\alpha-1)^p(\alpha^{1/p}-1)}{(\alpha^{1+1/p}-1)^{p+1}}.$$

Numerical experiments suggest that $f(\alpha,p)$ is increasing in α for fixed p, and decreasing in p for fixed α. Moreover, it is easy to show that $\lim_{\alpha \to 1^+} f(\alpha,p) = 0$. Some particular values of $f(\alpha,p)$ are given in Table 3.1. From the table, we can see that the values of $f(\alpha,p)$ are not sensitive to p, but are sensitive to α. It is advisable to preprocess the matrix A to achieve $\alpha \le 2$, since $f(\alpha,p)$ is then safely less than 1 and rapid convergence can be expected.

We develop the idea of preprocessing in the context of general A with possibly nonreal eigenvalues. Suppose the eigenvalues are ordered $|\lambda_n| \le \cdots \le |\lambda_1|$. A convenient way to reduce $\chi(A) := |\lambda_1|/|\lambda_n|$ is to take k_1 square roots of the triangular matrix R in the Schur form, which can be done using the method of Björck and Hammarling [5], or that of Higham [14] if R is real and quasitriangular. Since $\chi(A) = \chi(R) \le \kappa_2(R)$, in IEEE double precision arithmetic we can reasonably assume that $\chi(R) \le 10^{16}$, and then $k_1 \le 6$ square roots are enough to achieve $\chi(R^{1/2^{k_1}}) \le 2$. Write $p = 2^{k_0}q$ where q is odd. If $q = 1$, $R^{1/p}$ can be computed simply by k_0 square roots. If $q \ge 3$, we will take a total of $\max(k_0, k_1)$ square roots, compute the qth root by the Newton iteration, and finish with $k_1 - k_0$ squarings if $k_1 > k_0$. Taking $k_1 > k_0$ is justified by the operation counts if it saves just one iteration of the Newton process, because for triangular matrices the cost of a square root and a squaring is at most half of the cost of one Newton iteration. When R has nonreal eigenvalues we will increase k_1, if

necessary, so that the matrix $B = R^{1/2^{k_1}}$ to which we apply the Newton iteration has spectrum in the sector $\arg z \in (-\pi/8, \pi/8)$; in general we therefore require $k_1 \geq 3$. Then we take $c = \left(\frac{\mu_1+\mu_n}{2}\right)^{1/q}$, where $\mu_i = |\lambda_i|^{1/2^{k_1}}$. For any eigenvalue μ of B we have $\frac{2}{3} \leq (\frac{1}{c})^q|\mu| \leq \frac{4}{3}$, since $\mu_1/\mu_n \leq 2$, and thus $|1 - (\frac{1}{c})^q\mu| \leq |1 - \frac{4}{3}e^{i\frac{\pi}{8}}| \approx 0.56$. So the convergence of (3.2) is expected to be fast.

We now present our algorithm for computing the (inverse) principal pth root of a general A. We state the algorithm for real matrices, but an analogous algorithm is obtained for complex matrices by using the complex Schur decomposition.

ALGORITHM 3.3. *Given $A \in \mathbb{R}^{n\times n}$ with no eigenvalues on $\mathbb{R}^-$ this algorithm computes $X = A^{1/p}$ or $X = A^{-1/p}$, where $p = 2^{k_0}q$ with $k_0 \geq 0$ and q odd.*

1 Compute a real Schur decomposition $A = QRQ^T$.
2 if $q = 1$
3 $\quad k_1 = k_0$
4 else
5 $\quad$ Choose $k_1 \geq k_0$ such that $|\lambda_1/\lambda_n|^{1/2^{k_1}} \leq 2$,
$\quad$ where the eigenvalues of A are ordered $|\lambda_n| \leq \cdots \leq |\lambda_1|$.
6 end
7 If the λ_i are not all real and $q \neq 1$, increase k_1 as necessary so that
$\arg(\lambda_i^{1/2^{k_1}}) \in (-\pi/8, \pi/8)$ for all i.
8 Compute $B = R^{1/2^{k_1}}$ by k_1 invocations of the method of Higham [14] for the square root of a quasi-triangular matrix. If $q = 1$, goto line 21.
9 Let $\mu_1 = |\lambda_1|^{1/2^{k_1}}$, $\mu_n = |\lambda_n|^{1/2^{k_1}}$.
10 if the λ_i are all real
11 $\quad$ if $\mu_1 \neq \mu_n$
12 $\quad\quad$ determine c by (3.6) with λ_1, λ_n, p in (3.6) replaced by μ_1, μ_n, q
13 $\quad$ else
14 $\quad\quad c = \mu_n^{1/q}$
15 $\quad$ end
16 else
17 $\quad c = \left(\frac{\mu_1+\mu_n}{2}\right)^{1/q}$
18 end
19 Compute $\begin{cases} X = B^{-1/q} \text{ by (3.2)}, & \text{if } A^{-1/p} \text{ required,} \\ X = B^{1/q} \text{ by (3.3)}, & \text{if } A^{1/p} \text{ required.} \end{cases}$
20 $X \leftarrow X^{2^{k_1-k_0}}$ (repeated squaring).
21 $X \leftarrow QXQ^T$

The cost of the algorithm is about

$$\left(28 + \frac{2}{3}(k_1 + k_2) - \left(\frac{1}{3} + \frac{k_2}{2}\right)k_0 + \frac{k_2}{2}\log_2 p\right)n^3 \text{ flops},$$

where we assume that k_2 iterations of (3.2) or (3.3) are needed (the cost per iteration is the same for both for triangular matrices, except on the first iteration, where (3.2) requires $n^3/3$ fewer flops because X_1 does not require a matrix multiplication). When $k_0 = 0$, $k_1 = 3$, and $k_2 = 4$, for example, the flop count becomes $(32\frac{2}{3} + 2\log_2 p)n^3$, while the count is always $(28 + \frac{p-1}{3})n^3$ for Smith's method. Note, however, that the computational work can be reduced for Smith's method if p is not prime by applying the method over the prime factors of p (this is not beneficial for Algorithm 3.3). Our algorithm is slightly more expensive than Smith's method if p is small or highly

composite, but it is much less expensive than Smith's method if p is large and has a small number of prime factors.

Algorithm 3.3 can be modified to compute $A^{1/p}$ in a different way: by computing $X = B^{-1/q}$ in line 19 and replacing line 21 with $X \leftarrow QX^{-1}Q^T$, which is implemented as a multiple right-hand-side triangular solve followed by a matrix multiplication. The modified line 21 costs the same as the original, so the cost of the algorithm is unchanged. We will call this variant Algorithm 3.3a.

A key feature of Algorithm 3.3 is that it applies the Newton iteration to a (quasi)triangular matrix—one that has been "preconditioned" so that few iterations will be required. This can be expected to improve the numerical properties of the iteration, not least because for triangular matrices inversion and the solution of linear systems tend to be more accurate than the conventional error bounds suggest [16, Chap. 8].

4. An application to Markov models. Let $P(t)$ be a transition matrix for a time-homogeneous continuous-time Markov process. Thus $P(t)$ is a stochastic matrix: an $n \times n$ real matrix with nonnegative entries and row-sums 1. A generator Q of the Markov process is an $n \times n$ real matrix with nonnegative off-diagonal entries and zero row-sums such that $P(t) = e^{Qt}$. Clearly, Q must satisfy $e^Q = P \equiv P(1)$. If P has distinct, real positive eigenvalues then the only real logarithm, and hence the only candidate generator, is the principal logarithm, $\log P$. In general, a generator may or may not exist, and if it exists it need not be the principal logarithm of P [32].

Suppose a given transition matrix $P \equiv P(1)$ has a generator $Q = \log P$. Then Q can be used to construct $P(t)$ at other times, through $P(t) = \exp(Qt)$. For example, if P is the transition matrix for the time period of one year then the transition matrix for a month is $P(1/12) = e^{\frac{1}{12}\log P}$. However, it is more direct and efficient to compute $P(1/12)$ as $P^{1/12}$, thus avoiding the computation of a generator. Indeed, the standard inverse scaling and squaring method for the principal logarithm of a matrix requires the computation of a matrix root, as noted in Section 1. Similarly, the transition matrix for a week can be computed directly as $P^{1/52}$.

This use of matrix roots is suggested by Waugh and Abel [35], mentioned by Israel, Rosenthal, and Wei [20], and investigated in detail by Kreinin and Sidelnikova [24]. The latter authors, who are motivated by credit risk models, address the problems that the principal root and principal logarithm of P may have the wrong sign patterns; for example, the root may have negative elements, in which case it is not a transition matrix. They show how to optimally adjust these matrices to achieve the required properties, a process they term regularization. Their preferred method for obtaining transition matrices for short times is to regularize the appropriate matrix root.

Transition matrices arising in the credit risk literature are typically strictly diagonally dominant [20], and such matrices are known to have at most one generator [8]. For any strictly diagonally dominant stochastic matrix P, Gershgorin's theorem shows that every eigenvalue lies in one of the disks $|z - a_{ii}| \le 1 - a_{ii}$, and we have $a_{ii} > 0.5$, so the spectrum lies in $E(1, p)$ and the convergence of (3.2) and (3.3) (with $A = P$) is guaranteed with $c = 1$. Note, however, that faster convergence is possible by choosing $c < 1$ when P has eigenvalues close to 0. For $c = 1$, it is easy to see that $X_k e = e$ and $M_k e = e$ for each $k \ge 0$. Thus all approximations to $P^{1/p}$ obtained from (3.2) and (3.3) have unit row sums, though they are not necessarily nonnegative matrices.

To illustrate, consider the strictly diagonally dominant stochastic matrix [35]

$$P = \begin{bmatrix} 0.6 & 0.3 & 0.1 \\ 0.2 & 0.7 & 0.1 \\ 0.1 & 0.1 & 0.8 \end{bmatrix}.$$

Suppose we wish to compute $P(1/12)$ and $P(1/52)$. After (for example) four iterations of (3.3) with $c = 1$ we obtain (to four decimal places)

$$p = \frac{1}{12}: \qquad X = \begin{bmatrix} 0.9518 & 0.0384 & 0.0098 \\ 0.0253 & 0.9649 & 0.0098 \\ 0.0106 & 0.0089 & 0.9805 \end{bmatrix}, \qquad \|X^{12} - P\|_F = 4.7 \times 10^{-7}$$

and

$$p = \frac{1}{52}: \qquad X = \begin{bmatrix} 0.9886 & 0.0092 & 0.0023 \\ 0.0060 & 0.9917 & 0.0023 \\ 0.0025 & 0.0021 & 0.9954 \end{bmatrix}, \qquad \|X^{52} - P\|_F = 2.5 \times 10^{-7},$$

and both matrices are stochastic to the working precision of about 10^{-16}. Note that such a computation, requiring just matrix multiplication and the solution of multiple right-hand side linear systems, is easily carried out in a spreadsheet, which is a computing environment used by some finance practitioners.

In summary, Markov models provide an application of matrix roots that is little known to numerical analysts, and the Newton iterations (3.2) and (3.3) for computing these roots are well-suited to the application.

5. Numerical experiments. We present some numerical experiments to compare the behavior of Algorithm 1.1, Algorithm 3.3, and the Schur method of Smith [33]. First, we need to develop appropriate residual-based measures of numerical stability for pth roots and inverse pth roots.

Let $\widetilde{X} = X + E$ be an approximation to a pth root X of $A \in \mathbb{C}^{n\times n}$. Then $\widetilde{X}^p = A + \sum_{i=0}^{p-1} X^i E X^{p-1-i} + O(\|E\|^2)$. An obvious residual bound is $\|A - \widetilde{X}^p\| \le p\|X\|^{p-1}\|E\| + O(\|E\|^2)$. While this bound is satisfactory for $p = 2$ [14], for $p \ge 3$ it can be very weak, since $\|X^i\| \le \|X\|^i$ can be an arbitrarily weak bound. Therefore we use the vec operator, which stacks the columns of a matrix into one long column, and the Kronecker product [17, Chap. 4], to write

$$\operatorname{vec}(A - \widetilde{X}^p) = -\left(\sum_{i=0}^{p-1} \left(X^{p-1-i}\right)^T \otimes X^i\right) \operatorname{vec}(E) + O(\|E\|^2).$$

For the 2-norm, it follows that

$$\|A - \widetilde{X}^p\|_F \le \|E\|_F \left\| \sum_{i=0}^{p-1} \left(X^{p-1-i}\right)^T \otimes X^i \right\|_2 + O(\|E\|_F^2)$$

is a sharp bound, to first order in E. If we suppose that $\|E\|_F \le \epsilon \|X\|_F$, then

$$\frac{\|A - \widetilde{X}^p\|_F}{\|X\|_F \left\| \sum_{i=0}^{p-1} \left(X^{p-1-i}\right)^T \otimes X^i \right\|_2} \le \epsilon + O(\epsilon^2).$$

We conclude that if $\widetilde{X}$ is a correctly rounded approximation to a pth root $\widetilde{X}$ of A in floating point arithmetic with unit roundoff u, then we expect the *relative residual*

$$\rho_A(\widetilde{X}) := \frac{\|A - \widetilde{X}^p\|}{\|\widetilde{X}\| \big\| \sum_{i=0}^{p-1} \big(\widetilde{X}^{p-1-i}\big)^T \otimes \widetilde{X}^i \big\|}$$

to be of order u, where for practical purposes any norm can be taken. Therefore $\rho_A(\widetilde{X})$ is the appropriate residual to compute and compare with u. In [4] and [19] the scaled residual $\|A - \widetilde{X}^p\|/\|A\|$ was computed; this makes the interpretation of the numerical results therein difficult when the denominator of $\rho_A(\widetilde{X})$ is not of the same order as $\|A\|$.

For an approximate inverse pth root $\widetilde{X} \approx A^{-1/p}$ the situation is more complicated, as there is no natural residual. Criteria can be based on $A\widetilde{X}^p - I$, $\widetilde{X}^p A - I$, or indeed $\widetilde{X}^i A \widetilde{X}^{p-i} - I$ for any $i = 0\colon p$, as well as $\widetilde{X}^{-p} - A$ and $\widetilde{X}^p - A^{-1}$. Since they reduce to the pth root case discussed above, we will use the latter two residuals, which lead to the relative residuals $\rho_A(\widetilde{X}^{-1})$ and $\rho_{A^{-1}}(\widetilde{X})$. We compute the inverses in high precision to ensure that errors in the inversion do not significantly influence the computed residuals.

Iterations (3.2) and (3.3) can be terminated when $\|M_k - I\|$ is less than a suitable tolerance (nu in our experiments). This test has negligible cost and has proved to be reliable when used within Algorithm 3.3. In Algorithm 1.1 square roots were computed using the Schur method [14].

Our computational experience on a wide variety of matrices is easily summarized. The Schur method invariably produces a computed $\widehat{X} \approx A^{1/p}$ with $\rho_A(\widehat{X}) \approx u$, and $\rho_{A^{-1}}(\widehat{X}^{-1})$ is usually of order u but occasionally much larger. When computing $A^{-1/p}$, Algorithm 3.3 usually produces an $\widehat{X}$ with $\rho_A(\widehat{X}^{-1})$ order u, but occasionally this residual is a couple of orders of magnitude larger. When computing $A^{1/p}$, Algorithms 3.3 and 3.3a invariably yield $\rho_A(\widehat{X}) \approx u$.

We describe MATLAB tests with two particular matrices and $p = 5$. The first matrix is `gallery('frank',8)^5`, where the Frank matrix is upper Hessenberg and has real eigenvalues, the smaller of which are ill conditioned. The second matrix is a random nonnormal 8×8 matrix constructed as $A = QTQ^T$, where Q is a random orthogonal matrix and T, is in real Schur form with eigenvalues $\alpha_j \pm i\beta_j$, $\alpha_j = -j^2/10$, $\beta_j = -j$, $j = 1\colon n/2$ and $(2j, 2j+1)$ elements -450. The infinity norm is used in evaluating ρ. The results are summarized in Tables 5.1 and 5.2. The values for k_0, k_1, and the number of iterations are the same for Algorithms 3.3 and 3.3a. For the Frank matrix, $\rho_A(\widehat{X}^{-1}) \gg u$ but for the pth root approximation obtained using Algorithms 3.3 and 3.3a the residual is of order u. The five iterations required by the iterative phase of Algorithm 3.3 are typical. Both matrices reveal two weaknesses of Algorithm 1.1: it can require many iterations, making it significantly more expensive than the Schur method, and it can suffer from instability, as indicated by the relative residuals.

6. Conclusions. Our initial aim in this work was to strengthen existing convergence results for Newton's method for the inverse pth root. The analysis has led us to develop a hybrid algorithm—employing a Schur decomposition, matrix square roots, and two coupled versions of the Newton iteration—that computes either $A^{1/p}$ or $A^{-1/p}$. The new algorithm performs stably in practice and it is more efficient than the Schur method of Smith for large p that are not highly composite. Although the Newton iterations for $A^{1/p}$ and $A^{-1/p}$ have until recently rarely been used for

TABLE 5.1
Results for Frank matrix. $p = 5$, $\|A\|_2 = 4.3 \times 10^6$, $\|A^{1/p}\|_2 = 2.4 \times 10^1$, $\|A^{-1/p}\|_2 = 1.0 \times 10^4$.

Schur $\widehat{X} \approx A^{1/p}$	Inverse Newton $\widehat{X} \approx A^{-1/p}$, $\widehat{Y} \approx A^{1/p}$ (Alg. 3.3) $\widehat{Z} \approx A^{1/p}$ (Alg. 3.3a)	Newton (Alg. 1.1) $\widehat{X} \approx A^{1/p}$
$\rho_A(\widehat{X}) = 1.5\text{e-}16$ $\rho_{A^{-1}}(\widehat{X}^{-1}) = 1.8\text{e-}7$	$\rho_A(\widehat{X}^{-1}) = 2.5\text{e-}13$ $\rho_{A^{-1}}(\widehat{X}) = 1.8\text{e-}7$ $\rho_A(\widehat{Y}) = 8.2\text{e-}15$ $\rho_A(\widehat{Z}) = 9.8\text{e-}16$ $k_0 = 0$, $k_1 = 6$; 5 iterations	$\rho_A(\widehat{X}) = 1.8\text{e-}14$ $\rho_{A^{-1}}(\widehat{X}^{-1}) = 1.8\text{e-}7$ 19 iterations

TABLE 5.2
Results for random nonnormal matrix. $p = 5$, $\|A\|_2 = 4.5 \times 10^2$, $\|A^{1/p}\|_2 = 9.2 \times 10^5$, $\|A^{-1/p}\|_2 = 1.0 \times 10^6$.

Schur $\widehat{X} \approx A^{1/p}$	Inverse Newton $\widehat{X} \approx A^{-1/p}$, $\widehat{Y} \approx A^{1/p}$ (Alg. 3.3) $\widehat{Z} \approx A^{1/p}$ (Alg. 3.3a)	Newton (Alg. 1.1) $\widehat{X} \approx A^{1/p}$
$\rho_A(\widehat{X}) = 3.6\text{e-}18$ $\rho_{A^{-1}}(\widehat{X}^{-1}) = 4.1\text{e-}18$	$\rho_A(\widehat{X}^{-1}) = 5.0\text{e-}18$ $\rho_{A^{-1}}(\widehat{X}) = 9.7\text{e-}19$ $\rho_A(\widehat{Y}) = 1.5\text{e-}18$ $\rho_A(\widehat{Z}) = 5.4\text{e-}18$ $k_0 = 0$, $k_1 = 3$; 5 iterations	$\rho_A(\widehat{X}) = 3.1\text{e-}12$ $\rho_{A^{-1}}(\widehat{X}^{-1}) = 1.6\text{e-}11$ 21 iterations

$p > 2$, our work and that of Iannazzo [19] shows that these iterations are valuable practical tools, and that general-purpose algorithms can be built around them based on understanding of their convergence properties.

Acknowledgements. This work was carried out while the first author visited MIMS in the School of Mathematics at the University of Manchester; he thanks the School for its hospitality. Both authors thank the referees for their helpful comments.

REFERENCES

[1] D. H. BAILEY, *MPFUN: A portable high performance multiprecision package*, Technical Report RNR-90-022, NASA Ames Research Center, Moffett Field, CA, USA, Mar. 1990.
[2] ———, *A Fortran 90-based multiprecision system*, ACM Trans. Math. Software, 21 (1995), pp. 379–387.
[3] D. H. BAILEY, Y. HIDA, X. S. LI, AND B. THOMPSON, *ARPREC: An arbitrary precision computation package*, Technical Report LBNL-53651, Lawrence Berkeley National Laboratory, Berkeley, California, Mar. 2002.
[4] D. A. BINI, N. J. HIGHAM, AND B. MEINI, *Algorithms for the matrix pth root*, Numer. Algorithms, 39 (2005), pp. 349–378.
[5] Å. BJÖRCK AND S. HAMMARLING, *A Schur method for the square root of a matrix*, Linear Algebra Appl., 52/53 (1983), pp. 127–140.
[6] S. H. CHENG, N. J. HIGHAM, C. S. KENNEY, AND A. J. LAUB, *Approximating the logarithm of a matrix to specified accuracy*, SIAM J. Matrix Anal. Appl., 22 (2001), pp. 1112–1125.
[7] M. CORNEA-HASEGAN AND B. NORIN, *IA-64 floating-point operations and the IEEE standard for binary floating-point arithmetic*, Intel Technology Journal, Q4 (1999). `http://developer.intel.com/technology/itj/`.
[8] J. R. CUTHBERT, *On uniqueness of the logarithm for Markov semi-groups*, J. London Math. Soc., 4 (1972), pp. 623–630.

[9] J.-C. Evard and F. Jafari, *A complex Rolle's theorem*, Amer. Math. Monthly, 99 (1992), pp. 858–861.
[10] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, MD, USA, third ed., 1996.
[11] N. J. Higham, *Functions of a Matrix: Theory and Computation*. Book in preparation.
[12] ———, *Computing the polar decomposition—with applications*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 1160–1174.
[13] ———, *Newton's method for the matrix square root*, Math. Comp., 46 (1986), pp. 537–549.
[14] ———, *Computing real square roots of a real matrix*, Linear Algebra Appl., 88/89 (1987), pp. 405–430.
[15] ———, *Stable iterations for the matrix square root*, Numer. Algorithms, 15 (1997), pp. 227–242.
[16] ———, *Accuracy and Stability of Numerical Algorithms*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, second ed., 2002.
[17] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*, Cambridge University Press, 1991.
[18] B. Iannazzo, *A note on computing the matrix square root*, CALCOLO, 40 (2003), pp. 273–283.
[19] ———, *On the Newton method for the matrix pth root*, SIAM J. Matrix Anal. Appl., (2006). To appear.
[20] R. B. Israel, J. S. Rosenthal, and J. Z. Wei, *Finding generators for Markov chains via empirical transition matrices, with applications to credit ratings*, Mathematical Finance, 11 (2001), pp. 245–265.
[21] A. H. Karp and P. Markstein, *High-precision division and square root*, ACM Trans. Math. Software, 23 (1997), pp. 561–589.
[22] C. S. Kenney and A. J. Laub, *Condition estimates for matrix functions*, SIAM J. Matrix Anal. Appl., 10 (1989), pp. 191–209.
[23] Ç. K. Koç and B. Bakkaloğlu, *Halley's method for the matrix sector function*, IEEE Trans. Automat. Control, 40 (1995), pp. 944–949.
[24] A. Kreinin and M. Sidelnikova, *Regularization algorithms for transition matrices*, Algo Research Quarterly, 4 (2001), pp. 23–40.
[25] P. Laasonen, *On the iterative solution of the matrix equation $AX^2 - I = 0$*, M.T.A.C., 12 (1958), pp. 109–116.
[26] S. Lakić, *On the computation of the matrix k-th root*, Z. Angew. Math. Mech., 78 (1998), pp. 167–172.
[27] B. Meini, *The matrix square root from a new functional perspective: Theoretical results and computational issues*, SIAM J. Matrix Anal. Appl., 26 (2004), pp. 362–376.
[28] H.-O. Peitgen, H. Jürgens, and D. Saupe, *Fractals for the Classroom. Part Two: Complex Systems and Mandelbrot Set*, Springer-Verlag, New York, 1992.
[29] B. Philippe, *An algorithm to improve nearly orthonormal sets of vectors on a vector processor*, SIAM J. Alg. Discrete Methods, 8 (1987), pp. 396–403.
[30] M. Schroeder, *Fractals, Chaos, Power Laws: Minutes from an Infinite Paradise*, W. H. Freeman, New York, 1991.
[31] L. S. Shieh, Y. T. Tsay, and C. T. Wang, *Matrix sector functions and their applications to system theory*, IEE Proc., 131 (1984), pp. 171–181.
[32] B. Singer and S. Spilerman, *The representation of social processes by Markov models*, Amer. J. Sociology, 82 (1976), pp. 1–54.
[33] M. I. Smith, *A Schur algorithm for computing matrix pth roots*, SIAM J. Matrix Anal. Appl., 24 (2003), pp. 971–989.
[34] R. A. Smith, *Infinite product expansions for matrix n-th roots*, J. Australian Math. Soc., 8 (1968), pp. 242–249.
[35] F. V. Waugh and M. E. Abel, *On fractional powers of a matrix*, J. Amer. Statist. Assoc., 62 (1967), pp. 1018–1021.