SIAM J. MATRIX ANAL. APPL.
Vol. 14, No. 2, pp. 456–467, April 1993

# ON ACCURATE COMPUTATIONS OF THE PERRON ROOT*

L. ELSNER†, I. KOLTRACHT‡§, M. NEUMANN‡¶, AND D. XIAO‡#

**Abstract.** This paper establishes a new componentwise perturbation result for the Perron root of a nonnegative and irreducible matrix. The error bound is independent of the angle between left and right Perron eigenvectors. It is shown that a known inverse iteration algorithm with new stopping criteria will have a small componentwise backward error, which is consistent with the perturbation result. Numerical experiments demonstrate that the accuracy of the Perron root computed by the proposed algorithm is, indeed, independent of the angle.

**Key words.** nonnegative matrices, Perron root, sparse systems, backward error, componentwise perturbations, stable algorithms

**AMS(MOS) subject classifications.** 65F70, 15A06, 15A18

**1. Introduction.** In this paper we consider the problem of how to accurately compute the Perron root of a nonnegative and irreducible matrix $A$. It is well known (see Berman and Plemmons [3]) that the Perron root, which equals the spectral radius of $A$, is a simple eigenvalue to which there corresponds a positive eigenvector.

For general matrices, the sensitivity of a simple eigenvalue to perturbations depends on the angle between normalized left and right eigenvectors corresponding to the eigenvalue. The following result appears in Wilkinson's book [10]: *If $\lambda$ is a simple eigenvalue of a square matrix $A$ and $y$ and $x$ are corresponding normalized left and right eigenvectors, then for sufficiently small $\delta$ the matrix $A + \delta E$, with $\|E\|_2 = 1$, has a simple eigenvalue $\hat{\lambda}$ that satisfies the inequality*

$$|\lambda - \hat{\lambda}| \leqq \frac{\delta}{|y^T x|} + o(\delta).$$

Using a formula for the componentwise condition number for general continuous maps given in Gohberg and Koltracht [6], we shall obtain in § 2 the componentwise condition number of a simple eigenvalue of an $n \times n$ matrix $A$ and compare it with the usual normwise condition number. In the special case when $A$ is an $n \times n$ nonnegative and irreducible matrix and the eigenvalue in question is the Perron root, then for relatively small componentwise perturbations in $A$, i.e.,

$$(1.1) \qquad |E_{i,j}| \leqq \varepsilon A_{i,j}, \qquad i, j = 1, \ldots, n,$$

* Received by the editors August 7, 1991; accepted for publication (in revised form) November 12, 1991.

† Department of Mathematics, University of Bielefeld, P.O. Box 8640, 4800-Bielefeld 1, Germany. The research of this author was supported in part by Sonderforschungsbereich 343, "Discrete Structures in Mathematics."

‡ Department of Mathematics, University of Connecticut, Storrs, Connecticut 06269-3009.

§ The research of this author was supported in part by National Science Foundation grant DMS-9007030 and by Sonderforschungsbereich 343, "Discrete Structures in Mathematics," through the Department of Mathematics, University of Bielefeld.

¶ The research of this author was supported in part by U.S. Air Force Office of Scientific Research grant AFOSR-88-0047, by National Science Foundation grants DMS-8901860 and DMS-9007030, and by Sonderforschungsbereich 343, "Discrete Structures in Mathematics," through the Department of Mathematics, University of Bielefeld.

# The research of this author was supported in part by National Science Foundation grants DMS-8901860 and DMS-9007030.

we shall show that

$$(1.2)\qquad \frac{|\hat{\lambda}-\lambda|}{\lambda} \leqq \varepsilon + o(\varepsilon).$$

Hence the componentwise condition number of the Perron root is 1. In fact, we show in Theorem 1 of § 2 that a better result than (1.2) is possible. It is that

$$(1.3)\qquad \frac{|\hat{\lambda}-\lambda|}{\lambda} \leqq \varepsilon.$$

In view of (1.3) a natural question to ask is whether one can find an algorithm for computing the Perron root that is numerically stable in the sense of (1.1). To be precise, we are looking for an algorithm for computing the Perron root of $A$ for which the computed root is the exact Perron root of a perturbation $A + E$ of $A$ satisfying

$$(1.4)\qquad |E_{i,j}| \leqq f(n)uA_{i,j}, \qquad i,j = 1, \ldots, n,$$

where $u$ is the unit round-off error and $f(n)$ is a slowly growing function. Then the computed root will satisfy

$$(1.5)\qquad \frac{|\hat{\lambda}-\lambda|}{\lambda} \leqq f(n)u.$$

Since any algorithm for computing $\lambda$ is iterative, we must find an algorithm such that, subject to certain stopping criteria, the approximate root produced by the algorithm will be an exact root of $A + E$, where $E$ is small and satisfies (1.4).

One such algorithm is presented in § 3. It is based on a variant of the inverse iteration due to Noda [7], which was shown by Elsner [4] to be quadratically convergent to the Perron root and which involves, at each stage of the iteration, the solution of a linear system whose coefficient matrix is an M-matrix. Using a theorem of Skeel [9], we show that if at each stage of the algorithm the M-matrix system is solved by Gaussian elimination followed by an iterative refinement of the solution and if certain stopping criteria are satisfied, then this results in an algorithm such that the approximation to the Perron root of $A$ is an exact Perron root of $A + E$ with $E$ satisfying (1.4). This result is given in § 3.

In § 4 we present some numerical experiments that illustrate the stability of the proposed algorithm. In particular, we give an example of an M-matrix for which the standard QR algorithm breaks down, whereas the algorithm of § 3 computes the Perron root with high accuracy, as expected. These results also show that the power-squaring algorithm (see Friedland [5])

$$(1.6)\qquad \lambda = \rho(A) = \lim_{k\to\infty} \|A^{2^k}\|_{\infty}^{1/2^k},$$

where $\rho(\cdot)$ denotes the spectral radius, which is seemingly natural from the point of view of round-off error analysis, is inferior in its convergence speed. (However, it should be remarked that on a vector or parallel computer, the power-squaring method may become competitive because its rate of convergence could be compensated for by fast matrix–matrix multiplications.)[1] Our numerical experiments also show that the computed Perron vectors are highly accurate, although we do not yet have a full perturbation analysis for explaining this evidence.

[1] Friedland [5] also notes that the power-squaring algorithm could be used to compute the subdominant eigenvalue of a nonnegative and irreducible matrix.

**2. Perturbation results.** Suppose that the matrix $A \in R^{n,n}$ has a simple eigenvalue $\lambda = \lambda(A)$. It is well known that the map $F_E : \varepsilon \to \lambda(A + \varepsilon E)$ is analytic in a neighborhood of 0; see, for example, Wilkinson [10, pp. 66–67]. Therefore, the map $F : \hat{A} \to \lambda(\hat{A})$ has continuous partial derivatives with respect to each entry at $A$ and (see [10, Chap. 2])

$$\frac{\partial F}{\partial_{i,j}}(A) := \lim_{t \to 0} \frac{F(A + tE_{i,j}) - F(A)}{t} = \frac{y_i x_j}{y^T x}, \qquad i, j = 1, \dots, n.$$

Hence, as a map from $R^{n^2} \to R$, $F$ is differentiable at $A$ and

$$F'(A) = \left[\frac{\partial F}{\partial_{1,1}}, \dots, \frac{\partial F}{\partial_{1,n}}, \frac{\partial F}{\partial_{2,1}}, \dots, \frac{\partial F}{\partial_{n,n}}\right].$$

According to a formula of Gohberg and Koltracht [6], the sensitivity of $F(A)$ to componentwise perturbations in $A$ as in (1.1), for $\lambda \neq 0$, is characterized by the componentwise condition number of $F$ at $A$ given by

$$c(F, A) = \frac{\|F'(A)D_A\|_\infty}{\|F(A)\|_\infty} = \frac{\|F'(A)D_A\|_\infty}{|\lambda|},$$

where

$$D_A = \operatorname{diag}(a_{1,1}, \dots, a_{1,n}, a_{2,1}, \dots, a_{n,n}).$$

This means that

$$c(F, A) = \frac{\|(y_1 a_{1,1} x_1, \dots, y_1 a_{1,n} x_n, y_2 a_{2,1} x_1, \dots, y_n a_{n,n} x_n)\|_\infty}{|\lambda|},$$

where the infinity norm of the map $F'(A)D_A$ is, in fact, the 1-norm of the row vector that represents it. Thus

$$c(F, A) = \frac{\sum_{i,j=1}^{n} |y_i a_{i,j} x_j|}{|\lambda|\,|y^T x|} = \frac{|y|^T |A|\,|x|}{|\lambda|\,|y^T x|}, \tag{2.1}$$

where an absolute value of a matrix is the corresponding matrix of the absolute values of its entries. Thus if $E$ is a perturbation of $A$ which satisfies (1.1), the $\hat{\lambda} = \lambda(A + E)$ satisfies

$$\frac{|\hat{\lambda} - \lambda|}{|\lambda|} \leqq c(F, A)\varepsilon + o(\varepsilon), \tag{2.2}$$

where $o(\varepsilon)$ means that $\lim_{\varepsilon \to 0} / \varepsilon = 0$.

It is interesting to compare (2.1) with the usual condition number taken with respect to normwise perturbations in $A$. For example, let us consider perturbations in $A$ with respect to the Frobenius norm, in which case

$$k(F, A) = \frac{\|F'(A)\|_2 \|A\|_F}{|\lambda|},$$

where the 2-norm of $F'(A)$ is its norm as a map from $R^{n^2}$ to $R$ and therefore coincides with its usual 2-norm. Since $\|x\|_2 = \|y\|_2 = 1$, it is clear that

$$\|F'(A)\|_2 = \frac{1}{|y^T x|} \sum_{i,j=1}^{n} (x_i y_j)^2 = \frac{1}{|y^T x|},$$

and so

$$k(F, A) = \frac{\|A\|_F}{|\lambda|\,|y^Tx|}. \tag{2.3}$$

It is clear that $c(F, A) \leqq k(F, A)$. Moreover, it is possible that $k(F, A)$ is large, while $c(F, A)$ is much smaller. Indeed, this may be the case if the smallness of $y^Tx$ and/or $\lambda$ is offset by a small $|y^T|\,|A|\,|x|$.

Suppose now that $A$ is a nonnegative and irreducible matrix and $\lambda$ is its Perron root. Then, because $x$ and $y$ are positive vectors, $|y^T|\,|A|\,|x| = y^TAx = \lambda y^Tx$, and we see at once that $c(F, A) = 1$. According to (2.2),

$$\frac{|\hat{\lambda} - \lambda|}{|\lambda|} \leqq \varepsilon + o(\varepsilon).$$

We can, in fact, improve on this result.

THEOREM 1. *Suppose that $A$ is an $n \times n$ nonnegative and irreducible matrix, and suppose that $E$ is an $n \times n$ real matrix such that*

$$|E| \leqq \varepsilon A,$$

*where $\varepsilon \leqq 1$. Let $\lambda$ and $\lambda_E$ denote, respectively, the Perron roots of $A$ and $A + E$. Then*

$$\frac{|\lambda_E - \lambda|}{\lambda} \leqq \varepsilon. \tag{2.4}$$

*Proof.* The inequality (1.1) can be written as

$$0 \leqq A - \varepsilon A \leqq A + E \leqq A + \varepsilon A.$$

Since $\rho(\cdot)$ is monotone on the nonnegative matrices (see, for example, [3]), it follows that

$$\rho(A - \varepsilon A) \leqq \rho(A + E) \leqq \rho(A + \varepsilon A).$$

Since $\rho(A \pm \varepsilon A) = (1 \pm \varepsilon)\rho(A)$, we get that

$$(1 - \varepsilon)\lambda \leqq \lambda_E \leqq (1 + \varepsilon)\lambda.$$

Because $\lambda > 0$, the last inequality is equivalent to (2.4). □

**3. A stable algorithm for computing the Perron root.** Let $A$ be an irreducible nonnegative matrix, $p$ be the Perron vector of $A$ whose infinity norm is 1, and $\rho(A)$ be the Perron root of $A$. The basis for our algorithm is a certain inverse iteration due to Noda [7], which has been generalized and shown to be quadratically convergent by Elsner [4].

INVERSE ITERATION ALGORITHM. For a given $y_0 > 0$ define iteratively

$$\mu_s = \max\left(\frac{Ay_s}{y_s}\right),$$

$$x_s = (\mu_s I - A)^{-1} y_s,$$

and

$$y_{s+1} = \frac{x_s}{\|x_s\|_\infty}.$$

Then, for any $s$, $\rho(A) \leqq \mu_{s+1} \leqq \mu_s$ and

$$\mu_{s+1} - \rho(A) \leqq C_1(\mu_s - \rho(A))^2$$

and

$$\operatorname{osc}\left(\frac{y_{s+1}}{p}\right) \leqq C_2\left[\operatorname{osc}\left(\frac{y_s}{p}\right)\right]^2,$$

where $C_1$ and $C_2$ are some constants depending on $y_0$ and $A$ only, and where, for any two $n$-vectors $u = (u_1, \dots, u_n)^T$ and $v = (v_1, \dots, v_n)^T$, $v > 0$, by definition

$$\max\left(\frac{u}{v}\right) = \max_{1 \leq i \leq n} \frac{u_i}{v_i}, \qquad \min\left(\frac{u}{v}\right) = \min_{1 \leq i \leq n} \frac{u_i}{v_i},$$

and

$$\operatorname{osc}\left(\frac{u}{v}\right) = \max\left(\frac{u}{v}\right) - \min\left(\frac{u}{v}\right).$$

(For more background material concerning properties and applications of vector *oscillation*, see Seneta [8, Chap. 3.4].) Since $\mu_s > \rho(A)$ so long as $y_0$ is not a scalar multiple of $p$, it follows that the matrix $(\mu_s I - A)$ is an M-matrix. A certain version of the Gaussian elimination algorithm due to Ahac, Buoni, and Olesky [2] can be used to compute the solution of the equation $(\mu_s I - A)x_s = y_s$. According to [2], the pivots will always be found among the diagonal elements, and this saves some execution time. It is also shown in [2] that the computed solution is exact for a perturbed system whose relative distance from the original system is $O(n)u$, where $u$ is the machine precision. This distance, however, is measured normwise, whereas the bound of Theorem 1 requires componentwise perturbations. In order to get a small componentwise backward error, we add a one-step iterative refinement, as recommended by the following theorem of Skeel [9] concerning solution of $Ax = b$ by Gaussian elimination.

THEOREM 2 (Skeel [9]). *Let $u$ be the machine precision, and let the arithmetic be such that the floating-point result* fl $(a*b)$ *of the operation $a*b$, where $* \in \{+, -, \times, /\}$, satisfies* fl $(a*b) = (a*b)(1+e)$ *with $|e| \leqq u$. There is a function $f(A, b)$, typically behaving as $O(n)$, such that when the product of $\bar{k}(A) = \| |A| |A^{-1}| \|$ and $\sigma(A, x) = \max(|A||x|)/\min(|A||x|)$ is less than $(f(A, b)u)^{-1}$ and there is no overflow or underflow, then the following one-step refinement*:

*Solve $Ax = b$ using Gaussian elimination, obtaining solution $\hat{x}$, and saving the* LU *factors*;

*Compute the residual $r = A\hat{x} - b$ (using single-precision $u$)*;

*Solve $Ad = r$ for $d$ using the saved* LU *factors of $A$*;

*Update $\hat{x} = \hat{x} - d$*;

*gives a vector $\hat{x}$ that is the exact solution of the equation*

$$(A + \Delta A)\hat{x} = b + \Delta b,$$

*where*

$$|\Delta A_{i,j}| \leqq (n+1)u|A_{i,j}|, \qquad i, j = 1, \dots, n,$$

*and*

$$|\Delta b_j| \leqq (n+1)u|b_j|, \qquad j = 1, \dots, n.$$

We formulate the following algorithm for computing approximations to the Perron root $\rho(A)$ and the Perron vector $p$ of a nonnegative irreducible matrix $A$.

ALGORITHM. Let $u$ be the machine precision. Start with $y_0 > 0$ and $\mu_0 = \max(Ay_0/y_0)$. For $s = 0, 1, \ldots$

1. Compute the LU factorization

$$(\mu_s I - A) = L_s U_s,$$

and solve for $x_s$

$$L_s U_s x_s = y_s$$

by the Ahac, Buoni, and Olesky algorithm; save the LU factors;

2. Compute $r = Ax_s - y_s$;
3. Solve $Ad = r$ using the saved LU factors $L_s$ and $U_s$;
4. Update $\bar{x}_s = x_s - d$;
5. Compute

$$y_{s+1} = \frac{\bar{x}_s}{\|\bar{x}_s\|_\infty} \quad \text{and} \quad \mu_{s+1} = \max\left(\frac{Ay_{s+1}}{y_{s+1}}\right);$$

6. Proceed until

$$\|\bar{x}_s\|^{-1} \leqq u^{1/2} \quad \text{and} \quad \operatorname{osc}\left(\frac{y_s}{y_{s+1}}\right) \leqq u^{1/2}.$$

The following theorems characterize the numerical behavior of the algorithm and explain the stopping criteria in step 6 above. Variables computed by the algorithm are denoted by hats. The norm below always means the infinity norm. It is assumed for simplicity that the matrix $A$ is of the size $(n-1) \times (n-1)$.

THEOREM 3. *Suppose that the algorithm terminates when*

$$\|\hat{\bar{x}}_s\|^{-1} = \varepsilon_1 \quad \textit{and} \quad \operatorname{osc}\left(\frac{\hat{y}_s}{\hat{y}_{s+1}}\right) = \varepsilon_2$$

*and that*

$$4\eta(1+\eta) < \hat{\mu}_s \frac{1-nu}{1+nu}, \tag{3.1}$$

*where* $\eta = \max(\varepsilon_1, \varepsilon_2)$. *Suppose further that*

$$\|(\hat{\mu}_s I - A)^{-1} |\hat{\mu}_s I - A|\| \frac{\max(|\hat{\mu}_s - A|\hat{\bar{x}}_s)}{\min(|\hat{\mu}_s - A|\hat{\bar{x}}_s)} < (fu)^{-1}, \tag{3.2}$$

*where* $f = f(\hat{\mu}_s I - A, \hat{y}_s)$ *is as in Theorem* 2. *Then* $\hat{\mu}_s - \varepsilon_1$ *and* $\hat{y}_{s+1}$ *are the exact Perron root and Perron vector, respectively, of the matrix* $\bar{A}$, *where*

$$\bar{A}_{i,j} = A_{i,j}(1 + e_{i,j})(1 + \delta_i), \quad i, j = 1, \ldots, n-1, \quad i \neq j,$$

*and*

$$\bar{A}_{i,i} = A_{i,i}(1 + e_{i,i}), \qquad i = 1, \ldots, n-1,$$

*with*

$$|e_{i,j}| \leqq nu, \qquad |e_{i,i}| \leqq 2nu,$$

*and*

$$|\delta_i| \leqq \frac{4nu}{1-nu} + \frac{4\varepsilon_1(nu + \varepsilon_2 + nu\varepsilon_2)}{\hat{\mu}_s(1-nu)}.$$

*Proof.* It follows from the definition of $\hat{\bar{x}}_s$ in the algorithm and from Theorem 2 applied to the equation $(\hat{\mu}_s I - A)\hat{x}_s = \hat{y}_s$ that

$$(\hat{\mu}_s I - A - E)\hat{\bar{x}}_s = \hat{y}_s + \Delta y,$$

where

$$|E_{i,j}| \leqq nu|(\hat{\mu}_s I - A)_{i,j}| \tag{3.3}$$

and

$$|\Delta y_i| \leqq nu|(\hat{y}_s)_i|.$$

Therefore, dividing by the norm of $x_s$ and using the definition of $\varepsilon_1$ we can write

$$(\hat{\mu}_s I - A - E)\hat{y}_{s+1} = \varepsilon_1(I + D_\varepsilon)\hat{y}_s, \tag{3.4}$$

where $D_\varepsilon$ is a diagonal matrix such that $|f_i| \leqq nu$ with $D_\varepsilon(i, i) = f_i$. Note that the infinity norm of $\hat{y}_{s+1}$ and $\hat{y}_s$ equals 1, and that they are both positive vectors. Thus

$$\min\left(\frac{\hat{y}_s}{\hat{y}_{s+1}}\right) \leqq 1 \leqq \max\left(\frac{\hat{y}_s}{\hat{y}_{s+1}}\right).$$

From the equality

$$\max\left(\frac{\hat{y}_s}{\hat{y}_{s+1}}\right) = \operatorname{osc}\left(\frac{\hat{y}_s}{\hat{y}_{s+1}}\right) + \min\left(\frac{\hat{y}_s}{\hat{y}_{s+1}}\right)$$

it follows that

$$\max\left(\frac{\hat{y}_s}{\hat{y}_{s+1}}\right) \leqq 1 + \varepsilon_2 \quad \text{and} \quad \min\left(\frac{\hat{y}_s}{\hat{y}_{s+1}}\right) \geqq 1 - \varepsilon_2.$$

Therefore,

$$(1 - \varepsilon_2)\hat{y}_{s+1} \leqq \hat{y}_s \leqq (1 + \varepsilon_2)\hat{y}_{s+1},$$

and hence we can write $\hat{y}_s = (I + D_{\varepsilon_2})\hat{y}_{s+1}$, where $D_{\varepsilon_2}$ is a diagonal matrix such that $|h_i| \leqq \varepsilon_2$ with $D_{\varepsilon_2}(i, i) = h_i$. Substituting into (3.4), we get

$$(\hat{\mu}_s I - A - E)\hat{y}_{s+1} = \varepsilon_1(1 + D_\varepsilon)(1 + D_{\varepsilon_2})\hat{y}_{s+1}. \tag{3.5}$$

This, in turn, can be rewritten as

$$\sum_{j=1, j\neq i}^{n} (A_{i,j} + E_{i,j})\hat{y}_{s+1}(j) = ((\hat{\mu}_s - A_{i,i}) - E_{i,i} - \varepsilon_1(1 + f_i)(1 + h_i))\hat{y}_{s+1}(i). \tag{3.6}$$

Let us now define $e_{i,j} = E_{i,j}/A_{i,j}$ for $i \neq j$ (where $0/0 = 0$). Let $e_{i,i} = 0$ if $A_{i,i} < \hat{\mu}_s/2$, and let $e_{i,i} = E_{i,i}/A_{i,i}$ if $A_{i,i} \geqq \hat{\mu}_s/2$. For the case $A_{i,i} < \hat{\mu}_s/2$, let $\delta_i$ be such that

$$\sum_{j=1, j\neq i}^{n} (A_{i,j} + E_{i,j})\hat{y}_{s+1}(j)\delta_i = (-E_{i,i} - \varepsilon_1(f_i + h_i + f_i h_i))\hat{y}_{s+1}(i). \tag{3.7}$$

In this case it follows from (3.6) and (3.3) that

$$\begin{aligned} |\delta_i| &= \left|\frac{-E_{i,i} - \varepsilon_1(f_i + h_i + f_i h_i)}{(\hat{\mu}_s - A_{i,i}) - E_{i,i} - \varepsilon_1(1 + f_i)(1 + h_i)}\right| \\ &\leqq \frac{|E_{i,i}|}{\hat{\mu}_s(1 - nu)/2 - \hat{\mu}_s(1 - nu)/4} + \frac{\varepsilon_1(nu + \varepsilon_2 + nu\varepsilon_2)}{\hat{\mu}_s(1 - nu)/2 - \hat{\mu}_s(1 - nu)/4} \\ &\leqq \frac{4nu}{(1 - nu)} + \frac{\varepsilon_1(nu + \varepsilon_2 + nu\varepsilon_2)}{\hat{\mu}_s(1 - nu)/4}. \end{aligned}$$

If $A_{i,i} \geqq \hat{\mu}_s/2$, then let $\delta_i$ be such that

$$(3.8) \qquad \sum_{j=1}^{n} (A_{i,j} + E_{i,j})\hat{y}_{s+1}(j)\delta_i = -\varepsilon_1(f_i + h_i + f_i h_i)\hat{y}_{s+1}(i).$$

Again, it follows from (3.6) and (3.3) that

$$|\delta_i| = \left|\frac{-\varepsilon_1(f_i + h_i + f_i h_i)}{\hat{\mu}_s - \varepsilon_1(1+f_i)(1+h_i)}\right| \leqq \frac{\varepsilon_1(nu + \varepsilon_2 + nu\varepsilon_2)}{\hat{\mu}_s - \hat{\mu}_s(1-nu)/4}.$$

Using the definition of $\delta_i$ and $e_{i,j}$ and the equality (3.5), we finally obtain

$$(3.9) \qquad \sum_{j=1, j\neq i}^{n} (A_{i,j} + E_{i,j})(1+\delta_i)\hat{y}_{s+1}(j) + A_{i,i}(1+e_{i,i})\hat{y}_{s+1}(i) = (\hat{\mu}_s - \varepsilon_1)\hat{y}_{s+1}(i).$$

The theorem is proved. □

It follows from Theorem 1 that if $\eta = u^{1/2}$ and if the conditions (3.1) and (3.2) are satisfied, then $\hat{\mu}_s - \varepsilon_1$ is the exact Perron root of $\bar{A}$ such that $|A - \bar{A}| \leqq (u^{1/2} + O(u))A$. Indeed, it is easy to see from (3.1) that $|\delta_i| \leqq u^{1/2} + O(u)$. Thus, by Theorem 1, $|\hat{\mu}_s - \varepsilon_1 - \rho(A)| \leqq (u^{1/2} + O(u))\rho(A)$, and hence $|\hat{\mu}_s - \rho(A)| \leqq (1 + \rho(A))u^{1/2} + O(u)$. Since the sequence $\{\hat{\mu}_s\}$ converges quadratically to $\rho(A)$, it follows that

$$\hat{\mu}_{s+1} - \rho(A) \leqq C_1(1 + \rho(A))^2 u + o(u),$$

where $C_1$ is a constant defined in the inverse iteration algorithm.

We comment that condition (3.1) is not restrictive since $A$ can be replaced, e.g., by $I + A$. Let us argue that the condition (3.2) is not restrictive either. This condition is necessary to assure that the product of $\hat{k}(\mu_s I - A)$ and $\sigma(\mu_s I - A, x_s)$ is not too large relative to the machine precision for Skeel's theorem to apply. However, as was pointed out by Wilkinson [10, § 9.47], it is typical for the inverse iteration that the solution of $(\mu_s I - A)x_s = y_s$ is computed with high relative accuracy, while the coefficient matrix may be very ill conditioned. Further justification of this point will be given in Theorem 4 below. The second factor, $\sigma(\mu_s I - A, x_s)$, is necessary in Skeel's theorem to account for possible zeros of the solution vector and the right-hand side, as is discussed in detail in Arioli, Demmel, and Duff [1]. Since in our case both $x_s$ and $y_s$ are positive vectors, one can expect Skeel's result to hold even if the factor $\sigma(\mu_s I - A, x_s)$ is large. These arguments are supported by numerical evidence in § 4. The experiments also suggest that if only the Perron root, but not the Perron vector, needs to be calculated, then the stopping criterion of the algorithm should be changed to $\|x_s\|^{-1} < u^{1/2}$, since the second criterion, which assures accuracy of the Perron vector, may slow down the algorithm to some degree. In this case, as can be seen from the proof of the theorem, condition (3.1) can be replaced by the condition

$$4\varepsilon_1(1 + \varepsilon_2) < \hat{\mu}_s \frac{1 - nu}{1 + nu}.$$

It is now clear that $\varepsilon_2$ does not need to be very small for this condition to be satisfied.

In the next theorem we show when the condition (3.2) will always be satisfied.

THEOREM 4. *Let $\hat{y}_{s+1}$ and $\hat{\mu}_s$ denote the quantities computed in the algorithm for some value of $s$. Let $p$ be the Perron vector of $A$, and define*

$$S_p = \frac{\max(p)}{\min(p)}, \qquad S_y = \frac{\max(\hat{y}_{s+1})}{\min(\hat{y}_{s+1})},$$

*and*

$$\delta_m = \min_{1 \leq i \leq n} \sum_{i \neq j} A_{i,j}.$$

*Then*

$$\|(\hat{\mu}_s I - A)^{-1}|(\hat{\mu}_s I - A)|\|(\hat{\mu}_s - \rho(A)) \leqq 2S_p\hat{\mu}_s, \tag{3.10}$$

$$\frac{\max(|\hat{\mu}_s I - A|\hat{y}_{s+1})}{\min(|\hat{\mu}_s I - A|\hat{y}_{s+1})}\delta_m \leqq 2S_y\hat{\mu}_s. \tag{3.11}$$

*Proof.* Observe that

$$|(\hat{\mu}_s I - A)| = A - 2D_A + \hat{\mu}_s I,$$

where $D_A = \operatorname{diag}(A_{1,1}, \ldots, A_{n,n})$. Let $y = |\hat{\mu}_s I - A|\hat{y}_{s+1}$. Then clearly

$$y \leqq \hat{\mu}_s \hat{y}_{s+1} + A\hat{y}_{s+1}$$

and

$$y \geqq (\hat{\mu}_s I - D_A)\hat{y}_{s+1} + \delta_m \min(\hat{y}_{s+1}).$$

It follows from the inequality $\hat{\mu}_s \geqq \hat{\mu}_{s+1}$ and the definition of $\hat{\mu}_{s+1}$ that

$$\frac{\max(y)}{\min(y)} \leqq \frac{\hat{\mu}_s + \hat{\mu}_{s+1}}{\min_i(\hat{\mu}_s - A_{i,i}) + \delta_m} S_y \leqq \frac{2\hat{\mu}_s}{(\delta_m)} S_y.$$

Thus (3.11) is proved. To show (3.10), observe that for any $\mu > \rho(A)$,

$$(\mu I - A)^{-1}p = (\mu - \rho(A))^{-1}p.$$

Let $D_p = \operatorname{diag}(p(1), \ldots, p(n))$ and $e = (1, \ldots, 1)^T$. Then

$$D_p^{-1}(\mu I - A)^{-1}D_p e = (\mu - \rho(A))^{-1}e,$$

which implies that

$$\|D_p^{-1}(\mu I - A)^{-1}D_p\| = (\mu - \rho(A))^{-1}.$$

Therefore,

$$\|(\mu I - A)^{-1}\| \leqq (\mu - \rho(A))^{-1}S_p$$

and

$$\begin{aligned}\|(\mu I - A)^{-1}|(\mu I - A)|\| &\leqq \|(\mu I - A)^{-1}|(\mu I + A)|\|\\ &= \|(\mu I - A)^{-1}|(2\mu I - (\mu I - A))|\|\\ &= \|2\mu(\mu I - A)^{-1} - I\| \leqq 2\mu\|(\mu I - A)^{-1}\|.\end{aligned}$$

Thus, clearly,

$$\|(\mu I - A)^{-1}|(\mu I - A)|\| \leqq 2\mu(\mu - \rho(A))^{-1}S_p.$$

The theorem is now proved. □

Theorem 4 shows that if $4S_pS_y\hat{\mu}_s^2 u \leqq f^{-1}\delta_m(\hat{\mu}_s - \rho(A))$, where $u$ is the computer precision, then the condition (3.2) of Theorem 3 will be satisfied for $s$, for which the algorithm terminates. We remark that as long as $(\hat{\mu}_s I - A)\hat{x}_s \approx \hat{y}_s$, then we have that $\hat{\mu}_s - \rho(A) \approx \|\hat{x}_s\|^{-1}$. Therefore, with $\hat{\mu}_s - \rho(A) \approx u^{1/2}\hat{\mu}_s$, one can expect that the condition (3.10) will be satisfied when $\|\hat{x}_s\| \approx u^{1/2}$ (see also [10, § 9.47]).

## 4. Numerical experiments.

Test matrices whose Perron vectors have various oscillatory properties can be generated, for example, as follows. Take any matrix with positive entries, and divide each row by the sum of the elements of this row. This is the matrix $B$. Choose any $n$ positive numbers $d_1, d_2, \ldots, d_n$, and let

$$D = \operatorname{diag}(d_1, d_2, \ldots, d_n).$$

Then

$$A = DBD^{-1} \tag{4.1}$$

is positive (and hence irreducible), the Perron root of $A$ is equal to 1, and the Perron vector of $A$ is $(d_1, d_2, \ldots, d_n)^T$. One possible choice of $B$ is with all its entries equal to $1/n$ and $d_j = d^{j-1}, j = 1, \ldots, n$, with $d < 1$. In this case a right Perron vector is $p = (1, d, \ldots, d^{n-1})^T$ and the left Perron vector is $q = (d^{n-1}, \ldots, d, 1)^T$. It is clear that the usual condition number of the Perron root $k(F, A)$ given by (2.3) can be arbitrarily large in this case, since $1/q^Tp = n^{-1}d^{1-n}$.

We have extensively tested two algorithms on such matrices, the algorithm of § 3 and the power-squaring method given by (1.6). The reason for testing the latter algorithm is its obvious numerical stability: there are no subtractions of numbers with the same sign and, with appropriate scaling, no overflows in the course of this method. The power-squaring method requires $n^3$ multiplication operations for each iteration step, while the algorithm we suggested in the previous section requires about $\frac{2}{3}n^3$ operations of multiplication and division per step because of the elimination involved. The experiments were performed with a computer precision $u = 2^{-52} \approx 10^{-16}$ using MATLAB on the SUN 3/50 workstation.

We observed that the algorithm of § 3 always converged quadratically to the exact Perron root, and with $\|x_s\| > 10^8 = u^{-1/2}$ the error in the computed $\mu_{s+1}$ was of $O(u) = O(10^{-16})$. The power-squaring method always converged more slowly, sometimes significantly so, than the first algorithm, which is not surprising because it has only a geometric rate of convergence. The comparison of the two methods on a parallel or vector machine could change their relative performances but we have not yet carried out such experiments.

In the following tables we give some typical numerical examples. The iteration is stopped when $\|x_s\| \geq m_x$. In the tables, $\varepsilon_1$ denotes $1/\|x_s\|$ and $\varepsilon_2$ denotes $\operatorname{osc}(y_s/y_{s+1})$. In Table 1 we present a reproducible result, where $A$ is the $20 \times 20$ matrix

$$A = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ \varepsilon & 0 & 0 & \cdots & 0 \end{bmatrix}$$

TABLE 1
$\varepsilon = 0.5 \times 10^{20}$.

| $m_x$ | $\varepsilon_1$ | $\varepsilon_2$ | $\mu_s - \rho$ | $\lvert\mu_s - \varepsilon_1 - \rho\rvert$ | $\mu_{s+1} - \rho$ |
|---|---|---|---|---|---|
| $1.2 \times 10^4$ | $4.1 \times 10^{-5}$ | $5.1 \times 10^{-2}$ | $4.1 \times 10^{-5}$ | $1.4 \times 10^{-9}$ | $1.0 \times 10^{-8}$ |
| $1.2 \times 10^8$ | $1.0 \times 10^{-8}$ | $1.3 \times 10^{-7}$ | $1.0 \times 10^{-8}$ | $6.8 \times 10^{-16}$ | $7.7 \times 10^{-16}$ |
| $1.2 \times 10^{10}$ | $7.7 \times 10^{-16}$ | $8.4 \times 10^{-15}$ | $7.7 \times 10^{-16}$ | 0 | 0 |
| $1.2 \times 10^{12}$ | $7.7 \times 10^{-16}$ | $8.4 \times 10^{-15}$ | $7.7 \times 10^{-16}$ | 0 | 0 |

TABLE 2
$\varepsilon = 1.2 \times 10^{-16}$.

| $m_x$ | $\varepsilon_1$ | $\varepsilon_2$ | $\mu_s - \rho$ | $\lvert\mu_s - \varepsilon_1 - \rho\rvert$ | $\mu_{s+1} - \rho$ |
|---|---|---|---|---|---|
| $1.2 \times 10^4$ | $1.4 \times 10^{-4}$ | $5.8 \times 10^{-3}$ | $1.4 \times 10^{-4}$ | $1.3 \times 10^{-7}$ | $4.3 \times 10^{-7}$ |
| $1.2 \times 10^8$ | $3.6 \times 10^{-12}$ | $1.4 \times 10^{-10}$ | $3.6 \times 10^{-12}$ | $1.38 \times 10^{-18}$ | $2.7 \times 10^{-16}$ |
| $1.2 \times 10^8$ | $3.6 \times 10^{-12}$ | $1.4 \times 10^{-10}$ | $3.6 \times 10^{-12}$ | $1.38 \times 10^{-18}$ | $2.7 \times 10^{-17}$ |
| $1.2 \times 10^{12}$ | $2.7 \times 10^{-17}$ | $6.6 \times 10^{-16}$ | $2.7 \times 10^{-17}$ | $6.9 \times 10^{-18}$ | $2.7 \times 10^{-17}$ |

with $\varepsilon = (0.5)^{20} = 9.5367 \times 10^{-7}$. The initial vector for our algorithm was chosen to be $y_0 = (1, 1, \ldots, 1)^T$. It is easy to see that $\rho(A) = \varepsilon^{1/20}$, $p = (1, \varepsilon^{1/n}, \ldots, \varepsilon^{(n-1)/n})^T$, $q = (\varepsilon^{(n-1)/n}, \ldots, \varepsilon^{1/n}, 1)^T$, and $q^Tp = n\varepsilon^{(n-1)/n} \approx 10^{-5}$. It takes 11 steps to get the results on the first line of Table 1, 12 steps on the second line, and 13 steps on the third and fourth lines.

In Table 2 we give results for the same $A$ above, but with $\varepsilon = (0.16)^{20} = 1.2 \times 10^{-16}$. It takes 21 steps to get the results on the first line of Table 2, 23 steps on the second and third lines, and 24 steps on the fourth line.

We see that for $s = 23$, when $m_x = u^{-1/2}$, the computed value $\mu_{s+1}$ approximates $\rho(A)$ within the computer precision. The relatively large value of $q^Tp$ has no effect on the accuracy of the algorithm. We continued experiments with this matrix for decreasing $\varepsilon = 10^{-14}, 10^{-15}, 10^{-16}$, and $10^{-17}$. The computed Perron root was accurate, and the results were similar, e.g., the Perron root was computed exactly after the condition $m_x^{-1} < u^{1/2}$ was satisfied. We also tested the standard QR algorithm of MATLAB (the function eig $(A)$) for these values of $\varepsilon$. For $\varepsilon = 10^{-14}$ the QR algorithm lost two significant figures, for $\varepsilon = 10^{-15}$ it lost three, and for $\varepsilon = 10^{-16}$ it lost four. For $\varepsilon = 10^{-17}$ the QR algorithm computed no significant figures at all, namely, it gave 20 complex eigenvalues $\lambda_1, \ldots, \lambda_{20}$ such that $0.06 > |\mathrm{Re}(\lambda_i)| > 0.04$ and $0.06 > |\mathrm{Im}(\lambda_i)| > 0.028$, while the value of the Perron root in this case was $10^{-17/20} = 0.141\ldots$.

In Table 3 we present a typical example of experiments with $20 \times 20$ matrices of the form (4.1) in which $B$ was generated randomly. The results are very similar to those of Table 1.

These examples, which are representative of a large number of other experiments that we performed, demonstrate that the separation angle between the right and left eigenvectors has no effect on the accuracy of the computed Perron root. They also show that the stopping criterion $\|x_s\|^{-1} < u^{1/2}$ gives the maximum possible accuracy and that there is no need to use a threshold smaller than $u^{1/2}$.

TABLE 3

| $m_x$ | $\varepsilon_1$ | $\varepsilon_2$ | $\mu_s - \rho$ | $\lvert\mu_s - \varepsilon_1 - \rho\rvert$ | $\mu_{s+1} - \rho$ |
|---|---|---|---|---|---|
| $1.2 \times 10^4$ | $1.5 \times 10^{-5}$ | $1.0 \times 10^{-2}$ | $1.5 \times 10^{-5}$ | $2.2 \times 10^{-7}$ | $4.2 \times 10^{-10}$ |
| $1.2 \times 10^8$ | $1.3 \times 10^{-9}$ | $8.5 \times 10^{-5}$ | $1.3 \times 10^{-9}$ | $5.3 \times 10^{-14}$ | $2.2 \times 10^{-16}$ |
| $1.2 \times 10^{10}$ | $3.3 \times 10^{-11}$ | $8.4 \times 10^{-11}$ | $3.3 \times 10^{-11}$ | $5.4 \times 10^{-17}$ | $2.2 \times 10^{-16}$ |
| $1.2 \times 10^{12}$ | $1.3 \times 10^{-15}$ | $1.8 \times 10^{-13}$ | $2.0 \times 10^{-15}$ | $2.2 \times 10^{-16}$ | $4.4 \times 10^{-16}$ |

## REFERENCES

[1] M. ARIOLI, J. W. DEMMEL, AND I. S. DUFF, *Solving sparse linear systems with sparse backward error*, SIAM J. Matrix Anal. Appl., 10 (1989), pp. 165–190.

[2] A. A. AHAC, J. J. BUONI, AND D. D. OLESKY, *Stable* LU *factorization of H-matrices*, Linear Algebra Appl., 99 (1988), pp. 97–110.

[3] A. BERMAN AND R. PLEMMONS, *Nonnegative Matrices in Mathematical Sciences*, Academic Press, New York, 1979.

[4] L. ELSNER, *Inverse iteration for calculating the spectral radius of a nonnegative irreducible matrix*, Linear Algebra Appl., 15 (1976), pp. 235–242.

[5] S. FRIEDLAND, *Revisiting matrix squaring*, Linear Algebra Appl., 154/156 (1991), pp. 59–63.

[6] I. GOHBERG AND I. KOLTRACHT, *Componentwise mixed and structured condition numbers*, submitted.

[7] T. NODA, *Note on the computation of the maximal eigenvalue of a nonnegative irreducible matrix*, Numer. Math., 17 (1971), pp. 382–386.

[8] E. SENETA, *Non-negative Matrices and Markov Chains*, Springer-Verlag, New York, 1981.

[9] R. D. SKEEL, *Scaling for numerical stability in Gaussian elimination*, J. Assoc. Comput. Mach., 26 (1979), pp. 494–526.

[10] J. H. WILKINSON, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, U.K., 1965.