Abstract

CHANCHAN, PRAKASH. An Algorithm for Computing the Perron Root of a Nonnegative Irreducible Matrix. (Under the direction of Carl D. Meyer.)

We present a new algorithm for computing the Perron root of a nonnegative irreducible matrix. The algorithm is formulated by combining a reciprocal of the well known Collatz's formula with a special inverse iteration algorithm discussed in [10, *Linear Algebra Appl., 15 (1976), pp 235-242*]. Numerical experiments demonstrate that our algorithm is able to compute the Perron root accurately and faster than other well known algorithms; in particular, when the size of the matrix is large. The proof of convergence of our algorithm is also presented.

AN ALGORITHM FOR COMPUTING THE PERRON ROOT OF A NONNEGATIVE IRREDUCIBLE MATRIX

by

PRAKASH CHANCHANA

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

APPLIED MATHEMATICS

Raleigh, North Carolina

2007

APPROVED BY

Dr. Carl D. Meyer Chair of Advisory Committee	Dr. Ernie L. Stitzinger Committee Member
Dr. Zhilin Li Committee Member	Dr. Min Kang Committee Member

Biography

Prakash Chanchana was born on July 9th 1979 and raised in Hat Yai, a small city in Southern part of Thailand. In 1995, he got accepted to the Armed Forced Military Preparatory School (AFAP) in Bangkok, Thailand. He spent two years at AFAP and then moved to Nakorn Nayok to attend the Chulachomklao Royal Military Academy (CRMA). During his first year cadet at CRMA, Prakash was awarded with a full scholarship by the Royal Thai Army to attend the prestigious Military College of South Carolina, the Citadel.

At the Citadel, Prakash was a member of India company and served as a Platoon Leader during his senior year. He graduated from the Citadel in 2002 with a Bachelor of Science in Mathematics. After that he moved to Raleigh, North Carolina to pursue his graduate studies. He received a Master's in Applied Mathematics from NC State University in 2004. During his graduate studies at NC State University, he taught undergraduate mathematics courses and was awarded the University Outstanding Teaching Assistant Award in 2005.

Acknowledgements

I would like to thank my advisor, Dr. Carl D. Meyer for his guidance, encouragement and support through out my graduate studies. I am grateful for the time that he spent guiding my research, listening and answering problems I encountered and for providing constructive comments on this dissertation. I have learned so much from him. I would also like to thank the members of my committee: Dr. Ernie Stitzinger, Dr. Zhilin Li and Dr. Min Kang.

I owe many thanks to Dr. Amy Langville for her assistance and encouragement. Her advice and generosity were invaluable.

I thank you my fiancé, Benz Suanmali, for believing in me and supporting me in everyway.

To my parents, thank you very much for your continuous support through my studies in America.

Finally, thank you Dr. John Griggs, Dr. Jeff Scroggs, Denise Seabrooks, Carolyn Guntun, Brenda Currin, Di Bucklad and Seyma Bennett for providing a supportive atmosphere for teaching and learning.

Contents

List of Tables

Chapter 1

Introduction

Nonnegative matrices are often used to describe the behavior of many science and mathematical models which are involved in multiplicative processes. In a multiplicative process, we begin by multiplying a nonnegative input vector $x^{(0)}$ to a square nonnegative matrix A in order to obtain an output vector $x^{(1)}$. Then we use the vector $x^{(1)}$ as a new input vector and multiply it to A to get $x^{(2)}$. By continuing this process over some amount of time, the resulting vector is in fact an eigenvector corresponding to the spectral radius of A. If A is nonnegative and irreducible then the resulting vector after the normalization process is the eigenvector that associated with the Perron root of A.

Examples of such multiplicative processes can be found in the applications of a finite state Markov chains such as the branching processes [15], Markov rewards processes with exponential utility [16], Leontiev Input/Output Economic model, and many other models. Moreover, nonnegative matrices are also applied in other areas of science such as Statistical Mechanics [11], Low-Dimensional Topology [28], Matrix Iterative Analysis [32],[5]. In the Information Theory, the authors of [31] study the relationship between the Kullbac-Leibler distance and the Perron root and use it to develop a power control algorithms to provide desired quality of service.

Since there is a wide range of applications of nonnegative irreducible matrices and the size of the transition matrices are getting larger, speed and accuracy of the computation of the Perron root is necessary. In this dissertation, we aim to construct an algorithm that is capable of finding the Perron root of a large size nonnegative irreducible matrix accurately.

Chapter 2

Background and Theory

In this chapter, we introduce the definitions that are related to a nonnegative irreducible matrix. We also discuss background and certain related theories including previous work done on computing the Perron root by other mathematicians.

2.1 Irreducible Nonnegative Matrices

We define an $m \times n$ matrix $A = (a_{ij})$ for $1 \leq i \leq m$, and $1 \leq j \leq n$. An $m \times n$ matrix A is said to be *nonnegative* if for each $a_{ij} \geq 0$. We write $A \geq 0$. Consequently, for two matrices, $A \geq B$ if $a_{ij} \geq b_{ij}$ for all i and j. A is said to be a *positive matrix* if $a_{ij} > 0$ for all i and j. We denote P to be a square matrix of order n. P is called a *permutation matrix* if P can be obtained from the identity matrix of order n or I_n by interchanging its rows or columns.

Definition 2.1 *[23] A square matrix $A_{n\times n}$ is said to be reducible if there exist a permutation matrix P such that*

$$P^T AP = \begin{pmatrix} X & Y \\ 0 & Z \end{pmatrix},$$

where X and Z are both square; otherwise, A is said to be an irreducible matrix. The quantity $P^T AP$ is called a symmetric permutation of A.

Definition 2.2 *For a square matrix A, the quantity*

$$\rho(A) = \max_{\lambda \in \sigma(A)} |\lambda|$$

is called the spectral radius of A, where $\sigma(A)$ is a set of all eigenvalues of A .

The investigation of the properties of positive matrices has been successfully carried out by Oscar Perron in 1907. For a positive matrix A, the spectral radius $\rho(A)$ is a simple eigenvalue to a positive eigenvector, and $\rho(A) > \lambda$ for all other λ' s in $\sigma(A)$. Later in 1912, Frobenius gave the extension of Perron's results to irreducible nonnegative matrices.

Theorem 2.1 (Perron-Frobenius Theorem) *If A is a nonnegative irreducible square matrix of order n, then each of the following statements is true.*

- *$\rho(A) \in \sigma(A)$ and $\rho(A) > 0$.*
- *$\rho(A)$ is a simple eigenvalue.*
- *There exists an eigenvector $x > 0$ such that $Ax = \rho(A)x$.*
- *The unique vector defined by $Ap = \rho(A)p$, $p > 0$, and $||p||_1 = 1$, is called the Perron vector. There are no nonnegative eigenvectors for A except for the positive multiples of p, regardless of the eigenvalue.*

The proof of the *Perron-Frobenius* theorem can be found in [23].

2.2 Diagonal Transformation

A diagonal transformation is a method uses the fact that a nonnegative irreducible matrix has a real simple eigenvalue equal to its spectral radius. Let A be a nonnegative irreducible matrix, $\rho(A)$ be the spectral radius of A and p be an eigenvector associated with $\rho(A)$. Suppose D is a diagonal matrix whose diagonal elements are components of p, and we define $e = (1, 1, \ldots, 1)^T$. Then

$$D^{-1}ADe = D^{-1}Ap = D^{-1}\rho(A)p = \rho(A)D^{-1}p = \rho(A)e. \tag{2.1}$$

This implies $Be = \rho(A)e$ and $B = D^{-1}AD$, where its row sums equal to ρ. However, in many applications, an eigenvector p is unknown; thus, in order to use the fact in equation (2.1), we need a way to form a diagonal matrix D. The following methods show how to form D and use the diagonal transformation to compute the Perron root of A.

2.2.1 Brauer's Algorithm

Brauer [6] proved that for every positive number η, there exists a matrix $F(\eta)$ similar to A in which the difference between the maximum row sums R^* of $F(\eta)$ and the minimum row sums r^* satisfies

$$R^* - r^* < \eta.$$

The algorithm was derived from his previous algorithm used to compute the maximum eigenvalue of positive matrices. Suppose R and r are the maximum row sums and the minimum row sums of A respectively. If $R = r$ then, by the Collatz's formula [26], the spectral radius of A equals R. So, we let $R > r$. The algorithm begins by dividing the interval $[r, R]$ into four subintervals $I_1, I_2, I_3,$ and I_4 of equal width, so the width of each of the four subintervals is

$$d = \frac{R - r}{4}.$$

For each subinterval,

$$I_1 = \{r \leq x \leq r + d\},$$

$$I_2 = \{r + d < \ x \leq r + 2d\},$$

$$I_3 = \{r + 2d < x \leq r + 3d\}, \quad \text{and}$$

$$I_4 = \{r + 3d < \ x \leq r + 4d\}.$$

Then let $g = \frac{r+3d}{r+2d}$. If some of the row sums of A lie in the interval I_1 or I_2, then we multiply the elements of the corresponding rows by g and divide the elements of the corresponding columns by g. It is an analog of similarity transformation which transforms A to a similar matrix, B. Therefore, the Perron root of A and B are equal.

After applying one iteration of the transformation, the values of row sums of the matrix B that fall in the interval I_1 or I_2 are either remain unchanged from the values of the row sums of A or increased at least by $m(g - 1)$, where m is the smallest positive off-diagonal entry of A. If they are increased, they cannot grow big enough to be in the interval I_4. Meanwhile, the values of the row sums of B that fall in the interval I_3 or I_4 are either remain unchanged or decreased by at least $m(1 - g^{-1})$. If they are decreased, they cannot be small enough to be in the interval I_1. Hence, if there are no row sums of B that lie in I_1, then the difference between two row sums is less than or equal to $3d$.

Nevertheless, if there are some values of the row sums of B that lie in the interval I_1, we reapply the transformation on B using the original values R, r and g to obtain a similar matrix B_1. We continue the process until the values of the row sums of the resulting matrix are no longer in I_1. Let R' and r' be the maximum and the minimum row sums of the resulting matrix in the last iteration. The difference between the maximum row sums R'

and the minimum row sums r' is

$$R' - r' \ \leq \frac{3}{4}(R - r) = 3d.$$

After that, we divide the interval $[r', R']$ into four strips of equal width, so the width of each strip is

$$d' = \frac{R' - r'}{4}.$$

We repeat the same process to obtain a matrix that is similar to A, and the difference between its maximum row sums R'' and minimum row sums r'' is

$$R'' - r'' \ \leq \ \frac{3}{4}(4d') \ \leq (\frac{3}{4})^2 4d.$$

After k iterations, we have a matrix that is similar to A, and the difference between its maximum row sums and minimum row sums is

$$R^{(k)} - r^{(k)} \ \leq (\frac{3}{4})^k 4d.$$

Hence, for every positive number η, it is possible to choose a number k large enough so that $R^{(k)} - r^{(k)} < \eta$ [6].

This algorithm works well in theory. In practice, it converges slowly; particularly, when the difference between the maximum row sums and the minimum row sums of the resulting matrix approaches 0. In addition, there are a lot of computational work per iteration.

2.2.2 Hall and Porsching's Algorithm

In September of 1968, Hall and Porsching [13] came up with an algorithm for computing the Perron root. This algorithm is based on their previous work in which they used to compute the Perron root and the Perron vector of positive matrices. They construct a sequence of diagonal similarity transformations which transforms a given nonnegative irreducible matrix to a sequence of matrices whose row sums converge to the maximum eigenvalue [13, 14]. They also show in [13] that the rate of convergence of their algorithm does not depend on the ratio of the second largest and the largest eigenvalue.

Let A_k be the k^{th} matrix in the sequence of matrices; we denote the maximum row sums of A_k by $R^k = \max_i R_i^k$ and the minimum row sums of A_k by $r^k = \min_i R_i^k$, where R_i^k is the i^{th} row sums of A_k. Suppose $J_k = \{i \mid R_i^k = r^k\}$ be an index set and we define $b_i^k = \sum_{j \in J_k} (a_{ij})^k$ for all $1 \leq i \leq n$. Let T_k be a diagonal matrix of order n such that

$$T_k = \mathrm{diag}(d_1^k, \cdots, d_n^k),$$

where

$$d_i^k = \begin{cases} 1 & \text{if } i \notin \ J_k, \\ x_k & \text{if } i \in \ J_k. \end{cases}$$

Notice that $\{x_n\}$ is a random sequence of positive numbers. By [13], the matrix

$$A_{k+1} = T_k^{-1} A_k T_k \tag{2.2}$$

has row sums $R_i^{k+1} = R_i + b_i^k(x_k - 1)$ if $i \notin J_n$; otherwise, $R_i^{k+1} = b_i^k + (R_i^k - b_i^k)/x_k$.

Instead of using a random sequence of positive numbers x_k, the authors of [13] suggest a way to obtain x_k by using elements from a given nonnegative irreducible matrix A. Suppose ν and μ are numbers such that

$$\nu \in J_k, \ b_\nu^k = \min_{i \in J_k} b_i^k \text{ and } R_\mu^n = R^k,$$

and let $a = 4b_\mu^k,\ b = 2R^k - 6b_\mu^k - 2b_\nu^k, c = R^k + r^k - 2b_\mu^k - 2b_\nu^k$, then

$$x_k = \begin{cases} \frac{-b+(b^2-4ac)^{1/2}}{2a} & \text{if } a \neq 0 \\ b/c & \text{if } a = 0. \end{cases}$$

Using x_k to form the matrix T_k and iterating equation (2.2) until $R^{k+1} - r^{k+1} < tol$, we have $\rho(A) = \frac{R^{n+1}+r^{n+1}}{2}$. The complete proof of convergence for this algorithm is shown in [13], and $2n$ iterations of this algorithm are approximately equivalent to 3 iterations of the power method. Observe that the algorithm takes 40 iterations to calculate the Perron root of a given matrix A below with an error of 0.000009 [13].

$$A = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 2 & 1 & 0 & 0 \\ 0 & 2 & 1 & 0 \\ 0 & 0 & 2 & 1 \end{pmatrix}. \tag{2.3}$$

2.2.3 Markham's Algorithm

In October of 1968, Markham developed a practical method in [21] for computing the Perron root of a positive matrix by transforming a positive matrix $A_{n\times n}$ to a positive generalized stochastic matrix $S_{n\times n}$. A matrix $S_{n\times n}$ is said to be a positive generalized stochastic matrix if S is a positive matrix in which each row sum of S is equal to its spectral radius, $\rho(S)$.

Suppose A is a positive matrix of order n. Let R_i denotes the i^{th} row sum of A, $R = max_i\{R_i\}$ and $r = min_i\{R_i\}$. Suppose $Q = \text{diag}(R_1, R_2, \ldots, R_n)$ then, by a similarity transformation, we obtain a new positive matrix $B_1 = Q^{-1}AQ$, where each row sum of B_1 lies in the interval (r, R) [21, theorem 1]. Apply the similarity transformation to B_1 to obtain B_2. The difference between the maximum and the minimum row sums of B_2 is smaller than one in B_1. By continuing the iterative process, we obtain a finite sequence of matrices, B_1,

B_2, $\cdots$, B_n, $\dots$, in which the difference between the maximum and minimum row sums of each B_i is decreasing.

In addition, the sequence of matrices B_i converges to a positive generalized stochastic matrix S [21, theorem 2]. Since S is a positive generalized stochastic matrix that is similar to a positive matrix A, by the Frobenius theorem in section (3.1), we obtain the Perron root of S, which is also the Perron root of A. Nevertheless, this algorithm has two disadvantages. It only works for positive matrices, and it requires a large amount of computations. The author of [7] shows that this algorithm is equivalent to the power method. Therefore, if a given positive matrix A has a tight gap between the first and the second largest eigenvalues, Markham's algorithm converges very slowly.

2.2.4 Pham's Algorithm

Pham[2] proved that any positive matrix is similar to a quasi stochastic matrix; the proof can be found in [2]. Given below is a definition of a quasi stochastic matrix.

Definition 2.3 *[2] A matrix A is said to be a quasi stochastic matrix if A is nonnegative and $s_1 = s_2 = \ldots = s_n = \mu$, where*

$$s_i = \sum_{j=1}^{n} a_{ij}, \quad 1 \leq i \leq n$$

and μ is a characteristic number of the matrix A.

In fact, the characteristic number μ is the spectral radius of A [2]. Moreover, a vector $e = [1, 1, \ldots, 1]^T$ is an eigenvector corresponding to μ. In 1975, Pham proved that a nonnegative irreducible matrix can be transformed to a quasi stochastic matrix via similarity variation algorithm [3]. By applying the algorithm, we can find the maximum eigenvalue and the corresponding eigenvector of a nonnegative irreducible matrix.

Below is Pham's similarity variation algorithm for computing the Perron root μ and the Perron vector p of a nonnegative irreducible matrix A.

1. Let $A = A_0$ be a nonnegative matrix, from A_0 construct a sequence of matrices $\{A_k\}_{k=0}^{\infty}$ as follows,
2. $A_{k+1} = S_k^{-1} A_k S_k$ where A_k is a nonnegative matrix,
3. $S_k = diag(s_1^{(k)}, \ldots, s_n^{(k)})$, and $s_i = \sum_{j=1}^{n} a_{ij}, \quad i = 1, 2, \ldots, n$,
4. form $Q_{k+1} = T_0 T_1 \ldots T_k$ where $T_k = S_k / s_r^{(k)}$, and r is an arbitrary integer from 1 to n,
5. set $m_k = \min\{s_1^{(k)}, \ldots, s_n^{(k)}\}$ and $M_k = \max\{s_1^{(k)}, \ldots, s_n^{(k)}\}$.

Via the above algorithm, Pham proved in [3] that the sequences $\{A_k\}_{k=0}^{\infty}$, $\{m_k\}_{k=0}^{\infty}$, $\{M_k\}_{k=0}^{\infty}$ and $\{Q_k\}_{k=0}^{\infty}$ are convergent, and they hold the following properties:

Property 1. $\lim_{k\to\infty} A_k = \bar{A}$ is a quasi stochastic matrix.

Property 2. $\lim_{k\to\infty} M_k = \lim_{k\to\infty} m_k = \mu > 0$.

Property 3. $\lim_{k\to\infty} Q_k = Q = diag(q_1, \ldots, q_n) \quad \forall q_i > 0$.

Property 4. $\bar{A} = Q^{-1} A Q$.

Property 5. The quantity μ is the maximum eigenvalue, and a vector $q = (q_1, \ldots, q_n)^T$ is the corresponding eigenvector, of the matrix A.

Many numerical results from the algorithm can be found in [3]. Both Pham's algorithm and Markham's algorithm are equivalent [7]. In fact, they are equivalent to the power method of Von Mises [33]; it is known that the power method converges for primitive matrices with an arbitrary starting vector $x \neq 0$ [7, 33]. Since positive matrices and strictly irreducible matrices (i.e., all of diagonal entries are positive) are primitive, the algorithm converges. Nevertheless, they converge slowly if the magnitude of the subdominant eigenvalue is close to the Perron root of A.

2.2.5 Duan and Zhang's Algorithm

In 2006, Duan and Zhang [9] produced the algorithm that used the idea of a similarity transformation. Suppose $D_0 = diag(a_1^0, a_2^0, \ldots, a_n^0)$, where each $a_i^0 = \sum_{j=1}^n a_{ij}^0$ is the row sums of A. Let $A_{k+1} = D_k^{-1} A_k D_k$. Since A can be written as $A = \lambda I + B$, the Perron root of B is $\rho(B) = \rho(A) - \lambda$, where I is the identity matrix of order n, B is a nonnegative irreducible matrix and λ is any positive number [9].

The following is the steps of the algorithm:

Step 0: Given an $n \times n$ nonnegative irreducible matrix $A = a_{ij}$. Let $\epsilon > 0$ and let $k = 0$.

Step 1: Let $B = B_0 = I + A = b_{ij}^0$.

Step 2: Compute

$$b_i^k = \sum_{j=1}^n b_{ij}^k,\ r^k = \min_{1 \le i \le n} b_i^k,\ R^k = \max_{1 \le i \le n} b_i^k.$$

If $(R^k - r^k) < \epsilon$, go to step 5.

Step 3: Compute D_k =diag$(b_1^k, b_2^k, \cdots, b_n^k)$.

Step 4: Update. Let $B_{k+1} = D_k^{-1} B_k D_k$.

Set $k = k + 1$. Go back to step 2.

Step 5: Let $\rho = 1/2(r^k + R^k) - 1$. Stop.

The algorithm is feasible for any nonnegative irreducible matrix. Observe that the algorithm are similar to Markham's algorithm and Pham's algorithm. Thus, the algorithm always converges because the first step guarantees the primitivity. The slow rate of convergence occurs if the magnitude of the subdominant eigenvalue and the Perron root of A are close. Overall, this algorithm is the most powerful one that uses the idea of a diagonal similarity transformation.

In conclusion, all of the diagonal transformation methods mentioned in this section have one disadvantage in common. They consume time and use large amount of computation and organization. Next, we study a new approach for computing the Perron root. The approach is based on the Perron complementation technique.

2.3 Perron Complementation and Generalized Perron Complementation

In this section, we introduce the concept of Perron complementation. Carl Meyer first introduced the concept of Perron complementation [22, 1989] which concerns the computation of the unique normalized Perron vector π of a large scale problem. The idea is to partition a nonnegative irreducible matrix A into two or more smaller matrices – say $P_1, P_2, \ldots, P_k$ of order $r_1, r_2, \ldots, r_k$, where $\sum_{i=1}^{k} r_i = n$. In order to obtain the Perron vector π of A, the Perron vector $\pi^{(i)}$ of each P_i must be computed separately and $\pi = (\pi^{(1)}, \pi^{(2)}, \ldots, \pi^{(k)})$.

Let $A_{n\times n}$ be a nonnegative irreducible matrix with the spectral radius of A, $\rho(A)$, and its associated positive eigenvector π. Suppose α and β are disjoint nonempty ordered subsets of $\langle n\rangle = \{1, 2, \cdots, n\}$ such that $\alpha \cup \beta = \langle n\rangle$. Assuming, the elements of each ordered set are arranged in an increasing order. Let $|\alpha|$ denotes the cardinality of α, and $A[\alpha, \beta]$ denotes submatrix of A whose rows and columns are determined by α and β respectively. We denote $A[\alpha, \alpha]$, the principal submatrix of A based on α, as $A[\alpha]$. Using above notations, the Perron complement can now be defined as follows:

Definition 2.4 *Let A be an $n \times n$ nonnegative irreducible matrix with spectral radius ρ. For a certain α, the Perron complement of $A[\beta]$ in A is defined to be the matrix*

$$P(A/A[\beta]) = A[\alpha] + A[\alpha, \beta](\rho I - A[\beta])^{-1}A[\beta, \alpha]. \tag{2.4}$$

In addition, the following lemma related to the spectral radius of each Peron complementation. The complete proof can be found in [22].

Lemma 2.1 *If A is a nonnegative irreducible matrix with the spectral radius $\rho(A)$, then each Perron complement $P(A/A[\beta])$ is also a nonnegative irreducible matrix with the same spectral radius $\rho(A)$.*

In general, we do not know the value of $\rho(A)$ by looking at the matrix A. Lu, the author of [18] investigated the property of the Perron complement and combined the result from lemma 2.1 with the idea of the *generalized Perron complement* to approximate $\rho(A)$. The concept of the generalized Perron complement was introduced by Neumann [24].

Definition 2.5 *Let A be an $n \times n$ nonnegative irreducible matrix, the generalized Perron complement of $A[\beta]$ in A is defined to be the matrix*

$$P_t(A/A[\beta]) = A[\alpha] + A[\alpha, \beta](tI - A[\beta])^{-1}A[\beta, \alpha], \; t > \rho(A[\beta]). \tag{2.5}$$

One of the most important properties of the generalized Perron complement is that, for any $t > \rho(A[\beta])$, the quantity $P_t(A/A[\beta])$ is a nonnegative irreducible matrix [24]. The following lemma is in [18].

Lemma 2.2 *If A is a nonnegative irreducible matrix, then the Perron root $\rho(P_t(A/A[\beta]))$ of the generalized Perron complement is a strictly decreasing function of t on $(\rho(A[\beta]), \infty)$.*

2.3.1 Lu's Algorithm

In 2002, Lu [18] combined the idea of generalized Perron complementation and Newton iteration. The resulting algorithm is an alternative way of producing the Perron root of a nonnegative irreducible matrix A.

Lemma 2.3 *[Theorem 4] If A is a nonnegative irreducible matrix, then*

$$\rho(P_t(A/A[\beta])) \begin{cases} < \rho(A) & \text{if } t > \rho(A) \\ = \rho(A) & \text{if } t = \rho(A) \\ > \rho(A) & \text{if } \rho(A[\beta]) < t < \rho(A). \end{cases} \tag{2.6}$$

The generalized Perron complementation is applied to a given matrix A to obtain a good estimation of the upper bound and the lower bound of the Perron root of A. In order for $P_t(A/A[\beta])$ to be well defined, the value of t must be greater than $\rho(A[\beta])$ [24]; as a result, t is set to be the maximum row sums of $A[\beta]$. Using lemma 2.3, a good approximation of the Perron root can be determined, and the exact value of the Perron root can be obtained if the right t is chosen. In practice, it is difficult to determine the value of t that equals the Perron root of A. In stead of directly solving for the Perron root of A, Lu took the problem and considered it in another direction.

Let

$$f(t) \;=\; \rho(P_t(A/A[\beta])) \;=\; a_{ss} + A[s,\beta](tI - A[\beta])^{-1}A[\beta,s], \tag{2.7}$$

where $\beta = \langle n\rangle \setminus \{s\}$ in which $\{s\} = \{j \mid r_j = \text{minimum row sum of } A\}$. By lemma 2.3, $f(t)$ is a strictly decreasing continuous function of t on the interval $(\rho(A[\beta]), \infty)$. A new function $g(t) = f(t) - t$ is formed, and it has a unique root on the interval (b, c) if $\rho(A[\beta]) < b < \rho(A) < c$ [18, theorem 6]. From this set up, the root of $g(t)$ is equivalent to the Perron root of A. The following is Lu's algorithm.

Step 1: Calculate the row sums $r_j(A)$ of A and set $c = r_{max}(A)$.

Step 2: Determine β and α according to r_j and compute $P_c(A/A[\beta])$, then set $b = r_{min}P_c(A/A[\beta])$ if it is bigger than $r_{min}(A)$.

Step 3: Determine the $g(t)$.

Step 4: If $g((b+c)/2) > 0$ go to next step. Else start to choose the lower bound b such that $g(b) > 0$.

Step 5: Use the bisection method to reduce the length of (b, c).

Step 6: Apply Newton iteration to $g(t)$ on (b, c) to compute $\rho(A)$.

Although the algorithm produces the Perron root, two practical problems occur. It is hard to choose β and t properly. Secondly, the value of $P_c(A/A[\beta])$ must be computed for each iteration of the generalized Perron complement. Furthermore, Newton iterations converge quadratically, but $g(t)$ is a function of matrices and $g'(t) = 1 + A[s,\beta](tI - A[\beta])^{-2}A[\beta, s]$ must be computed in every Newton iteration. Therefore, it is expensive to compute the factor $(tI - A[\beta])^{-2}$. Despite the fact that a way to compute t_{k+1} is suggested in [18], the algorithm is certainly too expensive to compute the Perron root of A.

In addition, Shimming Yang and Ting-Zhu Huang computed the bounds of Perron root of a nonnegative irreducible matrix A using the idea of the Perron complementation.

2.3.2 Yang and Huang's Algorithm

Suppose A be a nonnegative irreducible matrix of order n, and $\rho(A)$ is a spectral radius of A. Let $R(A)$ denotes a maximum row sums of A, and $r(A)$ denotes a minimum row sums of A. Using the same notations as previously used in section 2.3.1, the following lemmas are mentioned and proved in [34] :

Lemma 2.4 *If k^{th} row has the minimum row sum in A and let $\beta = \{k\}$, then the maximum row sum of $A[\beta]$'s Perron complement is less than or equal to the maximum row sum of A. That is the inequality*

$$\rho(A) = \rho(P(A/A[\beta])) \leq R(P(A/A[\beta])) \leq R(A) \tag{2.8}$$

holds.

As a result, a new upper bound of A can be obtained from the maximum row sums of $P(A/A[\beta])$. The algorithm for finding a new upper bound of the Perron root of A is as follows:

Step 1: Calculate all the row sums $r_i(A)$ and set $r(A) = min_i(r_i(A))$. Let $\gamma = \{l \mid r_l = r(A)\},\ l \in< n >$; set $d = \infty$.

Step 2: Get one of the element from γ and assign its value to k. Then delete this element from γ. Update $\beta = \{k\}, \alpha = \langle n \rangle \backslash \beta$.

Step 3: Let $b_i = a_{kk} + r_i - a_{ik}$, $c_i = r_i a_{kk} - a_{ik} r(A)$. Set

$$d = \min\{\max_{i \in \beta} \frac{b_i + \sqrt{b_i^2 - 4c_i}}{2}, d\}.$$

Step 4: If γ is not empty, go to step 2; otherwise, the new upper bound is d.

Lemma 2.5 *If K^{th} row has the maximum row sum in A, let $\beta = \{K\}$, then the minimum row sum of $A[\beta]$' Perron complement is greater than or equal to the minimum row sum of A. That is the inequality*

$$r(A) = r(P(A/A[\beta])) \leq \rho(P(A/A[\beta])) \leq \rho(A) \tag{2.9}$$

holds.

The following is an algorithm that produces a new lower bound of the Perron root of A.

Step 1: Calculate all the row sums $r_i(A)$ and set $R(A) = max_i(r_i(A))$. Let $\gamma = \{l | r_l = R(A)\},\ l \in< n >$; set $d = 0$;

Step 2: Get one of the element from γ and assign its value to K. Then delete this element from γ. Update $\beta = \{K\}, \alpha =< n > \backslash \beta$;

Step 3: Let $b_i = a_{KK} + r_i - a_{iK}$, $c_i = r_i a_{KK} - a_{iK} R(A)$. Set

$$d = \max\{\min_{i \in \beta} \frac{b_i + \sqrt{b_i^2 - 4c_i}}{2}, d\};$$

Step 4: If γ is not empty, go to step 2; otherwise, the new lower bound is d.

Several examples that demonstrate how both algorithms work can be found in [34].

2.4 Iterative Method

During 1960's, while the idea of using the diagonal similarity transformation to find the Perron root of a nonnegative irreducible matrix A was well known, another idea emerged. Power method and inverse iteration are some of the simplest iterative methods that are used to find the largest eigenvalue of any matrix. Since the Perron root is the largest eigenvalue of a nonnegative irreducible matrix, the main ingredient for some of the iterative algorithms that we consider in this dissertation is a power method or an inverse iteration algorithm. The first iterative method was introduced by a Japanese scientist named Takashi Noda. His algorithm is based on an inverse iteration algorithm.

2.4.1 Noda's Algorithm

In 1971, Noda [25] introduced the algorithm that implemented the ideas of the Wielandt's method [27] and an inverse iteration method. Even though the Wielandt's method is applied to general matrices, the complete proof of the convergence of the method applied to a nonnegative irreducible matrix A can be found in [25]. Suppose λ is an approximation of the Perron root of A and assume that $\lambda > \rho(A)$. When applying an inverse iteration to the iterate matrix $(\lambda I - A)^{-1}$, the Perron root estimator λ converges to $\rho(A)$. Moreover, for an arbitrary positive vector x, the normalized vector x converges to a unique normalized eigenvector associated with $\rho(A)$.

The following is Noda's algorithm that used for computing the Perron root of a nonnegative irreducible matrix A.

Let λ^0 be a positive number that is greater than $\rho(A)$, and x^0 and v be positive vectors. Set $tol = 1e^{-16}$

For $n = 0, 1, 2 \ldots$

$$\begin{aligned}
x^{*(n+1)} &= (\bar{\lambda}^{(n)} I - A)^{-1} x^{(n)}, \\
\tau^{(n+1)} &= < x^{*(n+1)}, v > / < x^{(n)}, v >, \\
x^{(n+1)} &= x^{*(n+1)} / \tau^{(n+1)}, \\
\bar{\tau}^{(n+1)} &= \max_k ((x^{*(n+1)})_k / (x^{(n)})_k), \\
\bar{\lambda}^{(n+1)} &= \bar{\lambda}^{(n)} - (1/\bar{\tau}^{(n+1)}), \\
\lambda^{(n+1)} &= \bar{\lambda}^{(n)} - (1/\tau^{(n+1)}), \\
\underline{\tau}^{(n+1)} &= \min_k ((x^{*(n+1)})_k / (x^{(n)})_k), \\
\underline{\lambda}^{(n+1)} &= \bar{\lambda}^{(n)} - (1/\underline{\tau}^{(n+1)})
\end{aligned}$$

until $\bar{\lambda}^{(n)} - \underline{\lambda}^{(n)} < tol$, then the Perron root $\rho(A)$ is $\lambda^{(n)}$.

Since the Perron root estimator λ is greater than the Perron root of A, the iterate matrix $(\lambda I - A)^{-1}$ is always positive by theorem 3.9 in [32]. Apply the iterate matrix $(\lambda I - A)^{-1}$ to an arbitrary positive vector x, the vector $x^{*(n+1)} = (\lambda I - A)^{-1} x^{(n)}$ is always positive. Observe that for each iteration,

$$\underline{\lambda}^{(n+1)} \leq \lambda^{(n+1)} \leq \bar{\lambda}^{(n+1)}.$$

and

$$\lim_{n\to\infty} \bar{\lambda}^n = \lim_{n\to\infty} \underline{\lambda}^n = \lim_{n\to\infty} \lambda^n = \rho(A) \text{ and } \bar{\lambda}^0 > \bar{\lambda}^1 > \cdots > \bar{\lambda}^n > \bar{\lambda}^{n+1} > \cdots > \rho(A).$$

The numerical results of the this algorithm when applied to nonnegative irreducible matrices are presented in section 3.5. Up to this point, this algorithm use the least amount of time to compute the Perron root, and Elsner proved in 1976 that this algorithm converges to the Perron root quadratically [10].

2.4.2 Elsner's Algorithm

Elsner [10] used the idea of inverse iteration to calculate the Perron root of a nonnegative irreducible matrix A. The algorithm converges at the rate of at least quadratic. The proof of convergence using Hopf's inequality can be found in [10]. Suppose B is a nonnegative matrix of order n, and x, y be a pair of vectors such that $y > 0$. We define

$$\max(\frac{x}{y}) = \max_i(\frac{x_i}{y_i}),$$

$$\min(\frac{x}{y}) = \min_i(\frac{x_i}{y_i}),$$

and

$$osc(\frac{x}{y}) = \max(\frac{x}{y}) - \min(\frac{x}{y}).$$

Theorem 2.2 *[The Hopf's inequality] Let $B > 0$ be a positive matrix of order n. Then for any vector x and any positive vector y,*

$$osc(\frac{Bx}{By}) \leq \frac{\sqrt{K}-1}{\sqrt{K}+1} osc(\frac{x}{y})$$

where

$$K = \sup_{\substack{u \geq 0 \\ v \geq 0}} \left\{ \max(\frac{Au}{Av}) \max(\frac{Av}{Au}) \right\} \leq \frac{m^2}{M^2}$$

and

$$M = \max_{i,j} b_{ij}, \quad m = \min_{ij} b_{ij}.$$

The proof of this theorem 2.2 can be found in [20].

The following is Elsner's algorithm for computing the Perron root of a nonnegative irreducible matrix A.

Step 1: For a pair of vectors x, y with $y > 0$ defines

$$||x|| = \max(\frac{x}{y}).$$

Step 2: Let $\{B_n\}$ be a sequence of positive matrices commuting with A where $n = 0, 1, 2, \ldots$.

Step 3: Assume that there exists a number γ such that

$$N(B_n) \leq \gamma < 1, \text{ with } n = 0, 1, 2, \ldots$$

where $N(B_n) = \frac{\sqrt{K(B_n)}-1}{\sqrt{K(B_n)}+1}$.

Step 4: For given $x_0 > 0$, define iteratively

$$\tilde{x}_{n+1} = B_n x_n,$$

$$x_{n+1} = \frac{\tilde{x}_{n+1}}{||\tilde{x}_{n+1}||},$$

$$\bar{\lambda}_{n+1} = \max(\frac{Ax_{n+1}}{x_{n+1}}), \quad \underline{\lambda}_{n+1} = \min(\frac{Ax_{n+1}}{x_{n+1}})$$

where $\bar{\lambda}_0$ and $\underline{\lambda}_0$ are defined analogously.

From the above iterative procedure, the sequences of $\bar{\lambda}_n$ and $\underline{\lambda}_n$ converge to the Perron root $\rho(A)$ [10], where

$$\underline{\lambda}_n \leq \underline{\lambda}_{n+1} \leq \rho(A) \leq \bar{\lambda}_{n+1} \leq \bar{\lambda}_n. \tag{2.10}$$

2.4.3 Modified Elsner's Algorithm

Concerning the stability of Elsner's algorithm, the modified Elsner's algorithm [17] was formulated by L. Elsner, I. Koltracht, M. Neumann, and D. Xiao. The new algorithm is based on a special inverse iteration that was first proposed by Noda with a new stopping criteria. In fact, the idea of Elsner's procedure, which shown to converge to the Perron root quadratically, is implemented. The following is a stable algorithm for computing the Perron root of a nonnegative irreducible matrix:

1. Let u be the machine precision. Let y_0 be a positive vector in R^n and $\mu_0 = \max(Ay_0/y_0)$. For $s = 0, 1, \ldots$.

2. Compute the LU factorization

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

and solve for x_s in

$$L_s U_s x_s = y_s$$

by the Ahac, Bouni, and Olesky algorithm [1]; save the LU factors.

3. Compute $r = Ax_s - y_s$.

4. Solve $Ad = r$ using L_s and U_s.

5. Update $\bar{x}_s = x_s - d$; and compute

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

6. Proceed until

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

Note that the LU factorization used in step 2 can always find pivot elements when solving for x_s in the linear system

$$(\mu_s I - A)x_s = y_s.$$

However, the floating-point operation counts of this LU factorization are approximately $n^2/2$ more than operation counts of Gaussian elimination with either partial or no pivoting [1]. An iterative refinement is added due to the suggestion of Skeel [30] in step 3 to obtain a more accurate Perron root.

Chapter 3

Our Contribution

In this chapter, we present the main concept of our algorithm. The basis of this algorithm is combining the Collatz's formula and an inverse iteration algorithm. We shall discuss the formulation of this algorithm and introduce test matrices which are used to test for accuracy and convergence of our algorithm. The results for computing the Perron root using test matrices will be given later in this chapter. The proof of convergence will be given to establish the accuracy of the algorithm. After that, we will compare our results from this algorithm to some well known algorithms mentioned in Chapter 2.

3.1 Collatz's Formula

Lother Collatz, a German mathematician, discovered the formula for the Perron root of positive matrices in 1942. This formula was refined by Wielandt in 1950 to develop the Perron-Frobenius theory. The following is the ***Collatz-Wielandt formula*** for the Perron root of a positive matrix.

The Perron root of $A_{n\times n} > 0$ is given by

$$\rho(A) = \max_{x \in N} f(x),$$

where $f(x) = \min_{1\leq i\leq n} \frac{(Ax)_i}{x_i}$ and $N = \{x | x \geq 0 \text{ with } x \neq 0\}$ [23].

Throughout this chapter, we define $A = (a_{ij})$ to be an $n \times n$ nonnegative irreducible matrix, and let $x = (x_1, x_2, \ldots, x_n)^T$ be any positive vector. In addition, we define a function $f_i(x) = (Ax)_i/x_i$, where $1 \leq i \leq n$. Let $m(x) = \min_i f_i(x)$, and $M(x) = \max_i f_i(x)$. The following is the Collatz and Wielandt's theorem.

Theorem 3.1 *The spectral radius $\rho(A)$ of a nonnegative irreducible matrix satisfies either*

$$m(x) < \rho(A) < M(x) \tag{3.1}$$

or

$$m(x) = \rho(A) = M(x) \tag{3.2}$$

for any $x > 0$. If an equation (3.2) holds, then x is a positive eigenvector of A corresponding to $\rho(A)$.

Theorem 3.1 can be used to estimate the upper and lower bounds of the spectral radius $\rho(A)$ — see [6],[10],[13],[17], and [25]. Because of our objective is to formulate a fast algorithm for computing the Perron root, we employ the Collatz-Wielandt's theorem as our primary tool in our algorithm.

3.2 Stopping Criteria

Since our algorithm is involved finding a solution of a linear system whose coefficient matrix is a nonsingular $M-$matrix, the algorithm must be subjected to a certain stopping criteria. By the result of the Perron-Frobenius theorem, we know that the Perron root of a nonnegative irreducible matrix is a simple eigenvalue. Due to the perturbation results (for a sufficiently small δ, $A+\delta E$, $\|E\|_2 = 1$), the sensitivity of a simple eigenvalue depends on the angle between normalized left and right eigenvectors corresponding to the eigenvalue [17], [33], [8, Theorem 4.4 p.149].

However, the authors of [17] give the new result on a componentwise condition number of a simple eigenvalue of a nonnegative irreducible matrix A. It does not depend on the angle between normalized left and right eigenvectors.

Theorem 3.2 *For an $n \times n$ nonnegative and irreducible matrix A, and E is an $n \times n$ real matrix such that*

$$|E| \leq \epsilon A,$$

where $\epsilon \leq 1$. Let $\rho(A)$ be the Perron root of A and λ be the Perron root of $A + E$. Then

$$\frac{|\lambda - \rho(A)|}{\rho(A)} \leq \epsilon.$$

Therefore, we apply this result to our algorithm and use it as a stopping criteria.

3.3 Our Algorithm

Let A be a nonnegative irreducible matrix, p be the Perron vector of A, and $\rho(A)$ be the Perron root of A. Given below is our algorithm for computing the Perron root of a nonnegative irreducible matrix.

Algorithm. Let *tol* be the machine precision. Let $x^{(0)}$ be a positive vector, and set $\lambda^{(0)} = \max_i(\sum_{j=1}^{n} a_{ij})$, and $B^{(0)} = (\lambda^{(0)} I - A)^{-1}$.

For $i = 0, 1, 2, \ldots$.

1. Compute the LU factorization of

$$(\lambda^{(i)} I - A) = L^{(i)} U^{(i)},$$

and solve for $\tilde{x}^{(i)}$

$$L^{(i)} U^{(i)} \tilde{x}^{(i)} = x^{(i)}$$

2. Use the same LU factors solve for $x^{(i+1)}$

$$L^{(i)} U^{(i)} x^{(i+1)} = \tilde{x}^{(i)}.$$

3. Compute

$$\bar{\lambda}^{(i)} = \lambda^{(i)} - \min_j \left(\frac{\tilde{x}_j^{(i)}}{(B^{(i)} \tilde{x}^{(i)})_j} \right)$$

and

$$\underline{\lambda}^{(i)} = \lambda^{(i)} - \max_j \left(\frac{\tilde{x}_j^{(i)}}{(B^{(i)} \tilde{x}^{(i)})_j} \right)$$

where $1 \leq j \leq n$; Note that the quantity $(B^{(i)} \tilde{x}^{(i)})_j$ is $x_j^{(i+1)}$.

4. Set

$$\lambda^{(i+1)} = \bar{\lambda}^{(i)}.$$

5. Compute

$$error^{(i)} = (\bar{\lambda}^{(i)} - \underline{\lambda}^{(i)}) / \bar{\lambda}^{(i)}.$$

6. If $error > tol$ go back to step 1; otherwise, the Perron root of A is $\lambda^{(i+1)}$.

The following theorem characterizes the behavior of the algorithm including its convergence.

Theorem 3.3 *Let A be a nonnegative irreducible matrix and let $x^{(0)}$ be a positive vector such that*

$$x^{(0)} = \begin{pmatrix} x_1^{(0)} \\ x_2^{(0)} \\ \vdots \\ x_j^{(0)} \\ \vdots \\ x_n^{(0)} \end{pmatrix}.$$

Let $\lambda^{(0)} = max_i(\sum_{j=1}^{n}(a_{ij}))$ and let $B^{(0)} = (\lambda^{(0)}I - A)^{-1}$. Set

$$\bar{\lambda}^{(0)} = \lambda^{(0)} - \min_j \left(\frac{\tilde{x}_j^{(0)}}{\left(B^{(0)}\tilde{x}^{(0)}\right)_j} \right)$$

and

$$\underline{\lambda}^{(0)} = \lambda^{(0)} - \max_j \left(\frac{\tilde{x}_j^{(0)}}{\left(B^{(0)}\tilde{x}^{(0)}\right)_j} \right).$$

Let $\tilde{x}^{(0)} = B^{(0)}x^{(0)}$, and $x^{(1)} = B^{(0)}\tilde{x}^{(0)}$ set $\lambda^{(1)} = \bar{\lambda}^{(0)}$ and $B^{(1)} = (\lambda^{(1)}I - A)^{-1}$. Set

$$\bar{\lambda}^{(1)} = \lambda^{(1)} - \min_j \left(\frac{\tilde{x}_j^{(1)}}{\left(B^{(1)}\tilde{x}^{(1)}\right)_j} \right)$$

and

$$\underline{\lambda}^{(1)} = \lambda^{(1)} - \max_j \left(\frac{\tilde{x}_j^{(1)}}{\left(B^{(1)}\tilde{x}^{(1)}\right)_j} \right).$$

For $i = 2, 3, \ldots$. Let $\tilde{x}^{(i-1)} = B^{(i-1)}x^{(i-1)}$, and $x^{(i)} = B^{(i-1)}\tilde{x}^{(i-1)}$, Let $\lambda^{(i)} = \bar{\lambda}^{(i-1)}$ and

$B^{(i)} = (\lambda^{(i)} I - A)^{-1}$. *Set*

$$\bar{\lambda}^{(i)} = \lambda^{(i)} - \min_j \left(\frac{\tilde{x}_j^{(i)}}{(B^{(i)}\tilde{x}^{(i)})_j} \right)$$

and

$$\underline{\lambda}^{(i)} = \lambda^{(i)} - \max_j \left(\frac{\tilde{x}_j^{(i)}}{(B^{(i)}\tilde{x}^{(i)})_j} \right),$$

then

1. *For any fixed* i, $\underline{\lambda}^{(i)} \leq \rho(A) \leq \bar{\lambda}^{(i)}$.

2. $\bar{\lambda}^{(0)} > \bar{\lambda}^{(1)} > \ldots > \bar{\lambda}^{(i)} > \ldots \geq \rho(A)$.

3. $\lim_{i\to\infty} \bar{\lambda}^{(i)} = \rho(A)$.

4. $\lim_{i\to\infty} \underline{\lambda}^{(i)} = \rho(A)$.

We will examine the proof of each part separately.

Part 1

Proof

Since we define $B^{(i)} = (\lambda^{(i)} I - A)^{-1}$, we are able to obtain that

$$\rho(B^{(i)}) = \frac{1}{\lambda^{(i)} - \rho(A)}.$$

Now consider the Collatz's formula in equation (3.1), for any i = 0,1,2 ...,

$$\min_j\left(\frac{(B^{(i)}\tilde{x}^{(i)})_j}{\tilde{x}_j^{(i)}}\right) \;\leq\; \rho(B^{(i)}) \;\leq \max_j\left(\frac{(B^{(i)}\tilde{x}^{(i)})_j}{\tilde{x}_j^{(i)}}\right). \tag{3.3}$$

By lemma 3.11 part (vii) in [29], the equation (3.3) becomes

$$\max_j\left(\frac{\tilde{x}_j^{(i)}}{(B^{(i)}\tilde{x}^{(i)})_j}\right) \;\geq\; 1/\rho(B^{(i)}) \;\geq \min_j\left(\frac{\tilde{x}_j^{(i)}}{(B^{(i)}\tilde{x}^{(i)})_j}\right)$$

$$= \max_j\left(\frac{\tilde{x}_j^{(i)}}{(B^{(i)}\tilde{x}^{(i)})_j}\right) \ \geq \lambda^{(i)} - \rho(A) \ \geq \min_j\left(\frac{\tilde{x}_j^{(i)}}{(B^{(i)}\tilde{x}^{(i)})_j}\right)$$

$$\Rightarrow \ -\max_j\left(\frac{\tilde{x}_j^{(i)}}{(B^{(i)}\tilde{x}^{(i)})_j}\right) \ \leq \ \rho(A) - \lambda^{(i)} \ \leq -\min_j\left(\frac{\tilde{x}_j^{(i)}}{(B^{(i)}\tilde{x}^{(i)})_j}\right)$$

$$\Rightarrow \ \lambda^{(i)} - \max_j\left(\frac{\tilde{x}_j^{(i)}}{(B^{(i)}\tilde{x}^{(i)})_j}\right) \ \leq \ \rho(A) \ \leq \lambda^{(i)} - \min_j\left(\frac{\tilde{x}_j^{(i)}}{(B^{(i)}\tilde{x}^{(i)})_j}\right).$$

Hence, for each i, $\underline{\lambda}^{(i)} \leq \rho(A) \leq \bar{\lambda}^{(i)}$.

□

Part 2

Proof

By theorem 3.9 in [32], we obtain that for any fixed $i \geq 0$, if $\lambda^{(i)} > \rho(A)$, then $(\lambda^{(i)}I - A)^{-1} > 0$. With the result from part 1 and the facts that $\lambda^{(0)} = \max_i(\sum_{j=1}^{n} a_{ij})$ and $\tilde{x}^{(0)} > 0$, we then have

$$\bar{\lambda}^{(0)} = \lambda^{(0)} - \min_j\left(\frac{\tilde{x}_j^{(0)}}{(B^{(0)}\tilde{x}^{(0)})_j}\right) \geq \rho(A) > 0.$$

Hence, $\lambda^{(0)} > \bar{\lambda}^{(0)} \geq \rho(A) > 0$. Moreover, $B^{(0)} = (\lambda^{(0)}I - A)^{-1} > 0$ and $\tilde{x}^{(0)} > 0$; thereby, $\tilde{x}^{(1)} = B^{(0)}B^{(0)}\tilde{x}^{(0)} > 0$, which guarantees that the quantity $\min_j(\frac{\tilde{x}_j^{(1)}}{(B^{(1)}\tilde{x}^{(1)})_j}) > 0$.

Next consider $\bar{\lambda}^{(1)}$, because

$$\bar{\lambda}^{(1)} = \lambda^{(1)} - \min_j\left(\frac{\tilde{x}_j^{(1)}}{(B^{(1)}\tilde{x}^{(1)})_j}\right) = \bar{\lambda}^{(0)} - \min_j\left(\frac{\tilde{x}_j^{(1)}}{(B^{(1)}\tilde{x}^{(1)})_j}\right) \geq \rho(A) > 0,$$

it suffices to conclude that $\bar{\lambda}^{(0)} > \bar{\lambda}^{(1)} \geq \rho(A)$.

Recall that for any $i = 2, 3, 4, \ldots$, $\bar{\lambda}^{(i)} = \bar{\lambda}^{(i-1)} - \min_j(\frac{\tilde{x}_j^{(i)}}{(B^{(i)}\tilde{x}^{(i)})_j})$, and $\bar{\lambda}^{(i)} \geq \rho(A)$ for all i. By [32, theorem 3.9], $B^{(i)} > 0$ and $\tilde{x}^{(i)} > 0$. Hence, the quantity $\min_j(\frac{\tilde{x}_j^{(i)}}{(B^{(i)}\tilde{x}^{(i)})_j}) > 0$, and $\bar{\lambda}^{(i)} > \bar{\lambda}^{(i+1)} \geq \rho(A)$ for all i. It is enough to establish that

$$\bar{\lambda}^{(0)} > \bar{\lambda}^{(1)} > \ldots > \bar{\lambda}^{(i)} > \ldots \geq \rho(A).$$

□

Before we can prove the results of part 3 and part 4, we gather elementary facts in the following lemmas.

Lemma 3.1 *Let A be a nonnegative irreducible matrix and $\rho(A)$ be the Perron root of A. If $\lambda_1 > \lambda_2 > \rho(A)$, then $(\lambda_2 I - A)^{-1} > (\lambda_1 I - A)^{-1} > 0$.*

Proof

By Neumann series, we obtain

$$(\lambda_1 I - A)^{-1} = \frac{1}{\lambda_1}\left(I - \frac{A}{\lambda_1}\right)^{-1} = \frac{1}{\lambda_1}\sum_{r=0}^{\infty}\left(\frac{A}{\lambda_1}\right)^r = \frac{1}{\lambda_1}\left(I + \frac{A}{\lambda_1} + \frac{A^2}{{\lambda_1}^2} + \ldots\right), \text{ and}$$

$$(\lambda_2 I - A)^{-1} = \frac{1}{\lambda_2}\left(I - \frac{A}{\lambda_2}\right)^{-1} = \frac{1}{\lambda_2}\sum_{r=0}^{\infty}\left(\frac{A}{\lambda_2}\right)^r = \frac{1}{\lambda_2}\left(I + \frac{A}{\lambda_2} + \frac{A^2}{{\lambda_2}^2} + \ldots\right).$$

Observe that

$$(\lambda_2 I - A)^{-1} - (\lambda_1 I - A)^{-1} = \sum_{r=0}^{\infty}\left(\frac{1}{{\lambda_2}^{r+1}} - \frac{1}{{\lambda_1}^{r+1}}\right) A^r.$$

Since $\lambda_1 > \lambda_2 > \rho(A)$, we have $\left(\frac{1}{{\lambda_2}^{r+1}} - \frac{1}{{\lambda_1}^{r+1}}\right) > 0$ for all $r \geq 0$.

Hence, $(\lambda_2 I - A)^{-1} - (\lambda_1 I - A)^{-1} = \alpha_0 I + \alpha_1 A + \alpha_2 A^2 + \ldots$, where $\alpha_r = \left(\frac{1}{{\lambda_2}^{r+1}} - \frac{1}{{\lambda_1}^{r+1}}\right)$.

Let $B = \sum_{k=0}^{\infty} \alpha_k A^k$, where $\alpha_k > 0$. Since A is a nonnegative and irreducible matrix, then for some particular i, j there exists a positive integer k such that $a_{ij}^k > 0$, and k

depends on i, j. Let $k' = \max(k)$ such that $a_{ij}^k > 0$; hence, the quantity $\alpha_0 I + \alpha_1 A + \alpha_2 A^2 + \ldots + \alpha_{k'} A^{k'} > 0$.

Since

$$(\lambda_2 I - A)^{-1} - (\lambda_1 I - A)^{-1} = \sum_{r=0}^{\infty} \left(\frac{1}{{\lambda_2}^{r+1}} - \frac{1}{{\lambda_1}^{r+1}} \right) A^r \geq \alpha_0 I + \alpha_1 A + \alpha_2 A^2 + \ldots + \alpha_{k'} A^{k'} > 0,$$

we have

$$(\lambda_2 I - A)^{-1} > (\lambda_1 I - A)^{-1} > 0.$$

□

Lemma 3.2 *Let $\tilde{x}^{(0)}$ be a positive vector and let $\lambda^{(i)} > \lambda^{(i+1)} > \rho(A)$, where $i = 0, 1, 2, \ldots$. If $B^{(i)} = (\lambda^{(i)} I - A)^{-1}$, then*

$$\max_j \left(\frac{\tilde{x}_j^{(i)}}{(B^{(i)} \tilde{x}^{(i)})_j} \right) \geq \max_j \left(\frac{\tilde{x}_j^{(i+1)}}{(B^{(i+1)} \tilde{x}^{(i+1)})_j} \right) > 0$$

and

$$\lim_{i \to \infty} \left\{ \max_j \left(\frac{\tilde{x}_j^{(i)}}{(B^{(i)} \tilde{x}^{(i)})_j} \right) \right\} = 0,$$

where $\tilde{x}^{(i+1)} = B^{(i)} B^{(i)} \tilde{x}^{(i)}$.

Proof

Note that both matrices $B^{(i)}$ and $B^{(i+1)}$ are positive and nonsingular because $\lambda^{(i)} > \lambda^{(i+1)} > \rho(A)$. For any fixed i, we set

$$m^{(i)} = \min_j \left(\frac{(B^{(i)} \tilde{x}^{(i)})_j}{\tilde{x}_j^{(i)}} \right).$$

So that

$$m^{(i)} \leq \left(\frac{(B^{(i)} \tilde{x}^{(i)})_j}{\tilde{x}_j^{(i)}} \right) \text{ for all } j$$

and

$$m^{(i)}\tilde{x}_j^{(i)} \leq (B^{(i)}\tilde{x}^{(i)})_j \text{ for all } j.$$

We then have that

$$m^{(i)}\tilde{x}^{(i)} \leq B^{(i)}\tilde{x}^{(i)}. \tag{3.4}$$

Multiplying both sides of equation (3.4) by $B^{(i)}$, we obtain

$$m^{(i)}B^{(i)}\tilde{x}^{(i)} \leq B^{(i)}B^{(i)}\tilde{x}^{(i)} \tag{3.5}$$

and multiplying both sides of equation (3.5) by $B^{(i)}$, we have

$$m^{(i)}B^{(i)}B^{(i)}\tilde{x}^{(i)} \leq B^{(i)}B^{(i)}B^{(i)}\tilde{x}^{(i)}. \tag{3.6}$$

Observe that $B^{(i+1)} > B^{(i)} > 0$ for all i by lemma 3.1, then equation (3.6) becomes

$$0 < m^{(i)}B^{(i)}B^{(i)}\tilde{x}^{(i)} \leq B^{(i+1)}B^{(i)}B^{(i)}\tilde{x}^{(i)}, \tag{3.7}$$

then we have

$$0 < m^{(i)}(B^{(i)}B^{(i)}\tilde{x}^{(i)})_j \leq (B^{(i+1)}B^{(i)}B^{(i)}\tilde{x}^{(i)})_j \ \text{ for all } j. \tag{3.8}$$

Hence,

$$0 < m^{(i)} \leq \frac{\left(B^{(i+1)}B^{(i)}B^{(i)}\tilde{x}^{(i)}\right)_j}{(B^{(i)}B^{(i)}\tilde{x}^{(i)})_j} = \frac{\left(B^{(i+1)}\tilde{x}^{(i+1)}\right)_j}{\tilde{x}_j^{(i+1)}} \text{ for all } j. \tag{3.9}$$

As a result,

$$0 < \min_j \left(\frac{(B^{(i)}\tilde{x}^{(i)})_j}{\tilde{x}_j^{(i)}} \right) \leq \min_j \left(\frac{(B^{(i+1)}\tilde{x}^{(i+1)})_j}{\tilde{x}_j^{(i+1)}} \right). \tag{3.10}$$

Then by lemma 3.11 part (vii) in [29], equation (3.10) becomes

$$0 < \max_j \left(\frac{\tilde{x}_j^{(i+1)}}{(B^{(i+1)}\tilde{x}^{(i+1)})_j} \right) \leq \max_j \left(\frac{\tilde{x}_j^{(i)}}{(B^{(i)}\tilde{x}^{(i)})_j} \right). \tag{3.11}$$

Next, as $\lambda^{(i)} \to \rho(A)$, $B^{(i)}$ grows unbounded, which means that the sequence

$$\left\{ \min_j \left(\frac{(B^{(i)}\tilde{x}^{(i)})_j}{\tilde{x}_j^{(i)}} \right) \right\}_{i=0}^{\infty}$$

is a real positive monotone increasing unbounded sequence. Therefore, by theorem 3.6.3 in [4] we have

$$\lim_{i\to\infty} \left\{ \min_j \left(\frac{(B^{(i)}\tilde{x}^{(i)})_j}{\tilde{x}_j^{(i)}} \right) \right\} = \infty.$$

Hence,

$$\lim_{i\to\infty} \left\{ \max_j \left(\frac{\tilde{x}_j^{(i)}}{(B^{(i)}\tilde{x}^{(i)})_j} \right) \right\} = \lim_{i\to\infty} \left(\frac{1}{\left\{ \min_j \left(\frac{(B^{(i)}\tilde{x}^{(i)})_j}{\tilde{x}_j^{(i)}} \right) \right\}} \right) = 0.$$

The proof of lemma 3.2 is complete. $\square$

We are now ready to prove part 3 and part 4 of theorem 3.3.

Part 3.

Proof

From part 2, the sequence $\{\bar{\lambda}^{(i)}\}$ is real positive monotone decreasing and bounded below by $\rho(A)$. By Bolzano-Weierstrass theorem, there exists a subsequence $\{\bar{\lambda}^{(i_k)}\}$ of $\{\bar{\lambda}^{(i)}\}$ such that $\lim_{k\to\infty} \bar{\lambda}^{(i_k)} = \bar{\lambda}^*$.

In addition, $\bar{\lambda}^{(i)} = \lambda^{(i)} - \min_j \left(\frac{\tilde{x}_j^{(i)}}{(B^{(i)}\tilde{x}^{(i)})_j}\right)$, we then have that the sequence $\{\lambda^{(i)}\}$ is real positive monotone decreasing bounded below by $\rho(A)$. Hence, by the standard result of real analysis, we know that

$$\lim_{i\to\infty} \lambda^{(i)} = \lambda^* \text{ exists, and } \lambda^* \geq \rho(A). \tag{3.12}$$

Since $\underline{\lambda}^{(i)} = \lambda^{(i)} - \max_j \left(\frac{\tilde{x}_j^{(i)}}{(B^{(i)}\tilde{x}^{(i)})_j}\right)$ and $\rho(A) \geq \underline{\lambda}^{(i)}$ for all $i \geq 0$, we have

$$\rho(A) - \underline{\lambda}^{(i)} = \rho(A) - \lambda^{(i)} + \max_j \left(\frac{\tilde{x}_j^{(i)}}{(B^{(i)}\tilde{x}^{(i)})_j}\right) \geq 0.$$

So that

$$\rho(A) - \lambda^{(i)} \geq -\max_j \left(\frac{\tilde{x}_j^{(i)}}{(B^{(i)}\tilde{x}^{(i)})_j}\right),$$

and

$$\lim_{i\to\infty} (\rho(A) - \lambda^{(i)}) \geq \lim_{i\to\infty} \left(-\max_j \left(\frac{\tilde{x}_j^{(i)}}{(B^{(i)}\tilde{x}^{(i)})_j}\right)\right).$$

The right hand side becomes 0 by lemma 3.2, and we obtain

$$\lim_{i\to\infty} (\rho(A) - \lambda^{(i)}) \geq 0;$$

consequently,

$$\rho(A) \geq \lim_{i\to\infty} \lambda^{(i)} = \lambda^*. \tag{3.13}$$

Thereby, $\rho(A) = \lambda^*$ by equations (3.12) and (3.13) and $\lim_{i\to\infty} \lambda^{(i)} = \rho(A)$.

Next, by lemma 3.11 in [29],

$$\max_j \left(\frac{\tilde{x}_j^{(i)}}{(B^{(i)}\tilde{x}^{(i)})_j} \right) \geq \min_j \left(\frac{\tilde{x}_j^{(i)}}{(B^{(i)}\tilde{x}^{(i)})_j} \right) > 0 \text{ for all } i \text{ and}$$

$$\lim_{i\to\infty} \left(\max_j \left(\frac{\tilde{x}_j^{(i)}}{(B^{(i)}\tilde{x}^{(i)})_j} \right) \right) = 0,$$

we have

$$\lim_{i\to\infty} \left(\min_j \left(\frac{\tilde{x}_j^{(i)}}{(B^{(i)}\tilde{x}^{(i)})_j} \right) \right) = 0.$$

Hence, $\lim_{i\to\infty} \bar{\lambda}^{(i)} = \lim_{i\to\infty} \left(\lambda^{(i)} - \min_j \left(\frac{\tilde{x}_j^{(i)}}{(B^{(i)}\tilde{x}^{(i)})_j} \right) \right) = \lim_{i\to\infty} \lambda^{(i)} = \rho(A)$.

$\square$

Part 4.

Proof

Recall that we define

$$\underline{\lambda}^{(i)} = \lambda^{(i)} - \max_j \left(\frac{\tilde{x}_j^{(i)}}{(B^{(i)}\tilde{x}^{(i)})_j} \right),$$

by lemma 3.2, we have

$$\lim_{i\to\infty} \underline{\lambda}^{(i)} = \lim_{i\to\infty} \left(\lambda^{(i)} - \max_j \left(\frac{\tilde{x}_j^{(i)}}{(B^{(i)}\tilde{x}^{(i)})_j} \right) \right) = \lim_{i\to\infty} \lambda^{(i)} = \rho(A).$$

$\square$

3.4 Test Matrices

In this section, we present the results of using test matrices to test the convergence of our algorithm.

The first test matrix that we use is the inverse of a tridiagonal M-matrix. This matrix has a huge gap between its maximum row sums and its minimum row sums. Given below is an inverse tridiagonal M-matrix of order n.

$$A_n = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & 2 & 2 & \cdots & 2 \\ 1 & 2 & 3 & \cdots & 3 \\ \vdots & \vdots & \ddots & n-1 & n-1 \\ 1 & 2 & \cdots & n-1 & n \end{pmatrix}_{n\times n} = \begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 1 \end{pmatrix}_{n\times n}^{-1} . \tag{3.14}$$

One advantage of using A_n as a test matrix is that the largest eigenvalue can be determined explicitly by a formula referred in [19] as

$$\rho(A_n) = 1/(2 - 2\cos(\pi/(2n+1)). \tag{3.15}$$

The second test matrix is an $n \times n$ tridiagonal Toplitz matrix of the form

$$A = \begin{pmatrix} b & a & & & \\ c & b & a & & \\ & \ddots & \ddots & \ddots & \\ & & c & b & a \\ & & & c & b \end{pmatrix}_{n\times n} \quad \text{where } a, b, \text{ and } c \in R \text{ and } a \neq c \neq 0. \tag{3.16}$$

The eigenpairs of this matrix can be calculated by the formula given [23]. For each eigenvalue $\lambda_j,\ 1 \leq j \leq n$, we have

$$\lambda_j = b + 2a\sqrt{c/a}\cos(j\pi/n+1), \tag{3.17}$$

and for each eigenvector,

$$x_j = \begin{pmatrix} (c/a)^{(1/2)}\sin\left(1j\pi/(n+1)\right) \\ (c/a)^{(2/2)}\sin\left(2j\pi/(n+1)\right) \\ (c/a)^{(3/2)}\sin\left(3j\pi/(n+1)\right) \\ \vdots \\ (c/a)^{(n/2)}\sin\left(nj\pi/(n+1)\right) \end{pmatrix}.$$

Observe that from equation (3.17), A possesses a set of n distinct eigenvalues; thus, A is diagonalizable. The proof of the formula in equation (3.17) and the diagonalizability are mentioned in [23].

Next, we use an $n \times n$ perturbation matrix defined as

$$P_{n\omega} = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ \omega & 0 & 0 & \cdots & 0 \end{pmatrix}_{n\times n}. \tag{3.18}$$

The eigenvalues of $P_{n\omega}$ can be computed via the formula mentioned in [12],

$$\lambda_k = \sqrt[n]{\omega}e^{(2k\pi i/n)}, \quad k = 1, 2, \ldots, n. \tag{3.19}$$

In particular, the Perron root of this matrix is the maximum of λ_k or when $k = n$. Clearly, if we choose $\omega = (\epsilon)^n$, the Perron root of this matrix is ϵ. Moreover, if we keep decreasing the value of ω, not only we can test for the convergence but we can also test the algorithm for the accuracy. This matrix is also used by the authors of [17] to test for the accuracy of their algorithm.

The results using each of the test matrices as an input are shown in the next section.

3.5 Experiments and Results

We have extensively tested our algorithm on various nonnegative irreducible matrices. The experiments are performed using Matlab version 6.5 on a Power PC G4 with processor 868 MHz and memory 1.5 GB with stopping criteria $tol = 10^{-14}$. Our numerical results are presented and compared to Noda's algorithm in [25], Elsner's algorithm in [10], modified Elsner's algorithm in [17], and the power method with Collatz's formula.

Experiment I: Let A be a nonnegative irreducible matrix of order 8 given below. Note that this matrix is also used as a test matrix by the author of [18].

$$A = \begin{pmatrix} 8 & 6 & 3 & 5 & 7 & 0 & 7 & 1 \\ 0 & 7 & 3 & 8 & 5 & 6 & 4 & 1 \\ 1 & 2 & 6 & 1 & 3 & 8 & 8 & 7 \\ 2 & 8 & 4 & 0 & 7 & 7 & 8 & 2 \\ 2 & 4 & 6 & 2 & 5 & 7 & 6 & 5 \\ 4 & 1 & 0 & 4 & 8 & 4 & 8 & 2 \\ 3 & 1 & 6 & 6 & 4 & 5 & 5 & 0 \\ 0 & 1 & 1 & 6 & 7 & 0 & 3 & 4 \end{pmatrix}$$

Using eig(A) command in Matlab, this matrix has a single dominant eigenvalue denoted by $\lambda_1 = 33.2418$. All algorithms produce the same Perron root as the Matlab *eig* command. The power method with Collatz's formula perform the best since A is small and the gap between $\rho(A)$ and the subdominant eigenvalue is large.

Table 3.1: Results of experiment I.

Algorithm	Time(sec)	Iterations	Max Eigenvalue
Noda	0.01864	5	33.2418
Elsner	0.01343	4	33.2418
Mod Elsner	0.01932	3	33.2418
Our	0.01813	3	33.2418
Power Method	0.00161	21	33.2418

Experiment II: We use a positive random matrix which is generated by the $rand(n, n)$ command in Matlab. In addition, we randomly replace some entries by 0 to create a new random nonnegative irreducible matrix. Table 3.2 shows the results when we apply all algorithms to the nonnegative random matrix size of $n = 3000$.

Table 3.2: Results of experiment II.

Algorithm	Time(sec)	Iterations	Max Eigenvalue
Noda	114.5173	4	1.500039943045885e+03
Elsner	114.3617	4	1.500039943045886e+03
Mod Elsner	126.3965	4	1.500039943045898e+03
Our	58.0162	2	1.500039943045885e+03
Power Method	8.8956	8	1.500039943045885e+03

Experiment III: We use the matrix mentioned in the equation (3.14) of order n.

First, consider when $n = 6$, we see that

$$A_6 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 2 & 2 & 2 & 2 \\ 1 & 2 & 3 & 3 & 3 & 3 \\ 1 & 2 & 3 & 4 & 4 & 4 \\ 1 & 2 & 3 & 4 & 5 & 5 \\ 1 & 2 & 3 & 4 & 5 & 6 \end{bmatrix}.$$

Applying the formula from equation (3.15), we obtain that the exact value of $\rho(A_6) = 17.2068572$.

Table 3.3: Results of experiment III when $n = 6$.

Algorithm	Time(sec)	Iterations	Max Eigenvalue
Noda	1.4299e-03	5	1.720685726740094e+01
Elsner	1.2319e-03	5	1.720685726740094e+01
Mod Elsner	5.4510e-03	6	1.720685726740095e+01
Our	1.8220e-03	3	1.720685726740094e+01
Power Method	1.1249e-03	9	1.720685726740095e+01

Next, we consider the case when $n = 1000$. The exact value of $\rho(A_{1000}) = 405690.203$. Table 3.4 shows the results of all algorithms on A_{1000}.

Table 3.4: Results of experiment III when $n = 1000$.

Algorithm	Time(sec)	Iterations	Max Eigenvalue
Noda	7.4830	5	4.056902039587474e+05
Elsner	7.5037	5	4.056902039587469e+05
Mod Elsner	8.1742	5	4.056902039587484e+05
Our	4.7143	3	4.056902039587469e+05
Power Method	1.4475	12	4.056902039587474e+05

Observe that the value of $\rho(A_n)$ grows quickly as n increases. All algorithms are able to compute the actual Perron root. To see which algorithm obtains better results, we increase the size of this matrix to $n = 2000$, and $n = 3000$. The result are shown in table 3.5 and table 3.6 respectively.

Table 3.5: Results of experiment III when $n = 2000$.

Algorithm	Time(sec)	Iterations	Max Eigenvalue
Noda	54.5220	5	1.6219e+06
Elsner	46.9695	5	1.6219e+06
Mod Elsner	49.0124	5	1.6219e+06
Our	28.8379	3	1.6219e+06
Power Method	6.1329	13	1.6219e+06

Table 3.6: Results of experiment III when $n = 3000$.

Algorithm	Time(sec)	Iterations	Max Eigenvalue
Noda	174.8943	6	3.6488e+06
Elsner	146.6018	5	3.6488e+06
Mod Elsner	149.7228	5	3.6488e+06
Our	88.4873	3	3.6488e+06
Power Method	9.69139	8	3.6488e+06

As the size of A increases, we see that our algorithm gives a better result than other algorithms.

Experiment IV: We consider the results for the computation of the Perron root of a nonsymmetric tridiagonal Toplitz matrix in the equation (3.16) which can be represent by $T(c, b, a, n)$ in the section 3.4. In this experiment, we use $c = 2, b = 8, a = 5$, and $n = 800$. Refer to the formula in equation (3.16), the Perron root of $T(2, 8, 5, 800)$ is $\rho(T) = 1.432450667579053e + 01$, where the second largest eigenvalue is close to $\rho(T)$.

Table 3.7: Results of experiment IV.

Algorithm	Time(sec)	Iterations	Max Eigenvalue
Noda	99.131	119	1.432450667579053e+01
Elsner	104.5101	118	1.432450667579053e+01
Mod Elsner	104.2466	119	1.432450667579053e+01
Our	56.1995	66	1.432450667579053e+01
Power Method	Does not converge		

In addition, we extensively tested our algorithm on different tridiagonal Toplitz matrices with different sizes and different values of $a, b,$ and c. The results of these experiments are similar to one given in the table 7. The computational time and the number of iterations of our algorithm are roughly about half of other algorithms required. In addition, the power method with Collatz's formula does not converge.

In the next experiment, we run all algorithms on the perturbation matrix P in the equation (3.18) which is a nonnegative and irreducible matrix. For each case in this experiment, we increase the size n and decrease the value of ω. The authors of [17] also use these matrices to test the accuracy of their algorithm.

Experiment V: This is the perturbation matrix whose size $n = 20$ and $\omega = (0.5)^{20}$ (note that $(0.5)^{20}$=9.5367e-007) defined as

$$P_{20(0.5)^{20}} = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ (0.5)^{20} & 0 & 0 & \cdots & 0 \end{pmatrix}_{20\times 20} . \tag{3.20}$$

Using the formula given in the section 3.4, the Perron root of this matrix is $\rho = 0.5$.

Table 3.8: Results of experiment V on $P_{20(0.5)^{20}}$.

Algorithm	Time(sec)	Iterations	Max Eigenvalue
Noda	0.0236	14	0.5
Elsner	0.0172	13	0.5
Mod Elsner	0.0362	12	0.5
Our	0.0167	8	0.5
Power Method	Does not converge		

Next, we consider $P_{200(0.5)^{20}}$. Note that, the Perron root of this matrix is $\rho = 9.330329915368075e-01$.

Table 3.9: Results of experiment V on $P_{200(0.5)^{20}}$.

Algorithm	Time(sec)	Iterations	Max Eigenvalue
Noda	0.4187	13	9.330329915368074e-01
Elsner	0.4578	13	9.330329915368074e-01
Mod Elsner	0.3629	12	9.330329915368074e-01
Our	0.3528	8	9.330329915368073e-01
Power Method	Does not converge		

Interestingly, modified Elsner, Elsner, and Noda's algorithm converge at the rate of quadratic [10]. Our algorithm overcomes all algorithms especially for a larger size of a nonnegative irreducible matrix. Below is the result when we test all algorithms on the perturbation matrix $P_{500(0.5)^{20}}$ with the size of $n = 500$ and $\omega = (0.5)^{20}$. The exact Perron root is $\rho = 9.726549474122856e - 01$.

Table 3.10: Results of experiment V on $P_{500(0.5)^{20}}$.

Algorithm	Time(sec)	Iterations	Max Eigenvalue
Noda	3.1728	13	9.726549474122854e-01
Elsner	3.0301	12	9.726549474122828e-01
Mod Elsner	3.5694	13	9.726549474123013e-01
Our	1.8613	7	9.726549474122880e-01
Power Method	Does not converge		

Next, we look at the case when $n = 1000$ and $\omega = 1e - 16$. The exact Perron root of $P_{1000(1e-16)}$ is $\rho = 9.638290236239705e - 01$.

Table 3.11: Results of experiment V on $P_{1000(1e-16)}$.

Algorithm	Time(sec)	Iterations	Max Eigenvalue
Noda	31.3633	22	9.638290236239706e-01
Elsner	50.7471	35	9.638290236239713e-01
Mod Elsner	37.0038	24	9.638290236239708e-01
Our	19.4263	13	9.638290236239706e-01
Power Method	Does not converge		

The result from experiments I, II, and III indicate that if we have a nonnegative irreducible matrix whose Perron root is well conditioned, then the power method with Collatz's formula produce the best result as we expected. However, our algorithm also produces good results compare to the power method and other algorithms; especially, when it computes the Perron root of large size matrices.

Furthermore, the results from experiment IV and V demonstrate that for a nonnegative irreducible matrix whose eigenvalues are ill conditioned, the power method does not converge, while other algorithms converge to the Perron root quickly. Nevertheless, when we increase the size of the matrix, we observe that the results from our algorithm are better than Noda, Elsner, and modified Elsner's algorithm.

Chapter 4

Conclusions

In this dissertation we explored various methods for computing the Perron root of nonnegative irreducible matrices. We produced a new algorithm. The proposed algorithm is based on the reciprocal of Collatz's formula and the inverse iteration method. We have shown how our algorithm overcomes the three best known algorithms that use an inverse iteration technique which converge to the Perron root at the rate of quadratic [10]. Moreover, our algorithm converges to the Perron root faster than other algorithms that employed the diagonal transformation technique and the Perron complementation idea.

We then took a closer look at the computational time of all algorithms including our algorithm on nearly reducible matrices. From all experiments, we found that our algorithm produced the best results. The experiments suggested our algorithm converges to the Perron root of nonnegative irreducible matrices at least quadratically.

We also proved several results concerning the monotonicity of the sequence approximating the Perron root and the convergence of our algorithm. We hope that these results motivate the applications of the Perron root of nonnegative matrices. We predict, with increasingly use of nonnegative matrices in various applications, our algorithm will be one of the best choice among others.

Bibliography

[1] A. Ahac, J.J. Buoni, and D.D. Olesky. A stable lu-factorization of h-matrices. *Linear algebra appl.*, 99:97–110., 1988.

[2] Pham Van At. Reduction of a positive matrix to a quasistochastic matrix by a similar variation method. *U.S.S.R. Computation Mathematics and Mathematical Physics.*, 11:255–262., 1971.

[3] Pham Van At. Reduction of an irreducible non-negativ matrix to quasistochastic form by the method similarity variation. *U.S.S.R. Computation Mathematics and Mathematical Physics.*, 15:11–18., 1975.

[4] Robert G. Bartle and Donald R. Sherbert. *Introduction to Real Analysis Third Edition.* John Wiley and Sons, Inc, New York, 2000.

[5] Abraham Berman. *Nonnegative Matrices in the Mathematical Sciences.* Society for Industrial and Applied Mathematics, Philadelphia., 1994.

[6] A. Brauer. A method for the computation of the greatest root of a nonnegative matrix. *SIAM J. Numer. Anal*, 3:564–569., 1966.

[7] Wolfgang Bunse. A class of diagonal transformation methods for the computation of the spectral radius of a nonnegative matrix. *Siam J. Numer. Anal.*, 18:693–704., 1981.

[8] James W. Demmel. *Applied Numerical Linear Algebra*. Society for Industrial and Applied Mathematics, Philadelphia, 1997.

[9] Fujian Duan and Kecun Zhang. An algorithm of diagonal transformation for perron root of nnonnegative irreducible matrices. *Applied Mathematics and Computation*, 175:762–772., 2006.

[10] Ludwig Elsner. Inverse iteration for calculating the spectral radius of a non-negative irreducible matrix. *Linear Algebra Appl.*, 15:235–242., 1976.

[11] R.P. FEYNMAN. *Statistical Mechanics*. Benjamin,Reading, MA, 1972.

[12] Robert T. Gregory and David L. Karney. *A Collection of Matrices for Testing Computational Algorithms*. Wiley-Interscience, New York, 1969.

[13] C.A. Hall and T.A. Porsching. Computing the maximal eigenvalue and eigenvector of a nonnegative irreducible matrix. *SIAM J. Numer. Anal*, 5:470–474., 1968.

[14] C.A. Hall and T.A. Porsching. Computing the maximal eigenvalue and eigenvector of a positive matrix. *SIAM J. Numer. Anal*, 5:269–274., 1968.

[15] T.E. Harris. *The Theory of Branching Processes*. Spring-Verlag, Berlin, 1963.

[16] R.A. Howard and J.E. Matheson. Risk sensitive markov dicision processes. *Management Sci.*, 8:356–369, 1972.

[17] L.Elsner, I.Koltracht, M. Neumann, and D.Xiao. On accurate computations of the perron root. *SIAM J. Matrix Anal. Appl.*, 14(2), April 1993.

[18] Linzhang Lu. Perron complement and perron root. *Linear Algebra Appl.*, 341:239–248., 2002.

[19] Linzhang Lu and Michael K. Ng. Localization of perron roots. *Linear Algebra Appl.*, 392:103–117., 2004.

[20] M.S. Lynn and W.P. Timlake. Bounds for perron eigenvectors and subdominant eigenvalues of positive matrices. *Linear Algebra and Its Applications*, 2:143–152., 1969.

[21] T.L. Markahm. An iterative procedure for computing the maximal root of a positive matrix. *Math Comp.*, 22:869–871., 1968.

[22] C.D. Meyer. Uncoupling the perron eigenvector problem. *Linear Algebra Appl.*, 114/115:69–94., 1989.

[23] C.D. Meyer. *Matrix Analysis and Applied Linear Algebra.* Society for Industrial and Applied Mathematics, Philadelphia, 2000.

[24] Michael Neumann. Inverses of perron complements of inverse m-matrices. *Linear Algebra Appl.*, 313:163–171., 2000.

[25] Takashi Noda. Note on the computation of the maximal eigenvalue of a non-negative irreducible matrix. *Numer. Math*, 17:382–386, 1971.

[26] Tedja Santanoe Oepomo. A contribution to collatz's eigenvalue inclusion theorem for nonnegative irreducible matrices. *Electronic Journal of Linear Algebra*, 10:31–45., 2003.

[27] A.M. Ostrowski. On the convergence of the rayleigh quotient iteration for the computation of the characteristic roots and vectors. v. *Archive for Rational Mechanics and analysis*, 3:472–481, 1959.

[28] R.C. Penner. A construction of pseudo-anosov homeomorphisms. *Trans. Amer. Math. Soc.*, 310, 1988.

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

[30] R.D. Skeel. Scaling for numerical stability in gaussian elimination. *J. Assoc. Compt. Mach.*, 26, 1979.

[31] Slawomir Stanczak and Holger Boche. Information Theoretic Approach to the Perron Root of Nonnegative Irreducible Matrices. In *Proc. IEEE Information Theory Workshop (ITW 2004)*, pages 254–259, San Antonio, USA, October 2004.

[32] R.S. Varga. *Matrix iterative analsis.* Prentice-Hall Inc., Engwood Cliffs,N.J., 1962.

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

[34] Shiming Yang and Ting-Zhu Huang. A note on estimations of the spectral radius for nonnegative matrices. *Electronic Journal of Linear Algebra*, 13:352–358., 2005.