# EIGENVALUES AND EIGENVECTORS IN THE MAX-PLUS ALGEBRA

by

Misoo Chung

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado at Denver

in partial fulfillment

of the requirements for the degree of

Master of Science

Applied Mathematics

1995

This thesis for the Master of Science

degree by

Misoo Chung

has been approved

by

David C. Fisher

William E. Cherowitzo

Date

Chung, Misoo (M.S., Applied Mathematics)

Eigenvalues and Eigenvectors in the Max-Plus Algebra

Thesis directed by Associate Professor David C. Fisher

## Abstract

The max-plus algebra defined in the set $\Re \cup \{-\infty\}$ is an algebra with two binary operations $\oplus$ and $\otimes$ where $a \oplus b$ is the maximum of $a$ and $b$, and $a \otimes b$ is the sum of $a$ and $b$. These operations form a monoid-field (there is no inverse under $\oplus$). This paper implements algorithms for solving linear systems and computing eigenvalues and eigenvectors including the first known polynomial-time algorithm for finding eigenvalues of matrices in the max-plus algebra. Analogs to the characteristic equation and the Cayley-Hamilton theorem are presented.

This abstract accurately represents the content of the candidate's thesis. I recommend its publication.

Signed ____________________
David C. Fisher

# Contents


**1 The Max-Plus Algebra** **1**

1.1 Matrix Multiplication in $(\Re_{\max})^{n\times n}$ . . . . . . . . . . . . . . . . . . 3

1.2 An Application . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5

**2 Systems of Linear Equations in $(\Re_{\max})^n$** **8**

2.1 Solving $A\mathbf{x} \oplus \mathbf{b} = C\mathbf{x} \oplus \mathbf{d}$ . . . . . . . . . . . . . . . . . . . . . . . 8

2.2 Solving $\mathbf{x} = A\mathbf{x} \oplus \mathbf{b}$ . . . . . . . . . . . . . . . . . . . . . . . . . 9

2.3 Solving $A\mathbf{x} = \mathbf{b}$ . . . . . . . . . . . . . . . . . . . . . . . . . . . 15

**3 Eigenvalues and Eigenvectors** **16**

3.1 Existence and Uniqueness . . . . . . . . . . . . . . . . . . . . . . . 16

3.2 Maximum Cycle Mean Method . . . . . . . . . . . . . . . . . . . . 19

3.3 Power Method . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 22

3.4 Linear Programing Approach . . . . . . . . . . . . . . . . . . . . . 25

**4 Characteristic Polynomial and the Cayley-Hamilton Theorem** **28**

**5 References** **35**

# List of Figures


1 The precedence graph of $A$ . . . . . . . . . . . . . . . . . . . . . . . . 10

2 The precedence graph of $B$ . . . . . . . . . . . . . . . . . . . . . . . . 22

3 The graph of $x^4 = 2x^3 \oplus 5x^2 \oplus 10x \oplus 14$. . . . . . . . . . . . . . . . . 33

# List of Tables

1 Flops and Iterations by the Power Method . . . . . . . . . . . . . . 25

2 Flops by a Linear Program . . . . . . . . . . . . . . . . . . . . . . . 28

ACKNOWLEDGEMENTS

Professor Fisher has willingly lent his expertise and guidance in a caring and supportive manner. For his work in mentoring, I give my complete thanks. Professor Cherowitzo has also given me his time and talent without reservation earning my appreciation. Then, my thanks is given to Allen Holder and professor Jeffrey Stuart (University of Southern Mississippi) for their help with linear programming.

# 1 The Max-Plus Algebra

The thesis implements linear algebra algorithms for matrices under the operations $\oplus$ (pronounced 'o-plus') and $\otimes$ (pronounced 'o-times'). The symbol $\oplus$ in $a \oplus b$ means the maximum of the scalars $a$ and $b$ and the symbol $\otimes$ refers to addition. The role of $\otimes$ is generally played by conventional addition. If it is clear where the '$\otimes$'-symbols are used, they are sometimes omitted. This is exactly the same as the multiplication '$\times$' or '$\cdot$' symbol in conventional algebra.

**Definition 1 (Monoid-field):** *A set $K$ endowed with two binary operations $\oplus$ and $\otimes$ is a monoid-field if*

- $(K, \oplus)$ is a commutative monoid (the operation $\oplus$ is associative, commutative and has a zero element $\epsilon$);
- $(K - \epsilon, \otimes)$ is a group with an identity element $e$;
- for all $a, b, c \in K$, the left distributive law, $a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$, and the right distributive law, $(a \oplus b) \otimes c = (a \otimes c) \oplus (b \otimes c)$, hold;
- the operation $\otimes$ satisfies $\epsilon \otimes e = e \otimes \epsilon = \epsilon$.

*We say that the monoid-field is*

- idempotent if the operation $\oplus$ is idempotent, that is, if $a \oplus a = a, \forall\, a \in K$;

- commutative if the operation $\otimes$ is commutative.

**Definition 2 (The algebraic structure $\Re_{\max}$):** *The symbol $\Re_{\max}$ denotes the set $\Re \cup \{-\infty\}$ with $\max$ and $+$ as the two binary operations $\oplus$ and $\otimes$, respectively.*

We call this structure the max-plus algebra.

**Proposition 1:** *The algebraic structure $\Re_{\max}$ satisfies the following properties:*

- $\alpha \oplus (\beta \oplus x) = (\alpha \oplus \beta) \oplus x$ i.e., $\max(\alpha, \max(\beta, x)) = \max(\max(\alpha, \beta), x)$;

- $\alpha \oplus \beta = \beta \oplus \alpha$ i.e., $\max(\alpha, \beta) = \max(\beta, \alpha)$;

- $-\infty \oplus x = x \oplus -\infty = x$ i.e., $\max(-\infty, x) = \max(x, -\infty) = x$;

- $\alpha \otimes (\beta \otimes x) = (\alpha \otimes \beta) \otimes x$ i.e., $\alpha + (\beta + x) = (\alpha + \beta) + x$;

- $\alpha \otimes \beta = \beta \otimes \alpha$ i.e., $\alpha + \beta = \beta + \alpha$;

- $0 \otimes x = x \otimes 0 = x$ i.e., $0 + x = x + 0 = x$;

- $-x \otimes x = x \otimes -x = 0$ i.e., $-x + x = x + (-x) = 0$;

- $-\infty \otimes x = x \otimes -\infty = -\infty$ i.e., $-\infty + x = x + (-\infty) = -\infty$;

- $\alpha \otimes (x \oplus y) = (\alpha \otimes x) \oplus (\alpha \otimes y)$ i.e., $\alpha + \max(x, y) = \max(\alpha + x, \alpha + y)$;

- $(\alpha \oplus \beta) \otimes x = (\alpha \otimes x) \oplus (\beta \otimes x)$ i.e., $\max(\alpha, \beta) + x = \max(\alpha + x, \beta + x)$;

- $z^{\gamma} = \gamma z$ where $z,\ \gamma \in \Re$ and the multiplication on the right is that of $\Re$;

*for all* $\alpha, \beta, x, y \in \Re_{\max}$.

**Theorem 1:** *The algebraic structure* $\Re_{\max}$ *is an idempotent commutative monoid-field (There is no inverse under* $\oplus$*) with* $\epsilon = -\infty$ *and* $e = 0$.

If we compare the properties of $\oplus$ and $\otimes$ of $\Re_{\max}$ with those of $+$ and $\times$ of $\Re$, we see that:

- we have lost the inverse of addition (for a given $a$, an element $b$ does not exist such that $\max(b, a) = -\infty$ whenever $a \neq -\infty$);
- we have gained the idempotency of addition.

The relation "$\leq$" will have its usual meaning with the convention that $-\infty \leq \beta$ for all $\beta$ and $\alpha \leq -\infty$ only if $\alpha = -\infty$.

**Definition 3 (The algebraic structure $\bar{\Re}_{\max}$):** *The set* $\Re \cup \{-\infty\} \cup \{\infty\}$ *endowed with the operations* max *and* $+$ *as* $\oplus$ *and* $\otimes$ *and with the convention that* $(-\infty) + \infty = -\infty$ *is denoted* $\bar{\Re}_{\max}$.

## 1.1 Matrix Multiplication in $(\Re_{\max})^{n \times n}$

This paper is mainly interested in computations of matrices in the max-plus algebra. In this section, we define basic operations of matrices in the max-plus algebra including a program written in the Matlab language.

Let $(\Re_{\max})^{n \times n}$ be the set of $n \times n$ matrices with coefficients in $\Re_{\max}$ endowed with the following two internal operations:

- the componentwise addition denoted $\oplus$;
- the matrix multiplication denoted $\otimes$:

$$(A \otimes B)_{ij} = \bigoplus_{k=1}^{n} A_{ik} \otimes B_{kj};$$

and the external operations:

- $\forall \alpha \in \Re_{\max}, \forall A \in (\Re_{\max})^{n \times n}, \alpha A = (\alpha A_{ij})$.

The set $(\Re_{\max})^{n \times n}$ has

- the zero matrix, again denoted $-\infty$, which has all its entries equal to $-\infty$;
- the identity matrix, again denoted $\mathbf{0}$, which has the diagonal entries equal to 0 and the other entries equal to $-\infty$.

**Example 1:** Consider the following matrix multiplication.

$$\begin{bmatrix} 3 & 7 \\ 2 & 4 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 3 & 2 \end{bmatrix} = \begin{bmatrix} (3 \otimes 1) \oplus (7 \otimes 3) & (3 \otimes 0) \oplus (7 \otimes 2) \\ (2 \otimes 1) \oplus (4 \otimes 3) & (2 \otimes 0) \oplus (4 \otimes 2) \end{bmatrix} = \begin{bmatrix} 10 & 9 \\ 7 & 6 \end{bmatrix}.$$

**Program 1 (Matrix Multiplication):** If $A \in (\Re_{\max})^{m \times r}$ and $B \in (\Re_{\max})^{r \times n}$, then the following program computes $C = A \otimes B$.

```
function C = maxmult(A,B)
   [m,r]=size(A); [s,n]=size(B);
   if r==s C=-inf*ones(m,n);
      for i=1:m for j=1:n for k=1:r
         C(i,j)=max(C(i,j),(A(i,k)+B(k,j)));
      end end end
   else disp('Inner matrix dimensions must agree.') end
```

## 1.2 An Application

Where do we use max-plus algebra? One application can be found in calculating the 2-packing number of an $m \times n$ grid. The 2-packing number is the maximum size of a set $S$ of nodes of the grid such that every pair of nodes in $S$ are more than 2 apart. Fisher [**2**] found the 2-packing number of an $m \times n$ grid for all $m$ and $n$. However, the techniques used there were specific to this problem. Here we give an example of how the max-plus algebra can be used to solve problems of this type by finding the 2-packing number of a $4 \times n$ grid for all $n$ in finite time.

A node's status is given by a code: 0 if it is in the set $S$, 1 if it is above, below or to the right of a node in $S$, and 2 otherwise. Each column of the $4 \times n$ grid can be represented as a 4-digit trinary number. For example, 0112 means that the first node is in $S$, the second node and the third node are next (above, below, or right) to a node in $S$, and the forth is left of a node in $S$. Since substrings 02, 20, 00, 010, 1212, and 2121 are not allowed in each column and we cannot have 11 unless 0 is next to it on at least one side, there are 17 states: 0110, 0112, 0121, 0122, 1011, 1012, 1101, 1210, 1221, 1222, 2101, 2110, 2122, 2210, 2212, 2221, and 2222. Now we can form a $17 \times 17$ matrix $A$ indexed by the states. If $A = (A_{ij})$, then

$$A_{ij} = \begin{cases} -\infty & \text{if state } A_i \text{ cannot be right of state } A_j \\ 0 & \text{if state } A_i \text{ can be right of state } A_j \text{ and has no node in S} \\ 1 & \text{if state } A_i \text{ can be right of state } A_j \text{ and has one node in S} \\ 2 & \text{if state } A_i \text{ can be right of state } A_j \text{ and has two nodes in S.} \end{cases}$$

$$A = \begin{bmatrix}
- & - & - & - & - & - & - & - & - & - & - & - & 2 & - & 2 & - & 2 \\
- & - & - & - & - & - & - & - & - & - & 1 & - & - & - & - & - & - \\
- & - & - & - & - & - & - & - & - & - & - & 1 & - & 1 & - & - & - \\
- & - & - & - & - & - & - & - & - & - & - & - & 1 & - & 1 & 1 & 1 \\
- & - & - & - & - & - & - & 1 & - & - & - & - & - & 1 & - & - & - \\
- & - & - & - & - & - & - & - & 1 & 1 & - & - & - & - & 1 & 1 & 1 \\
- & - & 1 & 1 & - & - & - & - & - & - & - & - & - & - & - & - & - \\
- & 1 & - & 1 & - & - & - & - & - & - & - & - & - & - & - & - & - \\
0 & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - & - \\
- & 0 & 0 & 0 & - & - & - & - & - & - & - & - & - & - & - & - & - \\
- & - & - & - & - & - & - & - & 1 & 1 & - & - & 1 & - & - & 1 & 1 \\
- & - & - & - & - & 1 & - & - & - & - & - & - & - & - & - & - & - \\
- & - & - & - & 0 & 0 & - & - & - & - & - & - & - & - & - & - & - \\
- & - & - & - & - & - & - & - & - & - & - & - & 1 & - & 1 & - & 1 \\
- & - & - & - & - & - & 0 & - & - & - & 0 & - & - & - & - & - & - \\
- & - & - & - & - & - & - & 0 & - & - & - & 0 & - & 0 & - & - & - \\
- & - & - & - & - & - & - & - & 0 & 0 & - & - & 0 & - & 0 & 0 & 0
\end{bmatrix}.$$

We call this matrix the transfer matrix. We denote $-\infty$ as $-$ in the above matrix for ease of reading. Let $\mathbf{x}_1$ be an initial vector of length 17. If state $i$ can be the state for a $4 \times 1$ grid, then $x_i$ is the number of nodes in $S$ in state $i$. If the state $i$ cannot be a state for a $4 \times 1$ grid, then $x_i$ is $-\infty$. For example, $x_6$ is 1, because the 6th state 1012 can be a state of a $4 \times 1$ grid and it has one node in $S$. However, $x_7$ is $-\infty$, because the first node of the 7th state 1101 must be 2. So,

$$\mathbf{x}_1 = (2, -\infty, -\infty, 1, -\infty, 1, -\infty, -\infty, -\infty, -\infty, 1, -\infty, -\infty, 1, -\infty, -\infty, 0)^T.$$

We can compute $\mathbf{x}_k$ from $\mathbf{x}_{k-1}$ for $k > 1$ by the transfer matrix $A$; $\mathbf{x}_k = A \otimes \mathbf{x}_{k-1}$.

Then the 2-packing number, $\eta(P_4 \times P_n)$, is the maximum entry in $\mathbf{x}_n$. Here is an example,

$$\mathbf{x}_2 = A\mathbf{x}_1 = (2,2,2,1,2,1,2,2,2,1,1,2,1,1,1,1,0)^T$$

$$\mathbf{x}_3 = A\mathbf{x}_2 = (3,2,3,2,3,3,3,3,2,2,3,2,2,2,2,2,2)^T$$

$$\mathbf{x}_4 = A\mathbf{x}_3 = (4,4,3,3,4,3,4,3,3,3,3,4,3,3,3,3,2)^T$$

$$\mathbf{x}_5 = A\mathbf{x}_4 = (5,4,5,4,4,4,4,5,4,4,4,4,4,4,4,4,3)^T$$

$$\mathbf{x}_6 = A\mathbf{x}_5 = (6,5,5,5,6,5,6,5,5,5,5,5,4,5,4,5,4)^T$$

$$\mathbf{x}_7 = A\mathbf{x}_6 = (6,6,6,6,6,6,6,6,6,5,6,6,6,5,6,5,5)^T$$

$$\mathbf{x}_8 = A\mathbf{x}_7 = (8,7,7,7,7,7,7,7,6,6,7,7,6,7,6,6,6)^T$$

$$\mathbf{x}_9 = A\mathbf{x}_8 = (8,8,8,7,8,7,8,8,8,7,7,8,7,7,7,7,6)^T.$$

So, $\eta(P_4 \times P_2) = 2$, $\eta(P_4 \times P_3) = 3$, $\eta(P_4 \times P_4) = 4$, $\eta(P_4 \times P_5) = 5$, $\eta(P_4 \times P_6) = 6$, $\eta(P_4 \times P_7) = 6$, $\eta(P_4 \times P_8) = 8$, and $\eta(P_4 \times P_9) = 8$. Notice that $\mathbf{x}_9 - \mathbf{x}_2 = (6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6)^T$, i.e., $\mathbf{x}_9 = 6 \otimes \mathbf{x}_2$. Thus, $\mathbf{x}_{10} = A\mathbf{x}_9 = A(6 \otimes \mathbf{x}_2) = 6 \otimes A\mathbf{x}_2 = 6 \otimes \mathbf{x}_3$. We may easily establish that $\mathbf{x}_k = 6 \otimes \mathbf{x}_{k-7}$ for $k \geq 9$. Since $\eta(P_4 \times P_k)$ is the maximum entry of $\mathbf{x}_k$, we have $\eta(P_4 \times P_k) = \mathbf{y}^T \otimes \mathbf{x}_k$ where $\mathbf{y} = [0,0,\ldots,0]^T$. Then $\eta(P_4 \times P_k) = \mathbf{y}^T \otimes \mathbf{x}_k = \mathbf{y}^T \otimes (6 \otimes \mathbf{x}_{k-7}) = 6 \otimes \mathbf{y}^T \otimes \mathbf{x}_{k-7} = 6 + \eta(P_4 \times P_{k-7})$.

Chapter 2 deals with systems of linear equations in the max-plus algebra. It contains algorithms for solving linear systems. Chapter 3 talks about different methods of finding eigenvalues and eigenvectors of any square matrix. Chapter 4 shows some examples of characteristic equation and the Cayley-Hamilton theorem in the max-plus algebra. Chapter 5 includes references. The main objective of this thesis was to write computer programs for Matlab that perform matrix computations in the max-plus algebra. Underlying theory is mostly based on Baccelli, Cohen, Olsder, and Quadrat [**1**].

# 2 Systems of Linear Equations in $(\Re_{\max})^n$

In this chapter, we are mainly concerned with systems of linear equations. There are two kinds of linear systems in $\Re_{\max}$ for which we are able to compute solutions: $\mathbf{x} = A\mathbf{x} \oplus \mathbf{b}$ and $A\mathbf{x} = \mathbf{b}$ (the general system being $A\mathbf{x} \oplus \mathbf{b} = C\mathbf{x} \oplus \mathbf{d}$).

## 2.1 Solving $A\mathbf{x} \oplus \mathbf{b} = C\mathbf{x} \oplus \mathbf{d}$

In the max-plus algebra, the general system of equations is $A\mathbf{x} \oplus \mathbf{b} = C\mathbf{x} \oplus \mathbf{d}$, where $A$ and $C$ are $n \times n$ matrices and $\mathbf{b}$ and $\mathbf{d}$ are $n$-vectors.

**Definition 4** [**1**]**:** *The system $A\mathbf{x} \oplus \mathbf{b} = C\mathbf{x} \oplus \mathbf{d}$ is said to be in canonical form if $A$, $C$, $\mathbf{b}$, and $\mathbf{d}$ satisfy*

$$C_{ij} = -\infty \text{ if } A_{ij} > C_{ij}, \text{ and } A_{ij} = -\infty \text{ if } A_{ij} < C_{ij}$$

$$d_i = -\infty \text{ if } b_i > d_i, \text{ and } b_i = -\infty \text{ if } b_i < d_i.$$

The solutions of the canonical form are the same as the solutions of the original system.

**Example 2:** Consider the system

$$\begin{bmatrix} 5 & 2 \\ -\infty & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \oplus \begin{bmatrix} 1 \\ 3 \end{bmatrix} = \begin{bmatrix} 6 & 1 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \oplus \begin{bmatrix} 0 \\ 5 \end{bmatrix},$$

which can be put into canonical form as follows:

$$\begin{bmatrix} -\infty & 2 \\ -\infty & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \oplus \begin{bmatrix} 1 \\ -\infty \end{bmatrix} = \begin{bmatrix} 6 & -\infty \\ 2 & -\infty \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \oplus \begin{bmatrix} -\infty \\ 5 \end{bmatrix},$$

which implies

$$\begin{bmatrix} (-\infty \otimes x_1) \oplus (2 \otimes x_2) \\ (-\infty \otimes x_1) \oplus (2 \otimes x_2) \end{bmatrix} \oplus \begin{bmatrix} 1 \\ -\infty \end{bmatrix} = \begin{bmatrix} (6 \otimes x_1) \oplus (-\infty \otimes x_2) \\ (2 \otimes x_1) \oplus (-\infty \otimes x_2) \end{bmatrix} \oplus \begin{bmatrix} -\infty \\ 5 \end{bmatrix}$$

$$\Rightarrow \quad \begin{cases} (2 \otimes x_2) \oplus 1 = 6 \otimes x_1 \\ 2 \otimes x_2 = (2 \otimes x_1) \oplus 5 \end{cases} \quad \Rightarrow \quad 6 \otimes x_1 = (2 \otimes x_1) \oplus 5$$

$$\Rightarrow \quad 6 \otimes x_1 = 5 \quad \Rightarrow \quad x_1 = -1 \quad \Rightarrow \quad x_2 = 3.$$

This system has a solution. In general, a linear system may or may not have a solution. Moreover, even if a solution exists, it may not be unique.

## 2.2 Solving $\mathbf{x} = A\mathbf{x} \oplus \mathbf{b}$

This section shows how to solve the system $\mathbf{x} = A\mathbf{x} \oplus \mathbf{b}$.

**Definition 5 (Precedence graph):** *The precedence graph of a square $n \times n$ matrix $A$ with entries in $\Re_{\max}$ is a weighted digraph with $n$ nodes and an arc $(j, i)$ if $A_{ij} \neq -\infty$, in which case the weight this arc receives is the numerical value of $A_{ij}$. The precedence graph is denoted $G(A)$.*

**Example 3:** Let $A = \begin{bmatrix} -\infty & 2 & -\infty & -3 \\ 1 & -\infty & -\infty & -\infty \\ -\infty & 0 & -1 & -\infty \\ -\infty & 5 & -2 & -\infty \end{bmatrix}$. Then, the precedence graph of $A$ is shown in Figure 1.

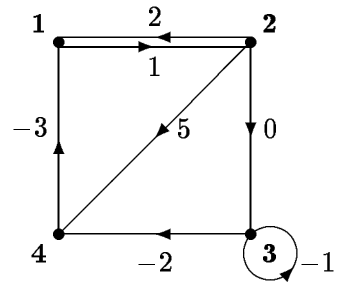

Figure 1: The precedence graph of $A$

To exclude degenerate cases, it is assumed that not all components of $A$ are identical to $-\infty$. Moreover, this paper considers mainly those $A$ in $(\Re_{\max})^{n\times n}$ whose precedence graph is strongly connected.

**Theorem 2** [1]: *If there are only circuits of nonpositive weight in $G(A)$, there is a solution to $\mathbf{x} = A\mathbf{x} \oplus \mathbf{b}$ which is given by $\mathbf{x} = A^*\mathbf{b}$, where $A^* = \mathbf{0} \oplus A \oplus \cdots \oplus A^n \oplus A^{n+1} \oplus \cdots$. Moreover, if the circuit weights are all negative, the solution is unique.*

Note that, in the related min-plus algebra, that is, $a \oplus b$ means the minimum of the scalars $a$ and $b$, if the circuit weights are nonnegative, then there is a solution.

*Proof:* If $A^*\mathbf{b}$ does exist, it is a solution:

$$A(A^*\mathbf{b}) \oplus \mathbf{b} = AA^*\mathbf{b} \oplus \mathbf{0}\mathbf{b} = (\mathbf{0} \oplus AA^*)\mathbf{b} = A^*\mathbf{b}.$$

**Existence of $A^*\mathbf{b}$.** The meaning of $(A^*)_{ij}$ is the maximum weight of all paths of any length from $j$ to $i$. Thus, a necessary and sufficient condition for the existence of $(A^*)_{ij}$ is that no strongly connected components of $G(A)$ have a circuit with positive weight. Otherwise, there would exist a path from $j$ to $i$ of arbitrarily large weight for all $j$ and $i$ belonging to the strongly connected component which includes the circuits of positive weight (by traversing this circuit a sufficient number of times).

**Uniqueness of the solution.** Suppose that $\mathbf{x}$ is a solution of $\mathbf{x} = A\mathbf{x} \oplus \mathbf{b}$. Then $\mathbf{x}$ satisfies

$$\mathbf{x} = \mathbf{b} \oplus A\mathbf{x},$$

$$\mathbf{x} = \mathbf{b} \oplus A(A\mathbf{x} \oplus \mathbf{b}),$$

$$\mathbf{x} = \mathbf{b} \oplus A\mathbf{b} \oplus A^2\mathbf{x},$$

$$\mathbf{x} = \mathbf{b} \oplus A\mathbf{b} \oplus \cdots \oplus A^{k-1}\mathbf{b} \oplus A^k\mathbf{x}, \tag{1}$$

and thus $\mathbf{x} \geq A^*\mathbf{b}$, where the order relation on the vectors is defined by $\mathbf{x} \geq \mathbf{y}$ if $\mathbf{x} \oplus \mathbf{y} = \mathbf{x}$. Moreover, if all the circuits of the graph have negative weights, then $A^k \to -\infty$ when $k \to \infty$. Indeed, the entries of $A^k$ are the weights of the paths of length $k$ which necessarily traverse some circuits of $A$ a number of times going to $\infty$ with $k$, but the weights of these circuits are all negative. Using this property in Equation (1) for $k$ large enough, we obtain that $\mathbf{x} = A^*\mathbf{b}$. $\square$

**Example 4:** Consider the following equation

$$\mathbf{x} = \begin{bmatrix} -1 & 1 \\ -3 & -2 \end{bmatrix} \mathbf{x} \oplus \begin{bmatrix} 0 \\ 2 \end{bmatrix}.$$

Then,

$$\mathbf{b} = \begin{bmatrix} 0 \\ 2 \end{bmatrix}, \quad A\mathbf{b} = \begin{bmatrix} 3 \\ 0 \end{bmatrix}, \quad A^2\mathbf{b} = \begin{bmatrix} 2 \\ 0 \end{bmatrix}, \quad A^3\mathbf{b} = \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \quad A^4\mathbf{b} = \begin{bmatrix} 0 \\ -2 \end{bmatrix} \ldots.$$

Notice that the entries of $A^k\mathbf{b}$ are decreasing as $k$ gets larger. Thus,

$$\mathbf{x} = \begin{bmatrix} 0 \\ 2 \end{bmatrix} \oplus \begin{bmatrix} 3 \\ 0 \end{bmatrix} \oplus \begin{bmatrix} 2 \\ 0 \end{bmatrix} \oplus \begin{bmatrix} 1 \\ -1 \end{bmatrix} \oplus \begin{bmatrix} 0 \\ -2 \end{bmatrix} \oplus \cdots = \begin{bmatrix} 3 \\ 2 \end{bmatrix}.$$

This is the unique solution, since all the circuit weights are negative.

**Program 2 (Finding the Weight of each Circuit):** If $A \in (\Re_{\max})^{m\times n}$, then the following program computes the weights of each circuit of $A$. If the weight of any circuit is positive, the program will be terminated and there will be no solution. The next program *linear2* uses this function call to check the positivity of any circuit of $A$.

```
function c=cycleok(A),
   [m,n]=size(A); c=-1;
   for i=1:n c=A(i,i); if c>0 break; end p(i)=i; end
   if c<=0 rcom=p;
      for r=2:n repeat=n;
         for k=1:r-1 repeat=repeat*(n-k); end
         for k=1:r repeat=repeat/k; end
         for i=1:r rcombi(i)=rcom(i); end
         tempy=0;
         for i=1:r-1 tempy=tempy+A(rcombi(i),rcombi(i+1)); end
         c=tempy+A(rcombi(r),rcombi(1)); if c>0 break; end
         factorial=r-1;
         for i=1:r-2 if r-2>0 factorial=factorial*(r-1-i); end end
         for i=1:r-1 rcombitemp(i)=rcombi(i+1); end
         for i=1:factorial-1 rcombitemp=perm(rcombitemp);
            for i=2:r rcombi(i)=rcombitemp(i-1); end
            tempy=0;
            for i=1:r-1 tempy=tempy+A(rcombi(i),rcombi(i+1)); end
            c=tempy+A(rcombi(r),rcombi(1));
            if c>0 break; end end
```

```
        for l=2:repeat i=r;
           while rcom(i)==n-r+i, if i>1 i=i-1; end end
           rcom(i)=rcom(i)+1;
           for j=i+1:r rcom(j)=rcom(i)+j-i; end
           for i=1:r rcombi(i)=rcom(i); end
           tempy=0;
           for i=1:r-1 tempy=tempy+A(rcombi(i),rcombi(i+1)); end
           c=tempy+A(rcombi(r),rcombi(1)); if c>0 break; end
           factorial=r-1;
           for i=1:r-2
              if r-2>0 factorial=factorial*(r-1-i); end end
           for i=1:r-1 rcombitemp(i)=rcombi(i+1); end
           for i=1:factorial-1 rcombitemp=perm(rcombitemp);
              for i=2:r rcombi(i)=rcombitemp(i-1); end
              tempy=0;
              for i=1:r-1
                 tempy=tempy+A(rcombi(i),rcombi(i+1)); end
              c=tempy+A(rcombi(r),rcombi(1)); if c>0 break; end
           end
        end
        rcom=p;
     end
  end
```

**Program 3 (Solution of $\mathbf{x} = A\mathbf{x} \oplus \mathbf{b}$):** If $A \in (\Re_{\max})^{m\times n}$ and $\mathbf{b} \in (\Re_{\max})^{n\times 1}$, then the following program finds the solution of $\mathbf{x} = A\mathbf{x} \oplus \mathbf{b}$ by using the algorithm of Theorem 2. There is no solution if the weight of any circuit is positive.

```
function x=linear2(A,b),
   [m,n]=size(A); [r,s]=size(b);
   if m==n if r==n if s==1 c=cycleok(A);
      if c>0 disp('There is no solution.')
      else temp=b; x=b; check=inf*ones(n,1);
         while x~=check, temp=maxmult(A,temp);
            for i=1:n x(i,1)=max(x(i,1),temp(i,1)); end
            check=maxmult(A,x);
            for i=1:n check(i,1)=max(check(i,1),b(i,1)); end
      end end end
   else disp('Column size for b must be 1.') end
   else disp('Inner matrix dimensions must agree.') end
   else disp('Matrix A must be square.') end
```

Program 3 uses program 2 and it can waste time if there is no circuit with a positive weight. Also checking the weight of all circuits is inefficient so the next program modifies this check.

**Program 4 (Solution of $\mathbf{x} = A\mathbf{x} \oplus \mathbf{b}$):** If $A \in (\Re_{\max})^{m\times n}$ and $\mathbf{b} \in (\Re_{\max})^{n\times 1}$, then the following program computes the solution of $\mathbf{x} = A\mathbf{x} \oplus \mathbf{b}$. If there is a solution, then $\forall\ i,\ A^i \leq \mathbf{0} \oplus A \oplus \cdots \oplus A^{i-1}$, since the weights of circuits are nonpositive. The order relation on the matrices is also defined by $A \leq B$ if $A \oplus B = B$. So $A^2\mathbf{b},\ A^3\mathbf{b}, \ldots$ are less than $\mathbf{b} \oplus A\mathbf{b}$. Therefore, if $A^i\mathbf{b}$ is greater than $A^{i-1}\mathbf{b}$, we can discontinue the computation of $A^k\mathbf{b}$ for $k > i$. This algorithm is much faster than program 3 as the size of $A$ increases.

```
function xval=maxlin2(A,b),
   [m,n]=size(A); [r,s]=size(b);
   if m==n if r==n if s==1
      check=inf*ones(n,1); temp=b; x=b; boo=1; boolean=0;
      while boo==1, tempy=temp; temp=maxmult(A,tempy);
         for i=1:n x(i,1)=max(x(i,1),temp(i,1)); end
         check=maxmult(A,x);
         for i=1:n check(i,1)=max(check(i,1),b(i,1)); end
         for i=1:n if x(i,1)==check(i,1) boo=0;
            else boo=1; break; end end
         if boo==1 for i=1:n if temp(i,1)<tempy(i,1)
            boolean=0; break; else boolean=1; end end end
         if boolean==1 break; end end
      if boolean==1 disp('There is no solution.')
      else xval=x; end
   else disp('Column size for b must be 1.') end
   else disp('Inner matrix dimensions must agree.') end
   else disp('Matrix A must be square.') end
```

## 2.3 Solving $A\mathbf{x} = \mathbf{b}$

The other class of linear systems for which we can obtain a general result consists of the systems $A\mathbf{x} = \mathbf{b}$. Unfortunately, we cannot guarantee that there exists a solution for any system $A\mathbf{x} = \mathbf{b}$. We need to weaken the notion of 'solution.' A *subsolution* of $A\mathbf{x} = \mathbf{b}$ is an $\mathbf{x}$ which satisfies $A\mathbf{x} \leq \mathbf{b}$.

**Theorem 3** [1]: *Given an $n \times n$ matrix $A$ and an $n$-vector* $\mathbf{b}$ *with entries in* $\bar{\Re}_{\max}$, *the greatest subsolution of $A\mathbf{x} = \mathbf{b}$ exists and is given by*

$$-x_j = \max_i(-b_i + A_{ij}).$$

In other words, $\mathbf{x} = -((-\mathbf{b})^T A)^T = -(A^T(-\mathbf{b}))$.

*Proof:* We have that

$$\begin{aligned} A\mathbf{x} \leq \mathbf{b} &\iff \bigoplus_j A_{ij} x_j \leq b_i, \ \forall\, i \\ &\iff x_j \leq b_i - A_{ij}, \ \forall\, i, j \\ &\iff x_j \leq \min_i(b_i - A_{ij}), \ \forall\, j \\ &\iff -x_j \geq \max_i(-b_i + A_{ij}), \ \forall\, j. \end{aligned}$$

Conversely, it can be checked similarly that the vector $\mathbf{x}$ defined by $-x_j = \max_i(-b_i + A_{ij})$, $\forall\, j$, is a subsolution. Therefore, it is the greatest one. $\square$

As a consequence, in order to attempt to solve the system $A\mathbf{x} = \mathbf{b}$, we may first compute its greatest subsolution and then check by inspection whether it satisfies the equality.

**Example 5:** Let us compute the greatest subsolution of the following equality:

$$\begin{bmatrix} 2 & 3 \\ 4 & 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 6 \\ 7 \end{bmatrix}.$$

By the theorem,

$$-\mathbf{x} = \begin{bmatrix} -6 & -7 \end{bmatrix} \begin{bmatrix} 2 & 3 \\ 4 & 5 \end{bmatrix} = \begin{bmatrix} -3 & -2 \end{bmatrix}^T.$$

Then the greatest subsolution is $(x_1, x_2) = (3, 2)$;

$$\begin{bmatrix} 2 & 3 \\ 4 & 5 \end{bmatrix} \begin{bmatrix} 3 \\ 2 \end{bmatrix} = \begin{bmatrix} 5 \\ 7 \end{bmatrix} \leq \begin{bmatrix} 6 \\ 7 \end{bmatrix}.$$

**Program 5 (Subsolution of $A\mathbf{x} = \mathbf{b}$):** If $A \in (\bar{\Re}_{\max})^{m \times n}$ and $\mathbf{b} \in (\bar{\Re}_{\max})^{n \times 1}$, then the following program computes the greatest subsolution of $A\mathbf{x} = \mathbf{b}$.

```
function x=linear(A,b),
   [m,n]=size(A); [r,s]=size(b);
   if m==n if r==n if s==1 x=-(maxmult(-b',A))';
   else disp('Column size for b must be 1.') end
   else disp('Inner matrix dimensions must agree.') end
   else disp('Matrix A must be square.') end
```

# 3 Eigenvalues and Eigenvectors

## 3.1 Existence and Uniqueness

Given an $n \times n$ matrix $A$ with entries in $\Re_{\max}$, we consider the problem of existence of eigenvalues and eigenvectors, that is, the existence of $\lambda$ and $\mathbf{x} \neq \mathbf{0}$ such that

$$A\mathbf{x} = \lambda\mathbf{x}.$$

In this section the maximum cycle mean will be defined and the existence of eigenvalues and eigenvectors will be proved. The entry $(i, j)$ of $A^* = \mathbf{0} \oplus A \oplus$

$\cdots \oplus A^n \oplus A^{n+1} \oplus \cdots$ denotes the maximum weight of all paths of any length which go from node $j$ to node $i$. The numerical value of $A_{ij}$ equals the weight of the arc from node $j$ to node $i$. If no such arc exists, then $A_{ij} = -\infty$. The meaning of $(A^k)_{ij}$ is the maximum weight with respect to all paths of length $k$ from $j$ to $i$. If no such path exists, then $(A^k)_{ij} = -\infty$. Let $\rho = (i_1, i_2, \ldots, i_k)$ be a path in a weighted graph of $A$. Then the weight of this path, $|\rho|_w$, is the product $A_{i_1 i_2} \otimes A_{i_2 i_3} \otimes \cdots \otimes A_{i_{k-1} i_k}$, and the length of this path, $|\rho|_l$, is $k-1$. Now we have the following definition.

**Definition 6 (Cycle mean):** *The mean weight of a path is defined as the sum of the weights of the individual arcs of this path, divided by the length of this path. If the path is denoted $\rho$, then the mean weight equals $|\rho|_w / |\rho|_l$. If such a path is a circuit one talks about the mean weight of the circuit, or simply the cycle mean.*

We are interested in the maximum of these cycle means, where the maximum is taken over all circuits in the matrix $A$. All circuits with a cycle mean equal to the maximum cycle mean are called critical circuits.

**Theorem 4 [1]:** *If $G(A)$ is strongly connected, there exists one and only one eigenvalue (but possibly several eigenvectors). This eigenvalue is equal to the maximum cycle mean of the graph:*

$$\lambda = \max_{\rho} \frac{|\rho|_w}{|\rho|_l},$$

*where $\rho$ ranges over the set of circuits of $G(A)$.*

*Proof:*

**Existence of x and $\lambda$.** Consider matrix $B = -\lambda A$, where $\lambda = \max_\rho |\rho|_w/|\rho|_l$. The maximum circuit weight of $G(B)$ is 0. Hence, $B^*$ and $B^+ = BB^*$ exist and matrix $B^+$ has some columns with diagonal entries equal to 0. Suppose a node $k$ is in the maximum circuit, then the maximum weight of paths from $k$ to $k$ is 0. Therefore, we have $0 = B^+_{kk}$. Let $B_k$ denote the $k$-th column of $B$. Then, since $B = -\lambda A$, $B^+ = BB^*$ and $B^* = \mathbf{0} \oplus B^+$, for that $k$,

$$B^+_k = B^*_k \Rightarrow BB^*_k = B^+_k = B^*_k \Rightarrow -\lambda AB^*_k = B^*_k \Rightarrow AB^*_k = \lambda B^*_k.$$

Hence $\mathbf{x} = B^*_k = B^+_k$ is an eigenvector of $A$ corresponding to the eigenvalue $\lambda$.

**Uniqueness of $\lambda$.** Let $\lambda_1$ and $\lambda_2$ be eigenvalues. Assume that $\lambda_1 < \lambda_2$. We know that there are $\mathbf{v}_1$ and $\mathbf{v}_2$ with $A\mathbf{v}_1 = \lambda_1\mathbf{v}_1$ and $A\mathbf{v}_2 = \lambda_2\mathbf{v}_2$. Pick $t \in \Re_{\max}$ large enough so that $t\mathbf{v}_1 \geq \mathbf{v}_2$ component by component. Then,

$$t\mathbf{v}_1 \oplus \mathbf{v}_2 = t\mathbf{v}_1,$$

$$A^n(t\mathbf{v}_1 \oplus \mathbf{v}_2) = A^n(t\mathbf{v}_1),$$

$$A^n t\mathbf{v}_1 \oplus A^n\mathbf{v}_2 = A^n t\mathbf{v}_1,$$

$$tA^n\mathbf{v}_1 \oplus A^n\mathbf{v}_2 = tA^n\mathbf{v}_1,$$

$$t\lambda_1^n\mathbf{v}_1 \oplus \lambda_2^n\mathbf{v}_2 = t\lambda_1^n\mathbf{v}_1.$$

For some $n$, $\lambda_2^n\mathbf{v}_2 \not\leq t\lambda_1^n\mathbf{v}_1$, since $\lambda_1 < \lambda_2$. So $t\lambda_1^n\mathbf{v}_1 \oplus \lambda_2^n\mathbf{v}_2 \neq t\lambda_1^n\mathbf{v}_1$. This is a contradiction. Therefore, $\lambda_1 = \lambda_2$. □

## 3.2 Maximum Cycle Mean Method

This section includes computer programs in the Matlab language which will find the eigenvalue of an $n \times n$ matrix $A$. This program is based on the algorithm of finding the maximum cycle mean. It involves numerous flops. Finding eigenvectors this way relates to finding critical circuits and paths with the maximum weight, and this approach is intractable for large $n$.

**Program 6 (Generating the permutation of $\{1, 2, \ldots, n\}$):** If $a$ is an $n$-vector, $a = [1,\ 2, \ldots, n]^T$, then the following program generates a permutation.

```
function permutation=perm(a)
   n=length(a); permutation=a;
   if n>1 j=n-1; while permutation(j) > permutation(j+1),
      j=j-1; end k=n;
      while permutation(j) > permutation(k),
         k=k-1; end
      temp=permutation(j); permutation(j)=permutation(k);
      permutation(k)=temp; r=n; s=j+1;
      while r > s, temp=permutation(r);
         permutation(r)=permutation(s);
         permutation(s)=temp; r=r-1; s=s+1; end
   end
```

**Program 7 (Generating the $r$-combinations of the set $\{1, 2, 3, \ldots, n\}$):** If $a$ is an $n$-vector, $a = [1,\ 2, \ldots, n]^T$, then the following program generates $r$-combinations, for $r = 1, 2, \ldots, n$.

```
function rcombination=rcombi(a),
   n=length(a); rcom=a; cycle=n;
   for i=1:n rcombi(i,1)=i; end
   for r=2:n repeat=n;
      for k=1:r-1 repeat=repeat*(n-k); end
      for k=1:r repeat=repeat/k; end
      for i=1:r rcombi(1+cycle,i)=rcom(i); end
```

```
    cycle=cycle+repeat;
    for l=2+cycle-repeat:cycle i=r;
       while rcom(i)==n-r+i, if i>1 i=i-1; end end
       rcom(i)=rcom(i)+1;
       for j=i+1:r rcom(j)=rcom(i)+j-i; end
       for i=1:r rcombi(l,i)=rcom(i); end
    end
    rcom=a;
  end
  rcombination=rcp(rcombi,n);
```

**Program 8 (Finding an eigenvalue):** If $A \in (\Re_{\max})^{n\times n}$, then the following program computes the eigenvalue of matrix $A$. This program uses program 6 and program 7 is built inside of this program for the sake of time. This program is based on the maximum cycle mean algorithm.

```
function eigenvalue=tmaxeig2(A),
   [m,n]=size(A);
   if m==n eigenvalue=0;
      for i=1:n p(i)=i;
         eigenvalue=max(eigenvalue,A(p(i),p(i))); end
      rcom=p; for r=2:n repeat=n;
      for k=1:r-1 repeat=repeat*(n-k); end
         for k=1:r repeat=repeat/k; end
         for i=1:r rcombi(i)=rcom(i); end
         tempy=0; for i=1:r-1
         tempy=tempy+A(rcombi(i),rcombi(i+1)); end
         tempy=tempy+A(rcombi(r),rcombi(1)); tempy=tempy/r;
         eigenvalue=max(eigenvalue,tempy); factorial=r-1;
         for i=1:r-2 if r-2>0 factorial=factorial*(r-1-i); end end
         for i=1:r-1 rcombitemp(i)=rcombi(i+1); end
         for i=1:factorial-1 rcombitemp=perm(rcombitemp);
            for i=2:r rcombi(i)=rcombitemp(i-1); end
            tempy=0; for i=1:r-1
            tempy=tempy+A(rcombi(i),rcombi(i+1)); end
            tempy=tempy+A(rcombi(r),rcombi(1)); tempy=tempy/r;
            eigenvalue=max(eigenvalue,tempy); end
         for l=2:repeat i=r;
            while rcom(i)==n-r+i,
               if i>1 i=i-1; end end
            rcom(i)=rcom(i)+1;
```

```
            for j=i+1:r rcom(j)=rcom(i)+j-i; end
            for i=1:r rcombi(i)=rcom(i); end
            tempy=0; for i=1:r-1
               tempy=tempy+A(rcombi(i),rcombi(i+1)); end
            tempy=tempy+A(rcombi(r),rcombi(1)); tempy=tempy/r;
            eigenvalue=max(eigenvalue,tempy); factorial=r-1;
            for i=1:r-2
               if r-2>0 factorial=factorial*(r-1-i); end end
            for i=1:r-1 rcombitemp(i)=rcombi(i+1); end
            for i=1:factorial-1 rcombitemp=perm(rcombitemp);
               for i=2:r rcombi(i)=rcombitemp(i-1); end
               tempy=0; for i=1:r-1
                  tempy=tempy+A(rcombi(i),rcombi(i+1)); end
               tempy=tempy+A(rcombi(r),rcombi(1)); tempy=tempy/r;
               eigenvalue=max(eigenvalue,tempy);
            end
         end
      rcom=p;
      end
   else disp('Matrix must be square.') end
```

**Example 6:** Let $A = \begin{bmatrix} -3 & -2 & 8 \\ 1 & 0 & 4 \\ 2 & 3 & -6 \end{bmatrix}$. Then the cycle means of $A$ are

length 1: $A_{11} = -3,\ A_{22} = 0,\ A_{33} = -6$

length 2: $A_{12} = \dfrac{-2+1}{2} = -\dfrac{1}{2},\ A_{13} = \dfrac{8+2}{2} = 5,\ A_{23} = \dfrac{4+3}{2} = \dfrac{7}{2}$

length 3: $A_{13} = \dfrac{-2+4+2}{3} = \dfrac{4}{3},\ A_{12} = \dfrac{8+3+1}{3} = 4.$

So, the eigenvalue is $\max(-3,\ 0,\ -6,\ -\frac{1}{2},\ 5,\ \frac{7}{2},\ \frac{4}{3},\ 4) = 5$. This implies that the critical circuit is $A_{13}$. In order to determine an eigenvector, we need to find $G(B)$ where

$$B = -\lambda A = -5 \begin{bmatrix} -3 & -2 & 8 \\ 1 & 0 & 4 \\ 2 & 3 & -6 \end{bmatrix} = \begin{bmatrix} -8 & -7 & 3 \\ -4 & -5 & -1 \\ -3 & -2 & -11 \end{bmatrix}.$$

Suppose that $\mathbf{x} = [x_1,\ x_2,\ x_3]^T$ is an eigenvector. Since node **1** and **3** determine the critical circuit in Figure 2, we may choose either $x_1$ or $x_3$ to be 0. The

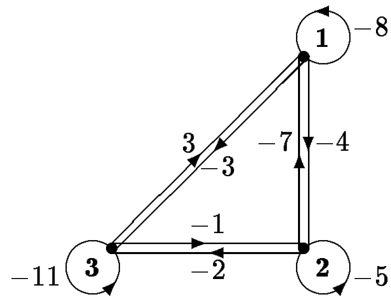

Figure 2: The precedence graph of $B$

maximum weight of a path from node **1** to node **2** is $-4$ and $-3$ from node **1** to node **3** in $G(B)$. So, $\mathbf{x} = B_1^+ = [0,\ -4,\ -3]^T$.

## 3.3 Power Method

Before the discussion of the power method, we need to consider the equation $\mathbf{x}_{t+1} = A\mathbf{x}_t$, where $t = 0, 1, 2, \ldots$ ($t$ can be time or iteration), $\mathbf{x} \in (\Re_{\max})^{n\times 1}$, and $A \in (\Re_{\max})^{n\times n}$. $\mathbf{x}_0$ is called the initial vector. Assume that $\mathbf{x}_0$ is an eigenvector of $A$. Then $\mathbf{x}_1 = A\mathbf{x}_0 = \lambda\mathbf{x}_0$, $\mathbf{x}_2 = A\mathbf{x}_1 = A\lambda\mathbf{x}_0 = \lambda^2\mathbf{x}_0, \ldots$ where $\lambda$ is the corresponding eigenvalue. More generally,

$$\mathbf{x}_t = \lambda^t\mathbf{x}_0,\ t = 0, 1, 2, \ldots. \tag{2}$$

Equation (2) yields $(\mathbf{x}_{t+1})_i - (\mathbf{x}_{t+1})_j = (\mathbf{x}_t)_i - (\mathbf{x}_t)_j$ and $(\mathbf{x}_{t+1})_i = \lambda \otimes (\mathbf{x}_t)_i$ for $i, j = 1, \ldots, n$. Thus the solution of these equations exhibits a kind of periodicity. In particular, $\mathbf{x}_{k+p} = \lambda^p \otimes \mathbf{x}_k$, where $p$ is a period, so $\lambda = \dfrac{(\mathbf{x}_{k+p})_i - (\mathbf{x}_k)_i}{p}$ for any $i = 1, \ldots, n$. Recall that this is the maximum cycle mean of $A$. An eigenvector

$\mathbf{x}$ can be represented as $\mathbf{x}_{k+p-1} \oplus \lambda\mathbf{x}_{k+p-2} \oplus \lambda^2\mathbf{x}_{k+p-3} \oplus \cdots \oplus \lambda^{p-1}\mathbf{x}_k$. Because,

$$\begin{aligned}
& A(\mathbf{x}_{k+p-1} \oplus \lambda\mathbf{x}_{k+p-2} \oplus \lambda^2\mathbf{x}_{k+p-3} \oplus \cdots \oplus \lambda^{p-1}\mathbf{x}_k) \\
= \; & A\mathbf{x}_{k+p-1} \oplus A\lambda\mathbf{x}_{k+p-2} \oplus A\lambda^2\mathbf{x}_{k+p-3} \oplus \cdots \oplus A\lambda^{p-1}\mathbf{x}_k \\
= \; & \mathbf{x}_{k+p} \oplus \lambda\mathbf{x}_{k+p-1} \oplus \lambda^2\mathbf{x}_{k+p-2} \oplus \cdots \oplus \lambda^{p-1}\mathbf{x}_{k+1} \\
= \; & \lambda\mathbf{x}_{k+p-1} \oplus \lambda^2\mathbf{x}_{k+p-2} \oplus \lambda^3\mathbf{x}_{k+p-3} \oplus \cdots \oplus \lambda^{p}\mathbf{x}_k \\
= \; & \lambda(\mathbf{x}_{k+p-1} \oplus \lambda\mathbf{x}_{k+p-2} \oplus \lambda^2\mathbf{x}_{k+p-3} \oplus \cdots \oplus \lambda^{p-1}\mathbf{x}_k).
\end{aligned}$$

**Example 7:** Let $\mathbf{x}_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ and $A = \begin{bmatrix} 3 & 7 \\ 2 & 4 \end{bmatrix}$. Then,

$$\mathbf{x}_1 = \begin{bmatrix} 3 & 7 \\ 2 & 4 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 7 \\ 4 \end{bmatrix},$$

$$\mathbf{x}_2 = \begin{bmatrix} 3 & 7 \\ 2 & 4 \end{bmatrix} \begin{bmatrix} 7 \\ 4 \end{bmatrix} = \begin{bmatrix} 11 \\ 9 \end{bmatrix},$$

$$\mathbf{x}_3 = \begin{bmatrix} 3 & 7 \\ 2 & 4 \end{bmatrix} \begin{bmatrix} 11 \\ 9 \end{bmatrix} = \begin{bmatrix} 16 \\ 13 \end{bmatrix},$$

$$\mathbf{x}_3 - \mathbf{x}_1 = \begin{bmatrix} 9 \\ 9 \end{bmatrix}.$$

This gives the eigenvalue $\lambda = \frac{9}{2} = 4.5$ (2 is the period), and the eigenvector $\begin{bmatrix} 11 \\ 9 \end{bmatrix} \oplus 4.5 \begin{bmatrix} 7 \\ 4 \end{bmatrix} = \begin{bmatrix} 11.5 \\ 9 \end{bmatrix}$.

Because of this periodicity, we call this method a power method.

**Program 9 (Finding eigenvalues and eigenvectors):** If $A \in (\Re_{\max})^{n \times n}$, then the following program computes the eigenvalue and an eigenvector of matrix $A$.

```
function [eigenvalue,eigenvector] = maxeig(A)
   [m,n] = size(A);
   if m==n oldx = zeros(n,1); storx = zeros(n,1);
      compx = zeros(n,1); iteration=2; ok=0;
      while ok==0, newx = maxmult(A,oldx);
```

```
        storx(1:n,iteration)=newx;
        for k=1:(iteration-1) ok=1;
           for i=1:n
              compx(i,k)=storx(i,iteration)-storx(i,iteration-k);
              if compx(1,k)~=compx(i,k) ok=0; break end
           end
           nu=compx(1,k); de=k; index=iteration-k;
           if ok==1 break end
        end
        iteration=iteration+1; oldx = newx;
     end
     eigenvalue=nu/de; eigenvector=storx(1:n,iteration-2);
     for i=1:(iteration-index-2)
        evec=maxmult(storx(1:n,iteration-i-2),i*eigenvalue);
        for j=1:n
           eigenvector(j,1)=max(eigenvector(j,1),evec(j,1));
     end end
     temp=inf;
     for i=1:n temp=min(temp,eigenvector(i,1)); end
     for i=1:n eigenvector(i,1)=eigenvector(i,1)-temp; end
  else disp('Matrix must be square.') end
```

Below is a summary table of flops and iterations of an $n \times n$ matrix, $n = 10, 20, 30, 40, 50$, by the power method. The data represents 100 random matrices for each $n$ with entries chosen from a uniform distribution on the interval $(0, 9)$. From the table, it can be seen that the program runs quickly and the number of iterations is small. The interesting observation is that the data is linear and the number of iterations is smaller when $n$ is bigger.

| | $n$=10 | | $n$=20 | | $n$=30 | | $n$=40 | | $n$=50 | |
|---|---|---|---|---|---|---|---|---|---|---|
| | flops | iter | flops | iter | flops | iter | flops | iter | flops | iter |
| Average | 893.4 | 7.11 | 2984.14 | 7.31 | 5199.68 | 6.19 | 5526.52 | 4.19 | 7801.77 | 3.92 |
| Median | 853 | 7 | 2539.5 | 6.5 | 3983 | 5 | 5200 | 4 | 7969 | 4 |
| Max | 2529 | 16 | 9984 | 21 | 18811 | 19 | 13628 | 9 | 24204 | 10 |
| Min | 289 | 3 | 1391 | 4 | 2029 | 3 | 1852 | 2 | 2812 | 2 |

Table 1: Flops and Iterations by the Power Method

## 3.4 Linear Programing Approach

Let $A \in (\Re_{\max})^{n\times n}$ and $A\mathbf{x} = \lambda\mathbf{x}$, where $\mathbf{x}$ is an eigenvector and $\lambda$ is the eigenvalue. If $A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$ and $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$, then

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \lambda \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}.$$

Using standard notation, this can be rewritten as

$$\max(a_{11} + x_1,\ a_{12} + x_2,\ a_{13} + x_3) = \lambda + x_1,$$

$$\max(a_{21} + x_1,\ a_{22} + x_2,\ a_{23} + x_3) = \lambda + x_2,$$

$$\max(a_{31} + x_1,\ a_{32} + x_2,\ a_{33} + x_3) = \lambda + x_3.$$

Since each entry $\lambda + x_i$ on the right hand side is the maximum of $a_{ij} + x_j$ for $i, j = 1, 2, 3$, it implies that

$$a_{11} + x_1 \leq \lambda + x_1,$$

$$a_{12} + x_2 \leq \lambda + x_1,$$

$$a_{13} + x_3 \leq \lambda + x_1,$$

$$a_{21} + x_1 \leq \lambda + x_2,$$

$$a_{22} + x_2 \leq \lambda + x_2,$$

$$a_{23} + x_3 \leq \lambda + x_2,$$

$$a_{31} + x_1 \leq \lambda + x_3,$$

$$a_{32} + x_2 \leq \lambda + x_3,$$

$$a_{33} + x_3 \leq \lambda + x_3.$$

This implies that $A\mathbf{x} = \lambda\mathbf{x}$ is equivalent to

$$\begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & -1 & 0 & 1 \\ 1 & 0 & -1 & 1 \\ -1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & -1 & 1 \\ -1 & 0 & 1 & 1 \\ 0 & -1 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \lambda \end{bmatrix} \geq \begin{bmatrix} a_{11} \\ a_{12} \\ a_{13} \\ a_{21} \\ a_{22} \\ a_{23} \\ a_{31} \\ a_{32} \\ a_{33} \end{bmatrix}. \qquad (3)$$

Let $H_n$ be an $n^2 \times (n+1)$ matrix in the conventional algebra, then the simplex method solves the linear programming problem

$$\text{Primal} \quad \max\ \mathbf{b}^T\mathbf{y} \text{ subject to } H_n^T\mathbf{y} = \mathbf{c},\ \mathbf{y} \geq 0, \text{ and}$$

$$\text{Dual} \quad \min\ \mathbf{c}^T\mathbf{z} \text{ subject to } H_n\mathbf{z} \geq \mathbf{b}. \qquad (4)$$

Notice that expression (4) is the same as the inequality (3). Theoretically, for any matrix $A \in (\Re_{\max})^{n\times n}$, we can use inequality (3) to find the eigenvalue and an eigenvector of $A$ by linear programming. In general $H_n$ depends on the size of matrix $A$, but not on $A$.

$$H_n = \begin{bmatrix} \mathbf{J}\mathbf{e}_1^T - I & \mathbf{1} \\ \mathbf{J}\mathbf{e}_2^T - I & \mathbf{1} \\ \vdots & \vdots \\ \mathbf{J}\mathbf{e}_n^T - I & \mathbf{1} \end{bmatrix},$$

where $\mathbf{J}$ is an $n \times 1$ vector of 1's, $I$ is an identity matrix in the conventional algebra, and $\mathbf{e}_i$ is the $i$th unit vector, i.e., $\mathbf{e}_i = [0, 0, \ldots, 0, 1, 0, \ldots, 0]^T$ where 1 is in the $i$th position.

We need to minimize $\mathbf{c}^T\mathbf{z}$ so that $H_n\mathbf{z} \geq \mathbf{b}$. Let $\mathbf{c} = [0, 0, \ldots, 0, 1]^T$. Then, $\mathbf{c}^T\mathbf{z} = \lambda$ and the sum of all entries of $H_n\mathbf{z}$ is equal to $n^2\lambda = n^2\mathbf{c}^T\mathbf{z}$.

$$\begin{aligned} H_n\mathbf{z} \geq \mathbf{b} \;\; &\Rightarrow \;\; n^2\mathbf{c}^T\mathbf{z} \geq \sum_{i=1}^{n^2} b_i, \\ &\Rightarrow \;\; \mathbf{c}^T\mathbf{z} \geq \frac{\sum_{i=1}^{n^2} b_i}{n^2}. \end{aligned}$$

So, if we minimize $\mathbf{c}^T\mathbf{z}$ and maximize $\frac{\sum_{i=1}^{n^2} b_i}{n^2}$, we can have an equality of these two expressions. Notice that $\frac{\sum_{i=1}^{n^2} b_i}{n^2}$ is $\mathbf{b}^T\mathbf{y}$, where $\mathbf{y} = [\frac{1}{n^2}, \frac{1}{n^2}, \ldots, \frac{1}{n^2}]^T$, and this $\mathbf{y}$ gives $H_n^T\mathbf{y} = \mathbf{c}$.

Linear programs assume that the rows of $H_n$ in (4) are linearly independent, because a linear dependency among the rows of $H_n$ would lead either to contradictory constraints or to a redundancy that could be eliminated. Therefore, it is crucial for $H_n$ to have a full rank. The matrix $H_n$ in (3) has rank 1. So without loss of generality, $H_n$ needs to be refigured by getting rid of the first column.

The power method finds an eigenvector and the eigenvalue of a matrix $A \in (\Re_{\max})^{n\times n}$. But we do not know whether the iteration converges or not, since the upper bound of the period is unknown. Linear programming guarantees that an eigenvector and the eigenvalue can be found in a polynomial time. Unfortunately the linear programs that I used will find the eigenvalues $\lambda$, but not the eigenvectors. However, they do give a primal solution which is the critical circuit that we use to find the eigenvalue. With this critical circuit, we can find an eigenvector by using the maximum cycle mean method. If the linear programs find an eigenvector, it is the greatest subeigenvector $\mathbf{x}$, i.e., $A\mathbf{x} \leq \lambda\mathbf{x}$.

Here is a table of flops for $B \in (\Re_{\max})^{n\times n}$, $n = 10, 20, 30$, when the eigenvalues are calculated by a linear program. 15 random matrices have been used for each $n$. The flops are higher than the corresponding flops of the power method.

| | $n$=10 flops | $n$=20 flops | $n$=30 flops |
|---|---|---|---|
| Average | 187819.5 | 2672503.7 | 13854109.1 |
| Median | 164130 | 2861503 | 13343509 |
| Max | 229428 | 3267824 | 15260416 |
| Min | 164062 | 2049414 | 11445719 |

Table 2: Flops by a Linear Program

# 4 Characteristic Polynomial and the Cayley-Hamilton Theorem

This chapter defines a characteristic polynomial, characteristic equation and provides an analogue of the Cayley Hamilton Theorem in the max-plus algebra.

**Definition 7:** *An elementary product from* $A \in (\Re_{\max})^{n\times n}$ *is a product of* $n$

*entries of $A$, exactly one from each row and each column. A positive elementary product is an elementary product $A_{1j_1}A_{2j_2}\cdots A_{nj_n}$ where the permutation $(j_1, j_2, \ldots, j_n)$ is even. A negative elementary product is an elementary product $-A_{1j_1}A_{2j_2}\cdots A_{nj_n}$ where the permutation $(j_1, j_2, \ldots, j_n)$ is odd.*

**Definition 8:** *The characteristic polynomial of $A \in (\Re_{\max})^{n\times n}$ is defined by*

$$p_A(x) = \det(xI - A), \tag{5}$$

*where $I$ is the identity matrix. $p_A(x)$ is the sum of all positive and negative elementary products from $xI - A$.*

*The characteristic equation is given by*

$$p_A^+(x) = p_A^-(x), \tag{6}$$

*where the positive determinant $p_A^+(x)$ is the sum of all positive terms from $p_A(x)$ and the negative determinant $p_A^-(x)$ is the sum of all negative terms from $p_A(x)$.*

If we consider the conventional algebra, this characteristic equation coincides with the equation obtained by setting the characteristic polynomial (5) equal to zero. But the crucial feature of (6) is that there are no terms with 'negative' coefficients, since the inverse of $\oplus$ does not exist.

**Example 8:** Let $A = (A_{ij})$ be a $3 \times 3$ matrix with $A_{ij} \in \Re_{\max}$. Then,

$$xI - A = \begin{bmatrix} x - A_{11} & -A_{12} & -A_{13} \\ -A_{21} & x - A_{22} & -A_{23} \\ -A_{31} & -A_{32} & x - A_{33} \end{bmatrix}.$$

The positive elementary products of $xI - A$ are $(x - A_{11})(x - A_{22})(x - A_{33})$, $(-A_{12})(-A_{23})(-A_{31})$, and $(-A_{13})(-A_{21})(-A_{32})$. The negative elementary products of $xI - A$ are $-(x - A_{11})(-A_{23})(-A_{32})$, $-(-A_{12})(-A_{21})(x - A_{33})$, and $-(-A_{13})(x - A_{22})(-A_{31})$. Then each elementary product can be simplified as following:

$$\begin{aligned}
(x - A_{11})(x - A_{22})(x - A_{33}) &= x^3 - (A_{11} + A_{22} + A_{33})x^2 \\
&\quad +(A_{11}A_{22} + A_{11}A_{33} + A_{22}A_{33})x - A_{11}A_{22}A_{33}, \\
(-A_{12})(-A_{23})(-A_{31}) &= -A_{12}A_{23}A_{31}, \\
(-A_{13})(-A_{21})(-A_{32}) &= -A_{13}A_{21}A_{32}, \\
-(x - A_{11})(-A_{23})(-A_{32}) &= -A_{23}A_{32}x + A_{11}A_{23}A_{32}, \\
-(-A_{12})(-A_{21})(x - A_{33}) &= -A_{12}A_{21}x + A_{12}A_{21}A_{33}, \\
-(-A_{13})(x - A_{22})(-A_{31}) &= -A_{13}A_{31}x + A_{13}A_{22}A_{31}.
\end{aligned}$$

Therefore,

$$\begin{aligned}
p_A^+(x) &= x^3 \oplus (A_{11}A_{22} \oplus A_{11}A_{33} \oplus A_{22}A_{33})x \\
&\quad \oplus A_{11}A_{23}A_{32} \oplus A_{12}A_{21}A_{33} \oplus A_{13}A_{22}A_{31}, \\
p_A^-(x) &= (A_{11} \oplus A_{22} \oplus A_{33})x^2 \oplus (A_{12}A_{21} \oplus A_{13}A_{31} \oplus A_{23}A_{32})x \\
&\quad \oplus A_{11}A_{22}A_{33} \oplus A_{12}A_{23}A_{31} \oplus A_{13}A_{21}A_{32},
\end{aligned}$$

where, as usual, the $\otimes$-symbols have been omitted.

**Example 9:** Let $A = (A_{ij})$ be a $4 \times 4$ matrix with $A_{ij} \in \Re_{\max}$. Then,

$$xI - A = \begin{bmatrix} x - A_{11} & -A_{12} & -A_{13} & -A_{14} \\ -A_{21} & x - A_{22} & -A_{23} & -A_{24} \\ -A_{31} & -A_{32} & x - A_{33} & -A_{34} \\ -A_{41} & -A_{42} & -A_{43} & x - A_{44} \end{bmatrix}.$$

By computing the positive and the negative elementary products of $xI - A$, the characteristic equation is $p_A^+(x) = p_A^-(x)$, where

$$\begin{aligned}
p_A^+(x) \quad = \quad & x^4 \oplus (A_{11}A_{22} \oplus A_{11}A_{33} \oplus A_{11}A_{44} \oplus A_{22}A_{33} \oplus A_{22}A_{44} \oplus A_{33}A_{44})x^2 \\
& \oplus(A_{11}A_{23}A_{32} \oplus A_{11}A_{24}A_{42} \oplus A_{11}A_{34}A_{43} \oplus A_{12}A_{21}A_{33} \oplus A_{12}A_{21}A_{44} \\
& \oplus A_{13}A_{22}A_{31} \oplus A_{13}A_{31}A_{44} \oplus A_{14}A_{22}A_{41} \oplus A_{14}A_{33}A_{41} \oplus A_{22}A_{34}A_{43} \\
& \oplus A_{23}A_{32}A_{44} \oplus A_{24}A_{33}A_{42})x \\
& \oplus A_{11}A_{22}A_{33}A_{44} \oplus A_{11}A_{23}A_{34}A_{42} \oplus A_{11}A_{24}A_{32}A_{43} \oplus A_{12}A_{21}A_{34}A_{43} \\
& \oplus A_{12}A_{23}A_{31}A_{44} \oplus A_{12}A_{24}A_{33}A_{41} \oplus A_{13}A_{21}A_{32}A_{44} \oplus A_{13}A_{22}A_{34}A_{41} \\
& \oplus A_{13}A_{24}A_{31}A_{42} \oplus A_{14}A_{21}A_{33}A_{42} \oplus A_{14}A_{22}A_{31}A_{43} \oplus A_{14}A_{23}A_{32}A_{41}, \\
p_A^-(x) \quad = \quad & (A_{11} \oplus A_{22} \oplus A_{33} \oplus A_{44})x^3 \oplus (A_{12}A_{21} \oplus A_{13}A_{31} \oplus A_{14}A_{41} \oplus A_{23}A_{32} \\
& \oplus A_{24}A_{42} \oplus A_{34}A_{43})x^2 \oplus (A_{11}A_{22}A_{33} \oplus A_{11}A_{22}A_{44} \oplus A_{11}A_{33}A_{44} \\
& \oplus A_{12}A_{23}A_{31} \oplus A_{12}A_{24}A_{41} \oplus A_{13}A_{21}A_{32} \oplus A_{13}A_{34}A_{41} \oplus A_{14}A_{21}A_{42} \\
& \oplus A_{14}A_{31}A_{43} \oplus A_{22}A_{33}A_{44} \oplus A_{23}A_{34}A_{42} \oplus A_{24}A_{32}A_{43})x \\
& \oplus A_{11}A_{22}A_{34}A_{43} \oplus A_{11}A_{23}A_{32}A_{44} \oplus A_{11}A_{24}A_{33}A_{42} \oplus A_{12}A_{21}A_{33}A_{44} \\
& \oplus A_{12}A_{23}A_{34}A_{41} \oplus A_{12}A_{24}A_{31}A_{43} \oplus A_{13}A_{21}A_{34}A_{42} \oplus A_{13}A_{22}A_{31}A_{44} \\
& \oplus A_{13}A_{24}A_{32}A_{41} \oplus A_{14}A_{21}A_{32}A_{43} \oplus A_{14}A_{22}A_{33}A_{41} \oplus A_{14}A_{23}A_{31}A_{42}.
\end{aligned}$$

If we consider

$$A = \begin{bmatrix} 1 & 2 & 2 & -6 \\ 0 & 1 & 4 & 2 \\ 0 & 1 & 2 & 5 \\ 3 & 1 & -2 & 1 \end{bmatrix}$$

in the algebraic structure $\Re_{\max}$, then the characteristic equation becomes

$$x^4 \oplus 3x^2 \oplus 6x \oplus 11 = 2x^3 \oplus 5x^2 \oplus 10x \oplus 14,$$

which can be simplified to

$$x^4 = 2x^3 \oplus 5x^2 \oplus 10x \oplus 14, \tag{7}$$

since the omitted terms are dominated by the corresponding terms on the other side of the equality. The equation (7) is equivalent to

$$4x = \max(3x + 2,\ 2x + 5,\ x + 10,\ 14).$$

The geometric interpretation of this equation is shown in Figure 3.

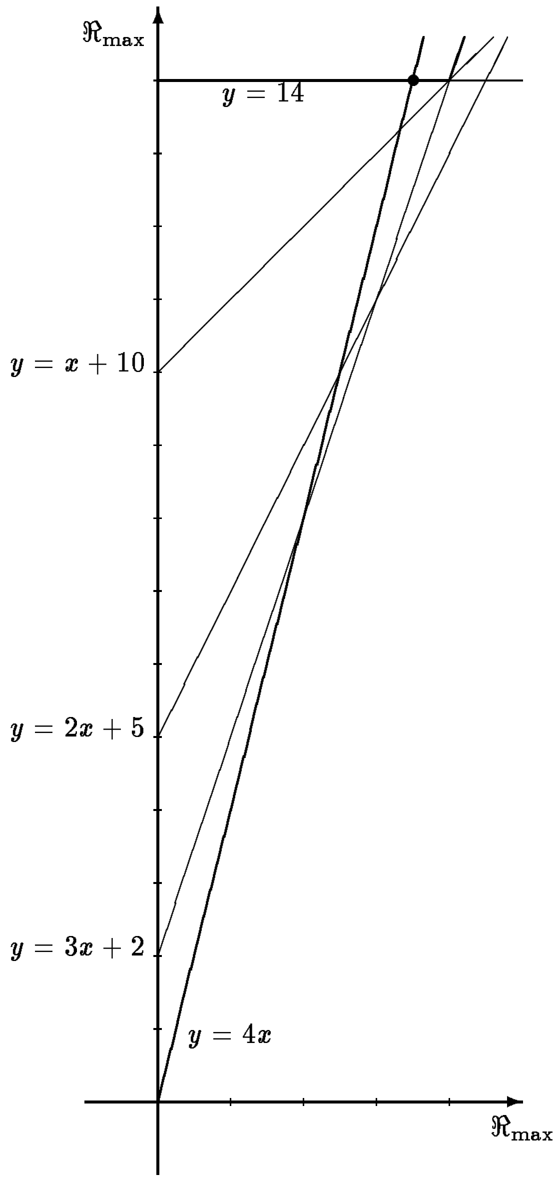


Figure 3: The graph of $x^4 = 2x^3 \oplus 5x^2 \oplus 10x \oplus 14$.

Figure 3 shows that the intersection point of $y = 4x$ and $y = 14$ is higher than the other three intersection points. This implies that $x = 3.5$ is the solution of $x^4 = 2x^3 \oplus 5x^2 \oplus 10x \oplus 14$. Therefore, $\lambda = 3.5$.

**Theorem 5 (The Cayley-Hamilton Theorem [1]):** *Let $p_A^+(x)$ be the positive determinant and $p_A^-(x)$ be the negative determinant of $A \in (\Re_{\max})^{n\times n}$. Then the following identity holds:*

$$p_A^+(A) = p_A^-(A).$$

From example 8, the characteristic equation of $A = \begin{bmatrix} 1 & 2 & 2 & -6 \\ 0 & 1 & 4 & 2 \\ 0 & 1 & 2 & 5 \\ 3 & 1 & -2 & 1 \end{bmatrix}$ is the equation (7). A simple calculation shows that if one substitutes $A$ in the equation (7), one obtains the identity:

$$\begin{bmatrix} 14 & 12 & 12 & 13 \\ 14 & 14 & 14 & 14 \\ 13 & 12 & 14 & 15 \\ 13 & 11 & 11 & 14 \end{bmatrix} = \begin{bmatrix} 12 & 10 & 10 & 13 \\ 14 & 12 & 11 & 13 \\ 12 & 12 & 12 & 12 \\ 8 & 8 & 11 & 12 \end{bmatrix} \oplus \begin{bmatrix} 7 & 8 & 11 & 12 \\ 10 & 10 & 11 & 14 \\ 13 & 11 & 10 & 12 \\ 9 & 10 & 10 & 8 \end{bmatrix}$$
$$\oplus \begin{bmatrix} 11 & 12 & 12 & 4 \\ 10 & 11 & 14 & 12 \\ 10 & 11 & 12 & 15 \\ 13 & 11 & 8 & 11 \end{bmatrix} \oplus \begin{bmatrix} 14 & -\infty & -\infty & -\infty \\ -\infty & 14 & -\infty & -\infty \\ -\infty & -\infty & 14 & -\infty \\ -\infty & -\infty & -\infty & 14 \end{bmatrix}.$$

## 5 References


1. Baccelli, F., Cohen, G., Olsder, G. J., and Quadrat, J. *Synchronization and Linearity.* New York: John Wiley and Sons Ltd., 1992.

2. Fisher, David C. "The 2-packing Number of Complete Grid Graphs." *Ars Combinatoria* **30** (1993): 261-270.