

Totally non-negativity of a family of change-of-basis matrices

David Galvin and Yufei Zhang*

August 29, 2022

Abstract

Let $\mathbf{a} = (a_1, a_2, \dots, a_n)$ and $\mathbf{e} = (e_1, e_2, \dots, e_n)$ be real sequences. Denote by $M_{\mathbf{e} \rightarrow \mathbf{a}}$ the $(n+1) \times (n+1)$ matrix whose (m, k) entry ($m, k \in \{0, \dots, n\}$) is the coefficient of the polynomial $(x - a_1) \cdots (x - a_k)$ in the expansion of $(x - e_1) \cdots (x - e_m)$ as a linear combination of the polynomials $1, x - a_1, \dots, (x - a_1) \cdots (x - a_m)$. By appropriate choice of \mathbf{a} and \mathbf{e} the matrix $M_{\mathbf{e} \rightarrow \mathbf{a}}$ can encode many familiar doubly-indexed combinatorial sequences, such as binomial coefficients, Stirling numbers of both kinds, Lah numbers and central factorial numbers.

In all four of these examples, $M_{\mathbf{e} \rightarrow \mathbf{a}}$ enjoys the property of total non-negativity — the determinants of all its square submatrices are non-negative. This leads to a natural question: when, in general, is $M_{\mathbf{e} \rightarrow \mathbf{a}}$ totally non-negative?

Galvin and Pacurar found a simple condition on \mathbf{e} that characterizes total non-negativity of $M_{\mathbf{e} \rightarrow \mathbf{a}}$ when \mathbf{a} is non-decreasing. Here we fully extend this result. For arbitrary real sequences \mathbf{a} and \mathbf{e} , we give a condition that can be checked in $O(n^2)$ time that determines whether $M_{\mathbf{e} \rightarrow \mathbf{a}}$ is totally non-negative. When $M_{\mathbf{e} \rightarrow \mathbf{a}}$ is totally non-negative, we witness this with a planar network whose weights are non-negative and whose path matrix is $M_{\mathbf{e} \rightarrow \mathbf{a}}$. When it is not, we witness this with an explicit negative minor.

1 Introduction

Let $\mathbf{a} = (a_1, a_2, \dots, a_n)$ and $\mathbf{e} = (e_1, e_2, \dots, e_n)$ be real sequences. Each gives rise to a basis for the space of polynomials in one variable (x , say) of degree at most n , in the following way: set

$$\mathcal{B}_{\mathbf{a}} = \{1, x - a_1, (x - a_1)(x - a_2), \dots, \prod_{i=1}^n (x - a_i)\}$$

and define $\mathcal{B}_{\mathbf{e}}$ analogously.

Associated with $\mathcal{B}_{\mathbf{a}}$ and $\mathcal{B}_{\mathbf{e}}$ there is an $(n+1) \times (n+1)$ change-of-basis matrix $M_{\mathbf{e} \rightarrow \mathbf{a}}$, whose (m, k) entry ($0 \leq m \leq n, 0 \leq k \leq n$) is the coefficient of the polynomial $(x - a_1) \cdots (x - a_k)$

*Department of Mathematics, University of Notre Dame, Notre Dame IN; dgalvin1, yzhang43@nd.edu. Galvin in part supported by a Simons Collaboration Grant for Mathematicians.

in the expansion of $(x - e_1) \cdots (x - e_m)$ as a linear combination of the polynomials in $\mathcal{B}_{\mathbf{a}}$. (We adopt the standard convention here that the empty product evaluates to 1).

Evidently $M_{\mathbf{e} \rightarrow \mathbf{a}}$ is lower-triangular, with 1's down the main diagonal. There are numerous combinatorial matrices that are instances of $M_{\mathbf{e} \rightarrow \mathbf{a}}$, for suitable choices of \mathbf{a} and \mathbf{e} . We present some examples now (see [10, Section 1] for the mostly standard justifications).

1. Taking $e_i = -1$ and $a_i = 0$ for all i yields $M_{\mathbf{e} \rightarrow \mathbf{a}} = \left[\binom{m}{k} \right]$, the matrix of binomial coefficients.
2. Taking $e_i = 0$ and $a_i = i - 1$ for all i yields $M_{\mathbf{e} \rightarrow \mathbf{a}} = \left[\left\{ \begin{matrix} m \\ k \end{matrix} \right\} \right]$, the matrix of Stirling numbers of the second kind.
3. Taking $e_i = -(i - 1)$ and $a_i = 0$ for all i yields $M_{\mathbf{e} \rightarrow \mathbf{a}} = \left[\left\{ \begin{matrix} m \\ k \end{matrix} \right\} \right]$, the matrix of (unsigned) Stirling numbers of the first kind.
4. Taking $e_i = -(i - 1)$ and $a_i = i - 1$ for all i yields $M_{\mathbf{e} \rightarrow \mathbf{a}} = [L(m, k)]$, the matrix of Lah numbers.
5. Taking $e_i = i - 1 - b_i$ and $a_i = i - 1$, the (m, k) entry of $M_{\mathbf{e} \rightarrow \mathbf{a}}$ is the Rook number $R_{m-k}(B_m)$, where b_1, b_2, \dots is any non-decreasing sequence of non-negative integers, B_m is the Ferrers board with m columns that has b_i cells in column i , and $R_k(B_m)$ is the number of ways of placing k non-attacking rooks on B_m .

A matrix is *totally non-negative* (TNN) if all its minors (determinants of square submatrices) are non-negative. Totally non-negative matrices occur frequently in combinatorics and have been the subject of much investigation. See for example [5, 6, 9, 11, 15] for an overview. In particular, all of the matrices listed above are known to be totally non-negative. Motivated by the first four of the examples above, in [10] Galvin and Pacurar asked the following question.

Question 1.1. For which pairs of real sequences (\mathbf{a}, \mathbf{e}) is $M_{\mathbf{e} \rightarrow \mathbf{a}}$ TNN?

A partial answer was given in [10]. Specifically, a characterization was given of those \mathbf{e} -sequences for which $M_{\mathbf{e} \rightarrow \mathbf{a}}$ is TNN, when \mathbf{a} is weakly increasing. We now briefly describe that characterization, which depends on a notion of *restricted growth* (introduced, as far as we are aware, by Arndt [4]).

Definition 1.2. For weakly increasing \mathbf{a} , say that \mathbf{e} is a *restricted growth sequence* relative to \mathbf{a} if for each $i \geq 1$ it holds that $e_i \leq a_{f(i)}$, where $f(1) = 1$ and for $i \geq 1$

$$f(i+1) = \begin{cases} f(i) & \text{if } e_i < a_{f(i)} \\ f(i) + 1 & \text{if } e_i = a_{f(i)}. \end{cases}$$

In other words, each e_i is at most a certain cap. The cap for e_1 is a_1 . If $e_1 < a_1$ then the cap for e_2 is also a_1 , while if $e_1 = a_1$ then the cap for e_2 is a_2 . In general, the cap for e_i is some $a_{i'}$, and if $e_i < a_{i'}$ then the cap for e_{i+1} is also $a_{i'}$, while if $e_i = a_{i'}$ then the cap for e_{i+1} is $a_{i'+1}$. If $\mathbf{a} = (0, 1, \dots, n-1, \dots)$ then a non-negative integer sequence \mathbf{e} is a restricted growth sequence relative to \mathbf{a} exactly if it is a restricted growth sequence in the usual sense, that is, one satisfying $e_1 = 0$ and $e_{i+1} \leq 1 + \max_{j=1, \dots, i} e_j$ for $i \geq 1$.

Theorem 1.3. ([10, Theorem 1.2]) Let \mathbf{a} be a weakly increasing sequence, and let \mathbf{e} be an arbitrary sequence. Then

1. the matrix $M_{\mathbf{e} \rightarrow \mathbf{a}}$ is TNN if and only if \mathbf{e} is a restricted growth sequence relative to \mathbf{a} , and
2. if \mathbf{e} is not a restricted growth sequence relative to \mathbf{a} then the failure of $M_{\mathbf{e} \rightarrow \mathbf{a}}$ to be TNN is witnessed by a negative entry in $M_{\mathbf{e} \rightarrow \mathbf{a}}$.

Galvin and Pacurar left open the major part of Question 1.1, that of determining when $M_{\mathbf{e} \rightarrow \mathbf{a}}$ is TNN when no restriction is placed on \mathbf{a} (or \mathbf{e}). In this note, we completely resolve this question. Our resolution involves extending the notion of restricted growth to the case when \mathbf{a} is arbitrary, via the following algorithm.

Algorithm 1.4. The input is a pair of real sequences $\mathbf{a} = (a_1, \dots, a_n)$ and $\mathbf{e} = (e_1, \dots, e_n)$.

1. Initialize $X = (e_1, \dots, e_n)$ (an ordered list), and $i = 1$.
2. While $i \leq n$, locate the first element, e' say, of X that is greater than or equal to a_i .
 - (a) If there is no such element, then delete the last element of X , and increment i by 1.
 - (b) If $e' = a_i$, then delete e' from X , and increment i by 1.
 - (c) If $e' > a_i$, then STOP and report the current value of i .
3. When $i = n + 1$, STOP and report the value $n + 1$.

Definition 1.5. For arbitrary real sequences \mathbf{a} and \mathbf{e} , say that \mathbf{e} is a *restricted growth sequence* relative to \mathbf{a} if Algorithm 1.4 (on input \mathbf{a} and \mathbf{e}) terminates with the report $n + 1$.

Example 1.6. With $\mathbf{a} = (3, 8, 7, 5, 2, 7)$ and $\mathbf{e} = (2, 1, 3, 7, 3, 4)$ (so $n = 6$), Algorithm 1.4 proceeds as follows:

- Initially $X = (2, 1, 3, 7, 3, 4)$, $i = 1$ and $a_1 = 3$. The first element of X which is greater than or equal to 3 is the third element, which has value 3, so $e' = 3$.
- Since $e' = a_1 = 3$, we update X to $(2, 1, 7, 3, 4)$ and set $i = 2$.
- Since there is no e' (in updated X) with $e' \geq a_2 = 8$, we update X to $(2, 1, 7, 3)$ and set $i = 3$.
- Now $e' = 7$, so we update X to $(2, 1, 3)$ and set $i = 4$.
- No element of X is greater than or equal to $a_4 = 5$, so X becomes $(2, 1)$ and i becomes 5.
- Now $e' = 2$, X updates to (1) and i to 6.
- In this final iteration there is no valid e' so X updates to the empty list, i to 7 ($= n + 1$), and the algorithm terminates with report 7. It follows that $(2, 1, 3, 7, 3, 4)$ is a restricted growth sequence relative to $(3, 8, 7, 5, 2, 7)$.

Example 1.7. With $\mathbf{a} = (11, 8, 3, 1)$ and $\mathbf{e} = (10, 9, 2, 1)$ (so $n = 4$), Algorithm 1.4 proceeds as follows:

- Initially $X = (10, 9, 2, 1)$ and $i = 1$; there is no e' with $e' \geq a_1$, so X updates to $(10, 9, 2)$.
- At $i = 2$ ($a_2 = 8$) we have $e' = 10 > 8$, so the algorithm terminates with report 2 ($< n + 1$). It follows that $(10, 9, 2, 1)$ is not a restricted growth sequence relative to $(11, 8, 3, 1)$.

It is easy to check that if \mathbf{a} is weakly increasing then the notions of restricted growth given in Definitions 1.2 and 1.5 coincide, so that the following theorem, the main point of this note, is an extension of item 1 of Theorem 1.3.

Theorem 1.8. *Let \mathbf{a} and \mathbf{e} be arbitrary real sequences of length n . The matrix $M_{\mathbf{e} \rightarrow \mathbf{a}}$ is TNN if and only if \mathbf{e} is a restricted growth sequence relative to \mathbf{a} .*

Example 1.9. (Continuing Example 1.6) With $\mathbf{a} = (3, 8, 7, 5, 2, 7)$ and $\mathbf{e} = (2, 1, 3, 7, 3, 4)$ we have, by Example 1.6 and Theorem 1.8, that

$$M_{\mathbf{e} \rightarrow \mathbf{a}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 2 & 8 & 1 & 0 & 0 & 0 & 0 \\ 0 & 42 & 12 & 1 & 0 & 0 & 0 \\ 0 & 42 & 42 & 10 & 1 & 0 & 0 \\ 0 & 210 & 210 & 62 & 9 & 1 & 0 \\ 0 & 840 & 840 & 272 & 44 & 12 & 1 \end{bmatrix}$$

is TNN (and this may be readily verified directly).

Example 1.10. (Continuing Example 1.7) With $\mathbf{a} = (11, 8, 3, 1)$ and $\mathbf{e} = (10, 9, 2, 1)$ we have that

$$M_{\mathbf{e} \rightarrow \mathbf{a}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 2 & 0 & 1 & 0 & 0 \\ 18 & 2 & 1 & 1 & 0 \\ 180 & 32 & 4 & 1 & 1 \end{bmatrix}$$

is not TNN. This may be readily verified directly, for example by noting that the minor corresponding to rows 1 and 2, columns 0 and 1 is -2 (recall that our indexing of rows and columns starts at 0). Note that this is an example of a non-TNN matrix of the form $M_{\mathbf{e} \rightarrow \mathbf{a}}$ where the failure to be TNN is not witnessed by a negative entry in the matrix; in the setting of [10], where \mathbf{a} is weakly increasing, such an example cannot occur.

Some remarks are in order. Firstly, note that an $(n + 1) \times (n + 1)$ matrix has $\Theta(4^n / \sqrt{n})$ square submatrices and that calculating the determinant of a $k \times k$ matrix requires $\Omega(k^2)$ operations. So the obvious naive algorithm for determining whether such a matrix is TNN requires $\Omega(n^{3/2}4^n)$ operations. Ando [3, Corollary 2.2] established that to test the total non-negativity of a lower-triangular matrix it suffices to test the $\Theta(n2^n)$ square submatrices

whose columns are consecutive and include the first column. Since a proportion $1 - o(1)$ of these submatrices have dimension $\Theta(n)$, using Ando's criterion still requires $\Omega(2^n)$ operations. Standing in contrast to this, Theorem 1.8 allows one to check whether $M_{\mathbf{e} \rightarrow \mathbf{a}}$ is TNN using $O(n^2)$ arithmetic operations.

Our next remark is that, unsurprisingly, the proof of Theorem 1.8 involves the theory of planar networks, and crucially uses the pivot operation of [10]. We postpone a detailed discussion of this to Section 2. For the moment, let us note that our proof passes through an intermediate algorithm that takes as input arbitrary real sequences \mathbf{a} and \mathbf{e} of length n , and outputs either

- a planar network with all non-negative weights whose path matrix is $M_{\mathbf{e} \rightarrow \mathbf{a}}$, witnessing that $M_{\mathbf{e} \rightarrow \mathbf{a}}$ is TNN

or

- a square submatrix of $M_{\mathbf{e} \rightarrow \mathbf{a}}$ (with consecutive rows and columns) whose determinant is negative, witnessing that $M_{\mathbf{e} \rightarrow \mathbf{a}}$ is not TNN.

(See Algorithm 2.4 and Claim 2.5). We will see that when $M_{\mathbf{e} \rightarrow \mathbf{a}}$ is not TNN, the witnessing negative minor can be produced in $O(n^2)$ steps.

Our final remark is that there are various alternate ways of describing the entries of the matrix $M_{\mathbf{e} \rightarrow \mathbf{a}}$; these were all observed in [10]. Denote by $M^{\mathbf{a}, \mathbf{e}}(m, k)$ the (m, k) entry of the matrix $M_{\mathbf{e} \rightarrow \mathbf{a}}$.

1. The numbers $M^{\mathbf{a}, \mathbf{e}}(m, k)$ satisfy the recurrence

$$M^{\mathbf{a}, \mathbf{e}}(m, k) = M^{\mathbf{a}, \mathbf{e}}(m-1, k-1) + (a_{k+1} - e_m)M^{\mathbf{a}, \mathbf{e}}(m-1, k) \quad \text{for } m, k > 0 \quad (1)$$

with initial conditions $M^{\mathbf{a}, \mathbf{e}}(0, 0) = 1$, $M^{\mathbf{a}, \mathbf{e}}(0, k) = 0$ for $k > 0$ and $M^{\mathbf{a}, \mathbf{e}}(m, 0) = \prod_{i=1}^m (a_1 - e_i)$ for $m > 0$. Various forms of this recurrence have appeared in the literature. As observed in [12], with suitable choices of \mathbf{a} and \mathbf{e} the recurrence (1) can encode

- some generalizations of the classical rook numbers [7],
- the normal order coefficients of the word $(VU)^n$ in the Weyl algebra generated by symbols V, U satisfying $UV - VU = hV^s$ [7],
- Hsu and Shiue's generalized Stirling numbers [13],
- the Jacobi-Stirling numbers (coefficients of the Jacobi differential operator) [7, 8],

as well as encoding Binomial coefficients, Stirling numbers of both kinds, Lah numbers and rook numbers. Also, with $e_m = 0$ for all m and $a_k = (k-1)^2$ for all k , (1) encodes the central factorial numbers of Riordan and Carlitz (see e.g. [2]).

2. We have the explicit expression

$$M^{\mathbf{a}, \mathbf{e}}(m, k) = \sum \prod_{i=1}^{m-k} (a_{s_i - i + 1} - e_{s_i})$$

where the sum is over all $S = \{s_1, \dots, s_{m-k}\} \subseteq \{1, \dots, m\}$ with $s_1 < \dots < s_{m-k}$.

3. We also have the explicit expression

$$M^{\mathbf{a}, \mathbf{e}}(m, k) = \sum_{\ell=0}^{m-k} (-1)^\ell h_{m-k-\ell}(a_1, \dots, a_{k+1}) s_\ell(e_1, \dots, e_m)$$

where $h_\ell(x_1, \dots, x_t)$ is the degree ℓ complete symmetric polynomial in x_1, \dots, x_t and $s_\ell(x_1, \dots, x_t)$ is the degree ℓ elementary symmetric polynomial in x_1, \dots, x_t .

In Section 2.1 we review the necessary background material from the theory of planar networks, and describe the pivot operation from [10]. In Section 2.2 we give the intermediate planar network algorithm. The proof that this algorithm is correct is given in Section 3.1, while the derivation of Theorem 1.8 from all that has preceded appears in Section 3.2.

2 Planar networks and the intermediate algorithm

2.1 Setup

We require some well-known results from the theory of totally non-negative matrices. (Note that much of what follows is taken from [10, Section 3]).

A *planar network* P is a directed acyclic planar graph with a subset $\{s_i : i = 0, 1, 2, \dots\}$ of vertices designated as sources and a subset $\{t_i : i = 0, 1, 2, \dots\}$ of vertices designated as sinks. We assume throughout that the sources and sinks are arranged around the outer face of the graph, and are ordered cyclically (clockwise) as $\dots, s_2, s_1, s_0, t_0, t_1, t_2, \dots$. A *weighted planar network* (P, w) is a planar network P together with a function $w : \vec{E}(P) \rightarrow \mathbb{R}$, which we think of as an assignment of weights to the edges of P . Figure 1 shows a particular weighted planar network. The vertices are the sources, the sinks, and the intersection points between the vertical and horizontal lines. All horizontal lines are oriented to the right and all vertical lines are oriented upward. All horizontal edge weights are 1, while the weights of the vertical edges are given by the x_{ij} 's.

The weighted planar network (G, w) shown in Figure 1 is the only one that we will consider in the sequel (with various choices of w). In figures we will use the compact array representation shown in Figure 2.

We will say that the edge of the planar network that is given weight x_{mk} in Figure 2 is the $[m, k]$ *edge* of the array (using square brackets to distinguish this from an entry in a matrix). For each $k = 1, \dots, n$, the collection of $[m, k]$ edges as m ranges from k to n is the k -th *column* of the planar network (so, for example, the set of edges with weights x_{33}, \dots, x_{n3} is the third column of the array in Figure 2).

The *path matrix* $M(P, w)$ of a weighted planar network (P, w) is the $(n+1) \times (n+1)$ matrix whose (m, k) entry (with rows and columns indexed by $\{0, 1, 2, \dots, n\}$) is the sum of the weights of all the directed paths from s_m to t_k , where the weight of any such path is the product, over all edges traversed, of the weight of the edge. For example, the $(3, 1)$ entry of the path matrix of the weighted planar network shown in Figure 1 is $x_{31}x_{21} + x_{31}x_{22} + x_{32}x_{22}$. Notice that the path matrix of the weighted planar network shown in Figure 1 is lower-triangular with 1's down the main diagonal. The following [14] (see also, for example, [1, Chapter 29], [15]) is a standard result in the theory of totally non-negative matrices.

$$\begin{array}{cccccccc}
a_1 - e_1 & & & & & & & \\
a_1 - e_2 & a_2 - e_1 & & & & & & \\
a_1 - e_3 & a_2 - e_2 & a_3 - e_1 & & & & & \\
a_1 - e_4 & a_2 - e_3 & a_3 - e_2 & a_4 - e_1 & & & & \\
a_1 - e_5 & a_2 - e_4 & a_3 - e_3 & a_4 - e_2 & a_5 - e_1 & & & \\
\vdots & \vdots & \vdots & \vdots & & & \ddots & \\
a_1 - e_n & a_2 - e_{n-1} & a_3 - e_{n-2} & a_4 - e_{n-3} & \cdots & a_{n-1} - e_2 & a_n - e_1 &
\end{array}$$

Figure 3: The array $A(\mathbf{a}, \mathbf{e})$.

$$\begin{array}{cccccccc}
3 - 2 & & & & & & & 1 \\
3 - 1 & 8 - 2 & & & & & & 2 \ 6 \\
3 - 3 & 8 - 1 & 7 - 2 & & & & & 0 \ 7 \ 5 \\
3 - 7 & 8 - 3 & 7 - 1 & 5 - 2 & & & & -4 \ 5 \ 6 \ 3 \\
3 - 3 & 8 - 7 & 7 - 3 & 5 - 1 & 2 - 2 & & & 0 \ 1 \ 4 \ 4 \ 0 \\
3 - 4 & 8 - 3 & 7 - 7 & 5 - 3 & 2 - 1 & 7 - 2 & & -1 \ 5 \ 0 \ 2 \ 1 \ 5
\end{array}$$

Figure 4: The array $A(\mathbf{a}, \mathbf{e})$ in the specific case $\mathbf{a} = (3, 8, 7, 5, 2, 7)$, $\mathbf{e} = (2, 1, 3, 7, 3, 4)$ (Example 1.6).

If it happens that all weights in $A(\mathbf{a}, \mathbf{e})$ are non-negative, then the total non-negativity of $M_{\mathbf{e} \rightarrow \mathbf{a}}$ follows immediately from Lemmas 2.1 and Lemma 2.2. If $A(\mathbf{a}, \mathbf{e})$ has some negative weights, then we will give a procedure that modifies the array by iteratively replacing negative weights with 0 weights, while not changing the path matrix of the array. We will show that this procedure succeeds in replacing all negative weights with 0 weights exactly when $M_{\mathbf{e} \rightarrow \mathbf{a}}$ is totally non-negative. To describe the procedure we need ([10, Lemma 3.9]), whose statement requires a little more notation. For an array A of the kind shown in Figure 2, we refer to the triangle consisting of the $[m + \ell_1, k + \ell_2]$ edges for $0 \leq \ell_1 \leq n - m$ and $0 \leq \ell_2 \leq \ell_1$ as the triangle *headed* at the $[m, k]$ edge (see Figure 5).

$$\begin{array}{cccccc}
[1, 1] & & & & & \\
[2, 1] & [2, 2] & & & & \\
[3, 1] & \mathbf{[3, 2]} & [3, 3] & & & \\
[4, 1] & \mathbf{[4, 2]} & \mathbf{[4, 3]} & [4, 4] & & \\
[5, 1] & \mathbf{[5, 2]} & \mathbf{[5, 3]} & \mathbf{[5, 4]} & [5, 5] & \\
\vdots & \vdots & & & \ddots & \ddots \\
[n, 1] & \mathbf{[n, 2]} & \mathbf{[n, 3]} & \mathbf{[n, 4]} & \cdots & \mathbf{[n, n-1]} \ [n, n]
\end{array}$$

Figure 5: The triangle headed at the $[3, 2]$ position (**bolded** entries).

Suppose that the triangle headed at the $[m, k]$ edge of A is an instance of $A(\mathbf{f}, \mathbf{g})$ for some sequences $\mathbf{f} = (f_1, \dots, f_{n-m+1})$, $\mathbf{g} = (g_1, \dots, g_{n-m+1})$. Denote by $A^p(m, k)$ the array

obtained from A by the following operations:

- Leave unchanged all weights of A that are on edges not in the triangle headed at the $[m, k]$ edge.
- Also leave unchanged the weight $(f_1 - g_1)$ on the $[m, k]$ edge. Replace the weights on the $[m + 1, k]$ and $[m + 1, k + 1]$ edges ($f_1 - g_2$ and $f_2 - g_1$, respectively), with $f_1 - g_1$ and $f_2 - g_2$, respectively.
- More generally, replace the weights on the $[m + \ell, k], [m + \ell, k + 1], [m + \ell, k + 2], \dots, [m + \ell, k + \ell]$ edges ($f_1 - g_{\ell+1}, f_2 - g_\ell, \dots, f_{\ell+1} - g_1$, respectively), with $f_1 - g_1, f_2 - g_{\ell+1}, \dots, f_{\ell+1} - g_2$, respectively, for $1 \leq \ell \leq n - m$.

We refer to $A^P(m, k)$ as the array obtained from A by *pivoting* on the $[m, k]$ edge. Note that in each row of the triangle headed at the $[m, k]$ edge, $A^P(\mathbf{f}, \mathbf{g})$ is obtained from A by moving the $-f_1$'s from the last position in the row to the first, and then shifting all other $-f_j$'s in the row one place to the right. Note also that if $f_1 = g_1$, then after pivoting the weights on the $[m, k']$ edges, for all $k \geq k'$, are 0.

$$\begin{array}{cccccccc}
3 - 2 & & & & & & & 1 \\
3 - 1 & 8 - 2 & & & & & & 2 \ 6 \\
\mathbf{3 - 3} & 8 - 1 & 7 - 2 & & & & & 0 \ 7 \ 5 \\
\mathbf{3 - 3} & \mathbf{8 - 7} & 7 - 1 & 5 - 2 & & & & 0 \ 1 \ 6 \ 3 \\
\mathbf{3 - 3} & \mathbf{8 - 3} & \mathbf{7 - 7} & 5 - 1 & 2 - 2 & & & 0 \ 5 \ 0 \ 4 \ 0 \\
\mathbf{3 - 3} & \mathbf{8 - 4} & \mathbf{7 - 3} & \mathbf{5 - 7} & 2 - 1 & 7 - 2 & & 0 \ 4 \ 4 \ -2 \ 1 \ 5
\end{array}
\quad \text{or}$$

Figure 6: The array from Figure 4 (Example 1.6), after pivoting on the $[3, 1]$ edge (**bolded**). The edges of the triangle headed at the $[3, 1]$ edge (the only edges whose weights might change by pivoting) are underlined.

$$\begin{array}{cccccccc}
3 - 2 & & & & & & & 1 \\
3 - 1 & 8 - 2 & & & & & & 2 \ 6 \\
3 - 3 & 8 - 1 & 7 - 2 & & & & & 0 \ 7 \ 5 \\
3 - 3 & 8 - 7 & 7 - 1 & 5 - 2 & & & & 0 \ 1 \ 6 \ 3 \\
3 - 3 & 8 - 3 & \mathbf{7 - 7} & 5 - 1 & 2 - 2 & & & 0 \ 5 \ 0 \ 4 \ 0 \\
3 - 3 & 8 - 4 & \mathbf{7 - 7} & \mathbf{5 - 3} & 2 - 1 & 7 - 2 & & 0 \ 4 \ 0 \ 2 \ 1 \ 5
\end{array}
\quad \text{or}$$

Figure 7: The array from Figure 6 (Example 1.6), after pivoting on the 0 at the $[5, 3]$ position (**bolded**).

Lemma 2.3. ([10, Lemma 3.9]) *For an array A of the kind shown in Figure 2, suppose that the triangle headed at the $[m, k]$ edge of A is an instance of $A(\mathbf{f}, \mathbf{g})$ for some sequence $\mathbf{f} = (f_1, \dots, f_{n-m+1})$, $\mathbf{g} = (g_1, \dots, g_{n-m+1})$.*

If $f_1 = g_1$ then $M(A^P(m, k)) = M(A)$ (that is, the path matrix is unchanged after pivoting on the $[m, k]$ edge, as long as the entry on the $[m, k]$ edge is 0).

It should be noted that when Lemma 2.3 was stated in [10], some assumptions were made about the weights in A on edges other than those edges in the triangle headed at the $[m, k]$ edge. However, those assumptions were not actually used in the proof, so the proof goes through unchanged in the setting of Lemma 2.3.

2.2 The intermediate algorithm

We are now in a position to describe the intermediate planar network algorithm alluded to in the introduction.

Algorithm 2.4. The input is a pair of real sequences $\mathbf{a} = (a_1, \dots, a_n)$ and $\mathbf{e} = (e_1, \dots, e_n)$. The algorithm proceeds by modifying the weighted planar network $A(\mathbf{a}, \mathbf{e})$ (see Figure 3).

- Initially, set $A = A(\mathbf{a}, \mathbf{e})$ and $k = 1$.
- As long as $k \leq n$, scan the weights in the k th column of A , from the $[k, k]$ edge down.
 - If no zero or negative weights are encountered in the column, leave A unchanged and increase k by 1.
 - If the first non-positive weight that is encountered is negative then STOP, output the current A and report that $M_{\mathbf{e} \rightarrow \mathbf{a}}$ is not TNN.
 - If the first non-positive weight that is encountered is zero, say on the $[m, k]$ edge:
 - * if the triangle headed at the $[m, k]$ edge of A is an instance of $A(\mathbf{f}, \mathbf{g})$ for some sequences $\mathbf{f} = (f_1, \dots, f_{n-m+1})$, $\mathbf{g} = (g_1, \dots, g_{n-m+1})$, then update A by pivoting on the $[m, k]$ position — that is, replace A by $A^p(m, k)$ — and increase k by 1. (See Figure 3 for an illustration of an instance of $A(\mathbf{f}, \mathbf{g})$.)
 - * if not, then STOP and report that the algorithm is inconclusive. (We will see in the analysis that this clause of the algorithm will never be invoked).
- If $k = n + 1$, then STOP, output the current A and report that $M_{\mathbf{e} \rightarrow \mathbf{a}}$ is TNN.

For the input $\mathbf{a} = (3, 8, 7, 5, 2, 7)$, $\mathbf{e} = (2, 1, 3, 7, 3, 4)$ (Example 1.6), the evolution of the planar network in Algorithm 2.4 is illustrated in Figures 4, 6 and 7. In this case the algorithm reports that $M_{\mathbf{e} \rightarrow \mathbf{a}}$ is TNN, and outputs the final network that is illustrated in Figure 7. For the input $\mathbf{a} = (11, 8, 3, 1)$ and $\mathbf{e} = (10, 9, 2, 1)$ (Example 1.7), Algorithm 2.4 terminates at $k = 2$ (without having done a pivot), and in this case the algorithm reports that $M_{\mathbf{e} \rightarrow \mathbf{a}}$ is not TNN. The final network that the algorithm outputs is illustrated in Figure 8.

$$\begin{array}{cccccc}
11 - 10 & & & & & 1 \\
11 - 9 & 8 - 10 & & & & 2 - 2 \\
11 - 2 & 8 - 9 & 3 - 10 & & \text{or} & 9 - 1 - 7 \\
11 - 1 & 8 - 2 & 3 - 9 & 1 - 10 & & 10 \quad 7 \quad -6 \quad -9
\end{array}$$

Figure 8: The final network in Algorithm 2.4 when $\mathbf{a} = (11, 8, 3, 1)$ and $\mathbf{e} = (10, 9, 2, 1)$ (Example 1.7).

Claim 2.5. For every input \mathbf{a} , \mathbf{e} :

1. Algorithm 2.4 terminates after at most $O(n^3)$ operations.
2. It never terminates with an inconclusive report.
3. If it terminates with the report that $M_{\mathbf{e} \rightarrow \mathbf{a}}$ is TNN, then the outputted weighted planar network A has all non-negative entries and has path matrix $M_{\mathbf{e} \rightarrow \mathbf{a}}$ (and so serves as a witness that $M_{\mathbf{e} \rightarrow \mathbf{a}}$ is TNN).
4. If it terminates with the report that $M_{\mathbf{e} \rightarrow \mathbf{a}}$ is not TNN, then there is an $O(n^2)$ algorithm that identifies, in the outputted weighted planar network A , a collection S of consecutive sources and a collection T of consecutive sinks with $|S| = |T|$, such that all collections of vertex disjoint paths from S to N have non-positive weight, and at least one has negative weight (which serves, via Lemma 2.1 to witness that $M_{\mathbf{e} \rightarrow \mathbf{a}}$ is not TNN).

3 Proofs of Claim 2.5 and Theorem 1.8

3.1 Proof of Claim 2.5

3.1.1 Item 1

Constructing the initial weighted planar network A requires $O(n^2)$ operations. The algorithm requires scanning $O(n^2)$ entries. At most $O(n)$ of these scans require performing a pivot, and each pivot requires $O(n^2)$ recalculations of weights. So in total, the algorithm requires $O(n^3)$ operations.

3.1.2 Item 2

This is an immediate corollary of the following straightforward observation:

Observation 3.1. Suppose that at some point in the algorithm the triangle headed at the $[k, k]$ edge is of the form $A(\mathbf{f}, \mathbf{g})$ for some sequences $\mathbf{f} = (f_k, \dots, f_n)$, $\mathbf{g} = (g_k, \dots, g_n)$. If we then pivot on the $[m, k]$ edge (so necessarily $f_k = g_m$), then after the pivot

- all the entries in positions $[m', k]$ for $m' \geq m$ are 0 (more precisely, they are all $f_k - g_m$), and
- the triangle headed at the $[k + 1, k + 1]$ position is of the form $A(\mathbf{f}', \mathbf{g}')$ where $\mathbf{f}' = (f_{k+1}, \dots, f_n)$ and $\mathbf{g}' = (g_k, \dots, g_{m-1}, \widehat{g_m}, g_{m+1}, \dots, g_n)$. (The $\widehat{g_m}$ here indicates the removal of that entry).

See Figures 6 and 7 for illustrations of Observation 3.1.

3.1.3 Item 3

Here Observation 3.1 is also a key ingredient. The proof of the statement in item 3 is by induction (on the column k being scanned).

By Lemma 2.2, the initial network has path matrix $M_{\mathbf{e} \rightarrow \mathbf{a}}$, and the triangle headed at the $[1, 1]$ edge is of the form $A(\mathbf{f}, \mathbf{g})$ (specifically with $\mathbf{f} = \mathbf{a}$ and $\mathbf{g} = \mathbf{e}$).

Suppose that we are at the point in the algorithm where we are scanning the k th column. Let us assume (by induction) that in the current weighted planar network A , the triangle headed at the $[k, k]$ edge is of the form $A(\mathbf{f}, \mathbf{g})$, and that all the entries in the first $k - 1$ columns are non-negative.

If column k has no zero entries, then we move on to column $k + 1$ without changing A ; so as we scan column $k + 1$, in the current weighted planar network A , the triangle headed at the $[k + 1, k + 1]$ edge is of the form $A(\mathbf{f}', \mathbf{g}')$ (with \mathbf{f}' obtained from \mathbf{f} by removing the first element of \mathbf{f} , and \mathbf{g}' obtained from \mathbf{g} by removing the last element of \mathbf{g}). Also, all the entries in the first k columns are non-negative.

If column k has its first zero at position $[m, k]$, for some $m \geq k$, then, by the observation above, after pivoting the k th column has all its entries non-negative, and (in the updated network) the triangle headed at the $[k + 1, k + 1]$ edge is of the form $A(\mathbf{f}', \mathbf{g}')$. Also (crucially), by Lemma 2.3, the new network still has path matrix $M_{\mathbf{e} \rightarrow \mathbf{a}}$.

So by induction (on k), when we get to the point where $k = n + 1$ (which we do reach, by assumption), the final weighted planar network A has all non-negative entries and has path matrix $M_{\mathbf{e} \rightarrow \mathbf{a}}$.

3.1.4 Item 4

Suppose that the first negative weight that is encountered in running the algorithm is at the $[m, k_0]$ edge.

Mark the $[m, k_0]$ edge. There is a path, p_m , from source s_m to sink t_k — move horizontally from s_m to the lower vertex of the marked edge, then move vertically until the path currently being constructed intersects with the unique path from s_k to t_k , and then move horizontally to t_k . This path only uses edges with non-zero weight (the $[m', k_0]$ edges, for $m' < m$, have positive weight), and in fact uses only edges with positive weight, except for the $[m, k_0]$ edge, which has negative weight.

Next, let k_1 be maximal subject to the condition that the $[m - 1, k_1]$ edge has positive weight (assuming such a k_1 exists — if not, the process terminates, as described in more detail in a moment), and mark the $[m - 1, k_1]$ edge. Note that there is a path, p_{m-1} , from source s_{m-1} to sink t_{k-1} — horizontal to the marked edge, then vertical to the unique path from s_{k-1} to t_{k-1} (the vertical stretch of the network goes at least as high as $k - 1$, since it is strictly to the left of a vertical stretch that goes at least as far as the unique path from s_k to t_k), then horizontal. This path is vertex disjoint from p_k . Also, this path only uses edges with positive weight — that the marked edge has positive weight forces all edges above it to have positive weight.

Repeat — mark (if it exists) the $[m - 2, k_2]$ edge, where k_2 is maximal subject to the that edge having positive weight. Note that there is a path, p_{m-2} say, from source s_{m-2} to sink t_{k-2} , that is vertex-disjoint from p_m, p_{m-1} , and again only uses edges with positive weight.

Suppose we have a path p'_m from s_m to t_k that avoids edges with weight zero and that makes its first vertical turn before the $[m, k_0]$ edge, i.e, before column k_0 , say at column $k'_0 < k_0$. Necessarily, $k'_0 \leq k_1$. So any path from s_{m-1} to t_{k-1} that avoids edges with weight zero and that is disjoint from p'_m must make its first vertical turn earlier than p_{m-1} does. Let p'_{m-1} be such a path, turning vertical first at column $k'_1 < k_1$. We have $k'_1 \leq k_2$, so any path from s_{m-2} to t_{k-2} that avoids edges with weight zero and that is disjoint from p'_{m-1} must make its first vertical turn earlier than p_{m-2} does. Iterating, we eventually get that if $p'_m, \dots, p'_{m-\ell+1}$ is any vertex disjoint path system from $\{s_m, \dots, s_{m-\ell+1}\}$ to $\{t_k, \dots, t_{k-\ell+1}\}$ that avoids edges with weight zero, then it must be the case that $p'_{m-\ell+1}$ makes its first vertical turn earlier than $p_{m-\ell+1}$ does. So any path from $s_{k-\ell}$ to $t_{k-\ell}$ that is disjoint from $p'_{m-\ell+1}$ and avoids edges with weight zero must make its first vertical turn before it reaches the $[m-\ell, k_\ell]$ edge. But no such vertical turn exists — if it did, the process of marking edges would not have stopped with the $[m-\ell, k_\ell]$ edge. This contradiction finishes the proof of the assertion that all the vertex disjoint path systems from $\{s_m, \dots, s_{m-\ell}\}$ to $\{t_k, \dots, t_{k-\ell}\}$, that avoid edges with weight zero, use the $[m, k_0]$ edge and no other negative edges, and finishes the verification of item 4.

3.2 Deriving Theorem 1.8 from Claim 2.5

First suppose that \mathbf{e} is a restricted growth sequence relative to \mathbf{a} . Then it is easy to see that Algorithm 2.4 terminates with the report that $M_{\mathbf{e} \rightarrow \mathbf{a}}$ is TNN, and moreover that we can recover the final weighted planar network A of Algorithm 2.4, by the following process: for $i = 1, \dots, n$, in the i th iteration of Algorithm 1.4,

- if no $e' \in X$ is found with $e' = a_i$ (so all $e' \leq a_i$), then for each $m = i, \dots, n$, at the $[m, i]$ edge of the weighted planar network put the weight $a_i - (X)_{m-i+1}$. Here $(X)_j$ indicates the j th entry of X (in its i th incarnation); and
- if there is an $e' \in X$ with $e' = a_i$, say $e' = (X)_{m'-i+1}$ for some m' , then for each $m = i, \dots, m'$, at the $[m, i]$ edge of the weighted planar network put the weight $a_i - (X)_{m-i+1}$, and in all other positions put the weight 0.

Note moreover that using this process, the weighted planar network that witnesses that $M_{\mathbf{e} \rightarrow \mathbf{a}}$ is TNN can be constructed from \mathbf{a} and \mathbf{e} in only $O(n^2)$ operations.

On the other hand, suppose that \mathbf{e} is not a restricted growth sequence relative to \mathbf{a} . If we execute the procedure described above (that associates a weighted planar network with a running of Algorithm 1.4), stopping it at the point where an $e' > a_i$ is first found, then the resulting partial weighted planar network agrees with the final network of Algorithm 2.4 at all the edges that are involved in the collections of vertex disjoint paths that witness a negative minor of $M_{\mathbf{e} \rightarrow \mathbf{a}}$, and so the collection of sources and sinks (rows and columns of $M_{\mathbf{e} \rightarrow \mathbf{a}}$) that witnesses a negative minor of $M_{\mathbf{e} \rightarrow \mathbf{a}}$ can be recovered from the output of Algorithm 1.4.

References

- [1] Martin Aigner and Günter M. Ziegler, *Proofs from THE BOOK* (4th edition), Springer Verlag, 2010.

- [2] Feryal Alayont and Nicholas Krzywonos, Rook polynomials in three and higher dimensions, *Involve* **6** (2013), 35–52.
- [3] T. Ando, Totally positive matrices, *Linear Algebra Appl.* **90** (1987), 165–219.
- [4] Jörg Arndt, *Matters Computational: Ideas, Algorithms, Source Code*, Springer, 2011.
- [5] Francesco Brenti, Combinatorics and total positivity, *J. Combin. Theory Ser. A* **71** (1995), 175–218.
- [6] F. Brenti, The applications of total positivity to combinatorics, and conversely, in *Total Positivity and its Applications*, (M. Gasca, C. Micchelli, eds.), Kluwer Academic Pub., Dordrecht, The Netherlands, 1996, 451–473.
- [7] Richell O. Celeste, Roberto B. Corcino and Ken Joffaniel M. Gonzales, Two approaches to normal order coefficients, *J. Integer Seq.* **20** (2017), article 17.3.5.
- [8] W. N. Everitt, K. H. Kwon, L. L. Littlejohn, R. Wellman, G. J. Yoon, Jacobi-Stirling numbers, Jacobi polynomials, and the left-definite analysis of the classical Jacobi differential expression, *J. Comput. Appl. Math.* **208** (2007), 29–56.
- [9] Sergey Fomin and Andrei Zelevinsky, Total positivity: tests and parameterizations, *Math. Intelligencer* **22** (2000), 23–33.
- [10] David Galvin and Adrian Pacurar, Total non-negativity of some combinatorial matrices, *J. Combin. Th. Ser. A* **172** (2020), article 105179.
- [11] Mariano Gasca and Charles A. Micchelli (eds.), *Total Positivity and its Applications*, Kluwer Academic Pub., Dordrecht, The Netherlands, 1996.
- [12] Ken Joffaniel M. Gonzales, Total positivity of a class of combinatorial matrices, unpublished manuscript.
- [13] Leetsch C. Hsu and Peter Jau-Shyong. Shiue, A unified approach to generalized Stirling numbers, *Adv. Appl. Math.* **20** (1998), 366–384.
- [14] Bernt Lindström, On the vector representations of induced matroids, *Bull. Lond. Math. Soc.* **5** (1973), 85–90.
- [15] Mark Skandera, Introductory notes on total positivity, retrieved from <http://www.math.lsa.umich.edu/~fomin/eidma.html>, July 31 2022.