



Max-convolution through numerics and tropical geometry

Taylor Brysiewicz¹ · Jonathan D. Hauenstein² · Caroline Hills²

Received: 7 June 2023 / Accepted: 20 September 2023

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

Abstract

The maximum function, on vectors of real numbers, is not differentiable. Consequently, several differentiable approximations of this function are popular substitutes. We survey three smooth functions which approximate the maximum function and analyze their convergence rates. We interpret these functions through the lens of tropical geometry, where their performance differences are geometrically salient. As an application, we provide an algorithm which computes the max-convolution of two integer vectors in quasi-linear time. We show this algorithm's power in computing adjacent sums within a vector as well as computing service curves in a network analysis application.

Keywords Numerics · Tropical geometry · Amoebas · Max-convolution · LogSumExp · Maximum approximation

1 Introduction

Given $v = (v_1, \dots, v_n) \in \mathbb{R}^n$, although computing the maximum $M = \max_{1 \leq i \leq n} v_i$ is an elementary task, the function $v \mapsto \max(v)$ is not differentiable. A common technique used in optimization [1, 2] and machine learning [3, 4] is to replace the precise computation of M with an approximate computation. This article investigates

✉ Caroline Hills
chills1@nd.edu

Taylor Brysiewicz
tbrysiew@uwo.ca

Jonathan D. Hauenstein
hauenstein@nd.edu

¹ Department of Mathematics, Western University, Middlesex College, London N6A 5B7, Ontario, Canada

² Department of Applied and Computational Mathematics, University of Notre Dame, 102 G Crowley Hall, Notre Dame, IN 46556, USA

three standard ways to smoothly approximate the maximum function. Equipped with

$$F_v(t) = \sum_{j=1}^n t^{v_j}, \quad L_v(t) = \log_t(F_v(t)), \quad R_v(t) = \frac{t F'_v(t)}{F_v(t)}, \quad \text{and} \quad ||v||_p = \left(\sum_{j=1}^n |v_j|^p \right)^{\frac{1}{p}},$$

we consider the approximations:

$$(\text{LogSumExp}) : M = \lim_{t \rightarrow \infty} L_v(t) \quad \text{and} \quad (\text{Ratio}) : M = \lim_{t \rightarrow \infty} R_v(t) \quad (1)$$

and, if each entry of v is non-negative,

$$(\text{p-norm}) : \quad M = \lim_{p \rightarrow \infty} ||v||_p = ||v||_\infty. \quad (2)$$

We drop the subscript v when the vector of interest is clear from context (Fig. 1).

When $t > 0$, the functions $L_v(t)$ and $R_v(t)$ are smooth as a function of v and approximate M as shown by the limits above. For $p \in \mathbb{R}_{>0}$, the function $||v||_p$ smoothly approximates M provided that each element of v is non-negative. Each of these functions can be expressed in terms of

$$\mathcal{L}_v(t) = \log(F_v(t)). \quad (3)$$

Proposition 1.1 For $v = (v_1, \dots, v_n) \in \mathbb{R}^n$, $t \in \mathbb{R}$ with $t > 1$, we have $u = \log(t) > 0$, and

$$L_v(t) = L_v(e^u) = \frac{1}{u} \mathcal{L}_v(e^u), \quad (4)$$

$$R_v(t) = R_v(e^u) = \frac{d}{du} \mathcal{L}_v(e^u), \quad (5)$$

$$||v||_{\log(t)} = ||v||_u = e^{L_{\log|v|}(t)} = e^{\frac{1}{u} \mathcal{L}_{\log|v|}(e^u)}, \quad (6)$$

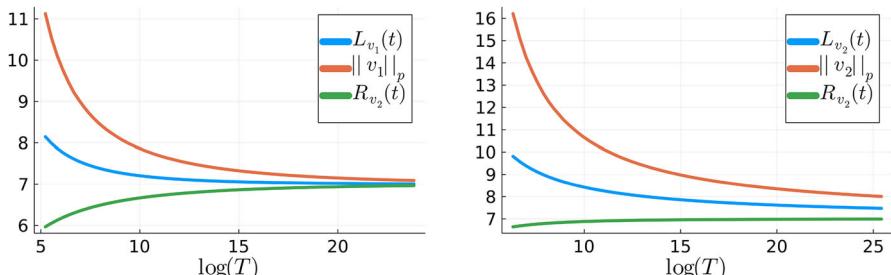


Fig. 1 Plots of the values of the smooth approximations $L(t)$, $||v||_p$, and $R(t)$ for the vectors $v_1 = (1, 2, 3, 4, 5, 6, 7)$ and $v_2 = (1, 2, 3, 4, 5, 6, 7, 7, 7, 7, 7, 7)$. These values are plotted against the natural logarithm of the largest absolute value T of a floating point number involved in the numerical evaluation of each function

where $\log |\cdot| : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is the componentwise log-absolute value map.

Throughout, we omit the proofs of statements, like Proposition 1.1, which follow from a direct calculation.

After providing some basic notation in Section 2, we derive convergence rates for each of these functions in Section 3 by analyzing $\mathcal{L}_v(t)$. Namely, for $\delta > 0$, we give bounds on t for which the absolute error of these approximations is smaller than δ . As a consequence, when the vector v is integral, one may use a rounding procedure, such as the floor or ceiling function, to provably compute M via a single evaluation of an approximating function.

In practice, when the entries of v are selected from a discrete set, such as the integers \mathbb{Z} , the maximum M is often obtained more than once. The number of times M is attained is the *multiplicity* of M in v , namely

$$\mu_M = \#\{i \mid v_i = M\}.$$

When the multiplicity of M is large, the ratio approximation (1) significantly outperforms the other approximations (see Section 5). In Section 3, we express μ_M as a limit of the aforementioned functions and derive analogous convergence rates result for computing μ_M .

In light of part (5) of Proposition 1.1, we generalize the approximation $R_v(t)$ using higher-order derivatives. In particular, for $k \geq 1$, we define

$$R_v^{(k)}(t) = -\frac{(-t)^k}{(k-1)!} \frac{d^k}{dt^k} \mathcal{L}_v(t). \quad (7)$$

Observe that $R_v^{(1)}(t) = R_v^{(1)}(e^u) = \frac{d}{du} \mathcal{L}_v(e^u) = R_v(e^u) = R_v(t)$, and that $R_v^{(k)}$ is related, but not equal, to $\frac{d^k}{dt^k} R_v$ (see also Remark 3.13). We show, in Section 3, that every $R_v^{(k)}(t)$ for $k \geq 1$ converges to M at the same rate. We discuss how to use these higher-order derivatives to numerically approximate other information about v (see Theorem 3.11).

In Section 4, we explore the geometry of $L_v(t)$ and $R_v^{(k)}(t)$ in terms of objects called amoebas from the world of tropical geometry. We realize the graph of the function $u \mapsto \mathcal{L}_v(e^u)$ as the upper boundary of a certain amoeba and provide a geometric interpretation of the performance differences of $R_v(t)$ and $L_v(t)$ when μ_M is large.

In Section 5, we conduct a series of experiments showcasing our theoretical results and the performance differences of the approximation techniques discussed. In particular, we provide empirical evidence showing the extent to which the bounds derived in Section 3 are tight. We illustrate how the ratio approximations perform significantly better than the others when the maximum appears with non-trivial multiplicity and that this feature persists in the presence of a noisy model.

In Section 6, we propose an algorithm for the max-convolution problem:

MAXCON: Given $a = (a_0, \dots, a_n) \in \mathbb{Z}^{n+1}$ and $b = (b_0, \dots, b_n) \in \mathbb{Z}^{n+1}$,
 compute $c = (c_0, \dots, c_{2n}) \in \mathbb{Z}^{2n+1}$ where (8)
 $c_k = \max_{\max\{0, k-n\} \leq i \leq \min\{k, n\}} (a_i + b_{k-i}).$

Since each c_k is a maximum of the integer vector $v^{(k)} = (a_i + b_{k-i})_{i=\max\{0, k-n\}}^{\min\{k, n\}}$ its value may be determined by an (appropriately large) evaluation of an approximation of that maximum. In particular, any algorithm which computes classical convolution coefficients may be used as an oracle for evaluating $L_{v^{(k)}}(t)$. The fast Fourier transform, for example, performs such a computation using $O(n \log(n))$ operations. By combining this fact with our bounds from Section 3, we obtain a quasi-linear time algorithm, i.e., the number of operations is $O(n \log(n))$, for the max-convolution problem (8). We end by applying our numerical approach to the maximum consecutive subsums problem and the computation of service curve constraints.

A short conclusion is provided in Section 7.

2 Notation and fundamental results

We begin by fixing the following notation

$v = (v_1, \dots, v_n)$: an n -tuple of real numbers

M : $\max(v)$

μ_c : multiplicity of a real number c in v , i.e., $\#\{i \mid v_i = c\}$

ℓ : number of distinct elements in v

$w = (w_1, \dots, w_\ell)$: decreasing list of unique elements in v i.e., $M = w_1 > \dots > w_\ell$

$g = (g_1, \dots, g_\ell)$: $g_i = M - w_i$ with $0 = g_1 < g_2 < \dots < g_\ell$.

Additionally, t will denote a variable which takes on positive real values whereas $u = \log(t)$ is its image under the natural logarithm.

Example 2.1 To illustrate notation, consider

$$v = (7, 7, -1, 0, 1, 1, 2.5, 2.5, 7, 7) \in \mathbb{R}^{10}.$$

Then we have,

$$M = 7, \quad \ell = 5,$$

$$\mu_7 = 4, \quad \mu_{2.5} = 2, \quad \mu_1 = 2, \quad \mu_0 = 1, \quad \mu_{-1} = 1,$$

$$w = (7, 2.5, 1, 0, -1), \quad g = (0, 4.5, 6, 7, 8).$$

□

In principle, the elements of v can be any real numbers; however, in practice, they are usually some floating point approximations. A floating point number, mathematically, may be interpreted as a rational number, even though it is not explicitly represented this way on a computer. Hence, we may take $v \in \mathbb{Q}^n$. Moreover, we may assume for our analyses that $v \in \mathbb{Z}^n$ since evaluating $\mathcal{L}_v(t)$ at a power t^k of t corresponds to scaling v by k :

$$\mathcal{L}_v(t^k) = \mathcal{L}_v(e^{ku}) = \mathcal{L}_{kv}(e^u) = \mathcal{L}_{kv}(t). \quad (9)$$

Remark 2.2 We note that $\mathcal{L}_v(t) = \mathcal{L}_{uv}(e)$. The function $\text{lse}(v) = \mathcal{L}_v(e)$ is a popular activation function in the field of machine learning traditionally called the *log-sum-exp* function [5]. Numerical methods for accurately evaluating it may be found in [6].

The following expansion of $\mathcal{L}_v(t)$ about $t = \infty$ is fundamental for our analysis.

Proposition 2.3 For $v \in \mathbb{Z}^n$, $\mathcal{L}_v(t)$ has the following expansion:

$$\mathcal{L}_v(t) = \log(t)M + \log(\mu_M) + \log \left(\sum_{i=1}^{\ell} \frac{\mu_{w_i}}{\mu_M} t^{-g_i} \right) \quad (10)$$

where the first term inside the logarithm is $\frac{\mu_{w_1}}{\mu_M} t^{-g_1} = 1$. In particular, by expanding the logarithmic term, there exist nonnegative real numbers $\{\alpha_j\}_{j=1}^{\infty}$ such that

$$\mathcal{L}_v(t) = \log(t)M + \log(\mu_M) + \sum_{j=1}^{\infty} \alpha_j t^{-j} \quad (11)$$

$$= uM + \log(\mu_M) + \sum_{j=1}^{\infty} \alpha_j e^{-ju} \quad (12)$$

where $\alpha_1 = \dots = \alpha_{g_2-1} = 0$ and $\alpha_{g_2} = \frac{\mu_{w_2}}{\mu_M}$.

We obtain similar expressions for $L_v(t)$, $R_v(t)$, and $R_v^{(k)}(t)$ by combining Propositions 1.1 and 2.3.

Proposition 2.4 For $v \in \mathbb{Z}^n$, let $\{\alpha_j\}_{j=1}^{\infty}$ be as in Proposition 2.3. Then,

$$\begin{aligned} L_v(t) &= M + \log_t(\mu_M) + \log_t \left(\sum_{i=1}^{\ell} \frac{\mu_{w_i}}{\mu_M} t^{-g_i} \right) \\ &= M + \frac{1}{\log(t)} \left(\log(\mu_M) + \sum_{j=1}^{\infty} \alpha_j t^{-j} \right) \end{aligned} \quad (13)$$

$$\begin{aligned}
&= M + \frac{1}{u} \left(\log(\mu_M) + \sum_{j=1}^{\infty} \alpha_j e^{-ju} \right), \\
R_v(t) &= M - \sum_{j=1}^{\infty} j \alpha_j t^{-j} \\
&= M - \sum_{j=1}^{\infty} j \alpha_j e^{-ju},
\end{aligned} \tag{14}$$

$$\begin{aligned}
R_v^{(k)}(t) &= M - \sum_{j=1}^{\infty} j \binom{j+k-1}{k-1} \alpha_j t^{-j} \\
&= M - \sum_{j=1}^{\infty} j \binom{j+k-1}{k-1} \alpha_j e^{-ju}.
\end{aligned} \tag{15}$$

In particular, $R_v(t)$ and, more generally, $R_v^{(k)}(t)$ are analytic at $t = \infty$. Additionally, if each $v_i \geq 0$,

$$||v||_{\log(t)} = e^{L_{\log|v|}(t)} = M \cdot e^{L_{\log|v|}(t) - \log(M)}. \tag{16}$$

Clearly, $L_v(t) \geq M$ and the $\log_t(\mu_M)$ term from (13) is responsible for a slow convergence rate of $L_v(t)$ to M . This logarithmic term is eliminated in $R_v(t)$ and, more generally, in $R_v^{(k)}(t)$. See Section 4 for a geometric explanation of this fact.

Remark 2.5 Since $R_v(t)$ and, more generally, $R_v^{(k)}(t)$ are analytic at $t = \infty$ when $v \in \mathbb{Z}^n$, Cauchy's integral formula yields that for each $k \geq 1$ there exists $r > 0$ such that

$$\frac{1}{2\pi\sqrt{-1}} \oint_{|t|=r} t^{-1} \cdot R_v^{(k)}(t^{-1}) \cdot dt = M. \tag{17}$$

Numerically, one can use the trapezoid rule [7] to approximate M from this integral.

Since $R_v(t)$ depends on $F'_v(t)$ which may be difficult to evaluate in practice (see Section 6), we show below how to approximate $R_v(t)$ from evaluations of $L_v(t)$.

Proposition 2.6 For $v \in \mathbb{R}^n$, $t > 1$, and $\alpha > 0$ with $\alpha \neq 1$, define

$$D_v(t, \alpha) = \log_{\alpha} \left(\frac{F_v(\alpha \cdot t)}{F_v(t)} \right) = \frac{\mathcal{L}_v(\alpha \cdot t) - \mathcal{L}_v(t)}{\log(\alpha)}. \tag{18}$$

Then,

$$\lim_{\alpha \rightarrow 1} D_v(t, \alpha) = R_v(t).$$

Proof Applying l'Hôpital's rule yields $\lim_{\alpha \rightarrow 1} D_v(t, \alpha) = \lim_{\alpha \rightarrow 1} R_v(\alpha \cdot t) = R_v(t)$. \square

3 Approximating quantities associated to v

Equipped with the expansions (13)–(16) and (18), the functions $L_v(t)$, $R_v^{(k)}(t)$, $D_v(t, \alpha)$, and $\|v\|_u$ may each be used to approximate certain information about v , such as M , μ_M , and g_2 .

For each such approximation of M , we derive a lower bound on t so that the absolute error is less than a given value $\delta > 0$. For integer vectors, we pay particular attention to the case where $\delta = 1$, since one can use the floor $\lfloor \cdot \rfloor$ and ceiling $\lceil \cdot \rceil$ functions to *provably* compute these values from their approximations.

3.1 Computing the maximum

We derive bounds on the absolute errors of $L_v(t)$, $R_v(t)$, $D_v(t, \alpha)$, and $\|v\|_u$ in Theorems 3.1, 3.3, 3.6, and 3.7 respectively.

Theorem 3.1 *Fix $v \in \mathbb{Q}^n$ and $\delta > 0$. Then $0 \leq L_v(t) - M < \delta$ whenever $t > 1$ and*

$$t^{\delta+g_2} - t^{g_2} \mu_M - (n - \mu_M) > 0.$$

If $v \in \mathbb{Z}^n$ and $\delta = 1$, the above bound is obtained when

$$t > \frac{\mu_M + \sqrt{\mu_M^2 + 4(n - \mu_M)}}{2}.$$

If additionally $\mu_M = 1$ then this bound further simplifies to $t > \frac{1}{2} + \sqrt{n}$.

Proof Assume, after reindexing, that $v_1 = \dots = v_{\mu_M} = M$. Thus,

$$L_v(t) - M = \log_t \left(\mu_M + \sum_{j=\mu_M+1}^n t^{v_j - M} \right).$$

Hence, $L_v(t) - M < \delta$ provided that the expression within the logarithm is smaller than t^δ . Since the function $x \mapsto t^x$ is monotonic for $t > 1$,

$$\mu_M + \sum_{j=\mu_M+1}^n t^{v_j - M} \leq \mu_M + (n - \mu_M)t^{-g_2},$$

completing the proof of the first statement since this value is less than t^δ when

$$t^\delta > \mu_M + (n - \mu_M)t^{-g_2}.$$

For $v \in \mathbb{Z}^n$, we have that $g_2 \geq 1$ so that a sufficient condition when $\delta = 1$ is

$$t^2 - \mu_M t - (n - \mu_M) > 0$$

yielding the second statement. The third statement follows immediately. \square

When v consists of integers and μ_M is known, Theorem 3.1 suggests an algorithm which provably computes M using one evaluation of $L_v(t)$:

Return $\lfloor L_v(t) \rfloor$ for t satisfying the inequality $2t > \mu_M + \sqrt{\mu_M^2 + 4(n - \mu_M)}$.

The largest t value required is when $\mu_M = n$ for which one can take $t = n + 1$. In particular, for any $v \in \mathbb{Z}^n$, one always has $\lfloor L_v(n + 1) \rfloor = M$.

The following example illustrates Theorem 3.1 on qualitatively different input.

Example 3.2 Consider the following integer vectors:

$$v_1 = (1, 2, 3, 4, 5, 6, 7) \quad \text{and} \quad v_2 = (1, 2, 3, 4, 5, 6, 7, 7, 7, 7, 7).$$

The maximum of both vectors is 7 which has multiplicity 1 and 5 in v_1 and v_2 , respectively. By Theorem 3.1, $L_{v_1}(t) \in [7, 8)$ when $t > 3$ and $L_{v_2}(t) \in [7, 8)$ when $t > 6$. Figure 2 is consistent with these bounds and illustrates the reduced convergence rate for v_2 due to the increased multiplicity of the maximum. \square

The worst-case scenario analysis for $R_v(t)$ is qualitatively distinct from that of $L_v(t)$. The fact which distinguishes these cases is that for a fixed $t > 1$, the function $x \mapsto xt^{-x}$ is decreasing only after reaching its maximum on $\mathbb{R}_{>0}$ at $x = \log(t)^{-1}$.

Theorem 3.3 Fix $v \in \mathbb{Q}^n$ and $\delta > 0$. Then $0 \leq M - R_v(t) < \delta$ when $t > e^{1/g_2}$ and

$$t > \left(\frac{(n - \mu_M)g_2}{\delta \cdot \mu_M} \right)^{\frac{1}{g_2}}.$$

If $v \in \mathbb{Z}^n$ and $\delta = 1$, this bound is attained when

$$t > \max \left(e, \frac{n - \mu_M}{\mu_M} \right).$$

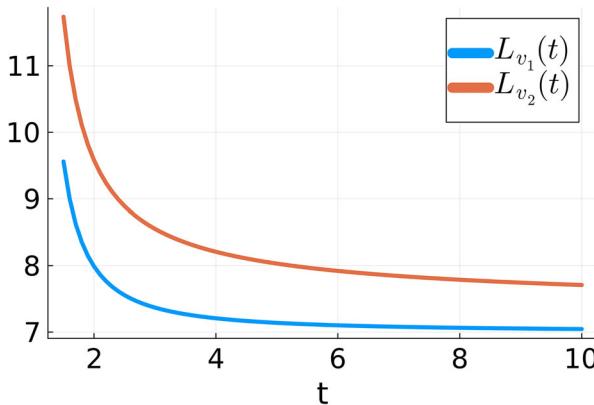


Fig. 2 The graphs of $L_{v_1}(t)$ and $L_{v_2}(t)$ as in Example 3.2

Proof From (14), a worst-case analysis with $t > e^{\frac{1}{g_2}}$ shows that

$$M - R_v(t) = \sum_{j=1}^{\infty} j \alpha_j t^{-j} < \frac{n - \mu_M}{\mu_M} g_2 t^{-g_2}.$$

Therefore, the main result follows from

$$M - R_v(t) < \delta \quad \text{whenever} \quad t^{g_2} > \frac{(n - \mu_M)g_2}{\delta \cdot \mu_M}.$$

When $v \in \mathbb{Z}^n$ and $\delta = 1$, this simplifies to $t^{g_2} > \frac{(n - \mu_M)g_2}{\mu_M}$. Since $t > e$ and $g_2 \geq 1$, this holds if additionally

$$t > \frac{n - \mu_M}{\mu_M}.$$

□

Example 3.4 For v_1 and v_2 as in Example 3.2, Fig. 3 compares the graphs of $L(t)$ and $R(t)$. Note that Theorems 3.1 and 3.3 guarantee that $L_{v_2}(t) \in [7, 8)$ when $t > 6$ and $R_{v_2}(t) \in (6, 7]$ when $t > e \approx 2.718$. □

Remark 3.5 For $v \in \mathbb{Z}^n$, one has

$$L_v(t) - M = \begin{cases} O(1/\log(t)) & \text{if } \mu_M > 1, \\ O(t^{-g_2}/\log(t)) & \text{if } \mu_M = 1, \end{cases} \quad \text{and} \quad M - R_v(t) = O(t^{-g_2}).$$

When $\mu_M = 1$, Theorem 3.1 requires $t = O(\sqrt{n})$ while Theorem 3.3 requires $t = O(n)$. However, the bound $\frac{n - \mu_M}{\mu_M}$ from Theorem 3.3 is smaller than the bound

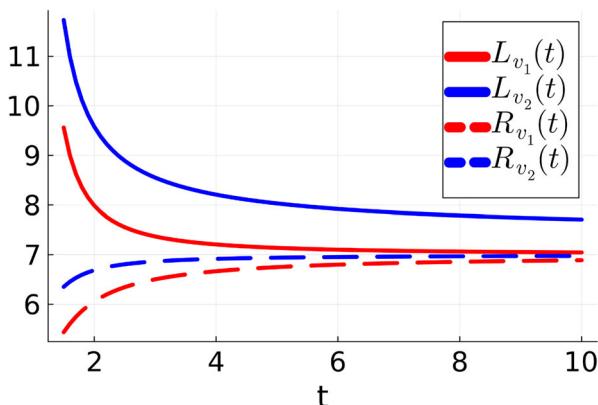


Fig. 3 The graphs of $L(t)$ and $R(t)$ applied to v_1 and v_2 from Example 3.2

$\frac{\mu_M + \sqrt{\mu_M^2 + 4(n - \mu_M)}}{2}$ from Theorem 3.1 whenever $\mu_M \geq \frac{1}{4}(\sqrt{8n + 1} - 1)$. For reference, this means that

$$(n, \mu_M) \in \{(10, 2), (105, 7), (1081, 23), (10153, 71), (100576, 224), \dots\}$$

are afforded equal t -bounds for $L_v(t)$ or $R_v(t)$ via Theorems 3.1 and 3.3, respectively. These bounds are derived from the worst-case scenarios where $M - 1$ appears with multiplicity $n - \mu_M$. However, on input vectors v sampled from the uniform distribution on $\{0, \dots, M\}$ with varying multiplicities μ_M , $L_v(t)$ consistently performs worse than $R_v(t)$. For more details, see the experiments in Section 5.

Based on the relationship between $D_v(t, \alpha)$ and $R_v(t)$ summarized in Proposition 2.6, the error is similar to Theorem 3.3. Here, the worst case analysis yields the function $x \mapsto \frac{t^{-x}(1 - \alpha^{-x})}{\log(\alpha)}$ which is decreasing after reaching its maximum on $\mathbb{R}_{>0}$ at $x = \frac{\log(\log(\alpha t)) - \log(\log(t))}{\log(\alpha)}$ which limits to $\log(t)^{-1}$ as $\alpha \rightarrow 1$.

Theorem 3.6 *Fix $v \in \mathbb{Q}^n$, $\delta > 0$, and $\alpha > 1$. Then, $0 \leq M - D_v(t, \alpha) < \delta$ when $t > e^{1/g_2}$,*

$$\begin{aligned} t &> \alpha \left(\frac{\alpha^{-g_2}}{1 - \alpha^{-g_2}} \right) \xrightarrow{\alpha \rightarrow 1} e^{1/g_2}, \quad \text{and} \\ t &> \left(\frac{n - \mu_M}{\delta \cdot \mu_M} \cdot \frac{1 - \alpha^{-g_2}}{\log(\alpha)} \right)^{\frac{1}{g_2}} \xrightarrow{\alpha \rightarrow 1} \left(\frac{(n - \mu_M)g_2}{\delta \cdot \mu_M} \right)^{\frac{1}{g_2}}. \end{aligned}$$

If $v \in \mathbb{Z}^n$ and $\delta = 1$, this bound is obtained when $\alpha > 1$ and

$$t > \max \left(e, \frac{n - \mu_M}{\mu_M} \right).$$

Proof The worst case analysis using the three assumptions on α and t that are independent of δ show that

$$M - D_v(t, \alpha) < \delta \quad \text{whenever} \quad t^{g_2} > \frac{n - \mu_M}{\delta \cdot \mu_M} \frac{1 - \alpha^{-g_2}}{\log(\alpha)}.$$

When $v \in \mathbb{Z}^n$ and $\delta = 1$, this simplifies to $t^{g_2} > \frac{n - \mu_M}{\mu_M} \frac{1 - \alpha^{-g_2}}{\log(\alpha)}$. Since $t > e$, $\alpha > 1$, and $g_2 \geq 1$, this holds if additionally $t > \frac{n - \mu_M}{\mu_M}$. \square

To analyze the p -norm case, following [8, 9], we assume the vector v has undergone a linear transformation so that each $v_j \in [0, 1]$.

Theorem 3.7 *If $v \in [0, 1]^n$ and $\delta > 0$, then $0 \leq ||v||_u - M < \delta$ when $u > 1$ and*

$$e^{u(\Delta + g_2)} - e^{ug_2} \mu_M - (n - \mu_M) > 0 \quad \text{where} \quad \Delta = \log \left(1 + \frac{\delta}{M} \right).$$

Proof By (16), $\|v\|_u = M \cdot e^{\epsilon(u)}$ where $\epsilon(u) = L_{\log|v|}(e^u) - \log(M)$ which is the error when using $L_{\log|v|}(e^u)$ to approximate $\log(M)$. Hence, $\|v\|_u - M < \delta$ if and only if $\epsilon(u) < \log(1 + \frac{\delta}{M}) =: \Delta$. By Theorem 3.1, this occurs whenever $t > 1$ and

$$t^{\Delta+g_2} - t^{g_2} \mu_M - (n - \mu_M) > 0.$$

Since $u = \log(t)$, changing coordinates gives the result. \square

Example 3.8 The following illustrates the differences between the approximations $L(t)$, $R(t)$, $D(t, \alpha)$, and $\|\cdot\|_p$ of M on our running examples of v_1 and v_2 from Example 3.2. First, similar to previous plots, Fig. 4 shows the difference of convergence rate for the p -norm approximation due to higher multiplicity. Next, Fig. 5 compares $R_v(t)$ with $D_v(t, 2)$ and $D_v(t, 1.5)$ for $v = v_1$ and $v = v_2$ showing comparable convergence rates. Finally, we compare the values of $\|v\|_p$, $L_v(t)$, $R_v(t)$, and $D_v(t, \alpha)$ when they require comparably large (in absolute value) floating point number for evaluation. Setting T to be the largest floating point number required, we plot these functions against $\log(T)$ in Figs. 6 and 7 for $v = v_1$ and $v = v_2$, respectively. \square

3.2 Computing the multiplicity

Due to the simplistic nature of the expansion in (13) for $L_v(t)$, we consider computing the multiplicity μ_M for the maximum M . In particular, it is easy to see from (13) that

$$\mu_M = \lim_{t \rightarrow \infty} t^{L_v(t) - M} \quad (19)$$

Of course, using this expression requires *a priori* knowledge of M which can be attained, for example, in the integer case by applying Theorem 3.1.

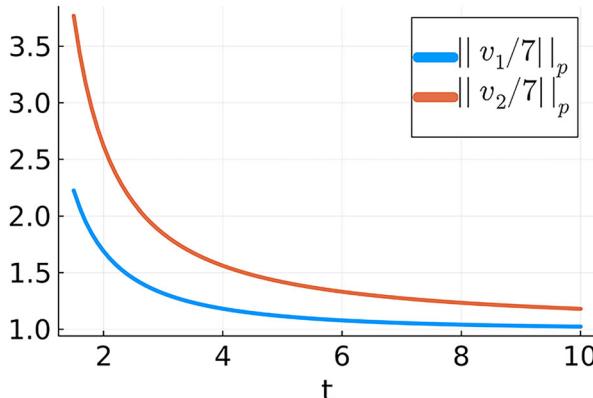


Fig. 4 The graphs of the p -norms of the vectors $v_1/7$ and $v_2/7$ with entries in $[0, 1]$, where v_1 and v_2 are from Example 3.2

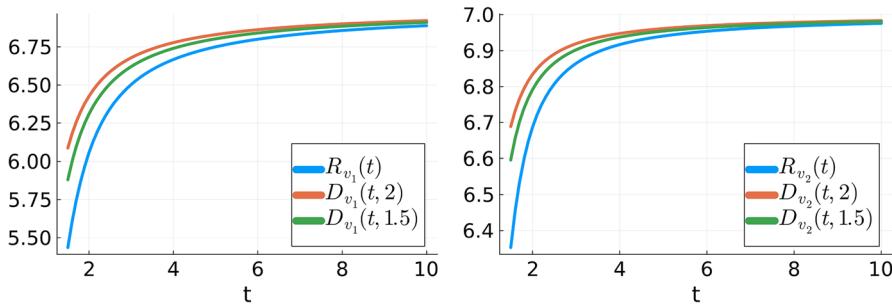


Fig. 5 Comparison of $D_v(t, \alpha)$ for v_1 and v_2 from Example 3.2 with various values of α

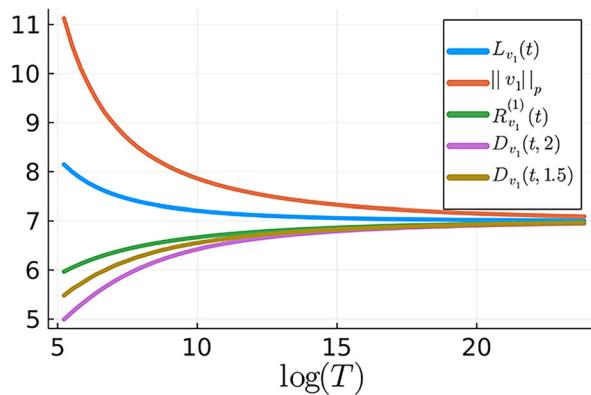


Fig. 6 A comparison of the smooth approximations $L_{v_1}(t)$, $R_{v_1}(t)$, $\|v_1\|_p$, $D_{v_1}(t, 2)$, and $D_{v_1}(t, 1.5)$ of the maximum of v_1 , plotted against the natural logarithm of T , the required absolute value of floating point numbers for evaluation

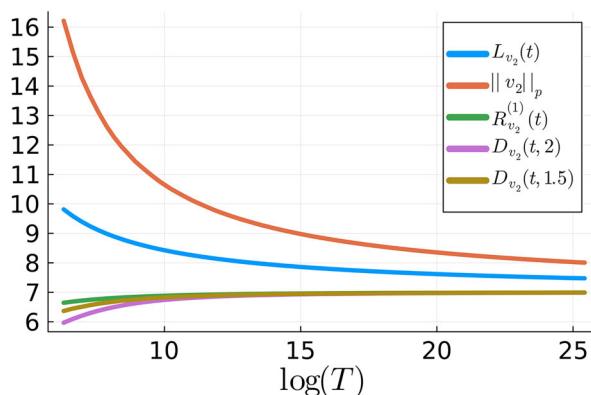


Fig. 7 A comparison of the smooth approximations $L_{v_2}(t)$, $R_{v_2}(t)$, $\|v_2\|_p$, $D_{v_2}(t, 2)$, and $D_{v_2}(t, 1.5)$ of the maximum of v_2 , plotted against the natural logarithm of T , the required absolute value of floating point numbers for evaluation

Theorem 3.9 Given $v \in \mathbb{Z}^n$, $\lfloor t^{L_v(t)-\lfloor L_v(t) \rfloor} \rfloor = \mu_M$ whenever

$$t > \max \left\{ n - \mu_M, \frac{\mu_M + \sqrt{\mu_M^2 + 4(n - \mu_M)}}{2} \right\}$$

Proof When $t > \frac{\mu_M + \sqrt{\mu_M^2 + 4(n - \mu_M)}}{2}$, Theorem 3.1 provides that $\lfloor L(t) \rfloor = M$. Using a worst-case analysis, one has

$$0 \leq t^{L_v(t)-M} - \mu_M \leq (n - \mu_M)t^{-1}$$

with the worst-case upper bound below 1 when $t > n - \mu_M$. \square

Example 3.10 Continuing with v_1 and v_2 from Example 3.2, Theorem 3.9 provides $t^{L_{v_1}(t)-\lfloor L_{v_1}(t) \rfloor} \in [1, 2)$ for $t > 6$ and $t^{L_{v_2}(t)-\lfloor L_{v_2}(t) \rfloor} \in [5, 6)$ for $t > 6$ with Fig. 8 showing convergence in advance of such worst-case bounds.

3.3 Combining $R_v^{(k)}(t)$ to improve convergence and compute g_2

Since all of the higher-order derivatives $R_v^{(k)}(t)$ have the same convergence rate, one can combine them in various ways to increase the convergence rate as well as extract other information about v . The following demonstrates a higher-order approximation of M along with approximating g_2 . The computation of g_2 and M produces, as a byproduct, the second largest element of v , namely $w_2 = M - g_2$.

Theorem 3.11 For $v \in \mathbb{Z}^n$, we have

$$\frac{2R_v^{(1)}(t)R_v^{(3)}(t) - R_v^{(2)}(t) \left(R_v^{(1)}(t) + R_v^{(2)}(t) \right)}{R_v^{(1)}(t) - 3R_v^{(2)}(t) + 2R_v^{(3)}(t)} = M + O(t^{-g_2-1}) \quad (20)$$

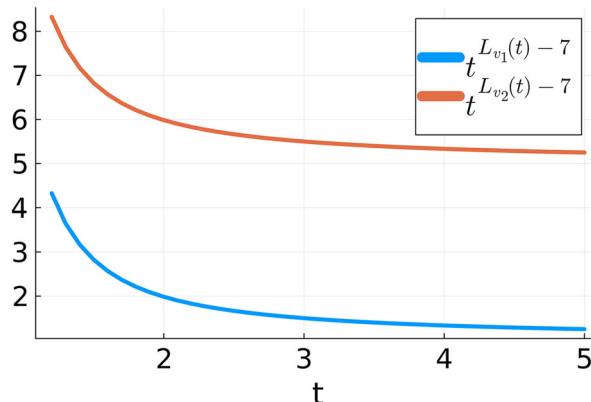


Fig. 8 The graphs of $t^{L(t)-M}$ applied to v_1 and v_2 from Example 3.2

and

$$\frac{R_v^{(1)}(t) - 3R_v^{(2)}(t) + 2R_v^{(3)}(t)}{R_v^{(2)}(t) - R_v^{(1)}(t)} = g_2 + O(t^{-1}). \quad (21)$$

Proof From Proposition 2.3 and (15),

$$\begin{aligned} R_v^{(1)}(t) &= M - g_2 \frac{\mu_{w_2}}{\mu_M} t^{-g_2} + O(t^{-g_2-1}), \\ R_v^{(2)}(t) &= M - g_2(g_2+1) \frac{\mu_{w_2}}{\mu_M} t^{-g_2} + O(t^{-g_2-1}), \\ R_v^{(3)}(t) &= M - \frac{g_2(g_2+1)(g_2+2)}{2} \frac{\mu_{w_2}}{\mu_M} t^{-g_2} + O(t^{-g_2-1}) \end{aligned}$$

and so the result follows by direct symbolic elimination. \square

Example 3.12 We illustrate Theorem 3.11 using v_1 and v_2 from Example 3.2. Figure 9 compares the convergence of $R_v^{(1)}(t)$, $R_v^{(2)}(t)$, $R_v^{(3)}(t)$, and the combined formula in (20) for $v = v_1$ and $v = v_2$ to $M = 7$ for both. For both cases, one sees faster convergence as expected from (20). Additionally, Fig. 10 shows the convergence of the combined formula in (21) for v_1 and v_2 to $g_2 = 1$ for both.

Remark 3.13 The functions $R_v^{(k)}(t)$ are linear combinations of the derivatives $\mathcal{D}^{(k)}(t) = \frac{d^k}{du^k} \mathcal{L}_v(e^u)$, e.g.,

$$\begin{aligned} R^{(1)}(t) &= \mathcal{D}^{(1)}(t), \\ R^{(2)}(t) &= -\mathcal{D}^{(1)}(t) + \mathcal{D}^{(2)}(t), \\ R^{(3)}(t) &= \frac{2\mathcal{D}^{(1)}(t) - 3\mathcal{D}^{(2)}(t) + \mathcal{D}^{(3)}(t)}{2}, \\ R^{(4)}(t) &= \frac{-6\mathcal{D}^{(1)}(t) + 11\mathcal{D}^{(2)}(t) - 6\mathcal{D}^{(3)}(t) + \mathcal{D}^{(4)}(t)}{6}. \end{aligned}$$

In particular, the linear transformation that maps the first r values of $\mathcal{D}^{(k)}(t)$ to the first r values of $R^{(k)}(t)$ is represented by an $r \times r$ lower triangular matrix $A^{(r)}$. For

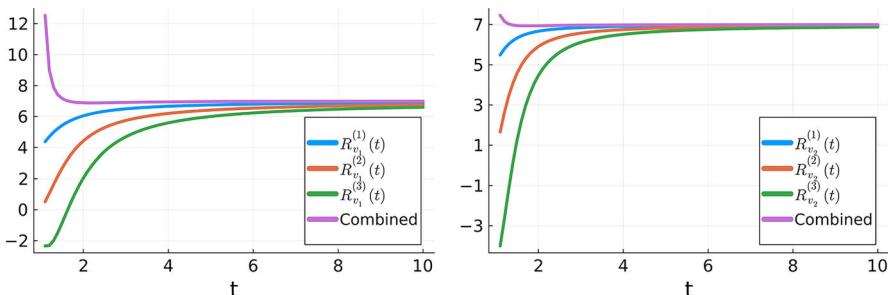


Fig. 9 Comparison of various methods to approximate M for v_1 and v_2

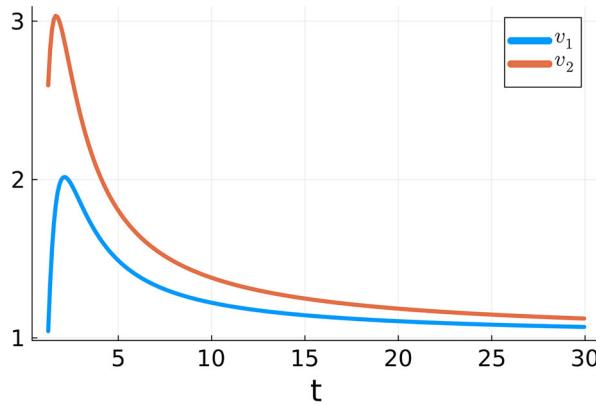


Fig. 10 The graphs of (21) for v_1 and v_2 which converge to $g_2 = 1$ for both cases

$j \leq i$, the (i, j) -entry of $A^{(r)}$ is

$$\frac{(-1)^{i+1}}{(i-1)!} \mathcal{S}_{ij}$$

where \mathcal{S}_{ij} is a Stirling number of the first kind. For example,

$$A^{(4)} = \begin{bmatrix} (-1)^2 1 & 0 & 0 & 0 \\ (-1)^3 1 & (-1)^3 (-1) & 0 & 0 \\ \left(\frac{(-1)^4}{2}\right) 2 & \left(\frac{(-1)^4}{2}\right) (-3) & \left(\frac{(-1)^4}{2}\right) 1 & 0 \\ \left(\frac{(-1)^5}{6}\right) 6 & \left(\frac{(-1)^5}{6}\right) (-11) & \left(\frac{(-1)^5}{6}\right) 6 & \left(\frac{(-1)^5}{6}\right) (-1) \end{bmatrix}.$$

4 The tropical viewpoint

We interpret our previous results geometrically using tools from *tropical geometry*. We stress that throughout this section, we consider specific families of tropical constructions which exist more generally. Namely, the varieties we consider are graphs of univariate Laurent polynomials with positive coefficients. Such varieties are quite special and so the results of this section may not hold in the more general setting. For an introduction to tropical geometry, we invite the interested reader to consult the standard reference [10].

To utilize the tropical geometry framework, we assume throughout this section that $v \in \mathbb{Z}^n$, define $\mathbb{C}^\times = \mathbb{C} \setminus \{0\}$, and consider the function

$$\begin{aligned} \varphi : \mathbb{C}^\times &\rightarrow \mathbb{C}^2 \\ t &\mapsto F_v(t). \end{aligned}$$

Let X_v be the graph of φ intersected with $(\mathbb{C}^\times)^2$. We note that X_v is the set of zeros of the polynomial $\mathcal{F}_v(t, y) = y - F_v(t)$:

$$X_v = \{(t, y) \in (\mathbb{C}^\times)^2 \mid \mathcal{F}_v(t, y) = 0\} \subset (\mathbb{C}^\times)^2.$$

The *Newton polygon* of $\mathcal{F}_v(t, y)$ is the convex hull $\mathcal{N}(\mathcal{F}_v(t, y))$ of the exponent vectors of $\mathcal{F}_v(t, y)$. In this case, $\mathcal{N}(\mathcal{F}_v(t, y))$ is simply the triangle Δ_v with vertices $(M, 0)$, $(\min(v), 0)$, and $(0, 1)$ as illustrated in Fig. 11a. The union of the outer normal rays of Δ_v along with the origin form a *polyhedral fan* called the *tropicalization* of X_v , denoted $\text{trop}(X_v)$ and illustrated in Fig. 11c. The fan $\text{trop}(X_v)$ is a *tropical curve* which encodes the asymptotic behavior of X_v near the coordinate axes $\mathbb{C}^2 \setminus (\mathbb{C}^\times)^2$. Note that in our specific situation, the Newton polytope $\mathcal{N}(\mathcal{F}_v(t, y))$, and hence the tropical curve $\text{trop}(X_v)$, depends only on $\min(v)$ and $\max(v)$.

An alternative construction of $\text{trop}(X_v)$, due to Bergman [11], involves the image $\mathcal{A}_\tau(X_v)$ of X_v under the log-absolute value map:

$$\begin{aligned} \text{Log}_\tau |\cdot| : (\mathbb{C}^\times)^2 &\rightarrow \mathbb{R}_{u,s}^2 \\ (t, y) &\mapsto (\log_\tau(|t|), \log_\tau(|y|)). \end{aligned}$$

The set $\mathcal{A}_\tau(X_v)$ is called the τ -amoeba of X_v . We remark that we use u and s for coordinates of the codomain and that the overlap of the symbol u with previous sections is intentional. Undecorated, the notation $\mathcal{A}(X_v) \subseteq \mathbb{R}_{u,s}^2$ refers to the e -amoeba of X_v as

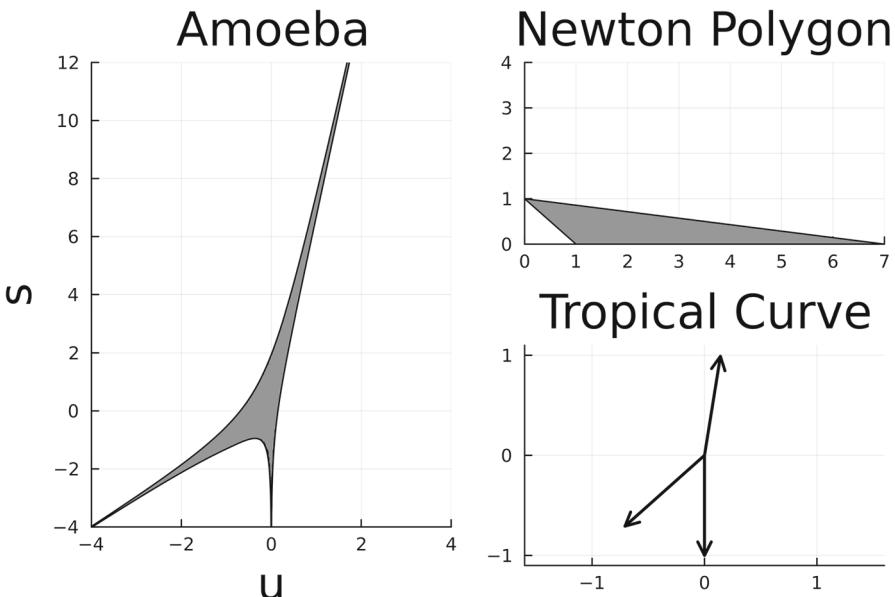


Fig. 11 **a** The Newton polygon of $\mathcal{F}_{v_1}(t, y)$ where v_1 is as in Example 3.2. **b** The amoeba $\mathcal{A}(X_{v_1})$. **c** The tropical variety $\text{trop}(X_{v_1})$

illustrated in Fig. 11b. Since $\mathcal{A}_\tau(X_v) = \frac{1}{\log(\tau)} \mathcal{A}(X_v)$, the set $\mathcal{A}(X_v)$ contains all of the information about all of the amoebas of X_v . Since the absolute values of coordinates of points in X_v may be arbitrarily large or small, the set $\mathcal{A}(X_v)$ is unbounded. The portions which approach infinity are loosely referred to as the *tentacles* of the amoeba. As $\tau \rightarrow \infty$, these tentacles limit to the rays of $\text{trop}(X_v)$. In this sense, $\text{trop}(X_v)$ contains asymptotic information about X_v and is sometimes referred to as the *logarithmic limit set* of the amoeba of X_v .

We call lines which intersect an amoeba $\mathcal{A}(X_v)$ in a ray tentacle lines. Up to translation, these rays are exactly those in $\text{trop}(X_v)$. Note that many tentacle lines may be associated to the same tropical ray (e.g., the two vertical lines of Fig. 12 both correspond to the ray $(0, -1)$). The following elementary facts relate the Newton polygon $\mathcal{N}(\mathcal{F}_v(t, y))$, amoeba $\mathcal{A}(X_v)$, and tropical curve $\text{trop}(X_v)$. We encourage the reader to refer to Fig. 12. These facts are specializations of a more general relationship between these three objects (see [10] for more details).

- (1) The tentacle lines of $\mathcal{A}(X_v)$ corresponding to the ray in $\text{trop}(X_v)$ spanned by $(0, -1)$ correspond to distinct moduli of complex roots of $F_v(t)$.
- (2) When the lowest order term of the Laurent polynomial $F_v(t)$ is a constant c , $\text{trop}(X_v)$ contains the ray spanned by $(-1, 0)$. There is one tentacle line of $\mathcal{A}(X_v)$ associated to that ray, which occurs at height $\log(c)$.

Additionally, the following facts relate $\mathcal{A}(X_v)$ and the function $u \mapsto \mathcal{L}_v(e^u)$.

- (3) The upper boundary \mathcal{U} of $\mathcal{A}(X_v)$ is the graph of $\mathcal{L}_v(e^u)$.
- (4) $L_v(t) = L_v(e^u) = \frac{\mathcal{L}_v(e^u)}{u}$ is the slope of the ray from the origin to the point $(u, \mathcal{L}_v(e^u)) \in \mathcal{A}(X_v)$.

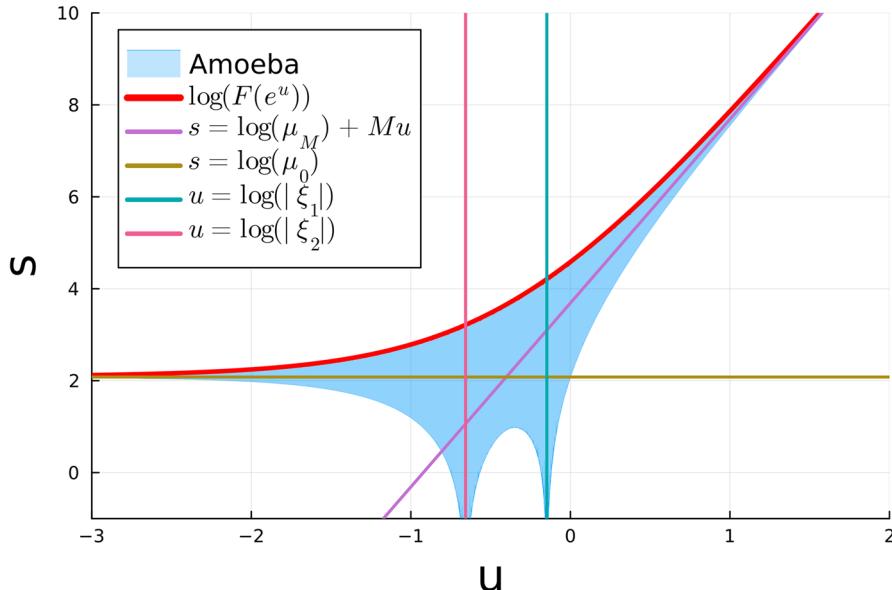


Fig. 12 The amoeba $\mathcal{A}(X_v)$ of the graph X_v where v is as in Example 4.2 along with its tentacle lines

(5) $\mathcal{D}_v^{(k)}(e^u) = \frac{d^k}{du^k} \mathcal{L}_v(e^u)$ is the k^{th} derivative of the function $u \mapsto \mathcal{L}_v(e^u)$.
 (6) $\mathcal{D}_v^{(1)}(e^u) = R_v^{(1)}(e^u)$ is the slope of the tangent line to the boundary of the amoeba at $(u, \mathcal{L}_v(e^u))$.

We point out that (3) follows from the fact that all of the coefficients of $F_v(t)$ are non-negative. In general, describing the boundaries of amoebas is challenging [12].

Proposition 4.1 *Let $v \in \mathbb{Z}^n$. The tentacle lines of $\mathcal{A}(X_v)$ are given by*

- (a) $s = \log(\mu_M) + M \cdot u$
- (b) $s = \log(\mu_m) + m \cdot u$ where $m = \min(v)$.
- (c) $u = \log(|\xi_i|)$ where $\{\xi_i\}_{i=1}^d \subset \mathbb{C}$ are the roots of $F(t)$.

Proof As observed already, the tentacles in the $(0, -1)$ direction correspond to roots ξ_i of $F_v(t)$ and they occur at $u = \log(|\xi_i|)$ which establishes (c). To see (a) and (b), suppose $m = \min(v) = 0$. Then, $F(t)$ has a constant term of μ_m and hence $\lim_{t \rightarrow 0} \log(F(t)) = \log(\mu_m)$. This limit indicates that $\mathcal{A}(X_v)$ has a horizontal tentacle occurring at $s = \log(\mu_m)$. However, translating v by $a \in \mathbb{Z}$ amounts to sheering the Newton polygon space by $(\alpha, \beta) \mapsto (\alpha, a\alpha + \beta)$ and the amoeba space by $(u, s) \mapsto (u - as, s)$. In particular, this transformation does not change the s -intercepts of the tentacle lines. Hence, the tentacle line associated to the minimum of v has the equation $s = \log(\mu_m) + m \cdot u$ whereas the line associated to the maximum M has the equation $s = \log(\mu_M) + M \cdot u$. \square

Proposition 4.1 gives a geometric interpretation of how the multiplicity of M in v contributes to a slower convergence rate of $L_v(t)$ but not for $R_v(t)$: multiplicity corresponds to a translation of the tentacle line of $\mathcal{A}(X_v)$ associated to the tropical ray spanned by $(1, M)$. This is geometrically displayed in Fig. 13.

Example 4.2 Let $v = [0_8, 1_5, 2_{40}, 3_5, 4_{40}]$, where a subscript indicates multiplicity. The amoeba $\mathcal{A}(X_v)$ is shown in Fig. 12. The polynomial

$$F_v(t) = 8 + 5t + 40t^2 + 5t^3 + 40t^4$$

has two pairs $\xi_1, \bar{\xi}_1$ and $\xi_2, \bar{\xi}_2$ of conjugate roots. Hence, there are two vertical tentacle lines of $\mathcal{A}(X_v)$. The tentacle line corresponding to the minimum multiplicity $\mu_m = \mu_0 = 8$ is horizontal at height $\log(8)$ and the equation of the remaining tentacle line is $s = \log(40) + 4u$. The upper boundary of the amoeba is the image of the positive part of X_v under the log-absolute value map.

Figure 12 illustrates a geometric interpretation of the values of $L_v(e^u)$ and $R_v^{(1)}(e^u)$ as slopes of rays. \square

One may also interpret the Cauchy integral in Remark 2.5:

$$\frac{1}{2\pi\sqrt{-1}} \oint_{|t|=r} t^{-1} \cdot R_v^{(k)}(t^{-1}) \cdot dt = M,$$

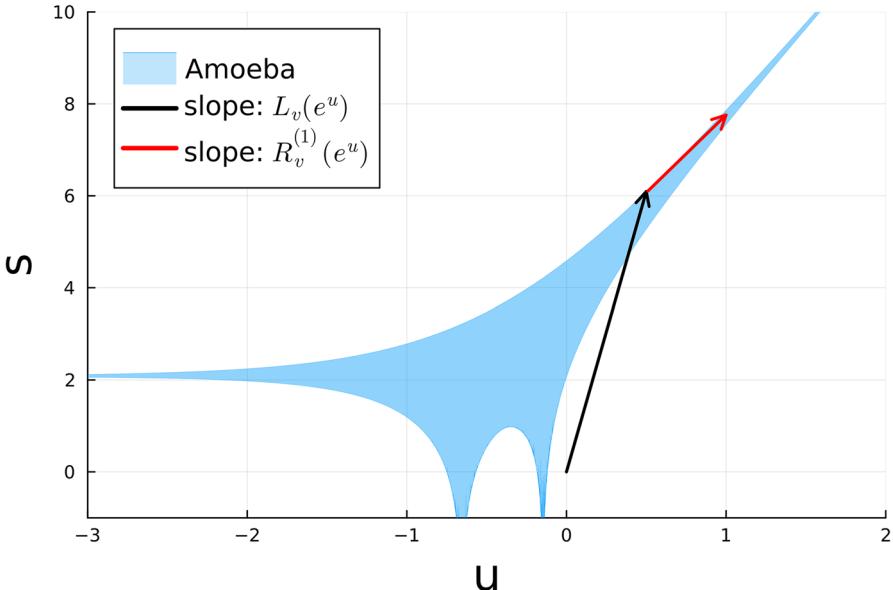


Fig. 13 The amoeba $\mathcal{A}(X_v)$ of the graph X_v for v as in Example 4.2 along with the ray from the $(0, 0)$ to $(u, L_v(e^u))$ and the ray from $(u, L_v(e^u))$ with slope $R_v^{(1)}(e^u) = \mathcal{D}_v^{(1)}(e^u)$

in terms of tropical geometry when $k = 1$. This is done via the order map:

$$\text{ord} : \mathbb{R}^2 \rightarrow \mathbb{R}^2$$

$$(u, s) \mapsto \left(\frac{1}{(2\pi\sqrt{-1})^2} \int_{\substack{\text{Log}|t|=u \\ \text{Log}|y|=s}} \frac{t F'_v(t)}{y - F_v(t)} \cdot \frac{dt dy}{ty}, \frac{1}{(2\pi\sqrt{-1})^2} \int_{\substack{\text{Log}|t|=u \\ \text{Log}|y|=s}} \frac{y}{y - F_v(t)} \cdot \frac{dt dy}{ty} \right).$$

The function ord is constant and \mathbb{Z}^2 -valued on connected components of the complement $\mathbb{R}^2 \setminus \mathcal{A}(X_v)$ of the amoeba [13]. In fact, ord maps these components to distinct integer points in the Newton polygon Δ_v . In particular, for any point (u, s) in the bottom right complement component, the first integral

$$\frac{1}{(2\pi\sqrt{-1})^2} \int_{\substack{\text{Log}|t|=u \\ \text{Log}|y|=s}} \frac{t F'_v(t)}{y - F_v(t)} \cdot \frac{dt dy}{ty}$$

degenerates as $|y| \rightarrow 0$ (or $s \rightarrow -\infty$) to the integral

$$\frac{1}{2\pi\sqrt{-1}} \oint_{\text{Log}|t|=u} \frac{t F'_v(t)}{F_v(t)} \cdot \frac{dt}{t} = \frac{1}{2\pi\sqrt{-1}} \oint_{\text{Log}|t|=u} t^{-1} R_v^{(1)}(t) \cdot dt = M$$

for sufficiently large u as in (17). The second integral, on the other hand, evaluates to zero. Since this value of ord on (u, s) is constant on connected components of the

complement of the amoeba, this shows that $\text{ord}(u, s)$ evaluates to the vertex $(M, 0)$ of Δ_v when (u, s) is in the bottom-right component of $\mathbb{R}^2 \setminus \mathcal{A}(X_v)$, in agreement with Remark 2.5.

5 Experiments

We compare each of the approximations discussed on a gallery of qualitatively different inputs v . In each of the following sections, we sample vectors v from some prescribed distribution. We then compare the approximations of M on these samples on average.

5.1 Integer numbers

We compared $L_v(t)$ and $R_v(t)$ in the integer case by defining a maximum M with multiplicity μ for an integer-valued vector $v \in [1, M]^n$. That is, we defined v of varying length $n = 1, \dots, 100$ with M appearing μ times, where $\mu \leq n$ and the remaining $n - \mu$ values of v were random integers sampled from $[1, M - 1]$. Figure 14 displays the four experiments comparing the performance of $L_v(t)$ and $R_v(t)$ for approximating a

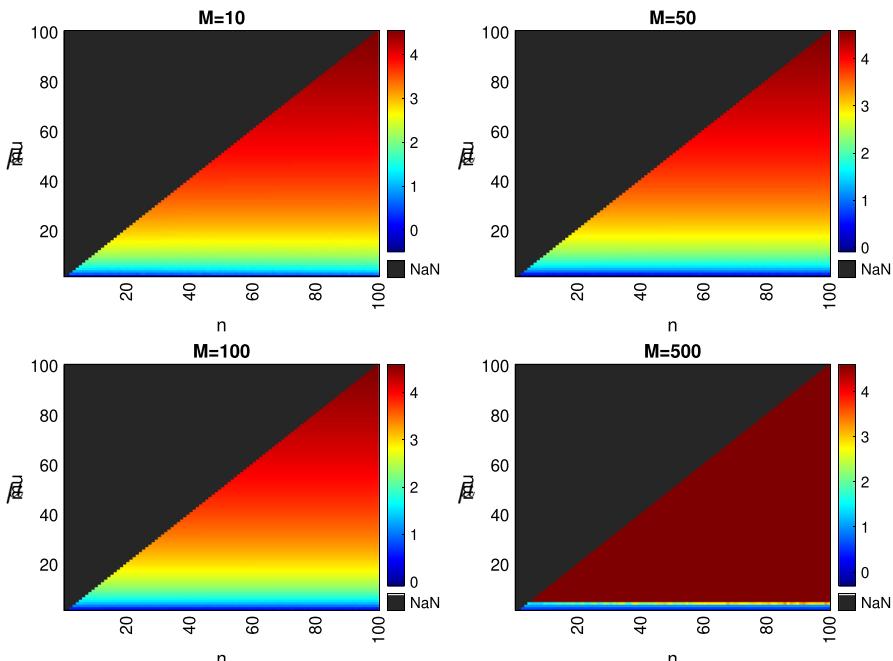


Fig. 14 Average results of the integer-valued vector v experiments. The x -axis and y -axis correspond to $n = 1, \dots, 100$ and $\mu = 1, \dots, n$ respectively. Each pixel is the average of 100 subexperiments measuring $\log(1 + t_{L_v}^* - t_{R_v}^*)$ where $t_{L_v}^*$ and $t_{R_v}^*$ are the t -values such that the absolute error of the corresponding function is less than 1. The maximum, M , are **a** 10, **b** 50, **c** 100, and **d** 500. Black pixels above the diagonal are when $\mu > n$

given maximum M , namely $M = 10, 50, 100$, and 500 , respectively. Performance is measured by t^* , the first value of t to approximate M up to an absolute error of 1 , so that the maximum is obtained from $L_v(t^*)$ or $R_v(t^*)$ through use of the floor or ceiling function, respectively. Each experiment consists of 100 subexperiments averaged over $\log(1 + t_{L_v}^* - t_{R_v}^*)$. The plotted values are logarithmic and offset by 1 since, if $t_{L_v}^* = t_{R_v}^*$, then $\log(1 + t_{L_v}^* - t_{R_v}^*) = \log(1) = 0$.

5.2 Uniformly distributed floating point numbers

We repeat the experiments of the above section with floating point vectors $v \in [0, 1]^n$ with $M = 1$. Figure 15 displays the results of comparing $L_v(t)$ and $R_v(t)$ on vectors whose elements are sampled from the uniform distribution on $[0, 1]$. As with the previous experiments, each pixel at coordinates (n, μ) represents the value of $\log(\alpha + t_{L_v}^* - t_{R_v}^*) - \log(\alpha)$, where t^* is the first value of t to approximate M up to a given absolute error. To model a significant g_2 gap, we constructed these vectors by choosing $n - \mu$ vectors uniformly from $[0, 1]$, multiplying them by $(n - 1)/n$ and appending them to a vector of length μ with coordinates all equal to 1 . The figures differ only in the absolute tolerance used to define t^* . The value $\alpha = \min(t_{L_v}^* - t_{R_v}^*) + 1$ offsets the results so that when the results are averaged and plotted logarithmically, the minimum

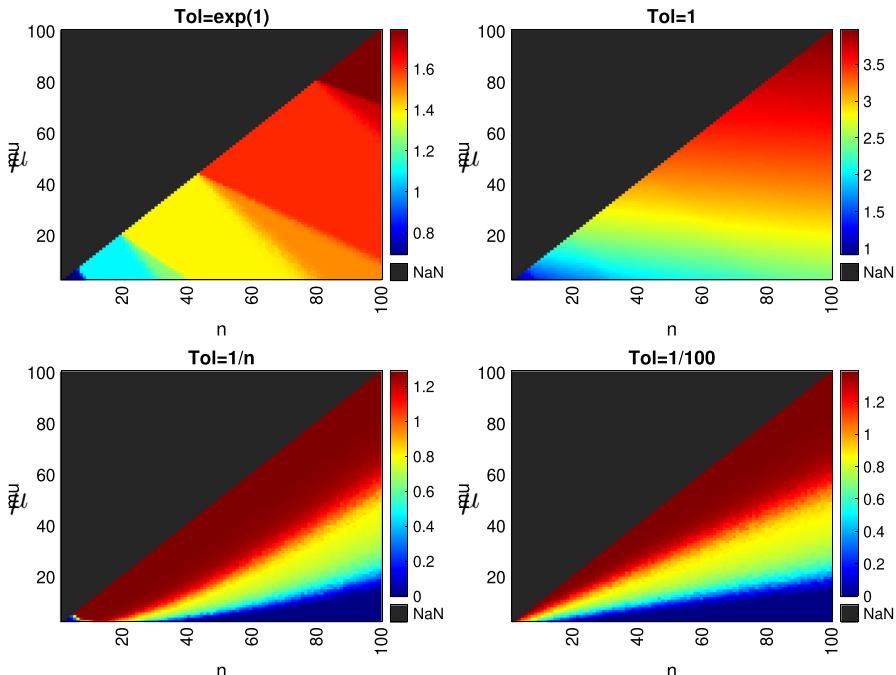


Fig. 15 Average results of the floating point-valued vector v experiments. Each pixel at position (n, μ) represents the value of $\log(\alpha + t_{L_v}^* - t_{R_v}^*) - \log(\alpha)$ averaged over 100 subexperiments. The x -axis and y -axis correspond to $n = 1, \dots, 100$ and $\mu = 1, \dots, n$ respectively. The tolerance used to define t^* are absolute error less than: **a** $\exp(1)$, **b** 1 , **c** $1/n$, and **d** $1/100$. Black pixels above the diagonal are when $\mu > n$

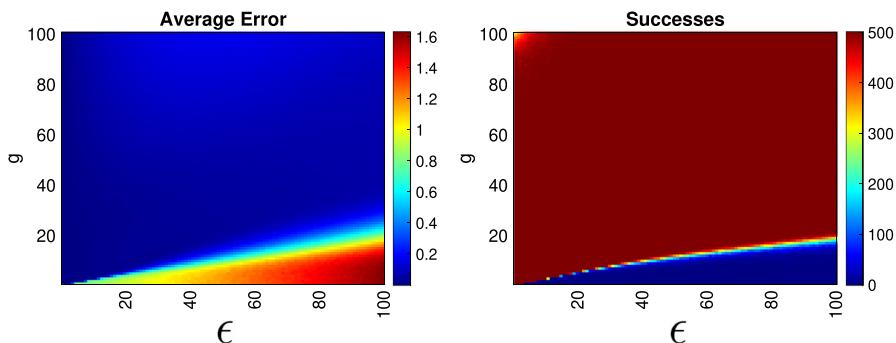


Fig. 16 For each (g, ϵ) and for 500 tries, (left) average error of $R_v(t^*)$ for t^* from Theorem 3.3 (right) number of approximations $R_v(t^*)$ within ϵ for t^* from Theorem 3.3

difference remains $\log(1) = 0$. The subtraction of the $\log(\alpha)$ term then better illustrates the subexperiments where $L_v(t)$ outperforms $R_v(t)$. This adjustment is accounted for in the uniform distribution examples as there are select instances in which $L_v(t)$ performs better than $R_v(t)$ by converging at a lesser t value, thus the difference is non-positive and less than -1 . This occurs most notably in the experiments with tolerance $1/n$ and $1/100$ of Fig. 15.

5.3 Clustering floating point numbers

We repeat a similar experiment with floating point numbers in the presence of noise. Our goal is to identify the scenarios where it is appropriate to apply Theorem 3.3 heuristically. Our setup is as follows. Suppose that 5 measurements, with values in $[0, 1]$, are to be taken, but the measuring device incurs some error $\pm\epsilon$. To rectify this, each measurement is performed 20 times. Heuristically, one may choose to apply Theorem 3.3 with the interpretation that v consists of 5 numbers, each occurring with multiplicity 20, with the goal of obtaining $\max(v)$ up to error ϵ . In this case, Theorem 3.3 specializes to $t > \left(\frac{4 \cdot g}{\epsilon}\right)^{\frac{1}{g}}$ where g is the gap between the top two *true* measurements.

After fixing ϵ and g , we model such a situation by the following procedure.

1. Pick 5 true measurements $w_1, \dots, w_5 \in [0, 1]$ by setting $w_5 = 1$, $w_4 = 1 - g$ and w_1, w_2 and w_3 , sampled uniformly at random from $[0, 1 - g]$.
2. For each w_i , sample 20 numbers uniformly from $[w_i - \epsilon, w_i + \epsilon]$. Collect all 100 numbers in v .
3. Evaluate $R_v(t^*)$ for $t^* = \left(\frac{4g}{\epsilon}\right)^{\frac{1}{g}}$ to obtain the absolute error $\text{err}_v = |1 - R(t^*)|$

For each pair (g, ϵ) , where $g = 0.01, \dots, 1$ and $\epsilon = 0, \dots, 1$, we repeat the above procedure 500 times and average the error obtained in step 3. Additionally, we deem an approximation a *success* if the error is smaller than ϵ . The two figures in Fig. 16 display, for each pair (g, ϵ) , the average error and number of successes.

As indicated by the experiments summarized in Fig. 16, a small gap g and large ϵ produces the largest errors with the fewest numbers of successes, as expected. Interestingly, a large gap and small ϵ , corresponding to the upper left corner of the figures also impacts the effectiveness of the heuristic. The large gap size means that w_2, \dots, w_5 are all chosen within a small interval $[0, 1 - g]$, and we suspect that this cluster behaves like the value $\frac{1-g}{2}$ appearing with high multiplicity. Additionally, the small ϵ value means it is difficult to achieve a “success” by having absolute error less than ϵ .

For these noisy experiments, there are two natural interpretations of what should be considered N and μ . The first is that N should be 5, the number of true measurements, whereas μ should be 1. The second interpretation is that N should be $20 \cdot 5 = 100$, the true length of v while μ should be 20, the size of the top cluster. In the $R_v(t)$ case, these distinctions cancel out in the bound provided by Theorem 3.3. In the $L_v(t)$ case, however, these interpretations give drastically different bounds when applying Theorem 3.1. The later interpretation often yields such enormous t bounds that an application of that result is not useful. In Fig. 17, we display the results of an experiment using the former interpretation. It did not happen that the $L_v(t)$ approximation with the interpreted bound from Theorem 3.1 achieved the expected accuracy. This suggests that the later interpretation, despite its lack of utility, is likely more appropriate. Our experiments also showcase the advantage of using the $R_v(t)$ approximation over $L_v(t)$, especially in noisy situations with high (approximate) multiplicity.

6 Max-convolution and applications

One way to use smooth approximations of the maximum function is to approximate the max-convolution of two vectors [9]. To that end, consider two integer vectors

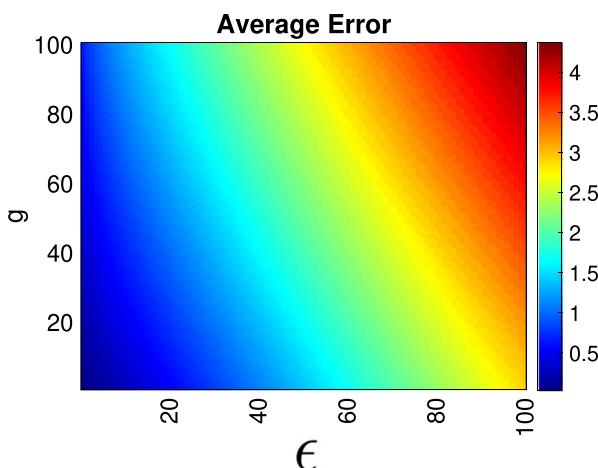


Fig. 17 For each (g, ϵ) , over 500 tries, the average error of $L_v(t^*)$ from Theorem 3.1

$a = (a_0, \dots, a_n)$ and $b = (b_0, \dots, b_n)$. The classical convolution problem asks to determine the vector of convolution coefficients $a \star b$ where

$$(a \star b)_k = \sum_{i=0}^k a_i \cdot b_{k-i} = \sum_{i=\max(0, k-n)}^{\min(k, n)} a_i \cdot b_{k-i}. \quad (22)$$

We remark that the middle description is sufficient if one takes $a_i = 0$ and $b_i = 0$ whenever they are undefined. With the same input, the problem of max-convolution, **MAXCON**, asks for the vector c of max-convolution coefficients, where

$$c_k = \max_{\max(0, k-n) \leq i \leq \min(k, n)} (a_i + b_{k-i}). \quad (23)$$

These coefficients can be obtained via (22) by replacing the operations $(\cdot, +)$ with $(+, \max)$, respectively. The form $c_k = \max_{0 \leq i \leq n} (a_i + b_{k-i})$ may be used if one replaces undefined a_i and b_i with $-\infty$. Equivalently, through constructing

$$A_t(x) = \sum_{i=0}^n t^{a_i} x^i, \quad B_t(x) = \sum_{i=0}^n t^{b_i} x^i \in \mathbb{Q}(t)[x],$$

the problem of **MAXCON** asks for the largest exponents in t appearing in the coefficients of

$$A_t(x) \cdot B_t(x) = \sum_{k=0}^{2n} \sum_{i=0}^k t^{a_i + b_{k-i}} x^k.$$

Setting $v^{(k)} = ((a_i + b_{k-i}) \mid \max(0, k-n) \leq i \leq \min(k, n))$, we rewrite this as

$$A_t(x) \cdot B_t(x) = \sum_{k=0}^{2n} F_{v^{(k)}}(t) x^k.$$

For fixed t , the values of $F_{v^{(k)}}(t)$ are *classical* convolution coefficients

$$((t^{a_0}, \dots, t^{a_n}) \star (t^{b_0}, \dots, t^{b_n}))_k$$

which can be computed by a quasi-linear time algorithm, e.g., using the fast Fourier transform (FFT) [14] requires $O(n \log(n))$ operations. Applying \log_t provides an $O(n \log(n))$ routine for evaluating $L_{v^{(k)}}(t)$, whereby with Theorem 3.1, this process computes $\max(v^{(k)}) = c_k$ when evaluated at a sufficiently large value of t , e.g., $t = n + 1$. If $M_a = \max_i |a_i|$ and $M_b = \max_i |b_i|$, then the largest value arising has bit-size at most $O((M_a + M_b + 1) \log_2(n))$ when $t = n + 1$. The following algorithm is a summary of this discussion.

Algorithm 1 MaxCon

Input: Two integer vectors $a = (a_0, \dots, a_n)$ and $b = (b_0, \dots, b_m)$.
Output: The max-convolution coefficients $c = (c_0, \dots, c_{n+m})$

- 1 Choose t^* satisfying the bounds of Theorem 3.1 (e.g., $t^* \geq \max(n, m)+1$)
- 2 Compute $\eta = ((t^*)^{a_0}, \dots, (t^*)^{a_n})$ and $\lambda = ((t^*)^{b_0}, \dots, (t^*)^{b_m})$
- 3 Compute $\ell(t^*) = \eta \star \lambda$
- 4 Apply $\lfloor \log_{t^*}(\cdot) \rfloor$ component-wise to $\ell(t^*)$ to obtain $c = \lfloor \log_{t^*}(\ell(t^*)) \rfloor$
- 5 **return** c

Example 6.1 Consider applying Algorithm 1 to compute the max-convolution coefficients of the vectors

$$a = (3, 1, 2, 4, 1, 2), \quad b = (5, 3, 0, 4).$$

Taking $t^* = 6$, we obtain

$$\eta = (216, 6, 36, 1296, 6, 36), \quad \lambda = (7776, 216, 1, 1296)$$

The classical convolution of η and λ is

$$\ell(6) = \eta \star \lambda = (1679620, 93312, 281448, 10365400, 334404, 329184, 1687400, 7812, 46656)$$

which under \log_6 evaluates to

$$\log_6(\ell(6)) = (8, 6.38685, 7.00301, 9.01571, 7.09923, 7.09045, 8.00258, 5.00258, 6).$$

Finally, by applying $\lfloor \cdot \rfloor$, we obtain the max-convolution coefficients

$$c = (8, 6, 7, 9, 7, 7, 8, 5, 6).$$

For completeness and interpretation of c , we provide $A_t(x)$, $B_t(x) \in \mathbb{Z}[t][x]$ below along with their product:

$$A_t(x) = t^3x^0 + t^1x^1 + t^2x^2 + t^3x^3 + t^1x^4 + t^2x^5, \quad B_t(x) = t^5x^0 + t^3x^1 + t^0x^2 + t^4x^3,$$

$$\begin{aligned} A_t(x) \cdot B_t(x) = & t^8x^0 + 2t^6x^1 + (t^7 + t^4 + t^3)x^2 + (t^9 + t^7 + t^5 + t)x^3 \\ & + (t^7 + t^6 + t^5 + t^2)x^4 + (t^7 + t^6 + 2t^4)x^5 + (t^8 + t^5 + t)x^6 + (t^5 + t^2)x^7 + t^6x^8. \end{aligned}$$

□

The key to Algorithm 1 lies in the ability to evaluate $F_{v^{(k)}}(t)$ via the fast Fourier transform in $O(n \log(n))$ by interpreting these values as classical convolution coefficients. A subsequent application of \log_t turns this into an evaluation of $L_{v^{(k)}}(t) =$

$\log_t(F_{v^{(k)}}(t))$ whereby one may apply Theorem 3.1. Similarly, with the aim to apply Theorem 3.6, one may use the approximation of Proposition 2.6 to wrap the ability to evaluate $F_{v^{(k)}}(t)$ into an algorithm which may require a smaller t evaluation.

Algorithm 2 MaxCon - using $D_{v^{(k)}}(t^*, \alpha^*)$

Input: Two integer vectors $a = (a_0, \dots, a_n)$ and $b = (b_0, \dots, b_m)$.

Output: The max-convolution coefficients $c = (c_0, \dots, c_{n+m})$

- 1 Choose t^* and α^* satisfying the bounds of Theorem 3.6 (e.g., $\alpha^* > 1$ and $t^* > \max(e, n-1, m-1)$)
- 2 Compute $\eta = ((t^*)^{a_0}, \dots, (t^*)^{a_n})$ and $\lambda = ((t^*)^{b_0}, \dots, (t^*)^{b_m})$
- 3 Compute $\eta' = ((\alpha^* t^*)^{a_0}, \dots, (\alpha^* t^*)^{a_n})$ and $\lambda' = ((\alpha^* t^*)^{b_0}, \dots, (\alpha^* t^*)^{b_m})$
- 4 Compute $\ell(t^*) = \eta \star \lambda$
- 5 Compute $\ell'(t^*) = \eta' \star \lambda'$ using FFT
- 6 Apply $\log_{\alpha^*}(\ell'(t^*)/\ell(t^*))$ component-wise to obtain $c = \lceil \log_{\alpha^*}(\ell'(t^*)/\ell(t^*)) \rceil$
- 7 **return** c

Remark 6.2 Algorithms 1 and 2, paired with their corresponding bounds from Section 3, give algorithms whose output constitute mathematical proofs provided that the convolution coefficients computed via FFT, as well as the ceiling of the logarithm, are computed exactly. Otherwise, the error introduced by the \star operation must be bounded by $1/2$, δ should be taken to be at most $1/2$ in the relevant theorems, and a two-sided rounding procedure should be applied rather than the ceiling or floor function.

Example 6.3 We apply Algorithm 2 to the vectors in Example 6.1 using $t^* = 6$ and $\alpha^* = e$. The values of $\ell(6)$ and $\ell'(6) = \ell(e \cdot 6)$ are

$$\begin{aligned}\ell(6) &= (1679620, 93312, 281448, 10365400, 334404, 329184, 1687400, 7812, 46656), \\ \ell'(6) &= (5006864730.36308, 37644747.57839, 307062197.51625, 81968557661.46344, \\ &\quad 326963800.35823, 325950992.03191, 5008018807.39795, 1154326.73120, 18822373.78920)\end{aligned}$$

so that $c = \lceil \log(\ell'(6)/\ell(6)) \rceil$ gives

$$c = \lceil (8.0, 6.0, 6.99486, 8.97562, 6.88525, 6.89789, 7.99561, 4.99561, 6.0) \rceil = (8, 6, 7, 9, 7, 7, 8, 5, 6).$$

We remark that if we use $t^* = 2$ and $\alpha^* = 1.05$, which do not necessarily meet the bounds of Theorem 3.6, we obtain the following results:

$$\begin{aligned}\ell(2) &= (256, 128, 152, 674, 228, 224, 290, 36, 64), \\ \ell'(2) &= (378.22859, 171.53224, 208.81795, 1017.32991, 311.12599, 304.77118, 421.16960, 45.25101, \\ &\quad 85.76612), \\ c &= \lceil (8.0, 6.0, 6.50915, 8.43831, 6.37121, 6.31101, 7.64815, 4.68754, 6.0) \rceil = (8, 6, 7, 9, 7, 7, 8, 5, 6).\end{aligned}$$

Hence, the correct max-convolution coefficients are recovered despite not satisfying the bounds of Theorem 3.6. \square

We conclude with two applications of max-convolution.

6.1 Maximum consecutive subsums problem

Given a single vector $v = (v_1, \dots, v_n) \in \mathbb{R}^n$, the problem of determining the largest consecutive sum $\sum_{i=1}^k v_{j_k+i}$ for each $k = 1, \dots, n$ is known as the *Maximum Consecutive Subsums Problem* (**M CSP**). As outlined in [15, § 7.1], **M CSP** directly reduces to an instance of **MAXCON** as follows. Taking $a, b \in \mathbb{R}^n$ to be $a_k = -\sum_{i=1}^k v_i$ and $b_{n-k+1} = \sum_{i=1}^k v_i$, the max-convolution coefficient c_{n-k} describes the largest sum of k consecutive entries of v .

Example 6.4 For $v = (1, 4, 2, 3, 8, 1, 1, 5, 6, 7, 5) \in \mathbb{Z}^{11}$, we have

$$\begin{aligned} a &= (-1, -5, -7, -10, -18, -19, -20, -25, -31, -38, -43), \\ b &= (43, 38, 31, 25, 20, 19, 18, 10, 7, 5, 1). \end{aligned}$$

The max-convolution of a and b is

$$c = (42, 38, 36, 33, 28, 24, 23, 18, 13, 8, 0, -1, -2, \dots).$$

For example, this shows that the largest sum of 2 and 5 consecutive entries of v is 13 and 24 obtained by $6+7$ and $1+5+6+7+5$, respectively. By convention, one may choose to prepend $c_0 = \sum_{i=1}^n v_i$ to c so as to include the subsum of n consecutive integers in the output as well. \square

We remark that even though Algorithms 1 and 2 are written for integer input, the algorithms work for floating point input as well, subject to different bounds (see Theorems 3.1 and 3.6). When using Algorithm 1, the output is an upper bound for the true max-convolution coefficients, subject to any error introduced by the FFT subroutine. Figure 18 shows the magnitude of error on the 100 outputs of a random **M CSP** problem on a vector $v \in [0, 1]^n$ for $n = 100$ and $n = 1000$ with each coordinate selected uniformly at random.

6.2 Service curve constraints

Convolution algorithms are integral in network calculus where systems model the data flow between networks [16]. The equation describing the incoming data is a monotonic input function $R(T)$, given in bits per second. The equation describing the outgoing data (after a time delay) is an output function, $R^*(T)$, also in bits per second. The function $R^*(T)$ is constrained by *service* constraints that state for any window of time, additional data outputted is bounded. The curves formed by these constraints are the result of a min-convolution between the service curve and input function $R(T)$ [16]. That is, a system with an input function $R(T)$ has an output function $R^*(T)$ that will

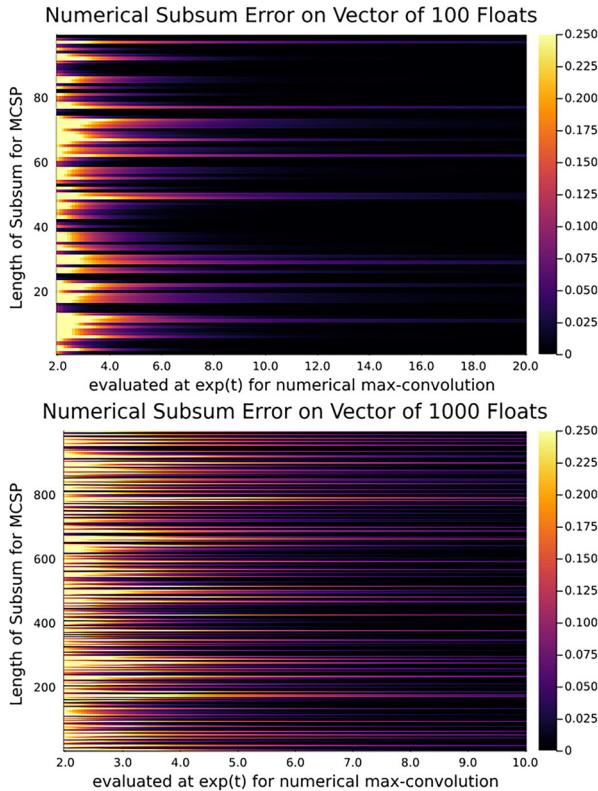


Fig. 18 Heatmap of the absolute errors of the numerical computation of **MCSP**. The numerical computations were performed on a vector of length 100 (top) and 1000 (bottom) with floating-point entries uniformly chosen in $[0, 1]$. The vertical axis indicates the output error on subsums of length k after evaluating at $\exp(t)$ (horizontal axis)

lie in the area bounded below by a service curve $\beta(T)$ and above by a maximum service curve $\gamma(T)$ such that

$$\inf_{s \leq T} \{R(T) + \beta(T - s)\} \leq R^*(T) \leq R(s) + \gamma(T - s), \quad s \leq T \quad (24)$$

Note that the input and output functions admit no subscript to avoid confusion with the ratio function $R_v(t)$. Additionally, we define the time variable for the service curve to be T rather than t which is the variable base used for the **MAXCON** algorithms.

Although the prior focus was on the maximum, it is very simple to reformulate everything to instead compute the minimum. That is, we take $t \rightarrow \infty$ when converging to the maximum while one can take $t \rightarrow 0^+$ to converge to the minimum.

As an example, we sought to recreate [16, Fig. 5.1] using Algorithm 2 to compute the discrete min-convolution of the input function and service curves. To obtain the numerical data points for this example, we first fit a polynomial curve to $R(T)$ from

the figure. In particular, we used the fitted septic polynomial

$$R(T) = 1.6738T - 0.7492T^2 - 0.08694T^3 + 0.1085T^4 - 0.01101T^5 - 0.001579T^6 + 0.0002085T^7.$$

The service curves are defined as $\beta(T) = T$ and $\gamma(T) = \begin{cases} 0 & \text{if } T \leq 3 \\ T - 3 & \text{if } T > 3, \end{cases}$ which corresponds with a 3 s time delay. To create a discrete problem, we evaluated these functions at equally spaced points.

We apply Algorithm 2 in the floating-point case, i.e., without rounding, for the computations with $\alpha = 1.01$ and $t = \left(\frac{1}{(n-1)}\right)^{\frac{1}{0.04}}$. Figure 19 shows the results of our

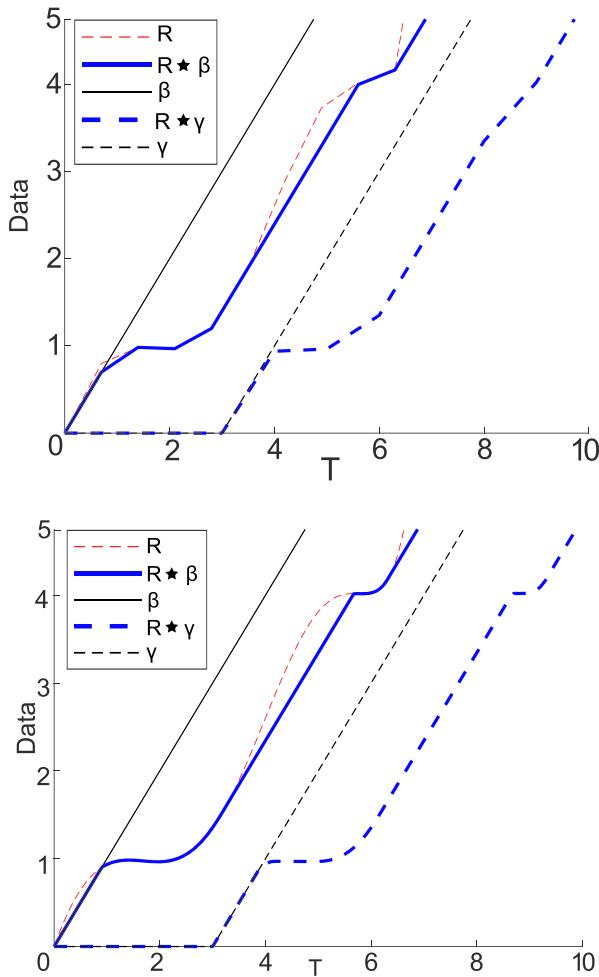


Fig. 19 Discretized service curves computed using Algorithm 2. The curve $R(T)$ and service constraints $\gamma(T)$ and $\beta(T)$ are discretized into 10 (top) and 100 (bottom) equally spaced points

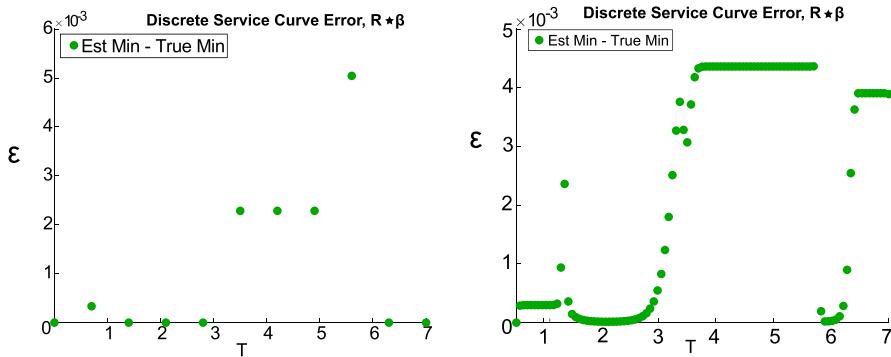


Fig. 20 Plots error of the computed points minus the actual points of the minimum convolution between the lines $R(T)$ and $\beta(T)$ for (left) 10 and (right) 100 discretized points

min-convolution using 10 and 100 discretized points to compute the corresponding bounds on $R^*(T)$ given in (24).

One can see that even with numerical discretization, the resulting curves in Fig. 19 exhibit satisfactory behavior in recreating the bounds of [16, Fig. 5.1]. Even for 10 discretization points, the computation captures the essential behavior of the convolution. Furthermore, using 100 discretization points better captures sharp transitions as well as flatter regions of the service curve bounds.

Note that this experiment employs two separate convolutions. We compared our estimated values to the actual minima computed via brute force, i.e., we computed the exact minima by computing the bounds given in (24) for each $T_i, i = 1, \dots, N$ where N is the number of discretized points. Figures 20 and 21 display the error between the computed points via Algorithm 2 and the actual points for the min-convolution between $R(T)$ and $\beta(T)$, and $R(T)$ and $\gamma(T)$, respectively. When using a time delay of 3 seconds, non-zero values first occur when $T > 3$ resulting in nearly zero error before then. Both discretizations have errors on the order of 10^{-3} or smaller. Additionally,

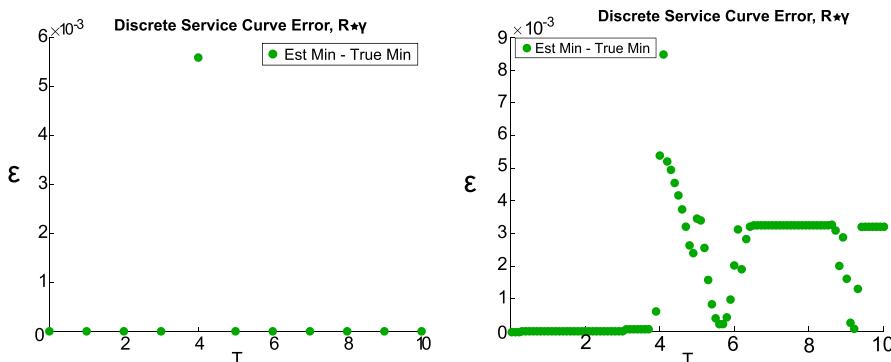


Fig. 21 Plots error of the computed points minus the actual points of the minimum convolution between the lines $R(T)$ and $\gamma(T)$ for (left) 10 and (right) 100 discretized points

Table 1 Time, in seconds, to calculate the min-convolution between $R(T)$ and $\beta(T)$

Number of discrete points	$D_v(t, \alpha)$	Brute force
10	0.0003841	0.002666
50	0.0003392	0.006488
100	0.0005358	0.01067
500	0.001053	0.04643
1000	0.001647	0.1114
5000	0.01183	1.3957
10000	0.02939	10.5240
100000	1.1738	1546.2364

the errors are nonnegative which highlights that our method slightly overestimates the values.

Tables 1 and 2 compare the time in seconds for one application of Algorithm 2 and a brute force computation between $R(T)$ and $\beta(T)$, and $R(T)$ and $\gamma(T)$, respectively, using a single processor. The difference between quasi-linear and quadratic time algorithms becomes apparent as n grows. Note that utilizing Algorithm 1 produced similar error and computational time.

7 Conclusion

Smooth approximations of the non-differentiable maximum function on vectors of real numbers are often used in optimization and machine learning. By using the lens of tropical geometry, one is able to see the differences between various smooth approximations. For example, Fig. 13 geometrically shows what slope is computed by a LogSumExp approximation and by a ratio approximation. In particular, when the maximum has multiplicity greater than 1, i.e., the maximum is repeated more than once in the vector, the LogSumExp function has a logarithmic term associated with the multiplicity that slows down convergence. A ratio approximation removes this logarithmic term associated with the multiplicity and thus converges faster when the multiplicity is greater than 1. Since ratio approximations may be difficult to evaluate

Table 2 Time, in seconds, to calculate the min-convolution between $R(T)$ and $\gamma(T)$

Number of discrete points	$D_v(t, \alpha)$	Brute force
10	0.0001468	0.002490
50	0.0001085	0.005987
100	0.0001927	0.007235
500	0.0007843	0.03623
1000	0.001421	0.1315
5000	0.006734	1.5642
10000	0.01953	5.2610
100000	0.7965	1239.4949

in practice, we propose a discretized approximation in (18) computed using two Log-SumExp evaluations that maintains the faster convergence of the ratio approximation when the multiplicity is greater than 1. In a noisy model where the values may not be exact, an exact high multiplicity maximum is perturbed into a cluster of points around the maximum. Experiments show that these results persist even with noise.

By reducing max-convolution (and min-convolution) to classical convolution, Algorithms 1 and 2 compute the max-convolution in quasi-linear time using the Log-SumExp and discretized approximation, respectively. Tables 1 and 2 clearly show the advantage of using a quasi-linear time, $O(n \log(n))$, algorithm over the brute force quadratic time, $O(n^2)$, algorithm as n increases when computing service curves.

Author contribution J. D. H. and T. B. developed the theories and algorithms and wrote the main manuscript. T. B. and C. H. tested the convergence rates; T.B. prepared Figs. 1–13 and 16–18. T.B. analyzed and wrote Section 4 regarding tropical geometry and wrote Section 6.1. C.H. tested and prepared Figs. 14–15 and 19–20 and wrote Sects. 5.1–5.2 and 6.2. All authors reviewed the manuscript.

Funding This material is based upon work supported by the National Science Foundation under grants CCF-1812746 and CMMI-2041789. The first author is partially supported by an NSERC Discovery Grant (Canada).

Data availability All data is described in the text.

Declarations

Ethical approval Not applicable

Conflict of interest The authors declare no competing interests.

References

1. Zang, I.: A smoothing-out technique for min-max optimization. *Math. Programming*. **19**(1), 61–77 (1980)
2. Zhao, G., Wang, Z., Mou, H.: Uniform approximation of min/max functions by smooth splines. *J. Comput. Appl. Math.* **236**(5), 699–703 (2011). The 7th International Conference on Scientific Computing and Applications, June 13–16, 2010, Dalian, China
3. Asadi, K., Littman, M.L.: An alternative softmax operator for reinforcement learning. In: Proceedings of the 34th International Conference on Machine Learning, Vol 70. ICML’17, pp. 243–252. JMLR.org, (Online) (2017)
4. Liu, W., Wen, Y., Yu, Z., Yang, M.: Large-margin softmax loss for convolutional neural networks. In: Balcan, M., Weinberger, K. (eds.) International Conference on Machine Learning, Vol 48. Proceedings of Machine Learning Research, vol. 48 (2016). 33rd International Conference on Machine Learning, New York, NY, JUN 20–22, 2016
5. Nielsen, F., Sun, K.: Guaranteed bounds on information-theoretic measures of univariate mixtures using piecewise log-sum-exp inequalities. *Entropy*. **18**(12) (2016). <https://doi.org/10.3390/e18120442>
6. Blanchard, P., Higham, D.J., Higham, N.J.: Accurately computing the log-sum-exp and softmax functions. *IMA J. Numer. Anal.* **41**(4), 2311–2330 (2020). <https://doi.org/10.1093/imanum/draa038>. <https://academic.oup.com/imajna/articlepdf/41/4/2311/40758053/draa038.pdf>
7. Trefethen, L.N., Weideman, J.A.C.: The exponentially convergent trapezoidal rule. *SIAM Review*. **56**(3), 385–458 (2014)
8. Pfeuffer, J., Serang, O.: A bounded p-norm approximation of max-convolution for sub-quadratic Bayesian inference on additive factors. *J. Mach. Learn. Res.* **17**, 36–13639 (2016)

9. Serang, O.: A fast numerical method for max-convolution and the application to efficient max-product inference in bayesian networks. *J. Comput. Biol. J. Mol. Cell. Biol.* **22** (2015). <https://doi.org/10.1089/cmb.2015.0013>
10. Maclagan, D., Sturmfels, B.: Introduction to tropical geometry. *Graduate Studies in Mathematics*, vol. 161. American Mathematical Society, Providence, Rhode Island (2015)
11. Bergman, G.M.: The logarithmic limit-set of an algebraic variety. *Trans. Am. Math. Soc.* **157**, 459–469 (1971)
12. DeWolff, T., Schroeter, F.: The boundary of amoebas. arXiv Preprint: - [arXiv:1310.7363](https://arxiv.org/abs/1310.7363) (2013)
13. Forsberg, M., Passare, M., Tsikh, A.: Laurent determinants and arrangements of hyperplane amoebas. *Adv. Math.* **151**, 45–70 (2000)
14. Moenck, R.T.: Practical fast polynomial multiplication. In: Proceedings of the Third ACM Symposium on Symbolic and Algebraic Computation. SYMSAC' 76, pp. 136–148. Association for Computing Machinery, New York, NY, USA (1976)
15. Cygan, M., Mucha, M., Wundefinedgrzycki, K., Włodarczyk, M.: On problems equivalent to $(\min, +)$ -convolution. *ACM Trans. Algorithms* **15**(1) (2019)
16. Van Bemten, A., Kellerer, W.: Network calculus: a comprehensive guide (2016). <https://doi.org/10.13140/RG.2.2.32305.89448>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.