Entropy **2014**, *16*, 627-644; doi:10.3390/e16020627

OPEN ACCESS

entropy

ISSN 1099-4300

www.mdpi.com/journal/entropy

Article

A Simplified Algorithm for the Topological Entropy of Multimodal Maps

José M. Amigó * and Ángel Giménez

Centro de Investigación Operativa, Universidad Miguel Hernández, Avda. de la Universidad s/n, Elche 03202, Spain; E-Mail: a.gimenez@umh.es

* Author to whom correspondence should be addressed; E-Mail: jm.amigo@umh.es; Tel.: +34-966658911.

Received: 5 December 2013; in revised form: 14 January 2014 / Accepted: 14 January 2014 / Published: 23 January 2014

Abstract: A numerical algorithm to compute the topological entropy of multimodal maps is proposed. This algorithm results from a closed formula containing the so-called min-max symbols, which are closely related to the kneading symbols. Furthermore, it simplifies a previous algorithm, also based on min-max symbols, which was originally proposed for twice differentiable multimodal maps. The new algorithm has been benchmarked against the old one with a number of multimodal maps, the results being reported in the paper. The comparison is favorable to the new algorithm, except in the unimodal case.

Keywords: topological entropy; multimodal maps; min-max symbols

1. Introduction

Let f be a continuous self-map of a compact interval, $[a, b] \subset \mathbb{R}$, with a finite number of turning (or critical) points. Such maps are generically called multimodal. Then, the topological entropy of f [1,2], $h(f)$, can be calculated (along with other possibilities) with the formula:

$$h(f) = \lim_{n\to\infty} \frac{1}{n} \log \ell_n \tag{1}$$

where ℓ_n is shorthand for the *lap number* of f^n, *i.e.*, the number of maximal monotonicity segments of f^n, the n-th iterate of f [3,4].

In [5] (Section 7), a numerical algorithm to compute the topological entropy of multimodal maps was proposed. Let us point out that this algorithm generalizes and hence includes a previous one for

unimodal maps published in [6]. The algorithm builds on Equation (1) by calculating ℓ_n with the help of the *min-max symbols* of f [5–8], a generalization of the kneading symbols [9,10]. The min-max symbols of a multimodal map not only locate the iterates of its critical values up to the precision set by the partition defined by its critical points, as the kneading symbols do, but they also display their minimum/maximum (or "critical") character. The interesting point is that such an additional information supposes virtually no extra computational cost. Indeed, it can be read recursively from a look-up table once the min-max symbols of the critical values are known.

In this paper, we propose a related algorithm, which actually approximates the value of $h(f)$ given by a closed formula involving also the min-max symbols of f. The new algorithm eliminates a formal restriction that, as it turns out, unnecessarily marred the applicability of the algorithm of [5]. At the same time, it simplifies the computation scheme of the latter. We elaborate upon these two points briefly.

With regard to the formal restriction, the theoretical results of [5] refer to twice differentiable multimodal maps only. However, numerical simulations with piecewise linear maps of constant slope (and alternating sign) suggested that the algorithm of [5] could be applied as well to just continuous maps. In this paper, we justify the extension of the results from smooth to just continuous maps. Although the proof turns out to be straightforward, this generalization was not explored in the previous papers [5,6], just because these followed the original approach in [7,8], which only considered twice differentiable maps for simplicity.

As for the simplification of the computation scheme, this has to do with the boundary conditions (or the lack of them). Indeed, the algorithm of [5] keeps track of the orbits of the boundary points, thus calculating the exact value of the lap number, ℓ_n, in each computation loop. On the contrary, the new algorithm dispenses with those orbits, because they do not affect the limit in Equation (1). In fact, Theorem 3 below shows that, as far as the computation of $h(f)$ is concerned, one may assume that f is boundary-anchored, *i.e.*, $f(\{a,b\}) \subset \{a,b\}$. The result is a compact expression for the lap number, ℓ_n, that makes possible a closed formula for $h(f)$.

In sum, we fill a theoretical gap in the application of the algorithm in [5] by showing that the continuity of the maps suffices. Moreover, we abridge the numerical scheme by approximating ℓ_n in Equation (1) with a formula, which is exact only for boundary-anchored maps, but which provides the correct limit in Equation (1) for $h(f)$.

This paper is organized as follows. To make the paper self-contained, we review in Section 2 all the basic concepts, especially the concept of min-max sequences, needed in the following sections. Most importantly, we extend in Theorem 1 the transition rules for min-max symbols from twice differentiable multimodal maps [5] to just continuous ones. In Section 3, we introduce some instrumental results, which lead, in Section 4, together with Theorem 3, to a closed formula for $h(f)$ containing the min-max symbols of f (Theorem 4). A formal proof of Theorem 3 has been shifted to the Appendix in order not to interrupt the flow of ideas. Section 5 contains the main result of the paper, namely, an algorithm for the topological entropy of (not necessarily smooth) multimodal maps, which approximates the value of $h(f)$ given in Theorem 4. As a way of illustration, this algorithm is put to the test in Section 6. First, the new, abridged algorithm is benchmarked in Sections 6.1 to 6.3 against the full-pledged one [5] using smooth uni-, bi- and tri-modal maps, respectively, borrowed from [5,6]. Finally, in Section 6.4, we also

compare both algorithms via piecewise linear, four- and five-modal maps of known topological entropy. It turns out that, except in the unimodal case, the new algorithm outperforms the old one.

2. Min-Max Sequences

For the reader's convenience, we use the same notation as in [5] throughout. Let I be a compact interval, $[a, b] \subset \mathbb{R}$, and $f : I \to I$ a piecewise monotone continuous map. Such a map is called l-modal if f has precisely l *turning points* (*i.e.*, points in (a, b), where f has a local extremum). Sometimes, we speak also of multimodal maps, in general, or of unimodal maps in the particular case $l = 1$. Furthermore, assume henceforth that f has local extrema at $c_1 < ... < c_l$ and is strictly monotone in each of the $l + 1$ intervals:

$$I_1 = [a, c_1), I_2 = (c_1, c_2), ..., I_l = (c_{l-1}, c_l), I_{l+1} = (c_l, b]$$

In this case, we write $f \in \mathcal{M}_l(I)$. When the interval, I, is clear from the context or unimportant for the argument, we write just $\mathcal{M}_l$.

Since the results we obtain below for the calculation of the topological entropy do not depend on the *shape* of f, *i.e.*, on whether $f(c_1)$ is a maximum (positive shape) or a minimum (negative shape), we assume, unless otherwise stated, that f has a positive shape. This implies that $f(c_{\text{odd}})$ are maxima, whereas $f(c_{\text{even}})$ are minima, where here and hereafter "even" and "odd" stand for even and odd subindices, respectively. Hence, f is strictly increasing on the intervals I_{odd} and strictly decreasing on the intervals I_{even}. The points, $f(c_i)$, $1 \leq i \leq l$, are called the critical values of f, although no differentiability of f at c_i is assumed when so doing.

Theorem 1. Let $f \in \mathcal{M}_l$ have a positive shape, and $n \geq 1$. Then:

(a) If $f^n(x) = c_{\text{odd}}$, then $f^{n+1}(x)$ is a maximum. If $f^n(x) = c_{\text{even}}$, then $f^{n+1}(x)$ is a minimum.

(b) If $f^n(x)$ is a minimum, then:

$$f^{n+1}(x) \text{ is a } \begin{cases} \text{minimum} & \text{if } f^n(x) \in I_{\text{odd}} \\ \text{maximum} & \text{if } f^n(x) \in I_{\text{even}} \end{cases}$$

(c) If $f^n(x)$ is a maximum, then:

$$f^{n+1}(x) \text{ is a } \begin{cases} \text{maximum} & \text{if } f^n(x) \in I_{\text{odd}} \\ \text{minimum} & \text{if } f^n(x) \in I_{\text{even}} \end{cases}$$

Proof. (a) This is a trivial consequence of f having a positive shape.

(b) Suppose that $f^n(x_0)$ is a minimum with $f^n(x_0) \in I_{\text{odd}}$. Therefore, there exists a neighborhood of x_0, $U(x_0)$, such that $f^n(x_0) \leq f^n(x)$ for all $x \in U(x_0)$. Without restriction, we may assume that $U(x_0) \subset f^{-n}(I_{\text{odd}})$. It follows that:

$$f^{n+1}(x_0) = f(f^n(x_0)) \leq f(f^n(x)) = f^{n+1}(x)$$

for all $x \in U(x_0)$, because $f^n(U(x_0)) \subset I_{\text{odd}}$, an interval where f is increasing. We conclude that $f^{n+1}(x_0)$ is a minimum.

If $f^n(x_0) \in I_{\text{even}}$, then we derive from $f^n(x_0) \leq f^n(x)$ that $f^{n+1}(x_0) \geq f^{n+1}(x)$ for all $x \in U(x_0)$, because this time, $f^n(U(x_0)) \subset I_{\text{even}}$, an interval where f is decreasing.

(c) This case follows similarly to (b). □

The *itinerary* of $x \in I$ under f is a symbolic sequence:

$$\mathbf{i}(x) = (i_0(x), i_1(x), ..., i_n(x), ...) \in \{I_1, c_1, I_2, ..., c_l, I_{l+1}\}^{\mathbb{N}_0}$$

($\mathbb{N}_0 \equiv \{0\} \cup \mathbb{N}$), defined as follows:

$$i_n(x) = \begin{cases} I_j & \text{if } f^n(x) \in I_j \ (1 \leq j \leq l+1) \\ c_k & \text{if } f^n(x) = c_k \ (1 \leq k \leq l) \end{cases}$$

The itineraries of the critical values,

$$\gamma^i = (\gamma_1^i, ..., \gamma_n^i, ...) = \mathbf{i}(f(c_i)), \ 1 \leq i \leq l$$

are called the *kneading sequences* [9,10] of f.

Definition 1 [5–8]. The *min-max sequences* of an l-modal map f,

$$\omega^i = (\omega_1^i, \omega_2^i, ..., \omega_n^i, ...), \ 1 \leq i \leq l$$

are defined as follows:

$$\omega_n^i = \begin{cases} m^{\gamma_n^i} & \text{if } f^n(c_i) \text{ is a minimum} \\ M^{\gamma_n^i} & \text{if } f^n(c_i) \text{ is a maximum} \end{cases}$$

where γ_n^i are kneading symbols.

Thus, the *min-max symbols* ω_n^i have an exponential-like notation, where the "base" belongs to the alphabet, $\{m, M\}$, and the "exponent" is a kneading symbol. The extra information of a min-max symbol, ω_n^i, as compared to a kneading symbol, γ_n^i, is contained, therefore, in the base, which tells us whether $f^n(c_i)$ is a minimum (m) or a maximum (M). Theorem 1 shows that once the symbol, ω_n^i, of a map with a positive shape is known, the symbol, ω_{n+1}^i, can be read from the table:

ω_n^i	$\rightarrow$	ω_{n+1}^i
$m^{c_{\text{even}}}, M^{c_{\text{even}}}$	$\rightarrow$	$m^{\gamma_{n+1}^i}$
$m^{c_{\text{odd}}}, M^{c_{\text{odd}}}$	$\rightarrow$	$M^{\gamma_{n+1}^i}$
$m^{I_{\text{odd}}}, M^{I_{\text{even}}}$	$\rightarrow$	$m^{\gamma_{n+1}^i}$
$m^{I_{\text{even}}}, M^{I_{\text{odd}}}$	$\rightarrow$	$M^{\gamma_{n+1}^i}$

(2)

Let us mention for completeness that if $f \in \mathcal{M}_l$ has a negative shape, then the transition rules from ω_n^i to ω_{n+1}^i read:

ω_n^i	$\rightarrow$	ω_{n+1}^i
$m^{c_{\text{even}}}, M^{c_{\text{even}}}$	$\rightarrow$	$M^{\gamma_{n+1}^i}$
$m^{c_{\text{odd}}}, M^{c_{\text{odd}}}$	$\rightarrow$	$m^{\gamma_{n+1}^i}$
$m^{I_{\text{odd}}}, M^{I_{\text{even}}}$	$\rightarrow$	$M^{\gamma_{n+1}^i}$
$m^{I_{\text{even}}}, M^{I_{\text{odd}}}$	$\rightarrow$	$m^{\gamma_{n+1}^i}$

(3)

instead. This follows *mutatis mutandis* as in the proof of Theorem 1.

The transition rules in Equations (2) and (3) substantiate our claim in the Introduction that, from the point of view of the computational cost, min-max sequences and kneading sequences are virtually equivalent.

Therefore, the kneading symbols of $f \in \mathcal{M}_l$, along with its *initial min-max symbols*, *i.e.*,

$$\omega_1^i = \begin{cases} M^{\gamma_1^i} & \text{if } i = 1, 3, ..., 2\left\lfloor \frac{l+1}{2} \right\rfloor - 1 \\ m^{\gamma_1^i} & \text{if } i = 2, 4, ..., 2\left\lfloor \frac{l}{2} \right\rfloor \end{cases} \tag{4}$$

and the transition rules (2) allow us to compute the min-max sequences of f in a recursive way. If f has a negative shape, swap M and m in the initial condition (4) and use the transition rules (3).

A final ingredient (proper of min-max sequences) is the following. Let the *i-th critical line*, $1 \leq i \leq l$, be the line $y = c_i$ in the Cartesian product $I \times I$. Min-max symbols split into *bad* and *good symbols* with respect to the i-th critical line. Geometrically, good symbols correspond to local maxima strictly above the line $y = c_i$ or to local minima strictly below the line $y = c_i$. All other min-max symbols are:

$$\mathcal{B}^i = \{M^{I_1}, M^{c_1}, ..., M^{I_i}, M^{c_i}, m^{c_i}, m^{I_{i+1}}, ..., m^{c_l}, m^{I_{l+1}}\}$$

for the set of bad symbols of $f \in \mathcal{M}_l$ with respect to the i-th critical line. There are $2(l+1)$ bad symbols and $2l$ good symbols with respect to a given critical line.

Bad symbols appear in all results of [5,6] concerning the computation of the topological entropy of $f \in \mathcal{M}_l$ via min-max symbols. In this sense, we may say that bad symbols are the hallmark of this approach.

3. Auxiliary Results

Let s_ν^i, $1 \leq i \leq l$, stand for the *number of interior simple zeros of* $f^\nu(x) - c_i$, $\nu \geq 0$, *i.e.*, solutions of $x - c_i = 0$ ($\nu = 0$) or solutions of $f^\nu(x) = c_i$, $x \in (a, b)$, with $f^\mu(x) \neq c_i$ for $0 \leq \mu \leq \nu - 1$ and $f^{\nu\prime}(x) \neq 0$ ($\nu \geq 1$). Geometrically, s_ν^i is the number of *transversal* intersections on the Cartesian plane (x, y) of the curve $y = f^\nu(x)$ and the straight line $y = c_i$, over the interval (a, b). Note that $s_0^i = 1$ for all i.

To streamline the notation, set:

$$s_\nu = \sum_{i=1}^{l} s_\nu^i \tag{5}$$

for $\nu \geq 0$. In particular,

$$s_0 = \sum_{i=1}^{l} s_0^i = \sum_{i=1}^{l} 1 = l \tag{6}$$

According to [5] (Equation (31)), the lap number of f^n, ℓ_n, satisfies:

$$\ell_n = 1 + \sum_{\nu=0}^{n-1} s_\nu = \ell_{n-1} + s_{n-1} \tag{7}$$

for $n \geq 1$. In particular, $\ell_1 = \ell_0 + s_0 = 1 + l$.

Furthermore, define:

$$K_\nu^i = \{(k, \kappa), 1 \leq k \leq l, 1 \leq \kappa \leq \nu : \omega_\kappa^k \in \mathcal{B}^i\} \tag{8}$$

($\nu \geq 1$, $1 \leq i \leq l$), that is, K_ν^i collects the upper and lower indices (k, κ) of the *bad* symbols with respect to the i-th critical line in all the initial blocks:

$$\omega_1^1, \omega_2^1, ..., \omega_\nu^1; \;\; \omega_1^2, \omega_2^2, ..., \omega_\nu^2; \;\; ...; \;\; \omega_1^l, \omega_2^l, ..., \omega_\nu^l$$

of the min-max sequences of f. We note for further reference that $K_{\nu-1}^i \subset K_\nu^i$, the set-theoretical difference being:

$$K_\nu^i \backslash K_{\nu-1}^i = \{(k, \nu), 1 \leq k \leq l : \omega_\nu^k \in \mathcal{B}^i\} \tag{9}$$

Finally, set:

$$S_\nu^i = 2 \sum_{(k,\kappa) \in K_\nu^i} s_{\nu-\kappa}^k \tag{10}$$

where $S_\nu^i = 0$ if $K_\nu^i = \emptyset$ and analogously to Equation (5),

$$S_\nu = \sum_{i=1}^{l} S_\nu^i \tag{11}$$

We say that $f \in \mathcal{M}_l$ is boundary-anchored if $f\{a, b\} \subset \{a, b\}$. This boundary condition boils down to:

$$f(a) = a, \text{ and } f(b) = \begin{cases} a & \text{if } l \text{ is odd} \\ b & \text{if } l \text{ is even} \end{cases} \tag{12}$$

for multimodal maps with a positive shape, and to:

$$f(a) = b, \text{ and } f(b) = \begin{cases} b & \text{if } l \text{ is odd} \\ a & \text{if } l \text{ is even} \end{cases} \tag{13}$$

for multimodal maps with a negative shape. As we will see shortly, boundary-anchored maps have some advantages when calculating the topological entropy.

Theorem 2. Let $f \in \mathcal{M}_l$ be boundary-anchored. Then:

$$s_\nu^i = 1 + \sum_{\mu=0}^{\nu-1} s_\mu - S_\nu^i \tag{14}$$

Proof. Suppose for the time being that f is twice differentiable on $[a, b]$ without any restriction at the boundaries. In this case, it was proved in ([5], Theorem 5.3) that:

$$s_\nu^i = 1 + \sum_{\mu=0}^{\nu-1} s_\mu - S_\nu^i - \alpha_\nu^i - \beta_\nu^i \tag{15}$$

where α_ν^i,β_ν^i are binary variables that vanish if f is boundary-anchored. Moreover, Equation (15) follows from the transition rules in Equation (2) (or (3) if f has a negative shape), which has been proven to hold true also for continuous multimodal maps in Theorem 1. It follows that Equation (15) holds for continuous, multimodal maps, as well. In particular, Equation (14) holds for the boundary-anchored ones. □

Therefore, the boundary conditions (12) or (13) allow us to express s_ν^i with the help of some s_0^k, s_1^k, ..., $s_{\nu-1}^k$, $1 \leq k \leq l$, via Equations (14) and (10). Sum the Equation (14) over i from one to l to obtain the compact relation:

$$s_\nu = l\left(1 + \sum_{\mu=0}^{\nu-1} s_\mu\right) - S_\nu \tag{16}$$

between $s_0 = l, s_1, ..., s_\nu$ and S_ν, for all $\nu \geq 1$. By Equation (7), this equation can we rewritten as $s_\nu = l\ell_\nu - S_\nu$, hence:

$$\ell_\nu = \frac{1}{l}(s_\nu + S_\nu) \tag{17}$$

4. A Closed Formula for the Topological Entropy of Multimodal Maps

According to [11] (Lemma 4.4), the topological entropy of a multimodal map depends only on the kneading sequences, *i.e.*, on the itineraries of the critical values, but not on the itineraries of the boundary points. This entails that one may assume without restriction boundary conditions (12) or (13) when calculating the topological entropy of l-modal maps with a positive or negative shape, respectively. A formal justification is given by the following theorem.

Theorem 3. Let $f \in \mathcal{M}_l(I)$. Then, there exists $F \in \mathcal{M}_l(J)$, where $J \supset I$, such that $h(F) = h(f)$ and F is boundary-anchored.

See [9] (Lemma 7.7) and [11] (proof of Lemma 4.4). For the reader's convenience, a proof of Theorem 3 is given in the Appendix.

This being the case, Equations (1) and (17) yield the following result.

Theorem 4. Let $f \in \mathcal{M}_l$. Then,

$$\begin{aligned} h(f) &= \lim_{\nu\to\infty} \frac{1}{\nu} \log \frac{s_\nu + S_\nu}{l} \\ &= \lim_{\nu\to\infty} \frac{1}{\nu} \log \frac{1}{l} \sum_{i=1}^{l} \left(s_\nu^i + 2 \sum_{(k,\kappa)\in K_\nu^i} s_{\nu-\kappa}^k \right) \end{aligned} \tag{18}$$

Equation (18) provides a closed expression for $h(f)$, which includes the min-max symbols of f.

5. A Simplified Algorithm for the Topological Entropy

An offshoot of the preceding section is that, when it comes to calculating the topological entropy of a multimodal map, one can resort to Equation (18), whether the map is boundary-anchored or not. Loosely speaking,

$$h(f) \simeq \frac{1}{\nu} \log \frac{s_\nu + S_\nu}{l} \tag{19}$$

for ν large enough.

As a matter of fact, the numerical algorithm below estimates $h(f)$ by $\frac{1}{\nu} \log \frac{s_\nu + S_\nu}{l}$ to the desired precision. The core of the algorithm consists of a loop over ν. Each time the algorithm enters the loop, the values of $s_{\nu-1}$ and $S_{\nu-1}$ are updated to s_ν and S_ν, and the current estimation of $h(f)$ is compared to the

previous one. Note that the computation of S_ν^i, $1 \leq i \leq l$, requires $s_0^i = 1, s_1^i, ..., s_{\nu-1}^i$, see Equation (10), while the computation of s_ν^i, $1 \leq i \leq l$, requires $s_0^i, s_1^i, ..., s_{\nu-1}^i$ and S_ν^i, see Equation (14).

We summarize next the algorithm resulting from Equation (18) in the following scheme ("$A \longrightarrow B$" stands for "B is computed by means of A").

(A1) Parameters: $l \geq 1$ (number of critical points), $\varepsilon > 0$ (dynamic halt criterion) and $n_{\max} \geq 2$ (maximum number of loops).

(A2) Initialization: $s_0^i = 1$, and $K_1^i = \{k, 1 \leq k \leq l : \omega_1^k \in \mathcal{B}^i\}$ $(1 \leq i \leq l)$.

(A3) First iteration: For $1 \leq i \leq l$,

$$\begin{aligned} s_0^i, K_1^i &\longrightarrow S_1^i, S_1 \quad \text{(use Equations (10) and (11))} \\ s_0^i, S_1^i &\longrightarrow s_1^i, s_1 \quad \text{(use Equations (14) and (16))} \end{aligned}$$

(A4) Computation loop. For $1 \leq i \leq l$ and $\nu \geq 2$, keep calculating K_ν^i, S_ν^i and s_ν^i according to the recursions:

$$\begin{aligned} K_{\nu-1}^i &\longrightarrow K_\nu^i \quad \text{(use Equations (9) and (2))} \\ s_0^i, s_1^i, ..., s_{\nu-1}^i, K_\nu^i &\longrightarrow S_\nu^i, S_\nu \quad \text{(use Equations (10) and (11))} \\ s_0^i, s_1^i, ..., s_{\nu-1}^i, S_\nu^i &\longrightarrow s_\nu^i, s_\nu \quad \text{(use Equations (14) and (16))} \end{aligned} \tag{20}$$

until (i)

$$\left| \frac{1}{\nu} \log \frac{s_\nu + S_\nu}{l} - \frac{1}{\nu - 1} \log \frac{s_\nu + S_\nu}{l} \right| \leq \varepsilon \tag{21}$$

or else, (ii) $\nu = n_{\max} + 1$.

(A5) Output. In case (i), output:

$$h(f) = \frac{1}{\nu} \log \frac{s_\nu + S_\nu}{l} \tag{22}$$

In case (ii), output "Algorithm failed".

As said above, Algorithms (A1)–(A5) simplify the original algorithm [5], which formally consists of the same five steps above, but is based on the exact value of the lap number, ℓ_ν. This entails that the new algorithm needs more loops to output $h(f)$ with the same parameter ε in the halt criterion shown in Equation (21), although this does not necessarily mean that the overall execution time will be longer, since now, less computations are required. In fact, we will find both situations in the numerical simulations of Section 6.

Furthermore, given a halt criterion, ε, the execution time depends, as well, on the units (*i.e.*, on the base of the logarithm), whichever algorithm is used. For instance, if logarithms to base e are used (*i.e.*, $h(f)$ in *nats*) and $\nu = n_{\text{nat}}$ is the first time that the halt criterion,

$$\left| \ln \frac{s_\nu + S_\nu}{l} - \ln \frac{s_{\nu-1} + S_{\nu-1}}{l} \right| \leq \varepsilon$$

happens to hold in the computation loop, then:

$$\begin{aligned} \left| \log_2 \frac{s_{n_{\text{nat}}} + S_{n_{\text{nat}}}}{l} - \log_2 \frac{s_{n_{\text{nat}}-1} + S_{n_{\text{nat}}-1}}{l} \right| &= \frac{1}{\ln 2} \left| \ln \frac{s_{n_{\text{nat}}} + S_{n_{\text{nat}}}}{l} - \ln \frac{s_{n_{\text{nat}}-1} + S_{n_{\text{nat}}-1}}{l} \right| \\ &\leq \frac{\varepsilon}{\ln 2} = 1.4427\varepsilon \end{aligned}$$

Therefore, if the the halt criterion:

$$\left|\log_2 \frac{s_\nu + S_\nu}{l} - \log_2 \frac{s_{\nu-1} + S_{\nu-1}}{l}\right| \leq \varepsilon$$

for the computation of $h(f)$ in bits does not hold when $\nu = n_{\text{nat}}$, *i.e.*,

$$\left|\ln \frac{s_{n_{\text{nat}}} + S_{n_{\text{nat}}}}{l} - \ln \frac{s_{n_{\text{nat}}-1} + S_{n_{\text{nat}}-1}}{l}\right| > (\ln 2)\varepsilon = 0.6932\varepsilon$$

then the algorithm will not exit the computation loop. We conclude that $n_{\text{bit}} \geq n_{\text{nat}}$ with both algorithms, where n_{bit} is the exit loop when logarithms to base 2 are employed.

Two final remarks:

R1. The parameter ε does not bound the error $\left|h(f) - \frac{1}{\nu}\log\frac{s_\nu+S_\nu}{l}\right|$, but the difference between two consecutive estimations; see Equation (21). The number of exact decimal positions of $h(f)$ can be found out by taking different ε's, as we will see in the next section. Equivalently, one can control how successive decimal positions of $\frac{1}{\nu}\log\frac{s_\nu+S_\nu}{l}$ stabilize with growing ν. Moreover, the smaller $h(f)$, the smaller ε has to be chosen to achieve a given approximation precision.

R2. According to [4] (Theorem 4.2.4), $\frac{1}{\nu}\log \ell_\nu \geq h(f)$ for any ν. We may expect, therefore, that the numerical approximations (22) converge from above to the true value of the topological entropy with ever more iterations, in spite of the relation $\ell_\nu = \frac{1}{l}(s_\nu + S_\nu)$ holding in general for boundary-anchored maps only.

6. Numerical Simulations

In this section, we compute the topological entropy of a variety of multimodal maps. To this end, a code for arbitrary l was written with PYTHON and run on an Intel(R) Core(TM)2 Duo CPU. All the numerical results will be given with six decimal positions for brevity.

Thus, in Section 6.1 to 6.3, we calculate the entropy of families of uni-, bi- and tri-modal maps, respectively, taken from [6] (unimodal case) and [5] (general case). Except for particular values of the parameters, these maps are not boundary-anchored. The purpose of our choice is to compare our entropy plots with the plots published in those references. To complete the picture, we will consider non-smooth maps in Section 6.4. The natural choices are piecewise linear maps of a constant slope, because, in this case, the exact value of the topological entropy is known. In all sections, we are going to compare numerically the performance of the algorithm presented in Section 5 with the general algorithm presented in [5] (Section 7) by means of single maps. For brevity, we shall refer to them as the new algorithm and the old one, respectively.

As for the units, the *nat* is the usual choice in applied mathematics and physics, while the *bit* is the standard unit in information theory and communication technologies. In the following subsections, we are actually going to use both of them despite the fact that, as shown in Section 5, computations with Napierian logarithms are faster, to a given precision. To be specific, we use *bits* in Sections 6.2 and 6.3 for the sake of comparison with the results published in [5], which are given in that unit.

6.1. Simulation with One-Modal Maps

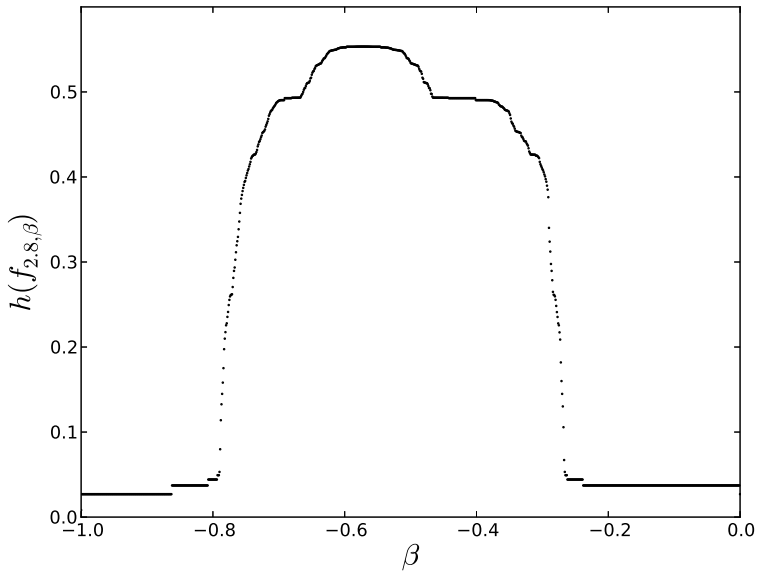
Let $\alpha > 0$, $-1 < \beta \leq 0$ and $f_{\alpha,\beta} : [-(1+\beta),(1+\beta)] \to [-(1+\beta),(1+\beta)]$ be defined as ([6], Equation (29)):

$$f_{\alpha,\beta}(x) = e^{-\alpha^2 x^2} + \beta$$

These maps have the peculiarity of showing direct and reverse period-doubling bifurcations when the parameters are monotonically changed (see Figure 3a in [6]).

Figure 1 shows the plot of $h(f_{2.8,\beta})$ *vs.* β calculated with the algorithm of Section 7. Here, $\varepsilon = 10^{-4}$, and the parameter, β, was increased in steps of $\Delta\beta = 0.001$ from $\beta = -0.999$ to $\beta = 0$. Upon comparing Figure 1 with Figure 3b of [6], we see that both plots coincide visually, except for the two vanishing entropy tails. We conclude that $\varepsilon = 10^{-4}$ is not small enough to obtain reliable estimations of the topological entropy for vanishing values of $h(f_{2.8,\beta})$. This fact can also be ascertained numerically by taking different values of ε, as we do in the table below.

Figure 1. Plot of $h(f_{2.8,\beta})$ in nats *vs.* β, $-1 < \beta \leq 0$ ($\varepsilon = 10^{-4}$, $\Delta\beta = 0.001$).



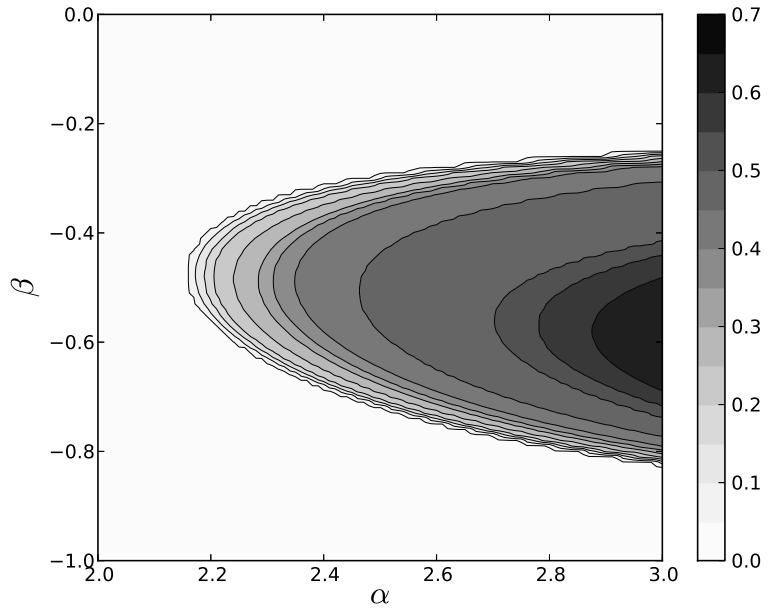
To compare the convergence speed and execution time of the old and the new algorithm, we have computed $h(f_{2.8,-0.5})$ with both algorithms for different ε's. The number of loops, n, needed to achieve the halt condition $\varepsilon = 10^{-d}$, $4 \leq d \leq 7$, and the execution time, t (in seconds), are listed in Table 1. The columns, h_{old}, n_{old} and t_{old}, were obtained with the old algorithm, while the columns, h_{new}, n_{new} and t_{new}, were obtained with the new one. For $\varepsilon = 10^{-4}$, it exceptionally holds $t_{old} > t_{new}$; otherwise $t_{old} < t_{new}$. Furthermore, we conclude from Table 1 that $h(f) = 0.52...$ nats with either algorithm and $\varepsilon = 10^{-6}$, both decimal digits being exact. If $\varepsilon = 10^{-7}$, the old algorithm fixes the third decimal digit, $h(f) = 0.524...$ nats, whereas the new algorithm does not.

Table 1. Comparison of performances when computing $h(f_{2.8,-0.5})$ in nats.

	h_{old}	n_{old}	t_{old}	h_{new}	n_{new}	t_{new}
$\varepsilon = 10^{-4}$	0.531968	81	0.031076	0.534106	101	0.021248
$\varepsilon = 10^{-5}$	0.526645	253	0.179558	0.527305	318	0.193149
$\varepsilon = 10^{-6}$	0.524935	797	1.684213	0.525142	1,004	1.912784
$\varepsilon = 10^{-7}$	0.524391	2,519	16.369158	0.524456	3,174	18.900032

Figure 2 depicts the values of $h(f_{\alpha,\beta})$ for $2 \leq \alpha \leq 3$, $-1 < \beta \leq 0$, $\varepsilon = 10^{-4}$ and $\Delta\alpha, \Delta\beta = 0.01$.

Figure 2. Level sets of $h(f_{\alpha,\beta})$ in nats *vs.* α, β, $2 \leq \alpha \leq 3$ and $-1 < \beta \leq 0$ ($\varepsilon = 10^{-4}, \Delta\alpha = \Delta\beta = 0.01$).



6.2. Simulation with Two-Modal Maps

Let $0 \leq v_2 < v_1 \leq 1$ and $f_{v_1,v_2} : [0,1] \to [0,1]$ be defined as ([5], Section 8.1):

$$f_{v_1,v_2}(x) = (v_1 - v_2)(16x^3 - 24x^2 + 9x) + v_2$$

These maps have convenient properties for numerical simulations, as they share the same fixed critical points,

$$c_1 = 1/4,\ c_2 = 3/4$$

the critical values are precisely the parameters,

$$f_{v_1,v_2}(1/4) = v_1,\ f_{v_1,v_2}(3/4) = v_2$$

and the values of f at the endpoints are explicitly given by the parameters as follows:

$$f_{v_1,v_2}(0) = v_2,\ f_{v_1,v_2}(1) = v_1$$

Figure 3 shows the plot of $h(f_{1,v_2})$ *vs.* v_2, $0 \leq v_2 < 1$, computed with the new algorithm, $\varepsilon = 10^{-4}$, and $\Delta v_2 = 0.001$. Again, this plot coincides visually with the same plot computed with the old algorithm [5] (Figure 4), except for the vanishing entropy tail, which indicates that $\varepsilon = 10^{-4}$ is too large a value for obtaining accurate estimates in that parametric region.

Figure 3. Plot of $h(f_{1,v_2})$ in bits *vs.* v_2, $0 \leq v_2 \leq 1$ ($\varepsilon = 10^{-4}, \Delta v_2 = 0.001$).

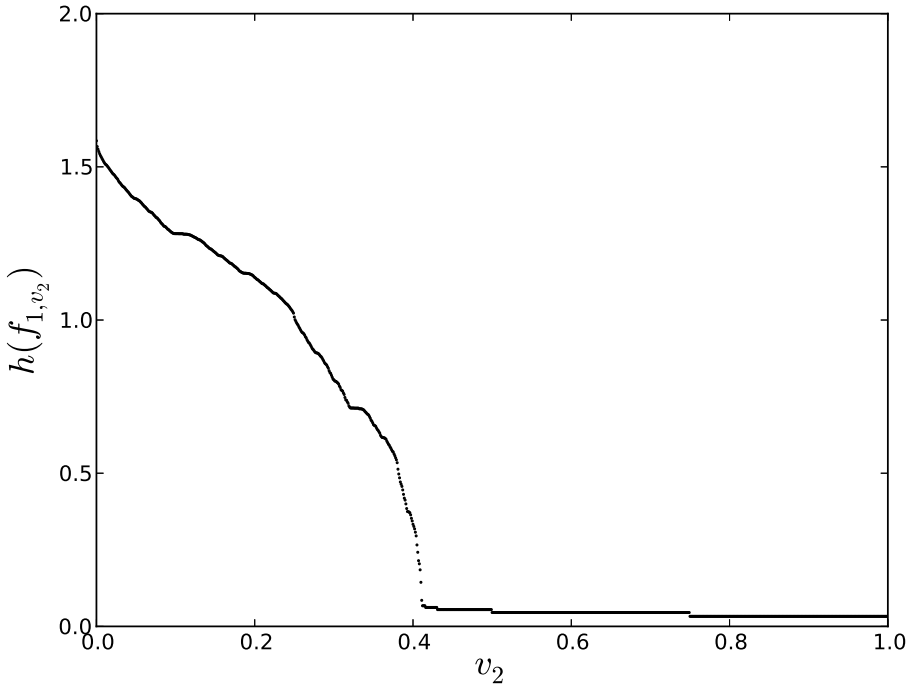


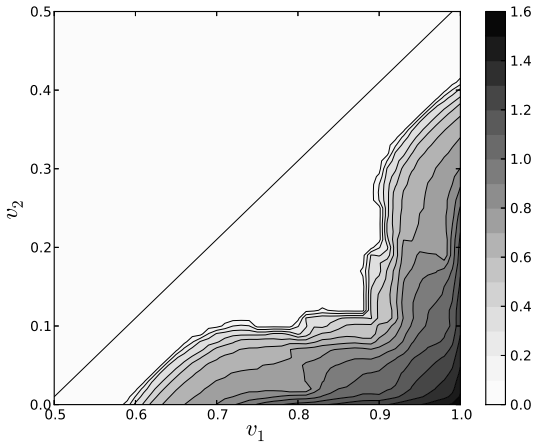
Table 2 displays the performance of the new algorithm as compared to the old one when computing $h(f_{0.9,0.1})$. This time, $t_{old} > t_{new}$ for $\varepsilon = 10^{-d}$, $4 \leq d \leq 7$ (as in Table 1). Furthermore, we obtain two correct decimal digits of the topological entropy, $h(f_{0.9,0.1}) = 0.60...$ bits, with both algorithms and $\varepsilon = 10^{-6}$.

Table 2. Comparison of performances when computing $h(f_{0.9,0.1})$ in bits.

	h_{old}	n_{old}	t_{old}	h_{new}	n_{new}	t_{new}
$\varepsilon = 10^{-4}$	0.619682	195	0.286922	0.622100	218	0.253133
$\varepsilon = 10^{-5}$	0.606568	613	2.665108	0.607310	688	2.485049
$\varepsilon = 10^{-6}$	0.602385	1938	26.238006	0.602622	2173	24.890648
$\varepsilon = 10^{-7}$	0.601062	6125	271.074381	0.601137	6871	265.198039

Figure 4 depicts the values of $h(f_{v_1,v_2})$ for $0 \leq v_2 \leq v_1 - 0.5$, $\varepsilon = 10^{-4}$, and $\Delta v_1, \Delta v_2 = 0.01$.

Figure 4. Level sets of $h(f_{v_1,v_2})$ in bits *vs.* v_1, v_2, $0 \leq v_2 \leq v_1 - 0.5$ ($\varepsilon = 10^{-4}$, $\Delta v_1 = \Delta v_2 = 0.01$).



6.3. Simulation with Three-Modal Maps

Consider next the three-modal maps, $f_{v_2,v_3} : [0,1] \to [0,1]$, defined by the quartic polynomials [5] (Section 8.2):

$$f_{v_2,v_3}(x) = \frac{4\left[\left(2\sqrt{2}-1\right)v_2 - 2v_3\right]x}{2(2\sqrt{2}+1)v_3 - 7v_2}\left[4\left(1+2\sqrt{2}\right)(x-1)(1-2x)^2 v_3 + \left(-56x^3 + 20\left(4+\sqrt{2}\right)x^2 - \left(37+18\sqrt{2}\right)x + 3\sqrt{2} + 5\right)v_2\right]$$

where $0 \leq v_2 < v_3 \leq 1$. The critical points of f_{v_2,v_3} are:

$$c_1 = \frac{-\sqrt{2}v_2 - 4v_2 + 12\sqrt{2}v_3 - 8v_3}{8\left(-7v_2 + 4\sqrt{2}v_3 + 2v_3\right)}, \quad c_2 = 1/2, \quad c_3 = \frac{1}{4}(2+\sqrt{2})$$

Moreover, this family verifies $f_{v_2,v_3}(0) = 0$, $f_{v_2,v_3}(c_2) = v_2$, $f(c_3) = v_3$ and:

$$f_{v_2,v_3}(1) = \frac{4\left(5\sqrt{2}-8\right)v_2\left(\left(2\sqrt{2}-1\right)v_2 - 2v_3\right)}{-7v_2 + 4\sqrt{2}v_3 + 2v_3}$$

Figure 5 shows the plot of $h(f_{v_2,1})$ *vs.* v_2, $0 \leq v_2 < 1$, computed with the new algorithm, $\varepsilon = 10^{-4}$, and $\Delta v_2 = 0.001$. Once more, this plot coincides visually with the same plot computed with the old algorithm (see Figure 7 (left) in [5]), except for the vanishing entropy tail, which again indicates that $\varepsilon = 10^{-4}$ is too large a value for obtaining accurate estimates in that parametric region.

Table 3 displays the performance of the new algorithm as compared to the old one when computing $h(f_{0.7,1})$. Furthermore, this time, $t_{old} > t_{new}$ for $\varepsilon = 10^{-d}$, $4 \leq d \leq 7$ (as in Tables 1 and 2). Furthermore, we obtain two correct decimal digits of the topological entropy, $h(f_{0.7,1}) = 0.69...$ bits, with both algorithms and $\varepsilon = 10^{-6}$.

Figure 6 depicts the values of $h(f_{v_2,v_3})$ for $v_2 + 0.3 \leq v_3 \leq 1$, $\varepsilon = 10^{-4}$ and $\Delta v_2, \Delta v_3 = 0.01$.

Figure 5. Plot of $h(f_{v_2,1})$ in bits *vs.* v_2, $0 \leq v_2 < 1$ ($\varepsilon = 10^{-4}, \Delta v_2 = 0.001$).

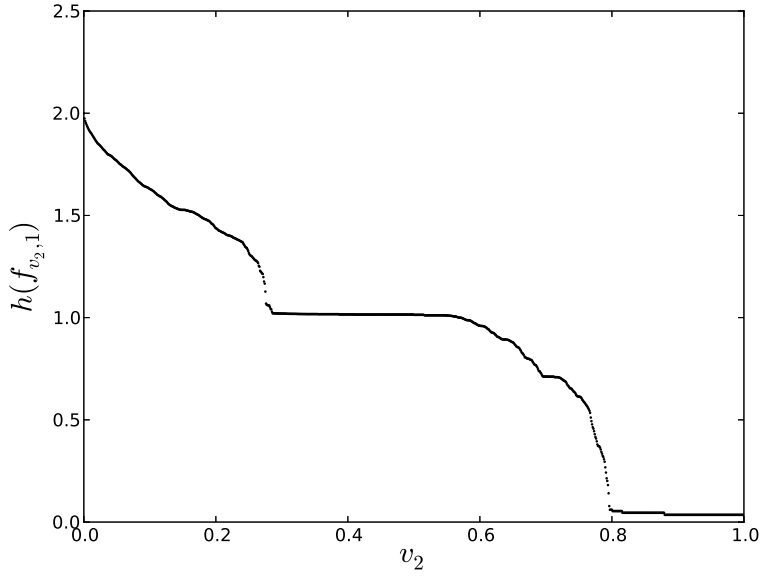
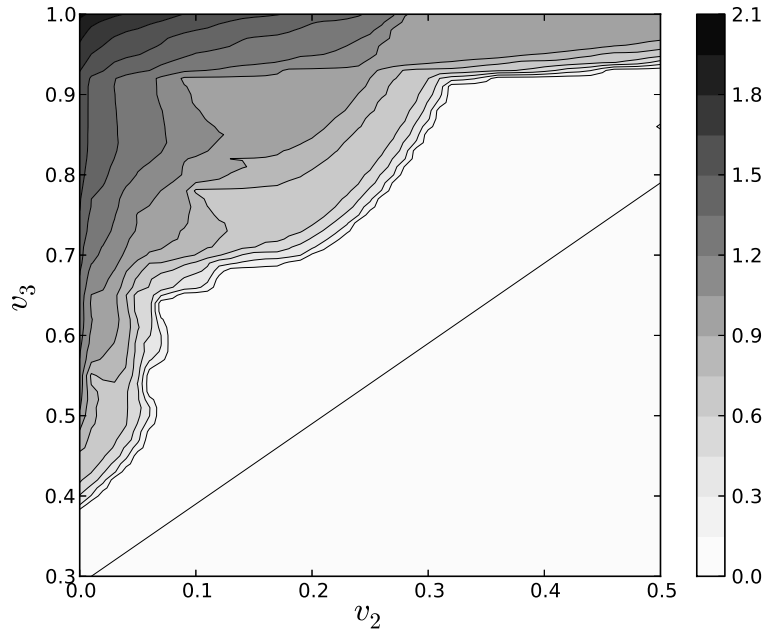


Table 3. Comparison of performances when computing $h(f_{0.7,1})$ in bits.

	h_{old}	n_{old}	t_{old}	h_{new}	n_{new}	t_{new}
$\varepsilon = 10^{-4}$	0.710321	162	0.422875	0.711709	177	0.351586
$\varepsilon = 10^{-5}$	0.699339	511	4.028703	0.699793	557	3.448455
$\varepsilon = 10^{-6}$	0.695855	1,615	39.882275	0.696000	1,759	34.853025
$\varepsilon = 10^{-7}$	0.694752	5,105	417.489291	0.694798	5,561	368.900286

Figure 6. Level sets of $h(f_{v_2,v_3})$ in bits *vs.* v_2, v_3, $v_2 + 0.3 \leq v_3 \leq 1$ ($\varepsilon = 10^{-4}$, $\Delta v_1 = \Delta v_2 = 0.01$).



6.4. Simulation with Higher Multimodal Maps

Last, but not least, we are going to compare the performances of the old and new algorithms with the four- and five-modal maps of Figure 7. These are piecewise linear maps on $[0, 1]$, with constant slope $s = \pm 1.5$, critical points:

$$c_1 = \tfrac{3}{10},\ c_2 = \tfrac{23}{60},\ c_3 = \tfrac{7}{15},\ c_4 = \tfrac{11}{20}$$

and critical values:

$$f(c_1) = f(c_3) = 0.450,\ f(c_2) = f(c_4) = 0.325$$

in the $l = 4$ case, while:

$$c_1 = 0.3,\ c_2 = 0.4,\ c_3 = 0.5,\ c_4 = 0.6,\ c_5 = 0.7$$

and:

$$f(c_1) = f(c_3) = f(c_5) = 0.45,\ f(c_2) = f(c_4) = 0.30$$

in the $l = 5$ case. By [4] (Corollary 4.3.13),

$$h(f) = \max\{0, \ln|s|\} = \ln 1.5 = 0.40547 \text{ nats}$$

in both cases. At variance with the previous examples in Sections 6.1 to 6.3, these two maps are non-smooth and boundary anchored.

Figure 7. A piecewise linear four-modal map **(left)** and five-modal map **(right)** with constant slope $s = \pm 1.5$.

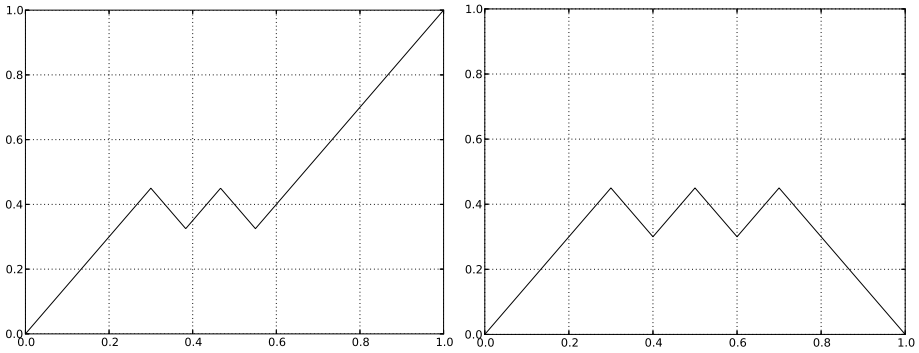


Table 4 summarizes the computational performance of both algorithms with the four-modal map. As happened with the two- and three-modal maps (Tables 2 and 3), the new algorithm needs more computation loops, but less execution time than the old one for all $\varepsilon = 10^{-d}$, $4 \leq d \leq 7$.

Table 4. Comparison of performances when computing $h(f)$ in nats with the four-modal map of Figure 7 (left).

	h_{old}	n_{old}	t_{old}	h_{new}	n_{new}	t_{new}
$\varepsilon = 10^{-4}$	0.421218	160	0.697776	0.422215	169	0.576800
$\varepsilon = 10^{-5}$	0.410476	503	6.583444	0.410776	533	5.747668
$\varepsilon = 10^{-6}$	0.407051	1,589	65.236068	0.407147	1,683	57.951979
$\varepsilon = 10^{-7}$	0.405967	5,021	678.706894	0.405997	5,321	616.59469

Likewise, Table 5 summarizes the computational performance of both algorithms with the five-modal map. It is worth noting that now, both algorithms need the same number of loops for all halt criteria ε, and yet, the new algorithm is faster.

Table 5. Comparison of performances when computing $h(f)$ in nats with the five-modal map of Figure 7 (right).

	h_{old}	n_{old}	t_{old}	h_{new}	n_{new}	t_{new}
$\varepsilon = 10^{-4}$	0.420542	152	0.848166	0.420542	152	0.644305
$\varepsilon = 10^{-5}$	0.410239	480	8.231152	0.410239	480	6.501711
$\varepsilon = 10^{-6}$	0.406978	1,515	81.429872	0.406978	1,515	65.307619
$\varepsilon = 10^{-7}$	0.405944	4,788	864.376277	0.405944	4,788	695.24749

As in the preceding simulations, we conclude from Tables 4 and 5 that both algorithms determine two correct decimal positions of the topological entropy of the corresponding map, $h(f) = 0.40...$ nats. However, this time, the halt criterion $\varepsilon = 10^{-6}$ does not suffice; here, one has to set $\varepsilon = 10^{-7}$ to achieve the same precision.

A concluding observation. As anticipated in remark R2 of Section 5 and illustrated in Tables 1–5, the values of h_{new} converge from above with ever more computation loops (or smaller values of the parameter, ε). This property follows for h_{old} from ([4] (Theorem 4.2.4).

7. Conclusions

The main contributions of this paper are the following.

(i) In Theorem 1, we proved that the transition rules for min-max symbols in Equations (2) and (3), which were derived in [5] for twice differentiable multimodal maps, actually hold true for just continuous ones.

(ii) As a result of Theorem 1, we conclude that the validity of formula (15), which was proved in ([5], Theorem 5.3) for twice differentiable multimodal maps, can be extended to continuous maps. For subsequent applications, only the particularization of Equation (15) to boundary-anchored maps (Theorem 2) is needed.

(iii) The results reviewed and proved in Sections 2 and 3 lead to closed formula (18) for the topological entropy of multimodal maps. Previously, we proved in Theorem 3 that, although ℓ_n clearly depends on the boundary conditions, the limit $h(f) = \lim_{n\to\infty} \frac{1}{n} \log \ell_n$ does not.

(iv) The numerical algorithm proposed in Section 5 for the computation of $h(f)$ amounts to a recursive scheme to approximate the limit in closed formula (18).

This algorithm is a simplification and, at the same time, a generalization of the recursion scheme proposed in [5] for $h(f)$. Indeed, it is a simplification, because Equation (15) was used in [5] to compute the lap number, ℓ_ν, while the abridged expression in Equation (14) is used here. In other words, the new algorithm does not track the orbits of the endpoints. Additionally, it is also a generalization, because we proved in Theorem 2 that Equation (14) (and Equation (15) for that matter) holds not only for twice

differentiable maps (as assumed in [5], (Theorem 5.3)) but also for just continuous ones. By the way, this point was numerically checked in Section 6.4.

The performances of both algorithms, old and new, were compared in Section 6.1 to 6.4 using smooth and non-smooth l-modal maps with $1 \leq l \leq 5$. In view of the results summarized in Tables 1 to 5, the old algorithm performs better in the unimodal case, while the opposite occurs in the other multimodal cases.

Acknowledgments

We thank our referees for their constructive criticism. We are also grateful to José S. Cánovas and María Muñoz Guillermo (Universidad Politécnica de Cartagena, Spain) for clarifying discussions and to Víctor Jiménez (Universidad de Murcia, Spain) for the elegant proof in the Appendix. This work was financially supported by the Spanish *Ministerio de Economía y Competitividad*, grant MTM2012-31698.

Conflicts of Interest

The authors declare no conflict of interest.

Appendix

Let $g : X \to X$ be a continuous map of a compact Hausdorff space, X, into itself. A point, $x \in X$, is non-wandering with respect to the map, g, if for any neighborhood, U, of x, there is an $n \geq 1$ (possibly depending on x), such that $f^n(U) \cap U \neq \emptyset$. Fixed and periodic points are examples of non-wandering points. The closed set of all non-wandering points of g is called its *non-wandering set* and denoted by $\Omega(g)$. According to ([4], Lemma 4.1.5),

$$h(g) = h(g|_{\Omega(g)}) \tag{A1}$$

Furthermore, if:

$$X = \bigcup_{i=1}^{k} Y_i$$

and all Y_i are closed and g-invariant (*i.e.*, $g(Y_i) \subset Y_i$), then ([4], Lemma 4.1.10),

$$h(g) = \max_{1 \leq i \leq k} h(g|_{Y_i}) \tag{A2}$$

To prove Theorem 3, suppose that f is an l-modal self-map of the compact interval, I, with a positive shape (the proof for maps with a negative shape is analogous).

Set $I = [a, b]$, and $J = [a', b']$ with $a' \leq a < b \leq b'$. If $f(a) = a$, choose $a' = a$; if $f(b) = a$ (l odd) or $f(b) = b$ (l even), choose $b' = b$. For definiteness, we suppose the most general situation, namely, $a' < a$ and $b < b'$. Let $F : J \to J$ be such that: (i) F is strictly increasing on $[a', a]$; (ii) $F|_{[a,b]} = f$; and (iii) F is strictly decreasing (l odd) or strictly increasing (l even) on $[b, b']$. In particular, F may be taken as piecewise linear on $[a', a] \cup [b, b']$. Thus, $F \in \mathcal{M}_l(J)$ has the same critical points and values as f, the same shape and is boundary-anchored. Note that the shape enters in how f is extended to F.

Moreover, it is easy to check that $\Omega(F) = \Omega(f) \cup C$, where C is a closed and F-invariant set that only contains fixed points. Thus, $h(F|_C) = 0$ and, according to Equations (A1) and (A2),

$$h(F) = h(F|_{\Omega(F)}) = \max\{h(F|_{\Omega(f)}), h(F|_C)\} = h(F|_{\Omega(f)}) = h(f|_{\Omega(f)}) = h(f) \ \square$$

References

1. Adler, R.; Konheim, A.; McAndrew, M. Topological entropy. *Trans. Am. Math. Soc.* **1965**, *114*, 309–319.
2. Walters, P. *An Introduction to Ergodic Theory*; Springer Verlag: New York, NY, USA, 2000.
3. Misiurewicz, M.; Szlenk, W. Entropy of piecewise monotone mappings. *Stud. Math.* **1980**, *67*, 45–63.
4. Alsedà, L.; Llibre, J.; Misiurewicz, M. *Combinatorial Dynamics and Entropy in Dimension One*; World Scientific: Singapore, Singapore, 2000.
5. Amigó, J.M.; Dilão, R.; Giménez, A. Computing the topological entropy of multimodal maps via min-max sequences. *Entropy* **2012**, *14*, 742–768.
6. Dilão, R.; Amigó, J.M. Computing the topological entropy of unimodal maps. *Int. J. Bifurc. Chaos* **2012**, *22*, 1250152.
7. Dias de Deus, J.; Dilão, R.; Taborda Duarte, J. Topological entropy and approaches to chaos in dynamics of the interval. *Phys. Lett. A* **1982**, *90*, 1–4.
8. Dilão, R. Maps of the Interval, Symbolic Dynamics, Topological Entropy and Periodic Behavior (in Portuguese). Ph.D. Thesis, Instituto Superior Técnico, Lisbon, Portugal, 1985.
9. Milnor, J.; Thurston, W. On Iterated Maps of the Interval. In *Dynamical Systems, Lectures Notes in Mathematics 1342*; Alexander, J.C., Ed.; Springer: Berlin, Germany, 1988; pp. 465–563.
10. De Melo, W.; van Strien, S. *One-Dimensional Dynamics*; Springer: New York, NY, USA, 1993.
11. Milnor, J.; Tresser, C. On entropy and monotonicity of real cubic maps. *Commun. Math. Phys.* **2000**, *209*, 123–178.