Periodic Orbits of the Logistic Map in Single and Double Precision Implementations

Zbigniew Galias, *Senior Member, IEEE*

***Abstract*—In finite precision implementations of chaotic maps all trajectories are eventually periodic. The goal of this brief is to develop methods for systematic study of effects of finite precision computations on dynamical behaviors of discrete maps and to carry out a study of the logistic map in this context. In particular, we are interested in finding all cycles when the logistic map is implemented in single and double precision and studying properties of these cycles including the size of the basin of attraction, and the maximum and average convergence times.**

***Index Terms*—Nonlinear dynamical systems, numerical simulation, floating-point arithmetic.**

I. Introduction

When chaotic maps are implemented using finite precision computations, the quantization causes dynamical degradation. The number of available states is finite and as a consequence all trajectories are eventually periodic (enter a cycle after the transient process). This may lead to wrong conclusions when using finite precision computations to study systems defined with infinite precision in a continuous domain. For example, when the maximum Lyapunov exponent is computed the result converges to the maximum Lyapunov exponent of one of the cycles, none of which is the correct result. Under certain assumptions chaotic attractors are densely filled by unstable periodic orbits. Therefore, one may expect that some short periodic orbits may be observed as steady states in finite precision computations. It may be difficult to find such orbits starting from randomly selected initial conditions due to possibly small basins of attraction of these orbits.

Implementations of chaotic maps in the digital domain are widely used in various applications, including pseudo random number generators [1], [2], [3], [4], chaos-based encryption [5] and secure communication [6], [7]. Finite precision computations may lead to unwanted phenomena, especially when the period of the steady state cycle is low. Various methods have been proposed to handle this problem including perturbing chaotic states or system parameters along trajectories or using cascades of chaotic systems [8], [9], [10].

Manuscript received April 7, 2021; accepted May 14, 2021. Date of publication May 18, 2021; date of current version October 28, 2021. This work was supported by the AGH University of Science and Technology. This brief was recommended by Associate Editor L. F. C. Alberto.

The author is with the Department of Electrical Engineering, AGH University of Science and Technology, 30-059, Kraków, Poland (e-mail: galias@agh.edu.pl).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCSII.2021.3081604.

Digital Object Identifier 10.1109/TCSII.2021.3081604

The problem of cycle detection in finite precision implementations of chaotic maps is therefore important from both theoretical and practical points of view. The most common approach to solve this problem is based on computing trajectories for randomly or uniformly selected initial conditions. In this approach one considers a large number of initial conditions and finds the corresponding steady states. This approach permits finding cycles (periodic orbits) with large basins of attraction and estimating the convergence probability. The problems related to cycle detection in finite precision representations of the logistic map have been studied extensively. Exhaustive search for cycles existing in 32-bit floating-point implementation of the logistic map is carried out in [11]. Cycles observed in 64-bit floating-point implementations of the logistic map are studied numerically in [6], [12], [13]. Statistical study of double precision errors in the logistic map is presented in [14]. Fixed-point n-bit representations with $n \leq 44$ are used in [15] to study the existence of cycles and transient lengths in the logistic map. Dynamical analysis of chaotic maps in the digital domain using the state-mapping network approach is presented in [16]. Properties of sequences generated by the logistic map over the finite field are studied in [17]. None of the existing methods can be used to find all cycles when the precision is high.

The goal of this brief is to develop methods to find all cycles existing in finite precision implementations of one-dimensional maps. As a representative example, we consider the logistic map [18] with 32-bit (single precision) and 64-bit (double precision) floating-point implementations. As it will be show, the research problem to find all cycles is very challenging due to the size of the state space. For the 32-bit implementation the size of the state space is relatively small (of order $10^7 - 10^9$ and the dynamics of the map can be represented using a graph structure. This approach allows us to efficiently find all cycles after visiting each possible state only once. For the 64-bit implementation the size of the state space is much larger (of order $10^{16} - 10^{19}$). In this case it is necessary to eliminate a large number of initial condition from the search space.

The remainder of this brief is organized as follows. In Section II the definition and certain properties of the logistic map are recalled and floating-point data types considered in this brief are described. In Section III two methods to find all cycles for finite precision implementations of one-dimensional maps are proposed and results obtained for single and double precision implementations of the logistic map are presented. The results including the number of cycles, their periods, sizes of basins of attraction, and average convergence times are reported. The last section concludes the study.

TABLE I
STATE SPACE OF THE LOGISTIC MAP IN FINITE PRECISION COMPUTATIONS

a	Ω	State space size S
single precision		
3.9	[0.09506219625, 0.9750000834]	28 764 580
4	[0.0, 1.0]	1 065 353 217
double precision		
3.9	[0.09506249999999996612, 0.975000000000000089]	15 442 843 122 253 457
4	[0, 1]	4 607 182 418 800 017 409

II. Finite Precision Implementations of the Logistic Map

The *logistic map* is a classical example of a simple nonlinear map with complex dynamics [18]. It is defined by

$$f_a(x) = ax(1-x), \tag{1}$$

where $x \in [0, 1]$ and $a \in [0, 4]$.

We consider two cases: $a = 3.9$ and $a = 4$, for which chaotic behavior is observed in computer simulations. For $a = 4$ the map f_a is topologically conjugate [19] to the tent map $g(x) = 1 - 2|x - 0.5|$ and is known to be chaotic [20]. The value $a = 3.9$ does not belong to any short periodic window and therefore the map $f_{3.9}$ is very likely to be chaotic (compare [21], [22]).

The logistic map has a single maximum at $c = 0.5$. The interval $I = [f_a^2(c), f_a(c)]$ is invariant [19], i.e., $f_a(I) = I$. It is the state space of the dynamical system defined by the map f_a. The invariant interval is $I = [0.0950625, 0.975]$ for $a = 3.9$ and $I = [0, 1]$ for $a = 4$.

The most natural way of implementing the logistic map is to use the computational formula $f_a(x) = a * x * (1 - x)$. Note that other computational formulas although equivalent from the mathematical point of view (for example $f_a(x) = a*(1-x)*x$) usually lead to different results in finite precision implementations (compare [13]).

In this brief we consider two of the most common floating-point formats used nowadays: the binary32 (single precision) and binary64 (double precision). These formats are defined in the IEEE 754 standard [23]. According to this standard basic arithmetic operations (addition, subtraction, multiplication and division) must round correctly. It follows that computations conforming to this standard should give reproducible results.

For finite precision implementations the state space is a set of representable numbers belonging to a certain interval Ω. Endpoints of these intervals for $a = 3.9$ and $a = 4$ when using single and double precision computations are given in Table I. For each case we also report the *state space size S*, which is the number of representable numbers (floats or doubles) which belong to Ω.

It is interesting to note that for $a = 3.9$ in both cases the maximum value (the right endpoint of Ω) is not obtained for $x = c = 0.5$. For example, when computations are carried out in single precision then $f_a(0.5) = 0.9750000238$ and $f_a(0.4999999106) = 0.9750000834 > 0.9750000238$.

For $a = 4$ the state space is much larger than for $a = 3.9$. This is related to the fact that representable numbers fill the intervals close to zero much more densely than intervals located far away from zero. For example, the number of floats in the intervals $[0, 0.1]$, $[0.1, 0.2]$, and $[0.2, 0.3]$ is 1036831950, 8388609, and 5033166, respectively.

In the following sections a systematic study of dynamical behaviors of the logistic map with $a = 3.9$ and $a = 4$ implemented in the single and double precision floating-point formats using the computational formula $f_a(x) = a*x*(1-x)$ is carried out.

III. Finding All Cycles and Their Basins of Attractions

In this section two methods to find all cycles and analyse their properties including the size of the basin of attraction [19], [24] and average convergence times are presented. The *basin of attraction* of a given cycle is defined as the subset of the state space containing points which converge to this cycle. The *basin size s* is defined as the number of points belonging to the basin of attraction. An important number characterizing the basin of attraction of a cycle is the *relative basin size r* defined as the ratio of the basin size s and the state space size S. The relative basin size is equal to the probability of convergence to a given cycle starting from random initial conditions belonging to the state space. For a given initial point x the convergence time $\tau(x)$ is defined as the number of iterations needed to reach the steady state.

The first method to find all cycles, which will be referred to as the *graph-based* method, is based on the construction of a graph structure representing the dynamics of the map over the state space. In this approach each point in the state space is a graph vertex. Graph edges correspond to transitions between points as defined by the action of the map in finite precision computations. For a given point x_k let us denote by b_k the index of the basin of attraction it belongs to and by $\tau_k = \tau(x_k)$ the convergence time. At the beginning of the procedure all graph vertices are marked as non-visited. During the procedure we select a non-visited vertex x_k and compute a trajectory starting from this vertex until we reach a visited vertex (denote it by x_l). Based on b_l and τ_l the values b_i and τ_i are updated for all vertices belonging to the trajectory starting at x_k and ending at x_l. The procedure ends when the set of non-visited vertices is empty. Once this is done, we can easily compute the number of cycles, their periods, basin sizes and the average convergence times. It will be shown that the graph based approach works well for single precision computations for which the size of the state space is small.

If it is impossible to store all states in a computer memory then another method has to be used. The simplest idea to find all cycles is to carry out an exhaustive search in which for each point in the state space the corresponding steady state (cycle) is found. This method is not feasible when the number of initial points is large due to extremely long computation times. To make this method work we need to limit the number of initial conditions to be considered. The first idea is to stop computation of a trajectory $(f^k(x_0))_{k=1,2,\ldots}$ when $f^k(x_0) > x_0$. This will significantly speed up computations since for many initial points the stop condition is satisfied for a small k. This idea can be extended to skip whole intervals of initial points using the interval arithmetic approach [25]. In interval arithmetic, all calculations are performed on intervals in such a way that the true result is always enclosed within the interval calculated by the computer. This is achieved by setting proper rounding modes when carrying out arithmetic operations on interval endpoints. The second method to find all cycles, which will be referred to as the *trajectory based* approach, works as follows. In the first step, the state space Ω is split into several test intervals. For each test interval $\mathbf{x}_0 = [\underline{x}_0, \overline{x}_0]$ enclosures $\mathbf{y}_k = [\underline{y}_k, \overline{y}_k]$ of $f^k(\mathbf{x}_0)$ for $k = 1, 2, \ldots, k_{\max}$ are computed using interval arithmetic methods. If for some $k \leq k_{\max}$ the condition $\underline{y}_k > \overline{x}_0$ is satisfied then the interval $\mathbf{x}_0$ is skipped. In the second step, for the remaining initial conditions x_0 trajectories $(f^k(x_0))_{k=1,2,\ldots}$ are computed until either $f^k(x_0) > x_0$ or a cycle is found.

Sizes of basins of attraction and average convergence times of cycles can be estimated using a statistical approach. In this approach n initial conditions are selected randomly. For each initial condition x_k the corresponding steady state is found, the basin index b_k to which x_k belongs to is identified and the convergence time τ_k is computed. Using this data one can estimate the relative basin size for the basin with the index b as $r = \sum_{k=1}^{n} \delta(b, b_k)/n$, where $\delta(i,j)$ denotes the Kronecker delta ($\delta(i,j) = 1$ if $i = j$ and $\delta(i,j) = 0$ otherwise). The average convergence time can be estimated as $\tau_{\text{aver}} \approx (\sum_{k=1}^{n} \tau_k \cdot \delta(b, b_k))/(\sum_{k=1}^{n} \delta(b, b_k))$.

The statistical approach works only for cycles with large basins. When the basin of attraction is small compared to the state space then usually none of the randomly selected initial points belongs to the basin. In this case one may use a graph based approach. In this approach one construct a graph representation of the basin. The basin is initialized as the set of points belonging to the cycle. During the procedure for each point in the basin its preimage under the map f_a is computed and added to the basin. The algorithm is continued until no more points can be added. This method permits finding the exact values of the relative basin size and the average convergence time. One has to be careful when computing f_a^{-1} in a finite precision. In the infinite precision the preimage of $y \in I$ can be computed as $x = f_a^{-1}(x) = \{0.5 \pm \sqrt{0.25 - y/a}\}$. $f_a^{-1}(y)$ is a single point for $y = f_a(c)$ and contains two points otherwise. In a finite precision the preimage of a single point can be empty. It can also be very large, especially for points close to $y = f_a(c)$.

A. Single Precision Computations

For the single precision case the size of the state space is relatively small (compare Table I) and the graph based method can be used. The results are presented in Table II. For each cycle we report its period p, the maximum position $x_{\max}$, the Lyapunov exponent λ, the basin size s, the relative basin size r, the maximum convergence time $\tau_{\max}$, and the average convergence time τ_{aver}. The Lyapunov exponent for the cycle $(x_k)_{k=0}^{p-1}$ is calculated using the formula $\lambda = p^{-1} \sum_{k=0}^{p-1} \log(f_a'(x_k))$.

For $a = 4$ the state space size is much larger than for $a = 3.9$ (compare Table I) and hence longer computation time and more computer memory are needed to solve the problem. The computation time is approximately 550 seconds for $a = 4$ and less than 10 seconds for $a = 3.9$.

For $a = 3.9$ there are 11 cycles with periods varying from 2 to 1415. Most trajectories converge to the period–1031 solution. The shortest orbit has the smallest basin of attraction. Lyapunov exponents computed along the cycles differ significantly. The smallest value $\lambda = 0.4196$ and the largest value $\lambda = 0.6134$ are observed for the period-65 cycle and the period-2 cycle, respectively.

TABLE II
CYCLES EXISTING FOR $a = 3.9$ AND $a = 4$ FOR SINGLE PRECISION COMPUTATIONS, p IS THE PERIOD, λ IS THE LYAPUNOV EXPONENT, s IS THE BASIN SIZE, r IS THE RELATIVE BASIN SIZE, τ_{MAX} AND τ_{AVER} ARE THE MAXIMUM AND AVERAGE CONVERGENCE TIMES

p	$x_{\max}$	λ	s	r	$\tau_{\max}$	τ_{aver}
			$a = 3.9$			
1415	0.974996209	0.5023	2598525	0.09034	1199	400
1031	0.974999428	0.4983	25783894	0.89638	5269	2551
192	0.974957347	0.5217	60632	0.00211	307	109
175	0.974936426	0.5372	24362	$8.47 \cdot 10^{-4}$	132	44
82	0.974431634	0.5198	11397	$3.96 \cdot 10^{-4}$	118	42
65	0.974998713	0.4196	241716	0.00840	731	214
32	0.974006951	0.5200	7118	$2.47 \cdot 10^{-4}$	211	91
22	0.972594261	0.5318	33845	0.00118	332	150
13	0.971583307	0.4889	2810	$9.77 \cdot 10^{-5}$	52	26
5	0.934898078	0.6119	240	$8.34 \cdot 10^{-6}$	18	9
2	0.897435904	0.6134	41	$1.43 \cdot 10^{-6}$	9	5
			$a = 4$			
4344	0.999999881	0.693085	713832287	0.67004	3377	1045
836	0.999932587	0.693148	118740620	0.11146	2149	900
436	0.999998629	0.693187	31822336	0.02987	941	335
143	0.999684513	0.693148	359961	$3.38 \cdot 10^{-4}$	257	88
136	0.999928474	0.693151	627693	$5.89 \cdot 10^{-4}$	257	96
5	0.937173307	0.693147	443	$4.16 \cdot 10^{-7}$	66	32
4	0.991486549	0.693147	5016	$4.71 \cdot 10^{-6}$	78	37
3	0.969846308	0.693147	696	$6.53 \cdot 10^{-7}$	66	33
1	0.75	0.693147	64	$6.01 \cdot 10^{-8}$	63	32
1	0	1.386294	199964101	0.18770	3040	1196

For $a = 4$ case there are 10 cycles. More than 66% of trajectories converge to the longest period–4344 cycle. Two cycles have period 1 ($x = 0$ and $x = 0.75$). They are unstable fixed points for the infinite precision system. Their basins of attraction are quite different. The basin of attraction of the fixed point $x = 0.75$ contains 64 points; the chance to reach this fixed point starting from random initial conditions is very low. On the other hand the basin of attraction of the fixed point $x = 0$ is relatively large. Almost 19% of initial conditions from the interval [0, 1] lead to this fixed point. For $a = 4$, Lyapunov exponents of all cycles are practically the same ($\lambda = 0.693$) with the exception of the fixed point $x = 0$ for which the Lyapunov exponent is $\lambda = 1.386$.

B. Double Precision Computations

For the double precision case the graph based approach cannot be used due to a very large state space.

The trajectory based approach is applied to find all cycles existing for $a = 3.9$. In the elimination step the number of initial conditions is reduced from $1.54 \cdot 10^{17}$ to $4.59 \cdot 10^{13}$ (more than 99.7% of initial conditions are eliminated). Trajectories for the remaining initial conditions are computed and all cycles are found. The total computation time using a single core 4.1 GHz processor is approximately 600 days. Parallel computations are utilized to speed up the process. A statistical approach with $n = 10^5$ initial conditions is used to estimate basin sizes and average convergence times for the first seven cycles. For the remaining cycles their basins are found using the graph based approach. The results obtained are reported in the first part of Table III.

For $a = 3.9$ there are 16 cycles with periods varying from 8 to 60858285. Four shortest cycles are shown in Fig. 1.

TABLE III
CYCLES EXISTING FOR $a = 3.9$ AND $a = 4$ FOR DOUBLE PRECISION COMPUTATIONS, p IS THE PERIOD, $x_{\max}$ IS THE MAXIMUM POSITION OF THE ORBIT, λ IS THE LYAPUNOV EXPONENT, s IS THE BASIN SIZE, r IS THE RELATIVE BASIN SIZE, τ_{MAX} AND τ_{AVER} ARE THE MAXIMUM AND AVERAGE CONVERGENCE TIMES

	index	p	$x_{\max}$	λ	s	r	$\tau_{\max}$	τ_{aver}
$a = 3.9$	1	60858285	0.97499999999999975575	0.496031		0.76412	64974305	21290635
	2	18143091	0.97499999999999809042	0.496129		0.11314	14881035	5465865
	3	9840432	0.97499999999999520384	0.496157		0.08652	23556226	9570955
	4	7724511	0.97499999999999686917	0.495782		0.03438	9192908	3072978
	5	1589143	0.97499999999985209609	0.495516		0.00168	2554331	1001401
	6	500711	0.97499999999908593118	0.495164		0.00015	932057	259766
	7	271845	0.97499999999323849753	0.496231		0.00001	23903	23903
	8	18086	0.97499999899347000731	0.496756	500509456	$3.24 \cdot 10^{-8}$	27547	6611
	9	906	0.97499991068666158611	0.504072	2634191	$1.71 \cdot 10^{-10}$	1337	374
	10	521	0.97499350313246058164	0.457566	876787	$5.68 \cdot 10^{-11}$	742	259
	11	400	0.97499901265892674473	0.490862	919817	$5.96 \cdot 10^{-11}$	1181	402
	12	318	0.97499951134441653622	0.481579	2367203	$1.53 \cdot 10^{-10}$	914	296
	13	130	0.97494809835736961912	0.472305	44984	$2.91 \cdot 10^{-12}$	219	93
	14	120	0.97482291442148838456	0.484493	81302	$5.26 \cdot 10^{-12}$	525	239
	15	13	0.97414223928824406062	0.470759	1644	$1.06 \cdot 10^{-13}$	77	34
	16	8	0.97484518785164275823	0.290500	21347	$1.38 \cdot 10^{-12}$	156	66
$a = 4$	1	14632801	0.99999999999999833467	0.693147		0.113170	28910266	8633648
	2	10210156	0.99999999999999611422	0.693147		0.017525	13468168	5891184
	3	5638349	0.99999999999925126559	0.693147		0.676795	101080968	54620675
	4	2625633	0.99999999999997657429	0.693147		0.014735	11746307	4086732
	5	2441806	0.99999999999869704226	0.693147		0.014405	11866292	5479868
	6	1311627	0.99999999998641386778	0.693147		0.000100	1027409	321274
	7	960057	0.99999999999176025778	0.693147		0.000210	2069934	1036259
	8	510250	0.99999999999727640088	0.693147		0.000035	228269	122879
	9	420909	0.99999999992764065926	0.693147		0.000145	1292467	653661
	10	234209	0.99999999975801878715	0.693147				
	11	66637	0.99999999937296479846	0.693147				
	12	4389	0.99999999999960476060	0.693147				
	13	3484	0.99999917060451570805	0.693147	5055899141	$1.10 \cdot 10^{-9}$	4330	1358
	14	1392	0.99999713694406588971	0.693147	9017978993	$1.96 \cdot 10^{-9}$	7427	3354
	15	1374	0.99999981511262558964	0.693147	2304383297	$5.00 \cdot 10^{-10}$	3592	1244
	16	84	0.99908941742182222345	0.693147	1226481	$2.66 \cdot 10^{-13}$	576	278
	17	63	0.99825679944966716484	0.693147	621982	$1.35 \cdot 10^{-13}$	566	272
	18	8	0.99548659390270277658	0.693147	22502	$4.88 \cdot 10^{-15}$	519	256
	19	1	0.75	0.693147	2	$4.34 \cdot 10^{-19}$	1	0.5
	20	1	0.0	1.386294		0.162880	27395187	9947016

Cycles with periods $p < 1000$ have very small basins of attraction. In consequence, the probability of reaching these cycles starting from random initial conditions is very small (below $2 \cdot 10^{-10}$). The largest basin of attraction is observed for the longest cycle. More than 76% of trajectories converge to this cycle. The longest convergence time observed is 64974305, while the average convergence time is below $3 \cdot 10^7$. Lyapunov exponents calculated along the cycles belong to the interval [0.2905, 0.5041]. The largest deviation from the average value $\lambda = 0.496$ is observed for the period–8 cycle ($\lambda = 0.2905$).

Similar computations are carried out for $a = 4$. In the elimination step the number of initial conditions is reduced from $4.61 \cdot 10^{18}$ to $6.29 \cdot 10^{13}$ (more than 99.998% initial conditions are eliminated). Remaining initial conditions are handled in the same way as for the case $a = 3.9$. The total computation time to find all cycles for $a = 4$ using a single core 4.1 GHz processor is approximately 500 days. The results obtained are reported in the second part of Table III. A statistical approach with $n = 10^5$ initial conditions is used to estimate basin sizes and average convergence times for the cycles with indices 1–9 and 20. For the cycles with indices 13–19 their basins are found using the graph based approach. For the cycles with indices 10–12 neither method works. Their basins are too small to be detected using the statistical approach and too large for the graph based approach. Similarly to the single precision case the Lyapunov exponents computed along the orbits are the same ($\lambda = 0.693147$) with the exception of the last cycle ($\lambda = 1.386294$).

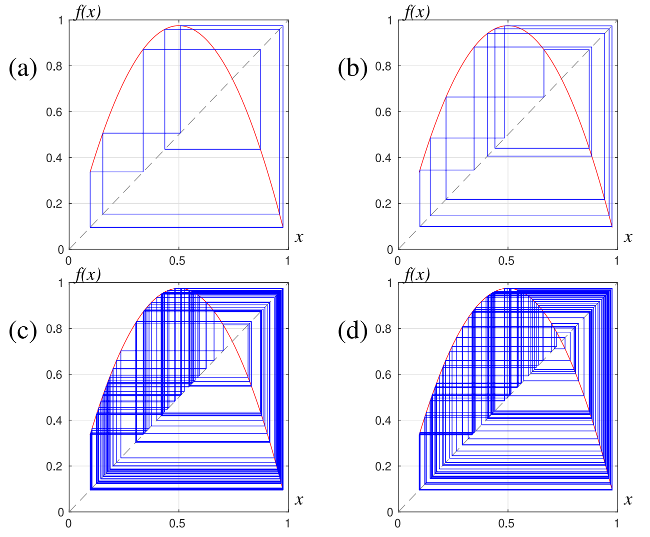

Fig. 1. Shortest periodic trajectories existing for $a = 3.9$ for double precision computations; (a) $p = 8$, (b) $p = 13$, (c) $p = 120$, (d) $p = 130$.

For $a = 4$ there exist 20 cycles. Most trajectories (above 67%) converge to the cycle with period 5638349 (index 3 in Table III). The case $a = 4$ has been studied by many researchers. For example, cycles with indices 1–7 and 20 are reported in [12]. In [6], the cycles with indices 1-10,12 and 20 are reported. The cycles with indices 1,3,4,5, and 20 are reported in [13]. The results obtained in [6], [12], [13] are based on computing trajectories starting from a small number of initial conditions (1000 or 10000). This method can find cycles with a large basin of attraction only.

Table III reports all cycles existing in the double precision implementation of the logistic map. These results may be useful for researchers studying the dynamics of the logistic map and other one-dimensional maps. Note that in order to find the majority of cycles (including those with very small basins of attraction) it is sufficient to consider initial conditions from a very small subset of the state space (in the examples considered $x \in [0.9741, 0.975]$ for $a = 3.9$ and $x \in [0.9954, 1]$ for $a = 4$). Additionally, when searching for short cycles (which are more difficult to find) it is sufficient to consider short trajectories because for short cycles the maximum convergence time is also short. These guidelines may be helpful in studying effects of finite precision computations on the dynamics of one-dimensional maps.

The results presented in Table III show that the double precision computations may lead to wrong results when studying the dynamics of one-dimensional maps. In both cases ($a = 3.9$ and $a = 4$) the average convergence time is below $6 \cdot 10^7$. Moreover, there exist initial conditions leading to extremely short periods. Relatively small convergence times and short cycle lengths may influence the results (such as Lyapunov exponents) produced in the double precision computations. An extra care has to be taken in verifying that a computer generated trajectory is not one of short cycles.

These results also indicate that double-precision implementations of the logistic map may cause serious disadvantages in chaos based applications like pseudo random number generators or chaos-based communication. A proper selection of initial conditions is crucial for the proper operation of applications based on chaotic maps.

IV. Conclusion

Systematic methods to find all periodic solutions of finite precision implementations of chaotic maps have been proposed. All cycles existing in single and double precision implementations of the logistic map for selected parameter values have been found. According to our knowledge, the results of this study are the first example reporting all cycles existing in the double precision implementation of the logistic map. Using the graph based approach basins of attraction and convergence times were found for cycles with small basins. The probability of convergence to a given cycle and average convergence times have been estimated for cycles with large basins using the statistical approach. It was shown that finite precision implementations of the logistic map may produce very short cycles. The proposed methods are general and can be applied without modifications to study other chaotic one-dimensional systems.

Acknowledgment

Computations were carried out using the AGH computational resources (84%) and the PL-Grid Infrastructure (16%).

References

[1] M. Andrecut, "Logistic map as a random number generator," *Int. J. Modern Phys. B*, vol. 12, no. 9, pp. 921–930, 1998.

[2] S.-L. Chen, T. Hwang, and W.-W. Lin, "Randomness enhancement using digitalized modified logistic map," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 57, no. 12, pp. 996–1000, Dec. 2010.

[3] Z. Hua and Y. Zhou, "One-dimensional nonlinear model for producing chaos," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 65, no. 1, pp. 235–246, Jan. 2018.

[4] M. Garcia-Bosque, A. Perez-Resa, C. Sanchez-Azqueta, C. Aldea, and S. Celma, "Chaos-based bitwise dynamical pseudorandom number generator on FPGA," *IEEE Trans. Instrum. Meas.*, vol. 68, no. 1, pp. 291–293, Jan. 2019.

[5] C. Li, D. Lin, B. Feng, J. Lu, and F. Hao, "Cryptanalysis of a chaotic image encryption algorithm based on information entropy," *IEEE Access*, vol. 6, pp. 75834–75842, 2018.

[6] S. Wang, W. Liu, H. Lu, J. Kuang, and G. Hu, "Periodicity of chaotic trajectories in realizations of finite computer precisions and its implication in chaos communications," *Int. J. Mod. Phys. B*, vol. 18, nos. 17–19, pp. 2617–2622, 2004.

[7] M. Garcia-Bosque, C. Sanchez-Azqueta, and S. Celma, "Secure communication system based on a logistic map and a linear feedback shift register," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2016, pp. 1170–1173.

[8] T. Addabbo, M. Alioto, A. Fort, S. Rocchi, and V. Vignoli, "A feedback strategy to improve the entropy of a chaos-based random bit generator," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 53, no. 2, pp. 326–337, Feb. 2006.

[9] Q. Wang *et al.*, "Theoretical design and FPGA-based implementation of higher-dimensional digital chaotic systems," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 63, no. 3, pp. 401–412, Mar. 2016.

[10] M. A. Dastgheib and M. Farhang, "A digital pseudo-random number generator based on sawtooth chaotic map with a guaranteed enhanced period," *Nonlinear Dyn.*, vol. 89, no. 4, pp. 2957–2966, 2017.

[11] K. Persohn and R. Povinelli, "Analyzing logistic map pseudorandom number generators for periodicity induced by finite precision floating-point representation," *Chaos Solitons Fractals*, vol. 45, no. 3, pp. 238–245, 2012.

[12] N. R. Wagner, "The logistic lattice in random number generation," in *Proc. 30th Annu. Allerton Conf. Commun. Control Comput.*, 1992, pp. 922–931.

[13] M. Yabuki and T. Tsuchiya, "Double precision computation of the logistic map depends on computational modes of the floating-point processing unit," 2013. [Online]. Available: arXiv:1305.3128.

[14] J. A. Oteo and J. Ros, "Double precision errors in the logistic map: Statistical study and dynamical interpretation," *Phys. Rev. E*, vol. 76, no. 3, 2007, Art. no.036214.

[15] I. Ozturk and R. Kilic, "Cycle lengths and correlation properties of finite precision chaotic maps," *Int. J. Bifurcation Chaos*, vol. 24, no. 9, 2014, Art. no. 1450107.

[16] C. Li, B. Feng, S. Li, J. Kurths, and G. Chen, "Dynamic analysis of digital chaotic maps via state-mapping networks," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 66, no. 6, pp. 2322–2335, Jun. 2019.

[17] B. Yang and X. Liao, "Period analysis of the logistic map for the finite field," *Sci. China Inf. Sci.*, vol. 60, no. 2, 2017, Art. no. 022302.

[18] R. M. May, "Simple mathematical models with very complicated dynamics," *Nature*, vol. 261, pp. 459–467, Jun. 1976.

[19] C. Robinson, *Dynamical Systems: Stability, Symbolic Dynamics, and Chaos*. Boca Raton, FL, USA: CRC Press, 1995.

[20] P. Collet and J.-P. Eckmann, *Iterated Maps on the Interval as Dynamical Systems*. Boston, MA, USA: Birkhauser, 2009.

[21] W. Tucker and D. Wilczak, "A rigorous lower bound for the stability regions of the quadratic map," *Physica D, Nonlinear Phenomena*, vol. 238, no. 18, pp. 1923–1936, 2009.

[22] Z. Galias, "Systematic search for wide periodic windows and bounds for the set of regular parameters for the quadratic map," *Chaos Interdiscipl. J. Nonlinear Sci.*, vol. 27, no. 5, 2017, Art. no. 053106.

[23] *IEEE Standard for Floating-Point Arithmetic*. IEEE Standard 754-2019, 2019.

[24] H.-D. Chiang and L. F. C. Alberto, *Stability Regions of Nonlinear Dynamical Systems*. Cambridge, U.K.: Cambridge Univ. Press, 2015.

[25] R. Moore, *Methods and Applications of Interval Analysis*. Philadelphia, PA, USA: SIAM, 1979.