



**GENERALIZED CHAOTIC MAPS AND
ELEMENTARY FUNCTIONS BETWEEN ANALYSIS
AND IMPLEMENTATION**

By

Wafaa Saber AbdelHalim Sayed

A Thesis Submitted to the
Faculty of Engineering at Cairo University
in Partial Fulfillment of the
Requirements for the Degree of
MASTER OF SCIENCE
in
Mathematical Engineering

FACULTY OF ENGINEERING, CAIRO UNIVERSITY
GIZA, EGYPT
2015

**GENERALIZED CHAOTIC MAPS AND
ELEMENTARY FUNCTIONS BETWEEN ANALYSIS
AND IMPLEMENTATION**

By

Wafaa Saber AbdelHalim Sayed

A Thesis Submitted to the
Faculty of Engineering at Cairo University
in Partial Fulfillment of the
Requirements for the Degree of
MASTER OF SCIENCE
in
Mathematical Engineering

Under the Supervision of

Prof. Abdel-Latif E. Hussien

Professor

Engineering Mathematics and Physics Department

Faculty of Engineering, Cairo University

Prof. Hossam A. H. Fahmy

Professor

Electronics and Communication Engineering Department

Faculty of Engineering, Cairo University

Assoc. Prof. Ahmed G. Radwan

Associate Professor

Engineering Mathematics and Physics Department

Faculty of Engineering, Cairo University

**FACULTY OF ENGINEERING, CAIRO UNIVERSITY
GIZA, EGYPT**

2015

**GENERALIZED CHAOTIC MAPS AND
ELEMENTARY FUNCTIONS BETWEEN ANALYSIS
AND IMPLEMENTATION**

By

Wafaa Saber AbdelHalim Sayed

A Thesis Submitted to the
Faculty of Engineering at Cairo University
in Partial Fulfillment of the
Requirements for the Degree of
MASTER OF SCIENCE
in
Mathematical Engineering

Approved by the Examining Committee:

Prof. Abdel-Latif E. Hussien, Thesis Main Advisor

Prof. Hossam A. H. Fahmy, Member

Assoc. Prof. Ahmed G. Radwan, Member

Prof. Salwa K. Abd-El-Hafiz, Internal Examiner

Prof. Nasser H. Sweilam, External Examiner
Professor, Faculty of Science, Cairo University

FACULTY OF ENGINEERING, CAIRO UNIVERSITY
GIZA, EGYPT

2015

Engineer's Name: Wafaa Saber AbdelHalim Sayed
Date of Birth: 20/3/1991
Nationality: Egyptian
E-mail: wafaa.s.sayed@eng.cu.edu.eg
Registration Date: 1/10/2013
Awarding Date: --/--/2015
Degree: Master of Science
Department: Engineering Mathematics and Physics



Supervisors:

Prof. Abdel-Latif E. Hussien
Prof. Hossam A. H. Fahmy
Assoc. Prof. Ahmed G. Radwan

Examiners:

Prof. Nasser H. Sweilam, Professor, Faculty of Science, Cairo University	(External examiner)
Prof. Salwa K. Abd-El-Hafiz	(Internal examiner)
Prof. Abdel-Latif E. Hussien	(Thesis main advisor)
Prof. Hossam A. H. Fahmy	(Member)
Assoc. Prof. Ahmed G. Radwan	(Member)

Title of Thesis:

Generalized Chaotic Maps and Elementary Functions between Analysis and Implementation

Key Words:

Logistic map; Tent map; Digital implementation; Finite precision; Power function

Summary:

In this thesis, we aim to bridge the gap between mathematical analysis of generalized one-dimensional discrete chaotic maps and their implementation on digital platforms. We propose several variations and generalizations on the logistic and tent maps and employ the power function in a general map that could yield each of them and other new maps. We present negative control parameter maps that provide wider alternating-sign output ranges which are controllable by scaling parameters. Moreover, the proposed general powering map is mathematically analyzed and verified for boundary cases. A transition map is presented that exhibits responses in the intermediate range from tent to logistic map. Finite precision logistic map is studied explaining the impact of finitude on its properties. In addition, floating-point implementations of the power function are tested on the occurrence of special values of the operands.

Acknowledgements

First and foremost, I am thankful to Almighty God for His uncountable grants upon all of us.

I owe sincere gratitude to Prof. Abdel-Latif E. Hussein, Prof. Hossam A. H. Fahmy, and Assoc. Prof. Ahmed G. Radwan for their valuable guidance and support. It is my pleasure to have this distinguished group of professors as thesis advisors, and I hope I managed to be as trustworthy as they expected. They gave me so much of their precious time and helped me out of many problems with their knowledge and experience.

I highly appreciate the too many hours that Prof. Hossam A. H. Fahmy and Assoc. Prof. Ahmed G. Radwan dedicated to our meetings which were undoubted fruitful towards presenting this work. They have always encouraged me to explore new areas of science whenever I come across them. They provided me with invaluable guidance in my early experiences in international publication. They answered my questions patiently and paid attention to every fine detail.

Moreover, I would like to thank Assoc. Prof. Ahmed G. Radwan for being confident in my capabilities, answering all my inquiries on the spot in spite of his busy schedule, and motivating me to exert relentless efforts in my work.

I am deeply indebted to Prof. Hossam A. H. Fahmy for his keenness on keeping my morale high. He has played a major role in the development of my personality and mind. In fact, he spares no effort to offer moral and scientific support to all his students. All words of gratitude fail to express how much I owe him for he always guides, helps, and encourages me to acquire more skills and experiences.

I thank my beloved mother for her invaluable support during my life. I could not have accomplished any success without her continual efforts to let me focus on my studying. Not to mention my beloved younger brother who is a source of inspiration and a motivation for success.

I am grateful to every teacher and professor who contributed to my knowledge and my decisions throughout the route towards my career.

Finally, I would like to thank my dear friends and colleagues; specifically those who are patiently always a source of support and kindness and those who have helped me since day one as a teaching assistant and postgraduate student until now. May Allah bless and reward all those who have supported me.

Table of Contents

Acknowledgements	i
Table of Contents	ii
List of Tables	vii
List of Figures	viii
List of Symbols and Abbreviations	xii
Abstract	xiii
1 INTRODUCTION	1
1.1 Motivation	1
1.1.1 Chaotic Maps and Numerical Computation	2
1.1.2 Power Function and Floating-Point Representation	4
1.2 Main Contributions	5
1.3 Publications out of This Work	6
1.4 Organization of the Thesis	6
2 REVIEW ABOUT THE STUDIED PROBLEMS	8
2.1 Chaos: Motivation and Mathematical Analysis	8
2.1.1 Historical Background	9
2.1.2 Conventional Logistic Map $x_{n+1} = \lambda x_n(1 - x_n)$	12
2.1.2.1 Exponential Versus Logistic Growth Models	12
2.1.2.2 Operands' Ranges	13
2.1.2.3 Main Definitions: Fixed Points and Periodic Points	14
2.1.2.4 Bifurcation Diagram	14
2.1.2.4.1 Fixed and Periodic Points and Their Stability	16
2.1.2.4.2 Different Phases on the Diagram	16
2.1.2.4.3 Key-points of the Bifurcation Diagram	17
2.1.3 Conventional Tent Map $x_{n+1} = \mu \min(x_n, 1 - x_n)$	17
2.1.3.1 Operands' Ranges	18
2.1.3.2 Bifurcation Diagram	18
2.1.4 Maximum Lyapunov Exponent	18
2.2 Negative Operands: How and Why?	19
2.2.1 The Negative Parameter Case First Visited	19
2.2.2 Enhancement of Randomness Measures	20
2.2.3 Wider Domain of Modeling With Improved Efficiency	21
2.3 Generalized Chaotic Maps	23
2.4 Digital Chaotic Maps	25
2.4.1 Multiple Precision Versus Fixed Precision	26
2.4.2 Previous Work	26
2.5 Elementary Functions: The Power Operation	29

2.5.1	IEEE Standard Definitions	30
2.5.1.1	Special Values	30
2.5.1.1.1	Signed Zero	30
2.5.1.1.2	Infinities	30
2.5.1.1.3	NaN	31
2.5.1.2	Operations	31
2.5.1.3	Operations Generating NaN (QNaN)	31
2.5.1.4	Exceptions and Pre-Substitutions	32
2.5.1.5	Rounding	32
2.5.2	Previous Implementations	33
3	SCALED POSITIVE, NEGATIVE, AND ALTERNATING SIGN MAPS	35
3.1	Generalizations of One-Dimensional Maps	35
3.1.1	The Proposed Maps	35
3.1.1.1	Variations on Logistic Map	36
3.1.1.2	Variations on Tent Map	37
3.1.2	Alternating Sign Maps	38
3.1.2.1	Mostly Positive Logistic Map	38
3.1.2.1.1	Parameters' Ranges	38
3.1.2.1.2	Fixed Points and Their Stability	41
3.1.2.2	Mostly Positive Tent Map	42
3.1.2.2.1	Parameters' Ranges	42
3.1.2.2.2	Fixed Points and Their Stability	42
3.1.2.2.3	Periodic Points and Their Stability	43
3.2	Scaled Logistic Map: Analysis and Results	43
3.2.1	Independent Scaling $x_{n+1} = \pm\lambda x_n(a \pm bx_n)$	43
3.2.1.1	Positive Logistic Map $f(x, \lambda, a, b) = \lambda x(a - bx)$	43
3.2.1.1.1	Range of λ	43
3.2.1.1.2	Fixed Points and Stability Condition	44
3.2.1.2	Mostly Positive Logistic Map $f(x, \lambda, a, b) = -\lambda x(a - bx)$	44
3.2.1.2.1	Range of λ	44
3.2.1.2.2	Fixed Points and Stability Condition	45
3.2.1.3	Maximum Lyapunov Exponent	45
3.2.2	Vertical Scaling $x_{n+1} = \pm\lambda x_n(1 \pm bx_n)$	46
3.2.2.1	Positive Logistic Map $f(x, \lambda, b) = \lambda x(1 - bx)$	46
3.2.2.1.1	Range of λ	46
3.2.2.1.2	Fixed Points and Stability Condition	46
3.2.2.1.3	Steady State Solutions Versus System Parameters	46
3.2.2.2	Mostly Positive Logistic Map $f(x, \lambda, b) = -\lambda x(1 - bx)$	48
3.2.2.2.1	Range of λ	48
3.2.2.2.2	Fixed Points and Stability Condition	48
3.2.2.2.3	Steady State Solutions Versus System Parameters	48
3.2.2.3	Maximum Lyapunov Exponent	49
3.2.3	Zooming $x_{n+1} = \pm\lambda x_n(a \pm x_n)$	50
3.2.3.1	Positive Logistic Map $f(x, \lambda, a) = \lambda x(a - x)$	51
3.2.3.1.1	Range of λ	51
3.2.3.1.2	Fixed Points and Stability Condition	51

3.2.3.1.3	Steady State Solutions Versus System Parameters	51
3.2.3.2	Mostly Positive Logistic Map $f(x, \lambda, a) = -\lambda x(a - x)$	52
3.2.3.2.1	Range of λ	52
3.2.3.2.2	Fixed Points and Stability Condition	52
3.2.3.2.3	Steady State Solutions Versus System Parameters	52
3.2.3.3	Maximum Lyapunov Exponent	54
3.3	General Schematic of Logistic Bifurcation Diagrams	55
3.3.1	Positive Logistic Map $x_{n+1} = \lambda x_n(a - bx_n)$	55
3.3.1.1	Bifurcation Diagram Versus λ	55
3.3.1.2	Bifurcation Diagram Versus a	55
3.3.2	Mostly Positive Logistic Map $x_{n+1} = -\lambda x_n(a - bx_n)$	56
3.3.2.1	Bifurcation Diagram Versus λ	56
3.3.2.2	Bifurcation Diagram Versus a	56
3.4	Logistic Map Design Procedure	56
3.4.1	Design Examples	57
3.4.2	Encryption System	58
3.5	Scaled Tent Map: Analysis and Results	60
3.5.1	Independent Scaling $x_{n+1} = \pm\mu \min(x_n, a - bx_n)$	60
3.5.1.1	Positive Tent Map	60
3.5.1.1.1	Parameters' Ranges	61
3.5.1.1.2	Fixed Points and Their Stability	61
3.5.1.2	Mostly Positive Tent Map	62
3.5.1.2.1	Parameters' Ranges	62
3.5.1.2.2	Fixed Points and Their Stability	63
3.5.1.2.3	Periodic Points and Their Stability	63
3.5.1.3	Maximum Lyapunov Exponent	64
3.6	General Schematic of Tent Bifurcation Diagrams	66
4	GENERAL POWERING MAP	67
4.1	Motivation	67
4.2	Proposed Map $x_{n+1} = r \min(x_n^\alpha (a - bx_n)^\beta, x_n^\beta (a - bx_n)^\alpha)$	68
4.2.1	Mathematical Analysis	68
4.2.1.1	Positive Control Parameter Case $r \geq 0$	69
4.2.1.1.1	Parameters' Ranges	69
4.2.1.1.2	Fixed Points and Their Stability	69
4.2.1.2	Negative Control Parameter Case $r < 0$	70
4.3	Transition Map $(\alpha, \beta) = (1, 0) \rightarrow (1, 1)$	71
4.3.1	Behavior From the Tent Map to the Logistic Map	71
4.3.2	Boundary Analysis	72
4.3.2.1	Tent Map	74
4.3.2.1.1	Positive Control Parameter Case $r \geq 0$	74
4.3.2.1.2	Negative Control Parameter Case $r < 0$	74
4.3.2.2	Logistic Map	74
4.3.2.2.1	Positive Control Parameter Case $r \geq 0$	74
4.3.2.2.2	Negative Control Parameter Case $r < 0$	75
4.3.3	Key-points of Bifurcation Diagram in the Transition Region	75
4.3.3.1	Positive Control Parameter Case $r \geq 0$	76

4.3.3.2	Negative Control Parameter Case $r < 0$	77
4.3.4	Maximum Lyapunov Exponent (MLE)	77
4.4	New Maps at Other Values for (α, β)	78
4.4.1	Sub-Tent Map: $(\alpha, \beta) = (0, 0) \rightarrow (0, 1)$	78
4.4.2	Sub-Logistic Map: $\beta = \alpha$ and $\alpha < 1$	79
4.4.3	Higher Order Map: $\alpha = 1$ and $\beta > 1$	80
4.4.4	Super-Logistic Map: $\beta = \alpha$ and $\alpha > 1$	82
4.5	A New Bifurcation Diagram	83
4.5.1	Positive Parameter $r \geq 0$	83
4.5.1.1	Range of β	83
4.5.1.2	Corresponding Allowed Values of r	84
4.5.2	Negative Parameter $r < 0$	84
4.6	Implementation Issues	85
5	FINITE-PRECISION LOGISTIC MAP	87
5.1	Digital Representation of the Logistic Map	87
5.1.1	Assumptions of Fixed-Point Binary Representation	87
5.1.2	Six Different Maps in Fixed-point Arithmetic	88
5.2	Properties of the Selected Maps	91
5.2.1	Key-points of the Bifurcation Diagram	91
5.2.1.1	Double-Precision Floating-Point	92
5.2.1.2	Fixed-Point Implementation	93
5.2.1.3	Sensitivity to Initial Conditions	94
5.2.1.4	Precision Threshold	94
5.2.2	Time Waveforms	94
5.2.2.1	Precision and Initial Point Effects at $\lambda = 3.9375$	95
5.2.2.2	Precision and Initial Point Effects at $\lambda = 3.984375$	97
5.2.2.3	Precision and Initial Point Effects at $\lambda = -1.9375$ and $\lambda = -1.984375$	97
5.2.2.4	Precision Threshold	98
5.2.3	Periodicity of the Generated Sequence	98
5.2.3.1	Positive Logistic Map (Positive Control Parameter Case)	100
5.2.3.2	Mostly Positive Logistic Map (Negative Control Param- eter Case)	100
5.2.4	Maximum Lyapunov Exponent	101
5.2.4.1	Analytical Derivative Formula	101
5.2.4.2	Numerical Approximation of First Derivative	102
6	ELEMENTARY FUNCTIONS: THE POWER OPERATION $Z = X^Y$	106
6.1	When do we Need to Perform the Calculation x^y ?	107
6.2	Definitions Proposed by IEEE 754-2008, C99, and C11 Standards	107
6.2.1	IEEE 754-2008 Standard for Floating-Point Arithmetic (IEEE Std 754-2008)	107
6.2.2	C99 and C11 Standards	109
6.3	Mathematical Discussion	109
6.3.1	The Base $x =$ Positive One ($+1^y$)	110
6.3.2	The Base $x =$ Positive Zero ($+0^y$)	111

6.3.3	The Base $x = \text{Positive Infinity } (+\infty^y)$	112
6.3.4	The Base $x = \text{NaN } (\text{NaN}^y)$	113
6.3.5	The Base $x = \text{Negative One } (-1^y)$	113
6.3.6	The Base $x = \text{Negative Zero } (-0^y)$	115
6.3.7	The Base $x = \text{Negative Infinity } (-\infty^y)$	116
6.3.8	The Cases of $y = \pm\infty$	117
6.3.9	Complex Power Function	118
6.4	Results of Different Programming Languages	120
6.4.1	Binary Real Power	120
6.4.1.1	Gcc Compiler Version 4.5.2 Along With Math Library .	120
6.4.1.2	Gcc Compiler Version 4.5.2 Along With MathCW Li- brary (Binary Format)	120
6.4.1.3	MATLAB 6.5 Using the Function “realpow”	121
6.4.1.4	MATLAB R2012a, R2012b, and R2014b Using the Function “realpow”	121
6.4.1.5	Octave 3.6.2 Using the Function “realpow”	121
6.4.1.6	Mathematica 8 and 9	122
6.4.2	Decimal Real Power	122
6.4.2.1	Gcc Compiler Version 4.5.2 Along With DecNumber Library (Version 3.68)	122
6.4.2.2	Gcc Compiler Version 4.5.2 Along With Intel Decimal Library (Release 2.0 Update 1)	124
6.4.2.3	Gcc Compiler Version 4.5.2 Along With MathCW Li- brary (Decimal Format)	124
6.5	Summary of the Results of the Studied Implementations Versus Current Standards	125
7	CONCLUSIONS AND SUGGESTIONS FOR FUTURE WORK	126
	References	130

List of Tables

2.1	Classification of chaotic generators	9
2.2	Orbits of the logistic map $f(x)$ at different values for λ	15
2.3	The properties of generalized logistic map with arbitrary power (2.29)	24
2.4	The properties of generalized tent map with arbitrary power (2.31)	25
2.5	The properties of the generalized scaled tent map (2.32)	25
3.1	Main differences between the proposed unity scaling logistic maps	37
3.2	Main differences between the proposed unity scaling tent maps	38
3.3	Comparison between the main aspects of the proposed generalized logistic maps	57
3.4	Constraints of the design problem	57
3.5	Four design cases of the logistic map	59
3.6	Encrypted HEX code for each design given $(\lambda_{fix}, a_{fix}, b_{fix})$	60
4.1	Parameters of tent and logistic maps	76
4.2	Key-points of the bifurcation diagram for sub-tent and sub-logistic maps	81
5.1	Relation between number of fractional bits and step length	88
6.1	Results for x^y for $x = +\infty, -\infty, +0, -0, +1, -1$ and NaN (QNaN) as defined in IEEE Std 754-2008, the result is QNaN for SNaN operands	108
6.2	Results for x^y for $y = +\infty$ and $-\infty$ as defined in IEEE Std 754-2008 where $\text{NaN}^{(\pm\infty)} = \text{NaN}$	108
6.3	Results for x^y for $x = +\infty, -\infty, +0, -0, +1, -1$ and NaN as defined in C99 and C11 standards	110
6.4	Results for x^y for $y = +\infty$ and $-\infty$ as defined in C99 and C11 standards where $\text{NaN}^{(\pm\infty)} = \text{NaN}$	110
6.5	Results for x^y for $x = +\infty, -\infty, +0, -0, +1, -1$ and NaN as proposed by our mathematical discussion, the result is QNaN for SNaN operands	117
6.6	Results for x^y for $y = +\infty$ and $-\infty$ as proposed by our mathematical discussion	118
6.7	Results for x^y for $x = +\infty, -\infty, +0, -0, +1, -1$ and NaN for different binary real power implementations, two more inconsistencies for Mathematica are: $(+\infty)^{(\pm 0)}$ yielding Indeterminate, $(+\infty)^{+\infty}$ and $(\pm 0)^{-\infty}$ yielding Complex Infinity	123
6.8	Results for x^y for $y = +\infty$ and $-\infty$ for different binary real power implementations	124
6.9	Summary for the binary real power function results	125
6.10	Summary for the binary real power function forms of inconsistency and incompatibility	126
6.11	The decimal power function results and forms of inconsistency and incompatibility, where MathCW is completely consistent with the standards	127

List of Figures

2.1	Attractor and orbit diagram of (a) Logistic map, (b) Tent map, (c) Gauss map, (d) Hénon map, and (e) Duffing map, (f) Lorenz attractor and its projections in three planes, and (g) Rössler attractor and one of its projections	10
2.2	Maximum Lyapunov exponent versus the system parameter (a) Logistic map and (b) Tent map	19
2.3	Existence of bifurcation diagram for negative system parameter (a) Logistic map and (b) Tent map	20
2.4	Algebraic completion of the real numbers according to IEEE 754-2008	30
2.5	Double precision floating-point binary representation	30
3.1	Bifurcation diagrams and graphs for the proposed unity scaling logistic maps	37
3.2	Bifurcation diagrams and graphs for the proposed unity scaling tent maps (a) $f_1(x) = \mu \min(x, 1 - x)$, (b) $f_2(x) = -\mu \min(x, 1 - x)$, (c) $f_3(x) = -\mu \min(-x, 1 + x)$, and (d) $f_4(x) = \mu \min(-x, 1 + x)$	38
3.3	Maximum value of λ for mostly positive logistic map (a) $\lambda = 1.6$, (b) $\lambda = 2$, and (c) $\lambda = 2.5$	39
3.4	Domain and range of $f(x) = -\lambda_{max}x(1 - x)$	39
3.5	Sensitive dependence on initial point for mostly positive logistic map at $\lambda = 2$ (a) Time waveform $x_0 = 0.05$, (b) Time waveform $x_0 = 0.06$, and (c) Initial point effect	41
3.6	Domain and range of $f(x) = -\mu_{max} \min(x, 1 - x)$	42
3.7	Bifurcation diagram vs. λ for independent scaling positive logistic map at (a) $b = 2$ and $a = \{0.25, 0.5, \dots, 2\}$ and (b) $a = 4$ and $b = \{0.25, 0.5, \dots, 2\}$	45
3.8	Bifurcation diagram vs. λ for independent scaling mostly positive logistic map at (a) $b = 2$ and $a = \{0.25, 0.5, \dots, 2\}$ and (b) $a = 4$ and $b = \{0.25, 0.5, \dots, 2\}$	46
3.9	MLE of independent scaling positive logistic map as a function of (a) λ and a at $b = 2$, (b) λ and b at $a = 4$, (c) a and b at λ_{max} , and (d) Full-range chaotic output versus a and b	47
3.10	MLE of independent scaling mostly positive logistic map as a function of (a) λ and a at $b = 2$, (b) λ and b at $a = 4$, (c) a and b at λ_{max} , and (d) Full-range chaotic output versus a and b	48
3.11	Function iterations of vertical scaling positive logistic map $f^m(x, \lambda, b)$ at $b = 2$ for $m = \{1, 2, 4, 6\}$	49
3.12	(a) Bifurcation diagram versus λ for different values of $b = \{0.2, 0.5, 5\}$ and Cobweb plot at $\lambda = 4, b = 10$, (b) Steady state solutions of x versus b for different values of $\lambda = \{2, 3.3, 3.83, 4\}$ for vertical scaling positive logistic map	49
3.13	(a) Bifurcation diagram versus λ for different values of $b = \{0.2, 0.5, 5\}$ and Cobweb plot at $\lambda = 2, b = 10$, (b) Steady state solutions of x versus b for different values of $\lambda = \{1.3, 1.5, 1.83, 2\}$ for vertical scaling mostly positive logistic map	50
3.14	Ten different snapshots of the bifurcation diagram versus λ for vertical scaling (a) positive and (b) mostly positive maps at $b = \{0.1, 0.2, \dots, 1\}$	50

3.15	MLE as a function of both λ and b for vertical scaling (a) positive and (b) mostly positive logistic maps	51
3.16	(a) Bifurcation diagram versus λ for different values of $a = \{0.2, 0.5, 2\}$ and Cobweb plot at $a = 4, \lambda = 1$, (b) Bifurcation diagram versus a for different values of $\lambda = \{0.5, 2, 4, 8\}$ for zooming positive logistic map	52
3.17	(a) Bifurcation diagram versus λ for different values of $a = \{0.2, 0.5, 2\}$ and Cobweb plot at $a = 4, \lambda = 0.5$, (b) Bifurcation diagram versus a for different values of $\lambda = \{0.5, 1, 2, 4\}$ for zooming mostly positive logistic map	53
3.18	Ten different snapshots of the bifurcation diagram versus λ for zooming (a) positive map and (b) mostly positive map at $a = \{0.2, 0.4, \dots, 2\}$	53
3.19	Ten different snapshots of the bifurcation diagram versus a for zooming (a) positive map and (b) mostly positive map at $\lambda = \{0.2, 0.4, \dots, 2\}$	54
3.20	MLE as a function of both λ and a for zooming (a) positive logistic map and (b) mostly positive logistic map	54
3.21	General bifurcation diagrams of independent scaling positive logistic map (a) versus λ and (b) versus a	55
3.22	General bifurcation diagrams of independent scaling mostly positive logistic map (a) versus λ and (b) versus a	56
3.23	Simple text encryption system (a) Encryption scheme and (b) Decryption scheme	58
3.24	System key used in the encryption scheme	60
3.25	Bifurcation diagram vs. μ for independent scaling positive tent map at (a) $b = 0.5$ and $a = \{0.25, 0.5, \dots, 2\}$, (b) $a = 4$ and $b = \{0.2, 0.3, \dots, 0.8\}$, (c) $b = 2$ and $a = \{0.25, 0.5, \dots, 2\}$, and (d) $a = 4$ and $b = \{2, 3, \dots, 8\}$	62
3.26	Bifurcation diagram vs. μ for independent scaling mostly positive tent map at (a) $b = 0.5$ and $a = \{0.25, 0.5, \dots, 2\}$, (b) $a = 2$ and $b = \{0.2, 0.3, \dots, 0.8\}$, (c) $b = 2$ and $a = \{0.25, 0.5, \dots, 2\}$, and (d) $a = 4$ and $b = \{2, 3, \dots, 8\}$	64
3.27	MLE of independent scaling positive tent map as a function of (a) μ and a at $b = 2$, (b) μ and b at $a = 4$, (c) a and b at μ_{max} , and (d) Full-range chaotic output versus a and b	65
3.28	MLE of independent scaling mostly positive tent map as a function of (a) μ and a at $b = 2$, (b) μ and b at $a = 4$, (c) a and b at μ_{max} , and (d) Full-range chaotic output versus a and b	65
3.29	General schematic of the bifurcation diagram vs. μ of independent scaling positive tent map (a) $b \leq 1$ and (b) $b > 1$	66
3.30	General schematic of the bifurcation diagram vs. μ of independent scaling mostly positive tent map (a) $b \leq 1$ and (b) $b > 1$	66
4.1	Bifurcation diagrams of general powering map for various values of α and β starting at initial point $x_0 = 0.01$ (a) $(\alpha, \beta) = (1, 0)$, (b) $(\alpha, \beta) = (1, 1)$, and (c) $(\alpha, \beta) = (1, 0.5)$	68
4.2	Domain and range of the proposed map with arbitrarily chosen parameters	71
4.3	Curves of transition map $f(x, \beta)$ and their fixed points at $\beta = \{0, 0.1, \dots, 1\}$ (a) at $r = (2)^{\beta+1}$ and (b) at $r = -1.9$	72
4.4	Surface plot of transition map $f(x, \beta)$ (a) at $r = (1.999)^{\beta+1}$ and (b) at $r = -1.9$	72
4.5	Bifurcation diagrams of the transition map for various values of β starting at initial point $x_0 = 0.01$	73

4.6	Cobweb plots of the proposed map at $\alpha = 1$ and various values of β and r starting at initial point $x_0 = 0.0005$	73
4.7	Key-points of bifurcation diagrams versus β at $\alpha = 1$ starting at $x_0 = 0.01$ (a) $r \geq 0$ and (b) $r < 0$	76
4.8	Eleven snapshots of the bifurcation diagrams of the proposed map at $\alpha = 1$ and $\beta = \{0, 0.1, \dots, 1\}$ starting at initial point $x_0 = 0.01$ (a) $r \geq 0$ and (b) $r < 0$	76
4.9	MLE as a function of both r and β for transition map (a) $r \geq 0$ and (b) $r < 0$	78
4.10	Real bifurcation diagrams of the sub-tent map for various values of β starting at initial point $x_0 = 0.05$	79
4.11	(a) Curves of sub-tent map $f(x, \beta)$ and their fixed points at $\beta = \{0.1, 0.2, \dots, 1\}$ at $r = (2)^\beta$ and (b) its surface plot at $r = (1.999)^\beta$	79
4.12	Real bifurcation diagrams of the sub-logistic map for various values of α starting at initial point $x_0 = 0.05$	80
4.13	(a) Curves of the sub-logistic map $f(x, \alpha)$ and their fixed points at $\alpha = \{0.1, 0.2, \dots, 1\}$ at $r = (2)^{2\alpha}$ and (b) its surface plot at $r = (1.999)^{2\alpha}$	80
4.14	Bifurcation diagrams of higher order map for various values of β starting at initial point $x_0 = 0.5$	81
4.15	(a) Curves of higher order map $f(x, \beta)$ and their fixed points at $\beta = \{1, 1.5, \dots, 5\}$ at $r = (2)^{\beta+1}$ and (b) its surface plot at $r = (1.999)^{\beta+1}$	81
4.16	Bifurcation diagrams of super-logistic map for various values of α starting at initial point $x_0 = 0.5$	82
4.17	(a) Curves of super-logistic map $f(x, \alpha)$ and their fixed points at $\alpha = \{1, 1.5, \dots, 5\}$ at $r = (2)^{2\alpha}$ and (b) its surface plot at $r = (1.999)^{2\alpha}$	83
4.18	Bifurcation diagrams versus the parameter β for different values of the parameter $r > 0$ starting at initial point $x_0 = 0.5$	83
4.19	The minimum value of β for which the bifurcation diagram exists (β_{min}) as a function of the system parameter r such that $r > 0$	84
4.20	Snap shots of the bifurcation diagram versus β starting at initial point $x_0 = 0.5$ for (a) $r = \{2, 3, \dots, 7\}$ and (b) $r = \{-7, -6, \dots, -1\}$	85
5.1	Bifurcation diagram versus the control parameter λ for the six maps starting at $x_0 = 0.5$ at $p = 9$	90
5.2	Bifurcation diagram versus the control parameter λ for the six maps starting at $x_0 = 0.125$ at $p = 9$	90
5.3	Bifurcation diagram of $f_3(x)$ versus the control parameter λ for different values of bus width starting at $x_0 = 0.5$	91
5.4	Bifurcation diagram of $f_6(x)$ versus the control parameter λ for different values of bus width starting at $x_0 = 0.5$	92
5.5	The key-points of the bifurcation diagram for both positive control parameter maps at different precisions (a) The first bifurcation point and (b) The average maximum value	93
5.6	The key-points of the bifurcation diagram for both negative control parameter maps at different precisions (a) The first bifurcation point, (b) The average minimum value, and (c) The average maximum value	94
5.7	Time waveforms of $f_3(x)$ at $\lambda = 3.9375$ starting at different initial conditions and various precisions	95

5.8	Time waveforms of $f_6(x)$ at $\lambda = 3.9375$ starting at different initial conditions and various precisions	96
5.9	Cobweb plots of $f_6(x)$ starting at different initial conditions (a) $x_0 = 0.125$, (b) $x_0 = 0.25$, (c) $x_0 = 0.375$, and (d) $x_0 = 0.5$	97
5.10	Time waveforms of $f_3(x)$ at $\lambda = 3.984375$ starting at different initial conditions and various precisions	98
5.11	Time waveforms of $f_6(x)$ at $\lambda = 3.984375$ starting at different initial conditions and various precisions	99
5.12	Time waveforms of $f_3(x)$ at $\lambda = -1.9375$ starting at different initial conditions and various precisions	100
5.13	Time waveforms of $f_6(x)$ at $\lambda = -1.9375$ starting at different initial conditions and various precisions	101
5.14	Time waveforms of $f_3(x)$ at $\lambda = -1.984375$ starting at different initial conditions and various precisions	102
5.15	Time waveforms of $f_6(x)$ at $\lambda = -1.984375$ starting at different initial conditions and various precisions	103
5.16	Maximum period obtained with different orders of execution plotted versus precision $p = 8 \rightarrow 13$ for (a) $\lambda > 0$ and (b) $\lambda < 0$	103
5.17	MLE evaluation in two different methods using $f_3(x)$ starting at $x_0 = 0.125$ at (a) $\lambda = 3.984375$ and (b) $\lambda = -1.984375$	105
5.18	MLE evaluation in two different methods using $f_6(x)$ starting at $x_0 = 0.125$ at (a) $\lambda = 3.984375$ and (b) $\lambda = -1.984375$	105

List of Symbols and Abbreviations

FPGA	Field Programmable Gate Array
ASIC	Application-Specific Integrated Circuit
PWLCM	Piece-Wise Linear Chaotic Maps
1D	One-Dimensional
2D	Two-Dimensional
LFSR	Linear Feedback Shift Register
FOMs	Figures of Merit
PRNG	Pseudo-Random Number Generator
ODE	Ordinary Differential Equation
IEEE	Institute of Electrical and Electronics Engineers
NaN	Not a Number
MLE	Maximum Lyapunov Exponent
λ	System parameter of logistic map
μ	System parameter of tent map
r	System parameter of general powering map

Abstract

Chaotic maps and elementary functions are highly required for scientific computation and utilized in many widely spread applications. For example, one-dimensional discrete chaotic maps are used for modeling and pseudo-random number generation and the power function $z = x^y$ is highly required for financial computations. Moreover, we utilize the power function in some of our proposed novel chaotic maps. Yet, the increased sensitivity of these functions makes them more subject to errors than most of the existing digitally implemented mathematical relations. In this thesis, we aim to bridge the gap between mathematical analysis of the conventional and generalized forms of these two sets of problems on one hand, and their implementation on digital platforms on the other hand.

First, we propose variations on the signs of parameters in the most famous one-dimensional discrete chaotic maps: the logistic and tent maps. Four different variations on each of the logistic and tent maps are proposed which allow output ranges to have single sign or alternating signs utilizing roughly the same equation with varied signs for the parameters. The new ranges allow modeling of additional phenomena and exhibit wider output ranges with longer sequences. Scaling parameters (a, b) are added that can be used to control the ranges of output responses and one of the most important properties of chaotic maps: the bifurcation diagram. Special cases of the proposed scaling technique are also discussed. Then, we introduce a general powering map with shaping parameters (α, β) employed as arbitrary powers that add the capability of controlling the shape of the map response and bifurcation diagram. The powers could be adjusted to obtain the tent map response, the logistic map response, and novel chaotic responses in between which we called “Transition map”. Other ranges of the powering parameters are studied as well.

On the implementation side, we analyze the behavior of logistic map, with either positive or negative parameter, in finite precision implementations and emphasize the impact of finitude on dynamical properties of the map while varying the used precision. Our approach in this study is novel as it uses the fixed-point toolbox of MATLAB to simulate FPGA hardware realizations in which the operations are executed individually in a sequential manner with the truncation step implemented between them, and the order of execution and its effects are discussed for the first time to the best of our knowledge. Accordingly, a precision threshold is recommended, in addition to indicating the advantages of our proposed map with negative control parameter over the conventional one.

For the power function, we propose a mathematically justified definition for its results on the occurrence of special values of the operands and test how different standards and software implementations deal with them. We present inconsistencies between the implementations and the standards and discuss incompatibilities between different versions of the same software. This study could aid in releasing implementations that guarantee reproducible programs, i.e., producing the same result in all implementations of a language. In addition, combining this study with the previous part could provide a framework for analyzing finite precision implementation impacts on the behavior of the newly proposed general powering map, after deciding the best way to implement it.

Chapter 1: Introduction

1.1 Motivation

The area of intersection between mathematics and computers is very rich in studies and explorations as it bridges the gap between two well-established fields. Mathematics and computer science are in fact complementary for each other in handling many problems. It is not surprising that many educational institutes contain departments that hold the name of mathematics and computer science. In addition, many researchers in either academic disciplines have a background somehow related to the other discipline and many others work in an interdisciplinary of both sciences. On the other hand, industrial research, mainly utilizing programmable methods of problem solving, indeed requires engineers with solid background in mathematics. In this thesis, we focus on a set of mathematical problems and handle them from two different, but highly related, aspects. First, we do our best to define the complete picture of the problem through widening the lens by which we view it and exploring areas or ranges that have not been encountered before. Then, possible methods of numerically representing these problems are surveyed and evaluated, whether on general purpose digital computers or specific purpose digital realizations. It is more common that researchers focus either on problem formulation and the proposed solution(s) and may be their implementations accompanied by a quick validation, or on verification techniques of various designs such as debugging/testing. Although computer simulations are used throughout the thesis to illustrate the general behavior of the studied problems, we do not blindly accept the results yielded by computer simulations and devote complete chapters to analyze and criticize some aspects of these results. Among all mathematical problems with calculations performed through software or hardware digital realizations, we select two sets of problems to study. In addition to the increased sensitivity of the selected problems relative to other digitally computed examples that are subject to errors, they are practically more important. Consequently, our selection criteria is practical need; namely the presence of widely utilized applications that massively employ these computations. The two sets of problems discussed in this thesis are:

1. Chaotic maps with the most widely utilized discrete one-dimensional (1D) examples in modeling and pseudo-random number generation: non-linear quadratic maps (the logistic map) and piece-wise linear maps (the tent map).
2. Elementary functions with the most complicated, yet highly required in financial computations: the power function $z = x^y$.

Another perspective of choosing these two sets of problems is that they represent examples on different classifications of functions: algebraic and transcendental functions. The studied discrete chaotic maps iterate a polynomial, i.e., they are classified as algebraic functions. On the other hand, elementary functions are transcendental functions which can not be fully represented in the form of a polynomial or cannot be defined as the root of a polynomial equation, i.e., they cannot be expressed in terms of a finite sequence of the basic operations: addition, subtraction, multiplication, division, and extraction of roots. Both sets of problems have been considered for implementation in many researches confined to the integer domain only, and even more positive values only. The reason behind

these limits was either to avoid the many questions that could arise when attempting to implement them in the domain of real numbers, or because their behavior for negative values was not explored before. In the analysis part of the thesis, we provide a framework for analyzing modifications or generalizations of 1D chaotic maps starting from variations on the signs of the included parameters, introducing extra degrees of freedom using scaling parameters, and proposing a completely new general map with arbitrary powers that could implement various 1D maps with different behavior utilizing the same resources. In addition, we propose a mathematical definition for the power function for special values of the operands. The proposed definition could pave the way towards more robust requirements, not just recommendations, to be added to programming standards that define correct results that implementations should yield. Moreover, we do not confine our analysis to the real domain as both sets of problems could yield complex results for some range(s) of the included parameters/operands.

1.1.1 Chaotic Maps and Numerical Computation

After the invention of personal computers with screens capable of displaying graphs, scientists and engineers have been able to see that important equations which interest them in their own fields had rather strange solutions, at least for some ranges of parameters that appear in the equations. This can be observed in experiments and in computer models of behavior in many fields that extend massively to include nearly all branches of science such as: mathematics, physics, chemistry, biology, microbiology, geology, engineering, computer science, robotics, population dynamics, finance, economics, algorithmic trading, and meteorology. Chaotic behavior is reported when observations or measurements of an explored system vary unpredictably with no discernible regularity or order. A chaotic system could exhibit an infinite number of periodic orbits, some of which could be of arbitrarily long period. This can be partially attributed to the presence of a kind of non-linearity associated with an iterative system. Such behavior was formerly explained as experimental error or noise before the evolution of the definition of “chaos”. Chaos is identified with non-periodicity and sensitive dependence on parameters and initial conditions and it is characterized by its complicated dynamics. A dynamical system consists of a set of possible states (represented by one or more real variables), in addition to a deterministic rule that determines the present state in terms of past states. A dynamical system displaying sensitive dependence on initial conditions on a closed invariant set (which consists of more than one orbit) is called chaotic [112]. Examples on both chaotic and non-chaotic iterative systems are given in Chapter 2 Subsubsection 2.1.2.1.

Owing to the continuous need for novel and more unpredictable chaotic generators, modified and generalized versions of many chaotic maps have been suggested through various ways. For 1D maps, a generalization could be achieved through adding an extra space dimension converting it to two-dimensional (2D) map, or increasing the operations count in the map equation. We think that it is always better to maintain simplicity and target modifications with the least, or even no extra software complexity or hardware resources needed. Keeping this principle in mind, we aimed at proposing generalizations along with considering implementation issues. The proposed modifications start from trying various variations for the signs of the included parameters. Then, a generalization based on introducing extra scaling parameters which are classified according to their direct

impact on the resulting behavior in a sense that has not been presented before to the best of our knowledge. The final generalization introduces a map with parameters that would determine its behavior. These parameters appear as powers in the map equation which imposed studying the power operation separately. They are not scaling parameters but “shaping” parameters that specify the type of the map where the shape of the defining function and the system response versus different parameters are not scaled but completely changed. The proposed general map with both the shaping and the scaling parameters could utilize the same implementation or resources to generate a huge amount of unique sequences, with different properties, according to the values of five different parameters (r , a , b , α , and β). For all the proposed maps, comprehensive mathematical analysis is presented and measures of different properties, that are characteristics of chaos theory which define every map, are provided.

Mathematically, it is possible to analyze the behavior of a chaotic system and define its response using the equations describing it. However, mapping these systems to digital software or hardware and solving them numerically would yield results that inevitably deviate from the expected analytical solutions. Many methods for pseudo-random number generation have been suggested as efficient and random-like based on simulations without specifying the utilized precision, or even based on mathematical analysis assuming infinite precision. These reasons are insufficient or cannot be considered an evidence for their efficiency or security when employed in digital chaotic ciphers. On the other hand, other methods have been claimed to be inefficient with no sufficient reasons or without attempting to enhance them on both formulation and implementation levels. After analyzing the behavior of the studied chaotic maps in new ranges and for different modifications and generalizations, we criticize limited precision implementations of these maps presenting a framework for studying the effect of finitude on the maps’ dynamical properties. The framework that gathers deteriorations in the behavior of the maps in fixed-point arithmetic can be extended to floating-point arithmetic as well.

Most existing numerical implementations, whatever robust, yield odd or strange results on the occurrence of certain computations. Errors that would be associated with numerical computation can be subdivided into different types according to the reason why they would occur

- Representational Errors: Errors in decimal-binary conversions or vice versa.
- Approximation Errors: Errors resulting from bad numerical quality of mathematical functions computed using approximations, e.g., polynomial approximations.
- Rounding Errors: On a computer, the infinite set of real numbers needs to be represented by a finite set of numbers called machine numbers. The error caused by rounding (round-off error), i.e., mapping the real result to a machine number is called rounding error. Truncation is one of the simplest rounding methods from the viewpoint of implementation, yet it could cause complete deviation from the expected properties especially in low precisions as in Chapter 5.
- Definitional Errors: These errors could arise when results of certain expressions are pre-substituted by certain values chosen by definition such as in the case of indeterminate expressions or special cases of the power function as in Chapter 6. These results are probably fetched by table-lookup and should be well-defined for

expressions such as those arising when passing a special value as an argument to the power function, which represents a difficulty in reaching an agreement that minimizes definitional errors.

- **Computational Errors:** Even if all terms constituting the mathematical expression were correctly rounded, there remains the errors arising from the cancellation due to finite precision, or the numerical computation method used to evaluate the function. This could also be viewed as an accumulation of numerical round-off. For complicated functions, there will probably be a long sequence of calculations each of which introducing some error.

1.1.2 Power Function and Floating-Point Representation

Chaos is not the only example on exchangeable suggestions between mathematicians and implementation engineers. Many other computational problems exist that involve endless arguments on how to do them in “exact real arithmetic.” These problems start from the basic operations: addition, subtraction, multiplication, division, and extraction of roots performed on the field of real numbers. They extend to be concerned with more complicated operations, e.g., elementary functions such as trigonometric, hyperbolic, and power functions. Nearly all the basic and the other operations can be extended to complex arithmetic though. Floating-point representation of numbers is very important on digital computers because nearly all the calculations involving fractions use it. The floating-point representation is similar to scientific notation; that is: $\text{fraction} \times (\text{radix})^{\text{exponent}}$. Having implementations that yield correct and consistent results for arithmetic functions according to clearly defined standards is an important topic in floating-point arithmetic. Yet, there are numerous bugs in the history of floating-point arithmetic designs that we can point to in order to show the frequent presence of deviations from correctness in various designs. Moreover, some strange behaviors can sometimes arise from difficult numerical problems (unstable, ill-conditioned, . . . etc.) even when the arithmetic is not flawed. The presence of these bugs is not surprising because mapping the continuous real numbers on a finite structure (the floating-point numbers or machine numbers) cannot be done without any trouble. Some famous bugs and their massive economic cost have been widely discussed; we recall some of them below in addition to other bugs that we report in various software implementations of the power function in both binary and decimal floating-point in Chapter 6 and throughout the rest of the thesis. For more examples and samples of user programs, refer to [79].

- The floating-point unit (FPU) of the Intel P5 Pentium processor may return incorrect decimal results due to the well-known Pentium FDIV bug (FDIV is the x86 assembly language mnemonic for floating-point division). Approximately \$475M were set aside by Intel in 1994 to cover costs arising from this issue [31].
- Some strange behavior of some versions of the Excel spreadsheet was reported in [57], in addition to the effects of a compiler’s optimization upon a program, e.g., in MATLAB and LAPACK. The effects of roundoff were reported too in MATLAB and FORTRAN.
- Bugs in the computer algebra system Maple releases 6.0 and 7.0 [82].

- The blast of the Ariane V space shuttle¹ in 1996 was caused by an overflow of a conversion from a floating-point to integer operation. That system was based on the ADA language which was designed to abort on the occurrence of any “arithmetic error”. The overflow that happened was not a serious error, but the system aborted and equipments that cost millions of dollars were completely lost [67].
- Some bugs do not require any programming error; they are due to poor specifications or misuse of units. For example, the Mars Climate Orbiter probe crash on Mars in September 1999, a space shuttle positioned itself to receive a laser beamed from the top of a mountain in June 1985, and a bridge between Germany and Switzerland did not fit at the border in January 2004 because the two countries use a different definition of the sea level [82].

1.2 Main Contributions

In this thesis, we attempt to answer the following questions: how much are well-established problems completely defined in all the possible ranges accompanied with a robust, unanimous mathematical analysis? How much does representation on digital platforms deteriorate the properties of a system or process/procedure and in what sense is deviation reported from its analytical expected results? Hence or otherwise, what is the best way and sufficient precision in which a specific class of problems can be correctly, or even fairly, represented? The main contributions of this work are:

1. Proposing variations on the signs of parameters in the most famous 1D discrete chaotic maps: the logistic and tent maps. New properties are discovered in the negative range(s) of parameters which are mathematically analyzed in detail. Scaling parameters are added that affect the ranges of output responses and one of the most important properties of chaotic maps: the bifurcation diagram.
2. Introducing a general map with shaping parameters employed as arbitrary powers that could be adjusted to obtain the tent map response or the logistic map response, and novel chaotic responses in between as well as the other ranges of the powering parameters. This map adds the capability of modifying the shape of the map response and bifurcation diagram. The capability of scaling is also maintained through adding extra scaling parameters to the general map.
3. Analyzing the behavior of conventional logistic map in finite precision implementations and emphasizing the impact of finitude on dynamical properties of the map. The effect of varying precision on several properties of the logistic map is studied and the differences from the analytical model are discussed. Our approach in this study is novel as it uses the fixed-point toolbox of MATLAB to simulate FPGA hardware realizations in which the operations are executed individually in a sequential manner with the truncation step implemented between them, and the order of execution is discussed for the first time to the best of our knowledge.
4. Proposing a mathematically justified definition for the correct results of the power function $z = x^y$ on the occurrence of the special values ± 0 , ± 1 , \pm infinity, and NaN as its operands, testing how different standards and software implementations for

the power function deal with these special values, and classifying the behavior of different programming languages from the viewpoint of how much they conform to the current standards and our proposed mathematical definition. We present inconsistencies between the implementations and the standards and we discuss incompatibilities between different versions of the same software. The study could aid in releasing implementations that guarantee reproducible programs, i.e., producing the same result in all implementations of a language.

1.3 Publications out of This Work

The following international journal and conference papers have been accepted for publication out of this work.

1. Sayed, W. S., Radwan, A. G., and Fahmy, H. A. H. Design of positive, negative, and alternating sign generalized logistic maps. *Discrete Dynamics in Nature and Society (DDNS)*, ARTICLE ID 586783, 2015.
2. Sayed, W. S. and Fahmy, H. A. H. What are the correct results for the special values of the operands of the power operation? *ACM Transactions on Mathematical Software (TOMS)*, accepted and to appear soon.
3. Sayed, W. S., Radwan, A. G., Fahmy, H. A. H., and Hussein, A. E. Scaling parameters and chaos in generalized 1D discrete time maps. *International Symposium on Nonlinear Theory and its Applications (NOLTA 2015)*, accepted and to appear soon.

1.4 Organization of the Thesis

The remainder of this thesis is organized as follows. First, Chapter 2 provides the mathematical background about non-linear dynamics and chaos theory, namely conventional 1D logistic and tent discrete maps; as well as elementary functions, namely the power function. In addition, basics and main definitions of fixed-point and floating-point arithmetic implementations are reviewed. Previous studies in mathematical analysis of the studied set of problems: chaotic maps and the power function are detailed. Moreover, a survey of previous attempts to come up with generalizations of chaotic maps is presented. A review on attempts to study finite precision logistic map is also included. In addition, previous implementations of the power function are presented and discussed.

Then, Chapter 3 provides four possible variations on the conventional logistic map with two parameters λ and x through varying their signs. The proposed maps are called: positive logistic map, mostly positive logistic map, negative logistic map, and mostly negative logistic map according to the maximum chaotic range of the output. These variations do not confine the output to a restricted range of fractions between 0 and 1 as the conventional map. The new range would be beneficial for many applications such as quantitative financial modeling, traffic, weather forecasting, and others. Moreover, a general design procedure is proposed utilizing two extra parameters (a, b) which may take one of three cases: (a, b) , $a, b \in R^+$ called the independent scaling case, $(1, b)$ called the vertical scaling case, and $(a, 1)$ called the zooming case. The proposed maps are

analyzed from the viewpoint of iteration effect, ranges of parameters, the fixed points, the bifurcation diagrams, and the maximum Lyapunov exponent (MLE) with respect to all system parameters. The general schematic of the bifurcation diagrams with respect to λ , in addition to a new bifurcation diagram with respect to a are provided. Four different design examples are presented to validate the proposed design procedure in addition to testing their efficiency in a simple text encryption application.

Then, Chapter 4 provides a new map with arbitrary powers α, β that could be considered a general form for 1D discrete maps with the tent and logistic maps as special cases. The effect of the presence of an elementary function in the proposed map is studied. The general map would utilize the same hardware resources to yield various shapes of the response and bifurcation diagram. A framework for analyzing the proposed map mathematically and predicting its behavior for various combinations of its parameters is introduced. In addition, the transition from tent map case to logistic map case is presented and explained. The possibility of generating real and imaginary bifurcations in this transition region in addition to the sub-tent region is checked. The map would be suitable even for maps whose iterative relations are not based on polynomials. Moreover, higher degree polynomials, as well as various combinations of the parameters, are investigated.

Then, Chapter 5 discusses how the conventional 1D logistic map can be represented in digital hardware realizations and the assumptions required to simulate this representation in software environments. Six different versions of the map are proposed based on the order of execution of the operations constituting its expression. Two of these versions are chosen primarily to conduct various experiments on them. The results obtained for the two chosen map versions are demonstrated which include: the bifurcation diagram, its key-points, time waveforms, periodicity of the generated sequence, and MLE.

Then, Chapter 6 provides a mathematical analysis of the power floating-point arithmetic operation that we utilized in the proposed general map. This chapter also includes testing the behavior of different implementations of the power functions when special values are encountered in its operands, compared to the behavior recommended by IEEE Standard for floating-point arithmetic 754-2008, and attempting to explain strange results.

Finally, Chapter 7 provides the main contributions of this work, our conclusions based upon the results that we have reached, and suggestions for future work.

Chapter 2: Review About the Studied Problems

This chapter provides a review about the mathematical problems studied throughout the thesis. First, the remarkable importance of chaos theory and its applications is indicated with more emphasis on the 1D discrete logistic and tent maps. The properties of the conventional form of these two maps are analyzed along with presenting the mathematical approach to studying non-linear dynamical systems and chaos. This introduction paves the way towards mathematically analyzing generalized maps that are proposed in the subsequent chapters. The simplest, yet one of the most efficient, generalization that we propose is allowing negative values for the system parameter. The motivation and suggested applications for this generalization is presented. Previous work in generalized maps is reviewed in order that we later on propose our generalizations in the light of this survey. Then, we move to the effect of digitally implementing chaotic maps on their dynamical properties. Several methods of digitization could come to mind such as: software implementations in floating-point arithmetic formats, simulations in fixed-point formats, hardware realizations either in ASIC or FPGAs, and other digital implementations. A literature survey on previous works in this field is presented. The last subsection of this chapter explains the complexity of evaluating the power function $z = x^y$, reviews previous implementations, and provides the main definitions of IEEE standard for floating-point arithmetic.

2.1 Chaos: Motivation and Mathematical Analysis

Chaos theory is a branch of mathematics, which is still in the process of development, classified under the category “applied mathematics to physical sciences”. Strange attractors, deterministic models, sensitivity to initial conditions, and fractals are all inherent to the development of this theory. All categories of applied mathematics are always in a state of continuous development in order to find their way towards formulating recent applications either theoretically or practically. However, the history of chaos theory goes back to the 17th century when there have been many arguments and debates whether every effect that is noted on a certain experiment or when observing natural phenomena can be precisely owed to a given reason or perhaps a list of reasons. Many of these experiments and phenomena can be described by dynamical systems, i.e., their modeling equations relate a quantity to its rate of change, and these are studied as differential equations. As a result, calculus that is classified under the category of pure mathematics is employed as a powerful tool used in investigating, understanding, and describing “change” in natural sciences and phenomena. The study of chaos enables mathematicians and physicists to describe various phenomena in the field of dynamics by the aid of equations and models. Chaos theory precisely describes many of the dynamical systems which exhibit unpredictable, yet deterministic, behavior. Chaotic generators can be classified into discrete time maps and continuous time differential equations. Table 2.1 shows a classification of some of the well-known real chaotic generators. We focus on discrete time maps, however, the most well-known continuous time chaotic Lorenz attractor must be included when handling chaos theory. Other continuous time differential equations that exhibit

chaotic behavior exist such as: Rössler attractor, Duffing equation, and forced Van der Pol. For each generator included in Table 2.1, the number of space dimensions, the popular parameter values used to generate chaotic behavior, as well as the characterizing plot that describes the system response are shown in Fig. 2.1. For 1D maps, the plot presents the map equation or the current iteration x_{n+1} as a function of the previous iteration x_n , in addition to the orbit diagram which shows how the steady state solution varies with respect to the system parameter. Examples for 1D maps are: the logistic, tent, and gauss maps. For two dimensional maps, such as Hénon map and Duffing map, the graph shows one of the involved space dimensions as a function of the other in addition to the orbit diagram. Finally, the solutions of Lorenz and Rössler systems are shown in Fig. 2.1(f) and (g) respectively. In our study, we concentrate on 1D discrete maps, specifically the logistic and tent maps whose properties are closely related since they are conjugate maps [7]. But at first, a historical background about the development of chaos theory is presented.

Table 2.1: Classification of chaotic generators

(a) Discrete time maps

Map	Space dimensions	Equation(s)	Parameter values	Plot
Logistic map	1	$x_{n+1} = \lambda x_n(1 - x_n)$	$\lambda = 4$	Fig. 2.1(a)
Tent map	1	$x_{n+1} = \mu \min(x_n, 1 - x_n)$	$\mu = 2$	Fig. 2.1(b)
Gauss map	1	$x_{n+1} = e^{-\alpha x_n^2} + \beta$	$\alpha = 6.2, \beta = -0.5$	Fig. 2.1(c)
Hénon map	2	$x_{n+1} = 1 - ax_n^2 + y_n$ $y_{n+1} = bx_n$	$a = 1.4$ $b = 0.3$	Fig. 2.1(d)
Duffing map	2	$x_{n+1} = y_n$ $y_{n+1} = -bx_n + ay_n - y_n^3$	$a = 2.75$ $b = 0.2$	Fig. 2.1(e)

(b) Continuous time differential equations

Equation	Space dimensions	Equation(s)	parameter values	Plot
Lorenz attractor	3	$\dot{x} = \sigma(y - x)$ $\dot{y} = x(\rho - z) - y$ $\dot{z} = xy - \beta z$	$\sigma = 10$ $\rho = 28$ $\beta = 8/3$	Fig. 2.1(f)
Rössler attractor	3	$\dot{x} = -y - z$ $\dot{y} = x + ay$ $\dot{z} = b + z(x - c)$	$a = 0.2$ $b = 0.2$ $c = 5.7$	Fig. 2.1(g)

2.1.1 Historical Background

The causality principle is the most basic principle of physics derived from the philosophy of René Descartes, it states that “Every effect has a cause.” In 1687, Isaac Newton applied this principle in order to calculate the planets’ trajectories, demonstrating that the observed motion of planets could be explained by the presence of a force of gravitational attraction among them. A force that is present between every two bodies which is directly proportional to the product of their masses and inversely proportional to the square of the

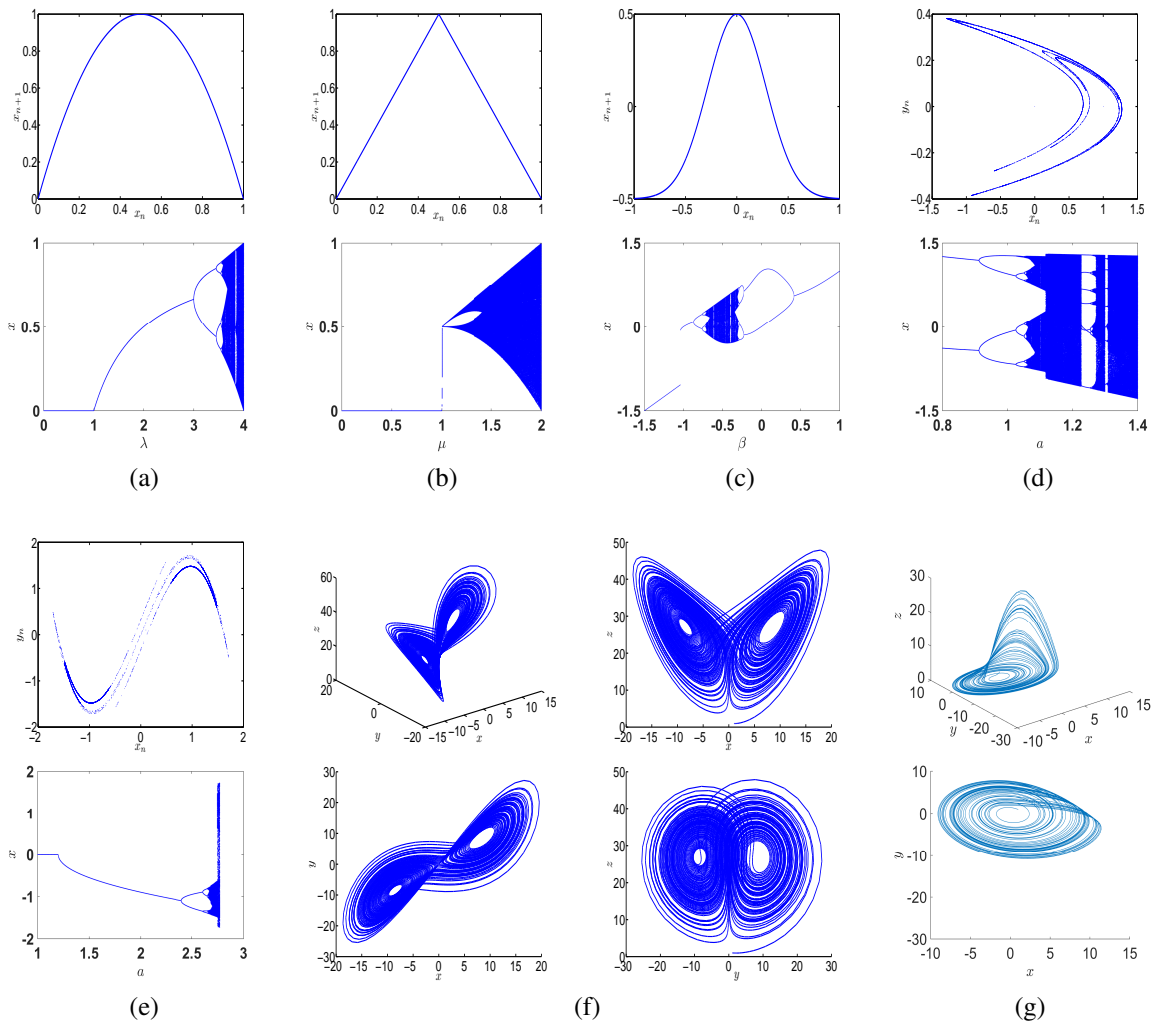


Figure 2.1: Attractor and orbit diagram of (a) Logistic map, (b) Tent map, (c) Gauss map, (d) Hénon map, and (e) Duffing map, (f) Lorenz attractor and its projections in three planes, and (g) Rössler attractor and one of its projections

distance between them. Simplifying the model to a two-body-problem considering the gravitational force of the sun on the earth, his calculations were consistent with the three laws of planetary motion that have been published by Johannes Kepler in 1609 and 1618. This research involved the invention of many calculus theories since fundamental equations of motion involve velocities and accelerations, which are derivatives of position. It is worth mentioning that similar ideas in the field of differential calculus have been developed concurrently by Leibniz. Later on, many other scientists extended and developed methods of using differential equations to describe the behavior of physical systems.

Throughout these years, there have been two well-known types of solutions for differential equations that describe motion: steady state solutions and oscillatory periodic or quasi-periodic solutions. However, physical variations cause real systems to be unpredictable such as friction and air resistance. Several systems with more complicated dynamics have been pointed out such as a pot of boiling water, or the molecules of air colliding in a room. For example, Maxwell suggested that the motion of gas molecules

could lead to progressive amplification of small changes due to an immense number of interacting particles and yield microscopic randomness. The phenomenon of sensitivity to initial conditions and long-term unpredictability were discovered by Poincaré in 1890. He wondered if the laws of nature and the state of the universe are exactly known at the initial moment, could we accurately predict the state of the same universe at a subsequent moment? or even just predict it approximately? Knowing that small differences in the initial conditions may generate very large differences in the final state and the error is enormously amplified, prediction then becomes impossible in such a “random” phenomenon. This was the birth of chaos theory. The work of Poincaré has been revisited and enriched by other scientists later on such as Kolmogorov whose contributions to mathematics are too numerous to be listed in our discussion. In some states of many dynamical systems, sensitive dependence on initial conditions is realized which means that points that start off close together can be widely separated at a later time.

Edward Lorenz is the official discoverer of chaos theory which he defined in the 1960s as [71]:

“Chaos: When the present determines the future, but the approximate present does not approximately determine the future.”

which means that there is chaos when no formula can tell us the value of the system output at a certain iteration n , even if we know the initial point, except by carrying out the successive iterations. Lorenz’s contributions are much related to the invention of high speed computers which allowed the conduction of simulations and numerical experiments on multiple non-linear differential equations as one way of discovering their complicated behavior. Lorenz was first working on weather prediction when he accidentally came across strange behaviors in which he became interested in. He developed the Lorenz system as a simplified mathematical model for weather prediction. The Lorenz attractor is a set of chaotic solutions of the Lorenz system. Plotting the solution of this system resembles figure eight as shown in Fig. 2.1(f). A few years later, the butterfly effect has been discussed in a meeting titled “Predictability: Does the Flap of a Butterfly’s Wings in Brazil Set Off a Tornado in Texas?” [108]. It has been wondered whether two particular weather situations differing as little as the immediate influence of a single butterfly will generally after sufficient time evolve into two situations differing as much as the presence of a tornado? In more technical language, is the behavior of the atmosphere unstable with respect to perturbations of small amplitude? Since then, numerical weather prediction has made enormous progress.

Later on, various laboratory experiments have been carried out in which very familiar settings exhibit unusual nonlinear effects and chaotic behavior. The concepts of dynamical systems became no longer confined to the microscopic world as proposed by Maxwell and others. They extended to include macroscopic systems such as: mechanism of fluids, some common electronic circuits and several types of low-energy lasers. All these are systems which were previously thought to be fairly well understood using the classical models. In this sense, chaos theory is applicable for both microscopic and macroscopic models. The outstanding work of Lorenz paved the way for the evolution of many studies in the field of chaos theory and discovering properties of various equations. Some of which have been thought to be well understood, while others exhibit unusual behaviors but with no supporting mathematical evidence. Subsequent researches have been conducted that constitute the base for all applications employing chaos in the present days. For

example, works of Robert M. May [78] and Mitchell J. Feigenbaum [37] on the logistic map, Mandelbrot on fractals [73], Ruelle and strange attractors [99], and many others. We shall highlight the contributions of various scientists in the fine field of discrete 1D maps throughout the rest of our discussion.

2.1.2 Conventional Logistic Map $x_{n+1} = \lambda x_n(1 - x_n)$

The conventional logistic map is a famous iterative map based on first order nonlinear difference equation which can model growth rate to study either reproduction or starvation phenomena, and is given by:

$$x_{n+1} = f(x_n, \lambda) = \lambda x_n(1 - x_n), \quad \lambda \in R^+, x_n \in [0, 1] \quad (2.1)$$

where λ is the population growth rate or fertility coefficient, and x_n is the relative population size at a discrete time instant n . Detailed analytical studies of logistic maps began in the 1950s. There had been a lot of arguments that such maps exhibit complicated properties beyond the simple oscillatory behavior that had been widely noted before. Yet, these studies had not been collectively demonstrated in a mature way until the map was popularized in a seminal 1976 paper by the biologist Robert May, [78] in part as a discrete-time demographic model analogous to the logistic equation first created by Pierre François Verhulst [110, 111]. A detailed mathematical analysis of the map and its properties has also been presented in [37]. In spite of the simplicity of the mathematical relation with which the logistic map is defined that uses simple and computationally fast operators, it is highly rich in information and indications that are very useful in the field of chaos theory and chaotic systems. Its properties as a smooth map constitute a large portion of the basic study of chaos theory. In addition, it is the most popular example in textbooks and elementary courses on chaotic dynamical systems. Specifically, the applications of the logistic map have increased during the last few decades. For example, in fields as biology, chemistry, physics [70, 101, 105], secure data and image transfer [87], pseudo-random number generation for chaos based communication [59, 61, 88, 90, 104], circuit applications [106], traffic [42], financial modeling, and business cycle theory [10, 52].

2.1.2.1 Exponential Versus Logistic Growth Models

Population growth as time progresses can be modeled by various relations, some of which are predictable ones with analytical solutions and others are not. We are concerned with two relations, the first relation is called the exponential growth model and is given by:

$$g(x, \lambda) = \lambda x, \quad \lambda, x \in R^+ \quad (2.2)$$

which is a linear system that is easy to understand, whereas the second relation is called the logistic growth model and is given by:

$$f(x, \lambda) = \lambda x(1 - x), \quad \lambda \in R^+, x \in [0, 1] \quad (2.3)$$

where λ is a system parameter known as the fertility coefficient, and x is the relative population expressed as the ratio of existing population to the maximum possible population. This model describes the behavior of a population with limited resources which may

increase, but saturation is achieved as the relative population x approaches one forcing the rate of growth λ to be upper bounded. The population at the next time instant is not only proportional to the existing relative population x , but also the remaining capacity $(1 - x)$. It should be noted that this equation provides no model for negative population as divergence occurs for negative values. Instead, the lower bound on x is zero which represents extinction. For a recursive relation, an initial value x_0 is needed, the resulting value will be the next input and so on. For instance, a relation $h(x)$ can be represented in a recursive form which is given by:

$$x_{n+1} = h(x_n), \quad (2.4)$$

where n is a non-negative integer that stands for discrete time instants and x_n is the n^{th} iteration of x_0 .

Another method of expressing the iterative property is through defining the composite function $h^n(x)$ such that

$$\begin{aligned} x_1 &= h(x_0), \\ x_2 &= h(x_1) = h(h(x_0)) = h^2(x_0), \\ &\vdots \\ x_n &= h(x_{n-1}) = h^n(x_0), \\ x_{n+1} &= h(x_n) = h^{n+1}(x_0). \end{aligned}$$

The set of all the iterations of a function $h(x)$ given an initial point x_0 is called the map of $h(x)$ or alternatively the orbit of x_0 . By iterating these relations, the discrete dynamics of the population that they model can be observed. In the former model, if the initial value of x (x_0) is greater than zero, the population will grow without bound. Although this might be correct for a certain range of populations, it may lose its applicability in other ranges. On the other hand, the latter model can have a finite limit where the population approaches an eventual limiting size, called steady state population. Reaching such a steady state with one fixed value, more than one value that follow a periodic exchange among them, or aperiodic behavior is controlled by the value of the system parameter λ . In the upcoming discussion, we are concerned with the logistic growth model.

2.1.2.2 Operands' Ranges

The logistic growth model could be given by the recursive formula (2.1), or by the bivariate function $f(x, \lambda)$, alternatively written as $f(x)|_\lambda$ or simply $f(x)$. To guarantee bounded output, the resulting value x_{n+1} or $f(x)$ should be included in the range $0 \leq x_n \leq 1$. This range is also owed to the definition of x as the ratio of existing population to the maximum possible population, and the application of the logistic model to concepts in biology, cryptography, traffic, finance and others related to probability theory. The critical point x_c of $f(x)$ is obtained by solving

$$f'(x_c) = 0 \rightarrow x_c = \frac{1}{2} \quad (2.5)$$

at which the value of the function is given by:

$$f(x_c) = \frac{\lambda}{4}. \quad (2.6)$$

Thus, $0 \leq \frac{\lambda}{4} \leq 1$ and the system parameter λ has the range $0 \leq \lambda \leq 4$. The map curve with the largest possible maximum value in the boundedness range is illustrated in Figure 2.1(a) that shows the graph of $f(x)$ at $\lambda = 4$ or maximum chaotic behavior. Briefly, for the logistic map, $x \in [0, 1]$ and $\lambda \in [0, 4]$. The map has an extremum at $x_c = \frac{1}{2}$ with a corresponding value of the function $f(x_c) = \frac{\lambda}{4}$ that is upper-bounded by 1.

2.1.2.3 Main Definitions: Fixed Points and Periodic Points

For the logistic growth model, fixed points are defined as the values of $x = x^*$ for which $f(x^*) = x^*$. For example, consider the map $f(x)|_{\lambda=1.5} = 1.5x(1-x)$, the map converges to a fixed point at $x^* = \frac{1}{3} = 0.333$ as shown in Table 2.2, where $f(0.333) = 0.333$. The same value results for any initial point $x_0 \in (0, 1)$. It should be noted that all the values are recorded in Table 2.2 to four decimal places. The eventual behavior of a map when it settles at a fixed value, for any initial point, is called an attractor. Let $x = x^*$ be a fixed point of the logistic map $f(x)$ at a specific value for λ , the stability conditions of the point x^* can be classified as follows.

- If $|f'(x^*)| < 1$, $x = x^*$ is a stable fixed point and it is called a sink (node).
- If $|f'(x^*)| > 1$, $x = x^*$ is a saddle fixed point and it is called a source.

The term $f'(x^*)$ is called the multiplier, its role in determining the stability of a fixed point can be explained using the limit definition of the first derivative and the forward difference formula. If a fixed point $x^* = f(x^*)$ is perturbed by a small quantity δx_{n-1} , the perturbation δx_n at the next iteration is given by:

$$\delta x_n = f(x^* + \delta x_{n-1}) - x^* = f'(x^*)\delta x_{n-1} + \mathcal{O}(\delta x_{n-1}). \quad (2.7)$$

Starting with an infinitesimally small δx_0 , the perturbation after n iterations is thus $\delta x_n \approx (\mu)^n \delta x_0$, where μ , the multiplier of the fixed point, is given by the map derivative at $x = x^*$, $f'(x^*)$. A fixed point is thus stable (unstable) when the absolute value of its multiplier is smaller (greater) than unity as previously proposed. The case $|f'(x^*)| = 1$ will be explained in the next subsection.

The points at which $f^k(x_p) = x_p$ are called periodic points of period- k , where k is the smallest positive integer satisfying the equation. For example, consider the map $f(x)|_{\lambda=3.3} = 3.3x(1-x)$, the map approaches a period-2 orbit such that $f^2(x) = x$ as shown in Table 2.2, where $f(0.8236) = 0.4794$ and $f(0.4794) = 0.8236$. Thus, $f^2(0.8236) = 0.8236$ and $f^2(0.4794) = 0.4794$. The same values result for any initial point $x_0 \in (0, 1)$. In this example, due to the different value of λ in the map, we do not have a stable fixed point (sink). Instead, a period-2 sink exists. The stability of periodic points depends on the absolute value of the first derivative of the composite map $|(f^k)'(x)|$ in a similar manner to how the stability of the fixed points depends on the first derivative of the map itself. Other values for periodic points are shown in Table 2.2.

2.1.2.4 Bifurcation Diagram

As previously defined, successive values of x evaluated using the recurrence (2.1) are called the orbit of x . This can also be viewed as applying the function $f(x) = \lambda x(1-x)$ and its composite functions $f^k(x)$ on x . The value at which the recurrence settles, alternatively

Table 2.2: Orbits of the logistic map $f(x)$ at different values for λ

n	$f(x) _{\lambda=1.5}$ (fixed-point)	$f(x) _{\lambda=3.3}$ (period-2)	$f(x) _{\lambda=3.83}$ (period-3)	$f(x) _{\lambda=3.5}$ (period-4)
0	0.5000	0.5000	0.5000	0.5000
1	0.3750	0.8250	0.9575	0.8750
2	0.3516	0.4764	0.1559	0.3828
3	0.3419	0.8232	0.5039	0.8269
4	0.3375	0.4804	0.9574	0.5009
5	0.3354	0.8237	0.1561	0.8750
6	0.3344	0.4792	0.5044	0.3828
7	0.3338	0.8236	0.9574	0.8269
8	0.3336	0.4795	0.1561	0.5009
9	0.3335	0.8236	0.5046	0.8750
10	0.3334	0.4794	0.9574	0.3828
11	0.3334	0.8236	0.1561	0.8269
12	0.3333	0.4794	0.5046	0.5009
13	0.3333	0.8236	0.9574	0.8750
14	0.3333	0.4794	0.1561	0.3828
\vdots	\vdots	\vdots	\vdots	\vdots

called the limit of the sequence or its steady state given by $\lim_{n \rightarrow \infty} f^n(x)$, is plotted for each value of λ as shown in Fig. 2.1(a). The sudden appearance of a qualitatively different solution for a system as some parameter is varied is called bifurcation. This appears in the form of fixed point, followed by a period doubling, quadrupling, etc., that accompanies the onset of chaos in what is called “Bifurcation diagram”. From the bifurcation diagrams of logistic and tent maps shown in Fig. 2.1(a) and (b) respectively, it is clear that they exhibit period doubling as a route to chaos that are characterized by certain numbers that do not depend on the nature of the map. For example, the ratio of the spacings between consecutive values of the system parameter at the bifurcation points approaches the universal “Feigenbaum” constant given by [37]

$$\delta = \lim_{k \rightarrow \infty} \frac{\lambda_{k+1} - \lambda_k}{\lambda_{k+2} - \lambda_{k+1}} = 4.6692\dots \quad (2.8)$$

Figure 2.1(a) shows the bifurcation diagram of logistic map, where different ranges for the system parameter λ cause the logistic map to exhibit different phases of behavior. At first, a steady state (sink) at the fixed point $x = 0$. Then, the non-trivial solution begins to appear once λ exceeds 1, where we may metaphorically call the point $\lambda = 1$ the first bifurcation point. The first non-trivial bifurcation point, alternatively called the second bifurcation point $\lambda = 3$ at which period-2 orbits begin to appear, followed by periodic orbits with multi-periods and so on till finally a chaotic behavior, where values cover the complete range $\in [0, 1]$ at $\lambda = 4$. This bifurcation effect appears gradually as λ increases.

Solving the equation $|f'(x^*)| = 1$ and getting the corresponding value(s) for λ determines the points at which bifurcation firstly occurs. Solving $|(f^k)'(x_p)| = 1$ and getting the corresponding value(s) for λ determines the points at which higher bifurcations occurs. These points are numbered as the first, second, third, ... bifurcation points and so on in

an ascending order with the corresponding values of λ . A detailed explanation of the resulting curve and the change in behavior in different ranges for λ can be carried out on the basis of the following discussion.

2.1.2.4.1 Fixed and Periodic Points and Their Stability The fixed points of the conventional discrete logistic map are given by

$$\lambda x^*(1 - x^*) = x^*. \quad (2.9)$$

which yields the two solutions

$$x_1^* = 0 \text{ and } x_2^* = 1 - \frac{1}{\lambda}. \quad (2.10)$$

The first derivative of $f(x)$ is given by

$$f'(x) = \lambda(1 - 2x), \quad (2.11)$$

in order to study the bifurcation points, we solve $|f'(x^*)| = 1$. For $x_1^* = 0$,

$$f'(x_1^*) = \lambda \rightarrow \lambda_{b1} = 1. \quad (2.12)$$

Therefore $x_1^* = 0$ is a sink for $\lambda < 1$. Hence, the trivial solution $x = 0$ prevails over the range $0 \leq \lambda < 1$. For $x_2^* = 1 - \frac{1}{\lambda}$,

$$f'(x^*) = 2 - \lambda \rightarrow \lambda_{b1} = 1, \lambda_{b2} = 3. \quad (2.13)$$

Therefore both $\lambda = 1$ and $\lambda = 3$ are bifurcation points. Just after $\lambda = 1$, the output reaches a steady state with only one fixed point at each value for λ . Moreover, at $\lambda = 3$, the solution is $f(x) = 1 - \frac{1}{3} = \frac{2}{3}$. Just after $\lambda = 3$, the output bifurcates to a period-2 solution according to the following analysis.

The next step is solving $f^2(x) = x$ and equating the absolute value of the first derivative of $f^2(x)$, $|(f^2)'(x)|$, at the resulting point to 1. We get the period-2 orbit which is stable for $3 < \lambda < 1 + \sqrt{6}$, and the next bifurcation point is $\lambda = 1 + \sqrt{6}$. The next periodic points and bifurcation points can be obtained following the same procedure.

2.1.2.4.2 Different Phases on the Diagram The remarkable indication of having a period-3 orbit is that it implies the existence of periodic orbits of all other periods. In addition, it implies sensitivity to initial data, i.e., the presence of chaos according to Sharkovskii's Theorem [7]. Observations on the change in behavior in different ranges for λ can be expanded as follows [111].

- In the range $0 \leq \lambda \leq 1$, the trivial solution prevails.
- In the range $1 \leq \lambda \leq 3$, a non-trivial fixed solution that depends on the value of the parameter λ starts to appear.
- In the range $3 \leq \lambda \leq 1 + \sqrt{6}$, the recurrence converges to a period-2 orbit.
- In the range $3.44949 \leq \lambda \leq 3.54409$ (approximately), the recurrence converges to a period-4 orbit.

- In the range $3.54409 \leq \lambda \leq 3.56995$ (approximately), the recurrence converges to a period-8 orbit.
- Starting at the value $\lambda = 3.56995$, the recurrence starts to exhibit chaotic behavior and sensitive dependence on initial conditions.
- In this range, there are still certain isolated ranges of λ that show non-chaotic behavior; these are sometimes called “islands of stability”. For instance, beginning at $1 + \sqrt{8}$ (approximately 3.82843), the recurrence converges to a period-3 orbit. For slightly higher values of λ , period-6 then period-12 exist and so on.
- Beyond the studied range, i.e., when $\lambda > 4$, the recurrence diverges for almost all initial values.

2.1.2.4.3 Key-points of the Bifurcation Diagram The key-points of the bifurcation diagram for the conventional 1D logistic map can be summarized as follows.

- The first bifurcation point $\lambda_{b1} = 1$. This is the point at which the first non-zero value for x appears.
- The second bifurcation point $\lambda_{b2} = 3$.
- The solution at λ_{b2} that equals to $x_{\lambda_{b2}} = 2/3$.
- The maximum value that the solution could reach $x_{max} = 1$.

2.1.3 Conventional Tent Map $x_{n+1} = \mu \min(x_n, 1 - x_n)$

The tent (or triangular) map is a piecewise linear chaotic map whose conventional form is given by either of the following forms [7]:

$$x_{n+1} = \mu \min(x_n, 1 - x_n), \quad \mu \in R^+, x_n \in [0, 1] \quad (2.14a)$$

$$x_{n+1} = \begin{cases} \mu x_n & x \leq x_k \\ \mu(1 - x_n) & x_k < x \end{cases}, \quad (2.14b)$$

where x_k is the point of intersection of the two straight lines that equals 0.5 as shown in Fig. 2.1(b). It could be also be regarded as

$$f(x) = \mu \min(x, 1 - x), \quad \mu \in R^+, x \in [0, 1] \quad (2.15)$$

Piecewise linear maps are perhaps the simplest kind of chaotic maps from the viewpoint of realization. Specifically, analog and digital realizations for the tent map have been proposed as in [29, 50]. Applications of the tent map include: cryptography [44, 114, 115], communication systems: channel coding and error correction [17], radar imaging [40], financial modeling [52] and others.

2.1.3.1 Operands' Ranges

The intersection point x_k is obtained by equating the left and right curves to get

$$x_k = \frac{1}{2}, \quad (2.16)$$

at which the value of the function is given by:

$$f(x_k) = \frac{\mu}{2}. \quad (2.17)$$

The map has an extrema at $x_k = \frac{1}{2}$ with a corresponding value of the function $f(x_k) = \frac{\mu}{2}$ that is upper-bounded by 1. Thus, for the tent map, $x \in [0, 1]$ and $\mu \in [0, 2]$.

2.1.3.2 Bifurcation Diagram

As previously mentioned, the tent map and the logistic map are conjugate [7] and thus their behaviors are identical under the effect of successive iterations. Depending on the value of the system parameter μ , the tent map demonstrates different dynamics that range from fixed and periodic predictable behaviors to chaotic behavior as shown in Fig. 2.1(b). The fixed points can be obtained as follows. Similarly to the analysis carried out for the logistic map:

$$x_1^* = 0 \text{ and } x_2^* = \mu/(1 + \mu), \quad (2.18)$$

$$f'(x^*) = \begin{cases} -\mu & x \leq x_k \\ +\mu & x_k < x \end{cases}, \quad (2.19)$$

$$|f'(x^*)| = 1 \rightarrow \mu_b = 1, \quad (2.20a)$$

$$x_b = 0. \quad (2.20b)$$

For $x_1^* = 0$, $f'(x^*) = \mu$, therefore $x_1^* = 0$ is a sink for $\mu < 1$. Hence, the trivial solution $x = 0$ prevails over the range $0 \leq \mu < 1$. Maximum chaotic behavior is obtained at $\mu = 2$.

2.1.4 Maximum Lyapunov Exponent

Since chaos also represents rapid divergence of nearby points, a quantity that measures the rate of this divergence would be quite useful. For a discrete map f of the real line R , the Lyapunov number $L(x_0)$ of the orbit x_0, x_1, x_2, \dots is defined to be

$$L(x_0) = \lim_{n \rightarrow \infty} (|f'(x_0)| \dots |f'(x_{n-1})|)^{1/n}, \quad (2.21)$$

which represents an average expanding rate. The maximum Lyapunov exponent (MLE) is given by

$$\text{MLE} = \ln L(x_0). \quad (2.22)$$

Hence,

$$\text{MLE} = \lim_{n \rightarrow \infty} \left(\frac{1}{n} \sum_{i=0}^{n-1} \ln |f'(x_i)| \right), \quad (2.23)$$

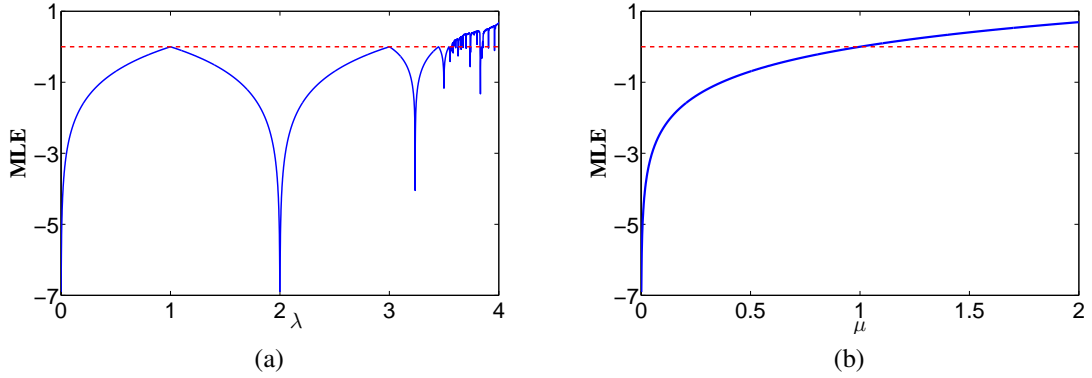


Figure 2.2: Maximum Lyapunov exponent versus the system parameter (a) Logistic map and (b) Tent map

where \ln is the natural logarithm and MLE is an indication whether the system exhibits chaotic behavior or not. Theoretically, a positive value for MLE proves chaotic behavior. Methods of MLE calculation in numerical simulations are further explained in Chapter 5, subsection 5.2.4. Figure 2.2(a) shows the maximum Lyapunov exponent MLE for the logistic map versus the system parameter λ , where positive values start to appear after the point of onset of chaos that occurs approximately at $\lambda = 3.56995$ accompanied with zero crossings and some negative values in between. The maximum chaotic behavior is recorded at $\lambda = 4$ where MLE equals $\ln 2$. For tent map, MLE is shown in Fig.2.2(b) versus the system parameter μ . The maximum chaotic behavior is recorded at $\mu = 2$ where MLE equals $\ln 2$ too. This value could be proved for tent map and by conjugacy for logistic map [7].

Proof. The maximum Lyapunov exponent for the chaotic tent map $f(x) = 2 \min(x, 1 - x)$ equals $\ln 2$.

In this case, $f'(x)$ equals either 2 or -2 . Thus, $|f'(x_i)| = 2$ which is constant value. From (2.23), we get $\text{MLE} = \lim_{n \rightarrow \infty} (\frac{1}{n} \ln 2) = \ln 2$. \square

Thus, according to this proof the plot of MLE for tent map is the same as \ln function as shown in Fig. 2.2(b) as if $\ln \mu$ is plotted versus μ .

2.2 Negative Operands: How and Why?

Recalling the equations of the conventional logistic (2.3) and tent maps (2.15), there has been a constraint on the value of λ to be positive real. Moreover, this constraint has been applied in the calculation of the bifurcation points, where at some step it was claimed that the solution of $|\lambda| = 1$ (or $|\mu| = 1$) is λ or μ equals positive one. Yet, if this constraint is removed, we could proceed with the negative solution and reach surprising results.

2.2.1 The Negative Parameter Case First Visited

Conducting a simple computer simulation that calculates the steady state solutions sweeping a range of the system parameter, we get the bifurcation diagrams shown in Fig. 2.3 for

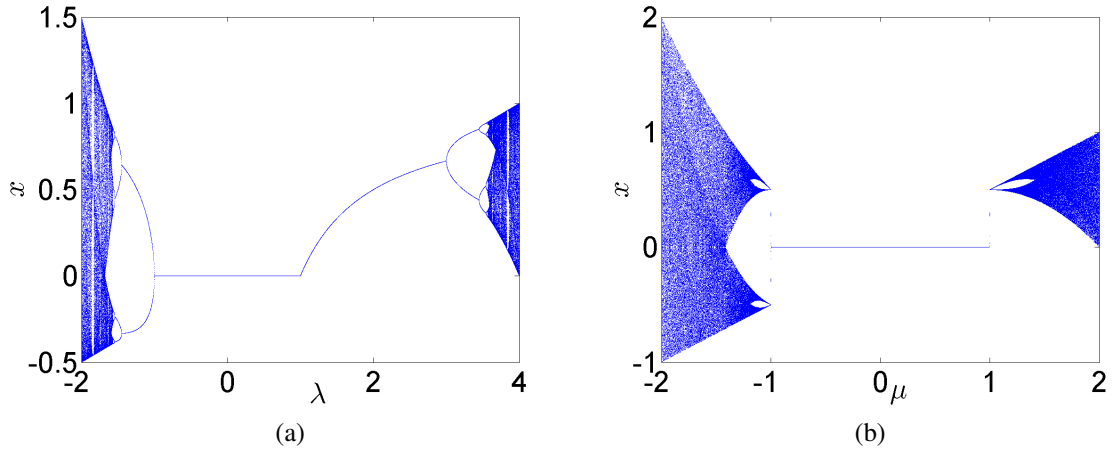


Figure 2.3: Existence of bifurcation diagram for negative system parameter (a) Logistic map and (b) Tent map

both logistic and tent maps. The diagrams are obtained through allowing negative values for the system parameter and exploring the behavior in this region. The system response in the negative side constructs an arrangement that could be described as a new bifurcation diagram. It is clear that they exhibit period doubling as a route to chaos similar to their positive parameter corresponding maps. This is an indication that Feigenbaum universal constant does not only occur for maps that have a global maximum, but also those that have a global minimum. The diagram of the proposed map with negative system parameter exhibits different values for the key-points and the range of outputs. The first difference is that the first bifurcation represents a change from trivial solution to period-2 without a non-trivial fixed-point unlike the case of positive system parameter. Another important difference is that the output range has a minimum negative value instead of zero, and a maximum positive value greater than one. For the logistic map $[x_{min}, x_{max}] = [-1/2, 3/2]$, while for the tent map it is $[x_{min}, x_{max}] = [-1, 2]$. A detailed analysis of this new region (negative system parameter) in both maps and methods of getting the solutions and predicting the behavior mathematically is presented in the next Chapter 3 and discussed in [100]. Negative and alternating sign maps are presented which have not been mathematically covered before in the literature to the best of our knowledge in addition to a study of their main properties. Negative parameter range has been quickly tackled as in [8, 15] without a detailed mathematical analysis and it was claimed to have no real world applications. A detailed analysis of three different parameterized versions for each of the proposed maps is also provided. But first, we devote the rest of this section to present a quick, but not exhaustive, review of related real world applications. The merits of the proposed maps for both pseudo-random number generation and modeling applications are discussed.

2.2.2 Enhancement of Randomness Measures

The wider output range and the asymmetry recognized in the resulting bifurcation diagrams of these maps could also be added to their advantages as they imply more unpredictability, in addition to the earlier onset of chaos for negative parameter compared to the case of positive parameter. We proceed in proving the enhancement of the dynamical properties

induced by the map with negative parameter in Chapter 5, specifically for limited precision implementations. Yet, the added value of the proposed map could be viewed from another aspect where their presence means the capability of operating the same hardware to get completely different responses. Many possibilities of “chaos” could be generated through varying the sign and magnitude of the system parameter too not only changing the seed. It is worth mentioning that the proposed map provides a wider range of seeds, i.e., initial points that enhance the system security for chaos-based cryptography and communication where the possibility of hacking the system through a pattern of seeds or randomly selected ones decreases, increasing the system’s robustness. Moreover, further generalizations proposed in Chapter 3 and Chapter 4 introduce new parameters that increase the degrees of freedom of the map and allow the construction of more efficient encryption keys with extra parameters. In conclusion, proposed generalizations on existing maps enhance their unpredictability and increase their reliability in secure communication and encryption.

2.2.3 Wider Domain of Modeling With Improved Efficiency

The output ranges of the proposed maps allow more flexibility in chaotic modeling of finance [10, 52], traffic, weather forecasting [86, 108], and many other fields. For instance, measures of relative humidity can exceed 100% in case of “supersaturation” such as nuclear physics experiments in Wilson cloud chamber [43]. Further enhancements are added in Chapter 3 and Chapter 4 that allow more control of the properties of each map and its bifurcation diagram. Various experiments have shown that some phenomena exhibit dynamical behaviors which could only be modeled with generalized maps with modulated parameters as conventional maps are not sufficient to model [24, 26]. We believe that there is still a gap which needs to be filled between areas of research in which some biologists, physicists, and economists propose new models and the field of non-linear dynamics and chaos in which researchers could come up with chaotic systems suitable for predicting the behavior of these models. Examples of models that could employ the proposed maps are discussed below.

Axioms of Kolmogorov’s probability theory [62] have the rule $0 \leq P(A) \leq 1$ since $P(A)$ describes the probability of occurrence of event A , or the probability of the outcome of an experiment, which must be a positive real number. However, these axioms do not provide the complete picture of stochastic reality as emphasized by Kolmogorov himself. Although his model, the measure-theoretic model with probability space or sample space of probability one, is a successful one for many domains of sciences, no model can be ever described as completely valid for all domains. The widely-spread Kolmogorov’s model has neglected the context of carrying out experiments. Moreover, the setup of many experiments has been prepared and observed with a prior hypothesis that probabilities are only confined to the range $[0, 1]$. Possibilities to extend the probability theory to describe numerous physical models with negative probabilities, and even probabilities more than one, have been mathematically discussed based on frequency probability model of Von Mesis, as in [60]. The frequency probability model starts with random sequences, collectives rather than directly with probabilities where probabilities are defined as limits of relative frequencies for results of observations.

Negative probabilities have been used in solving several problems [23, 45, 51], and paradoxes, e.g., Einstein-Podolsky-Rosen paradox [103]. These out of the expected range

probabilities are allowed by quasi-probability distributions that may apply to unobservable events or conditional probabilities. The idea of negative probabilities first arised in physics and particularly in quantum mechanics based on ideas of E. Wigner [113], P. Dirac [27,80], and R. Feynman [39]. Later on, many physicists and scientists [38,41,80,97] have argued for the necessity of extended probabilities in quantum theories and that its appearance is one of the main differences between classical and quantum theories. For instance, the probabilities of existence of positive and negative-energy photons, or the emissions and absorptions of photons, equal to 2 and -2 respectively [28,80]. The strong relation between these discoveries and the chaos theory could be understood better through reviewing further details on chaotic properties of electric transport in a semiconductor, those could be found in [101]. In this book, the applications of bifurcation theory and Hopf bifurcation, that was first realized in biochemical systems, in semi-classical and quantum physics are explained. Instabilities due to nonlinear processes of generation and recombination of carriers (electrons or holes) in semiconductors and their ionization across the bandgap or from localized levels are discussed. Chaotic behavior of semiconductors, lasers, current filaments, and others are presented.

Unsurprisingly, such extended probabilities exhibited by several physical phenomena and environmental processes found their way to commercial manufacturing. The previously raised concept of “supersaturation” relevant to meteorological and atmospheric studies is defined as a state of a solution in which the dissolved material is more than the amount that could be dissolved by the solvent under normal conditions. The higher energy state of supersaturated solutions lead to crystallization under specific circumstances. Several examples on commercial products, especially foodstuff, utilize this concept such as: honey, rock candy, sodas, and seltzer water.

Negative probabilities have more recently been applied to mathematical finance. In quantitative finance, most probabilities are not real probabilities but theoretical ones called pseudo-probabilities. The concept of risk-neutral or pseudo-probabilities is a popular concept in finance which has been numerously applied as in [30,53]. It has been shown how negative probabilities can be applied to financial option pricing. For example, calculating the probability of a price going up or down could be simplified by allowing such pseudo-probabilities to be negative in certain cases as pointed out by Haug [48]. A generalized version of the well-known Cox, Ross, and Rubinstein [21] binomial tree (CRR tree) has been considered. The tree is often used to price a variety of derivatives instruments where different probabilities could be yielded which might lie outside the interval $[0, 1]$. For example, one can get a risk-neutral up probability of 1.3689 and a down probability of -0.3689 . A rigorous mathematical definition of negative probabilities in finance and their properties has been recently derived by Burgin and Meissner [16]. The authors provide several situations in which negative probabilities occurred in finance, as well as negative interest rates: over-generated electricity in Norway that resulted in its price going negative a few hours during the night, the lender paying the bank interest rate in addition to the money in the 1970s in Switzerland, ‘repos’, i.e., repurchase agreements traded at negative interest rates which took place in Japan and USA in 2003, and negative nominal interest rates that occurred in the worldwide 2008/2009 financial crisis.

In conclusion, the new behavior of 1D maps allowing negative system parameter could permit their usage as discretized models for many applications even those that have been confined to the continuous domain. Combining these models along with considering quasi-

probability distributions which have been used for solving many problems as previously mentioned could be such a breakthrough.

2.3 Generalized Chaotic Maps

Conventional chaotic systems, especially discrete 1D maps, are highly rich in information and indications that constitute a large portion of the basic study of chaos theory. However, modeling dynamics in natural phenomena and stochastic processes, as well as demand on pseudo-random number generation for multiple purposes, requires novel chaotic systems. Consequently, generalized maps have been proposed in previous researches [32,68,77,109]. Although the starting equation is mostly generic, the subsequent analysis forces the parameters to some fixed values. For instance, the parameters of generalized chaotic function proposed in [77] have been adapted to fit a specific encryption algorithm. The proposed generalized map is given by:

$$g(\lambda, \alpha, \beta, x) = \lambda x(\alpha - x)^\beta, \quad \lambda, \alpha, \beta \in R^+ \quad (2.24)$$

Matthews [77] solved the proposed map for the critical value given by

$$g'(x_c) = 0 \rightarrow x_c = \frac{\alpha}{\beta + 1}, \quad (2.25)$$

and the first non-trivial fixed point that equals

$$x^* = g(x^*) \rightarrow x^* = \alpha - \frac{1}{\lambda^{\frac{1}{\beta}}}. \quad (2.26)$$

To guarantee bounded closed set responses for the map in the interval $x \in [0, \alpha]$, there is an upper bound on the value of λ . However, the author did not proceed in the general analysis deeper than this. Instead, the parameter α has been set to 1 to avoid extremely low responses, while the parameter λ has been set to its maximum corresponding value that equals $(\beta + 1)(1 + 1/\beta)^\beta$. Thus, two parameters only remained (x and β) and the map has not been mathematically analyzed in a generic way. The different chaotic properties of the proposed map had not been illustrated.

Another type of the listed research work depends on rather complicated equations that are classified as 2D maps. An example is the map presented in [32] which is given by:

$$x_{n+1} = ax_n(1 - x_n), \quad y_{n+1} = (b + cx_n)y_n(1 - y_n). \quad (2.27)$$

The parameters' ranges allowed by the map and their classification into fixed, periodic solution, or chaotic behavior have been detailed. The resulting bifurcation diagrams exhibit either different shape in the same range of output response of the conventional map $x \in [0, 1]$, or even different ranges. One of the diagrams for $0 \leq a \leq 4$, $b = -2$, and $c = 2$ generates output responses in the range $x \in [-0.5, 1.5]$.

Other generalized logistic, tent, and sine maps have been recently investigated in [91–93] where effects of generalized parameters on the behavior of the logistic map have been thoroughly studied. In [91], three different generalizations of the logistic map with arbitrary powers α and β have been proposed which are given by:

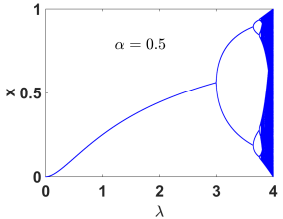
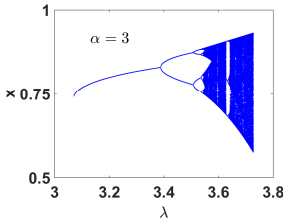
$$x_{n+1} = \lambda x_n^\alpha (1 - x_n^\beta), \quad (2.28)$$

where (α, β) can equal one of three cases (α, α) , $(\alpha, 1)$, or $(1, \alpha)$. The main factors, aspects, and properties of the proposed maps have been compared. For each map, the fixed points and their critical values, ranges, the bifurcation diagrams with respect to the two parameters α and λ , the effect of the function iterations, the iterated outputs, the cobweb plot, and the time evolution of the MLE have been discussed. The generic case is given by:

$$x_{n+1} = \lambda x_n^\alpha (1 - x_n^\alpha), \quad \lambda, \alpha \in R^+ \quad (2.29)$$

with properties which provide extra degrees of freedom than the conventional logistic map. Its properties are summarized in Table 2.3.

Table 2.3: The properties of generalized logistic map with arbitrary power (2.29)

Property	Value
Critical Point	$f'(x_c) = 0 \rightarrow x_c = \sqrt[\alpha]{0.5}$
Fixed Points	$\lambda(x^*)^{2\alpha} - \lambda(x^*)^\alpha + x^* = 0$
Bifurcation Points	$\lambda_b = \frac{1}{\alpha(x^*)^{\alpha-1}(1-2(x^*)^\alpha)}$
Bifurcation Diagram vs. λ ($\alpha < 1$)	
Bifurcation Diagram vs. λ ($\alpha > 1$)	

One of the other special cases given by

$$x_{n+1} = \lambda x_n^\alpha (1 - x_n), \quad \lambda, \alpha \in R^+ \quad (2.30)$$

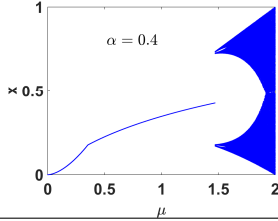
is similar to one of the modified maps presented in [68] but Levinsohn has been confining α to positive integers greater than one only. The map was suggested for use in analyzing a different ecological system. The analyses, although implying different steps, are consistent in their results for parameters' ranges. However, in [91] further properties have been studied such as the bifurcation diagram versus the new parameter α and allowing fractional values of this parameter.

A similar generalization for the tent map has been introduced in [92], which is given by either of the following forms.

$$x_{n+1} = \min(\mu x_n^\alpha, \mu(1 - x_n^\alpha)), \quad \mu, \alpha \in R^+ \quad (2.31a)$$

$$x_{n+1} = \begin{cases} \mu x_n^\alpha & x \leq x_k \\ \mu(1 - x_n^\alpha) & x_k < x \end{cases}, \quad (2.31b)$$

Table 2.4: The properties of generalized tent map with arbitrary power (2.31)

Property	Value
Intersection Point	$x_k = (0.5)^{\frac{1}{\alpha}}$
Fixed Points	$x_1^* = 0$, $x_2^* = (1/\mu)^{\frac{1}{\alpha-1}}$, and $x_3^* + \mu x_3^{*\alpha} - \mu = 0$
Bifurcation Points	$(\mu_b, x_b) = ((\frac{1}{\alpha})(\alpha + 1))^{1-1/\alpha}, (\frac{1}{\alpha+1})^{1/\alpha}$
Bifurcation Diagram vs. μ	

where $x_k = (0.5)^{\frac{1}{\alpha}}$. The properties of the map proposed in [92] are summarized in Table 2.4.

Another generalization for the tent map has been proposed in [92] too that employs other parameters in a different sense which is scaling instead of powering. A generalized version with two extra parameters has been introduced in [92] which is given by

$$x_{n+1} = \begin{cases} \mu x_n & x \leq x_k \\ \mu(a - bx_n) & x_k < x \end{cases}, \quad (2.32)$$

where $x_k = \frac{a}{1+b}$. The properties of the map proposed in [92] are summarized in Table 2.5. See also [98], [84], and [49] for other generalizations and/or modifications on the tent map.

Table 2.5: The properties of the generalized scaled tent map (2.32)

Property	Value
Intersection Point	$x_k = \frac{a}{1+b}$
Fixed Points	$x_1^* = 0$ and $x_2^* = a\mu/(1 + b\mu)$
Bifurcation Points	$(\mu_{b1}, x_{b1}) = (1, 0)$ $(\mu_{b2}, x_{b2}) = (1/b, a/(2b))$ for $b < 1$ only
Parameters' Ranges	$(\mu_{max}, x_{max}) = (1 + 1/b, a/b)$

Further generalizations have been utilized in [93] to build an image encryption scheme, such as the generalized sine map given by

$$x_{n+1} = f(x_n, r, \alpha, \beta, \gamma) = r \sin^\gamma(\alpha \pi x_n^\beta) \quad (2.33)$$

The paper summarizes the properties of the proposed generalized maps and utilizes the added degrees of freedom through the extra parameters in constructing an effective and sensitive encryption key.

2.4 Digital Chaotic Maps

The remarkable importance of chaotic iterated maps in both modeling and information processing in many fields explains the need for their hardware analog and digital realizations,

e.g., [9, 22, 72, 94–96, 117, 118]. Since we are concerned with digital implementations, we present a review of several studies [6, 13, 19, 69, 89] that have been conducted on the effect of finite-precision on the properties of chaotic systems and are much related to our proposed work. The problem of simulating or implementing digital chaos is composed of two parts: finite time and finite precision. This implementation is done probably in some sort of digital calculations on computers. In this case, sentences like steady state or the limit as the number of discrete time steps approaches infinity no longer carry the same meaning. Practically, we can only record the behavior of a limited number of time samples: hundreds, thousands, or even millions, but there is no “infinite” time. The same applies to precision, there is nothing practical that is equivalent to infinite precision. It is either multiple or fixed precision as explained in the next subsection.

2.4.1 Multiple Precision Versus Fixed Precision

Multiple precision or arbitrary precision arithmetic means performing calculations on numbers whose length is limited only by the available memory of the system. This allows the user to choose the precision of each calculation. On the other hand, fixed-precision arithmetic, which is significantly faster, is the arithmetic upon which most arithmetic logic unit (ALU) hardware, e.g., FPGAs is based. Such hardware buses typically offer between 8 and 64 bits of precision. The IEEE compliant software implementations and hardware designs follow a fixed-precision system. In some applications, accumulated round-off errors that result due to using fixed-precision arithmetic could be catastrophic, and the use of arbitrary-precision arithmetic becomes useful. For example, floating-point programs dedicated to electronic control units of aircrafts, astronomical applications, or in nuclear reactors. Multiple precision calculators can be subdivided into three types. First, languages with built-in support for multiple precision in which arbitrary precision is a part of the standard library of the language. Second, languages that have the capability of linking with an external library that performs arbitrary precision calculations. Examples on arbitrary precision libraries include: CRlibm [83], MPFR [46], and SOLLYA [18]. The third type of multiple precision calculators is stand-alone application software tools; in particular most computer algebra systems such as Mathematica and Maple. Other arbitrary precision tools may be based on decimal arithmetic such as: basic calculator (bc) [85]. The rest of our discussion focuses on computationally efficient fixed-precision implementations of chaos, specifically 1D maps. In the next section, previous work on analyzing the effect of this finitude on the dynamical properties of chaotic generators is reviewed.

2.4.2 Previous Work

R. M. Corless in [19] discusses numerical simulations of chaotic dynamical systems and how much they should be trusted. He suggests that the computed orbit and the accompanied value for maximum Lyapunov exponent (MLE) obtained through finite time and finite precision simulations could be falsely interpreted as chaotic, or non-chaotic. This could be owed to the ill-conditioned nature of chaotic dynamical systems such that small errors in initial conditions or involved operations are exponentially amplified with time. Consequently, no measure exists of how much the actual response obtained shall deviate from the expected behavior, and this deviation cannot be tracked as time progresses.

Many of these claims, even those not aided by sufficient simulations, could be noticed to be true from our simulations in Chapter 5 and for different maps. To overcome the possibility of misinterpretation, Corless recommends using backward error analysis techniques to uncover the real properties of the digitally implemented chaos. For this purpose, the problem is sub-divided into levels as follows: actual physical system, continuous time mathematical model (this level is skipped for discrete maps), discretized model, and finally finite-precision results. The latter results are mostly surprising in how they deviate from the previous levels.

Li, Chen and Mou in [69] present a quite comprehensive study on degradation of digitized chaos and ways to reduce its negative effect in chaos-based digital implementations. Their work is mainly concentrated on 1D piece-wise linear chaotic maps (PWLCM) such as the asymmetric tent map. There are two aspects of regarding digitally implemented chaos, either redefining the equation in a digital form and confining outputs to the integer domain [13] or digitally implementing it in finite-precision. Our interest is in the latter aspect, especially fixed-point representations. The reason why pseudo-random number chaotic generators are suggested frequently without paying attention to the effect of finite precision could be owed to the β -shadowing lemma. This lemma ensures that there exists an exact chaotic orbit close to the pseudo-orbit with only a small error [14]. However, there is an argument with strong evidences that this lemma can be applied to digital chaos. Previous research efforts have been reported whether attempting to come up with a theoretical formulation of the problem and its consequences, or experimentally obtaining results and analyzing them in the aim of acquiring full understanding of the problem. In this experimental approach, several properties of chaotic systems are used as indicators that can be used to uncover security weakness hidden inside some digital chaotic ciphers. The effect of rounding direction has also been considered. Suggested ways to reduce dynamical degradation of digital PWLCM include: using higher finite precision, cascading multiple chaotic systems, and pseudo-random perturbation for either the system parameter or its iterated variable with different configurations. Developing a robust theoretical analysis of digital chaotic maps, specifying error bounds on different arithmetic functions employed in chaotic maps iterations calculations, and formulating the relationship between the digital versions of traditional chaotic properties (such as Lyapunov exponent) and the system parameters in a digital chaotic system still remain open topics in this interesting field.

Performance analysis of digital tent map has been presented in [6]. Its behavior as a pseudo-random number generator has been compared to that of linear feedback shift registers (LFSR). In this paper, software simulations of digital chaotic maps are criticized for being unsuitable for direct application to hardware FPGAs which imply specific assumptions. In our work in Chapter 5, we overcome this viewpoint of criticism. The map equation has been re-written to describe the map in fixed-point representation, and the allowed ranges of parameters have been proved. A judgment of the behavior of the map considers different figures of merit (FOMs): minimum period length, isolated sets of seed that should be avoided, and statistical characteristics of the generated sequences. Hardware complexity has also been considered.

In the field of chaos based communication and for practical considerations, a design guide of a computationally efficient pseudo-random number generator (PRNG) using the logistic map with finite precision is needed. For this purpose, the effect of finite precision

on the periodicity of a PRNG based on the logistic map has been explored in [89]. A comparison has been made with some conventional methods of generating pseudo-random numbers. The cyclic properties of the sequence generated by the logistic map have been discussed in terms of the number, delay, and period of the orbits at varying degrees of precision (3 to 23 bits). It has been shown that PRNG based on the logistic map performs exponentially worse than conventional PRNGs for such precisions in terms of pathological seeds and effective bit length. A finite precision period calculation (FPPC) approach has been introduced to provide a general method for analyzing the period lengths of recurrence relations implemented in finite precision. For 4-bit precision, the results of each seed have been calculated including the sequence up to the first duplicate, the length of the periodic cycle, how many numbers are generated before the cycle (delay), and the total length of the sequence before the generator starts to repeat terms. The maximum period obtained was 5 which is less than the ideal of 15. Moreover, many of the periods were even shorter which represents an extremely inefficient use of the available seed space. A generalized algorithm has been used to calculate these results for all precisions in the studied range using parallel computing, keeping memory requirements in mind. The algorithm employs truncation in a single-precision floating-point environment after converting the binary32 format to a denormalized binary fraction. Results have been demonstrated in multiple curves that show: total period length versus precision, frequency of occurrence of periods with different lengths, occurrence of each delay versus total cycle length, parameter space utilization, i.e., the total sequence lengths normalized to the maximum possible length versus precision, and effective precision, i.e., the number of bits utilized by the logistic map PRNG versus actual precision. A question arises here: is it correct to consider the effect of truncation only after the execution of the whole expression takes place, considering the subtraction followed by two multiplication operations in the logistic map equation as a single operation?

Moreover, several studies for the effect of limited precision on the properties of other digitally implemented maps have been conducted. For example, in [75], fully digital implementation of a 3rd order ODE-based chaotic oscillator with signum nonlinearity has been proposed and the effect of precision (bus width) on its properties has been studied. The system output has been investigated for periodicity through viewing the oscilloscope output of digital X-Y attractors. Maximum Lyapunov exponent has been calculated using 250,000 iterations for all precisions up to 64 bits. The threshold minimum precision required for chaos has been decided to be 12 bits (8 fractional bits), below which the output is periodic. Furthermore, this threshold yields a positive value for maximum Lyapunov exponent (MLE); indicating chaotic behavior. Similar analysis has been conducted on fully digital implementations of four different systems in the 3rd order jerk-equation based chaotic family using Euler approximation [74]. Experimental results have shown that the minimum fractional bits needed for chaotic behavior are 10, 8, 10, and 11 bits respectively for the four proposed systems. However, it is expected in advance that the system based on the 1D discrete logistic map needs a higher minimum threshold precision. The logistic map is governed by a much simpler relation and exhibits less dynamics. The response variation is expected to be slower varying the used precision.

2.5 Elementary Functions: The Power Operation

Some of the proposed generalizations of 1D discrete chaotic maps utilize other arithmetic operations rather than the basic addition/subtraction, and multiplication operations. One of these is the power operation ($z = x^y$) which has always been somewhat problematic in modern computation. For the proposed generalized maps, simulations are performed by MATLAB for their different behaviors. Software implementations such as MATLAB are supposed to execute floating-point arithmetic according to well-defined standards. Floating-point arithmetic in general provides fairly accurate results and rapid execution for many numerically intensive applications. Almost all modern computers and languages follow the IEEE 754 standard for floating point arithmetic in its current version released in 2008 that includes nearly all of the original IEEE 754-1985 standard for binary floating-point arithmetic and the IEEE 854-1987 standard for radix-independent floating-point arithmetic. Since then, there have been many studies on improving the efficiency and enhancing the performance of IEEE-compliant floating-point computations [34, 35]. Picking the power function among the elementary functions recommended by IEEE standard for floating-point arithmetic (IEEE Std 754-2008 [2]) represents a challenge due to the difficulty in introducing an unanimous mathematical definition for the various cases of this transcendental function. Computation-wise, returning correct results for non-special inputs is rather hard, because of numerous sensitivities based on values of both operands x and y . Moreover, it has a wide domain since both operands have a very wide interval of values $[-\infty, +\infty] \times [-\infty, +\infty]$. Among all the standard elementary functions, the power function is the only one with two arguments that causes difficulty, because of the problem of error magnification (sensitivity of z to small changes in either x or y - specially the base x). The other functions with two arguments are either various forms of power functions such as $\text{compound}(x, n)$, $\text{rootn}(x, n)$, $\text{pown}(x, n)$ and $\text{powr}(x, y)$ whose differences from pow will be discussed later on. The rest of the functions with two arguments which are $\text{atan2}(y, x)$ (and $\text{atan2Pi}(y, x)$) and $\text{hypot}(x, y)$ functions generally require only careful attention to the sign and finite/infinite value of the ratio y/x , as well as avoidance of underflow and overflow.

Ten is the base or radix that humans use in calculations in a decimal number system while a binary system with radix two is more related to computers. Interest in decimal arithmetic increased considerably in recent years especially decimal floating-point implementations. They have many advantages over binary floating-point in financial and commercial applications whether as software or hardware implementations. Decimal is used extensively in banking, billing, and other financial applications. Simple decimal fractions such as $1/10$, representing a tax amount or a sales discount, have an infinite number of bits in their binary representation and can not be accurately represented in a binary number system. In a large billing application, such an error may be up to \$5 million per year [36]. Using decimal floating-point could be a solution for some source of representational error, or inexactness. Several researches have been conducted on decimal floating-point units that perform the basic operations. The proposed designs have been validated and tested indicating their various figures of merit as in [33, 36]. Previous work on the evaluation of elementary functions on computers is to be reviewed. But first, in the next subsection, we review some of the definitions of IEEE standard for floating-point arithmetic that are also encountered later on in Chapter 6.

2.5.1 IEEE Standard Definitions

The IEEE standard for floating-point arithmetic 754-2008 [2] specifies formats and methods for binary and decimal floating-point arithmetic, in addition to handling the special values and exception conditions. The standard's floating-point numbers approximate the familiar field of real numbers R algebraically completed by defining the signed zero ± 0 and the adjunction of $\pm\infty$ and NaN as shown in Fig. 2.4. The standard defines both

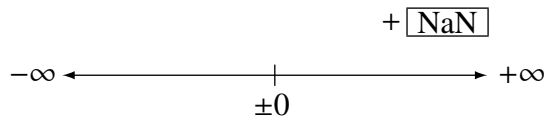


Figure 2.4: Algebraic completion of the real numbers according to IEEE 754-2008

basic formats (for computations) and interchange formats (to exchange data between different implementations). There are five basic formats: binary32, binary64, binary128, decimal64 and decimal128. For the floating-point simulations throughout the thesis, the default MATLAB binary64 or double precision representation is used. A number in this representation shown in Fig. 2.5 equals $(-1)^s \times 1.f \times 2^{e-1023}$, where s is the sign, f is the fractional significand with an implicit integer bit of value 1, and e is the biased exponent where the bias equals 1023. On the other hand, the special cases handled in Chapter 6 are usually defined irrespective of the precision for either binary or decimal implementations.

2.5.1.1 Special Values

The importance of the special values defined in IEEE Std. 754-2008 is that they conserve the closure property of the real field as discussed and shown in Fig. 2.4. They were proposed mainly to have well-defined results on encountering some computations which could be considered erroneous. Thus, on some systems, they might lead to aborting the execution unless a well-defined result is agreed upon for all systems to follow.

2.5.1.1.1 Signed Zero A zero result on a computer might not be an exact zero, but an infinitesimal value rounded to zero. The definition of the number zero associated with a sign resembles the mathematical concept of limits; whether this tiny value approaches zero from below(left) $x \rightarrow 0^-$ or above(right) $x \rightarrow 0^+$. Generally $+0 = -0$, except for some special cases such as: when they are operands of the logarithm function or divisors in a division operation with a finite dividend. This notation becomes significant for some scientific uses too, e.g., a “-0” measure of temperature on a Celsius scale means below freezing.

2.5.1.1.2 Infinities IEEE infinities are similar to their mathematical concept and are encountered in case of overflow as discussed in subsection 2.5.1.4.

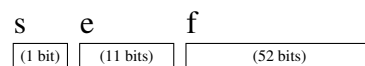


Figure 2.5: Double precision floating-point binary representation

2.5.1.1.3 NaN In floating-point computing, NaN, standing for not a number, is a numeric data type representing an undefined or unrepresentable value. For example, $0/0$ is undefined as a real number, thus represented by NaN; the square root of a negative number is imaginary which is not representable as a real floating-point number, thus represented by NaN (other examples are discussed in Chapter 6 in section 6.3). NaNs may be used to represent missing values in computations in order to obtain a mathematically closed system under the defined arithmetic operations. The standard states that two different kinds of NaN, signaling NaN (SNaN) and quiet NaN (QNaN), should be supported in all floating-point operations. Quiet NaNs either mean errors resulting from invalid operations, or they mean invalid or unavailable data and results. Signaling NaNs provide the representation for uninitialized variables and values that are not in the scope of the standard such as complex-affine infinities. The problem with forcing a result into this special value is that a NaN result might cause exceptions in the consequent operations, e.g., an “order-comparison with NaN” exception. This leads to having unordered number system by adding a fourth case to the trichotomy (three cases of comparison): less than, greater than, and equal.

2.5.1.2 Operations

The standard proposes definitions for the correct results of the basic operations (addition/subtraction, multiplication, division and square root, in addition to fused multiply-add) implemented in both binary and decimal formats, in addition to some other recommended functions. The recommended functions almost cover the most common mathematical functions (simple algebraic functions such as $1/\sqrt{x}$ in addition to transcendental functions such as sine, cosine, exponentials and logarithms of radices e , 2, and 10, and power . . . etc.). These elementary functions appear everywhere in scientific computing and are important for many applications. A conforming implementation for a basic operation should return results correctly rounded for the applicable rounding direction for all operands in its domain. Correct rounding means returning a floating-point result rounded (as if) infinite intermediate precision was used to evaluate the result. The general definition for correct rounding is to be within ± 0.5 ulp for rounding to nearest modes and 1 ulp for directed rounding modes. “ulp” stands for unit in the last place, i.e., the weight of the last bit/digit (bits in the case of binary floating-point and digits in the case of decimal floating-point) in a fixed precision floating-point representation. On the other hand, for recommended functions, the standard does not specify a given accuracy, i.e., it is left to be language-defined or platform-dependent.

2.5.1.3 Operations Generating NaN (QNaN)

The following are some examples of operations that can return NaN (QNaN):

1. Operations with a NaN as at least one operand.
2. Indeterminate forms:
 - (a) The divisions $\frac{\pm 0}{\pm 0}$ and $\frac{\pm \infty}{\pm \infty}$.
 - (b) The multiplications $\pm 0 \times \pm \infty$ and $\pm \infty \times \pm 0$.
 - (c) The additions $\infty + (-\infty)$, $(-\infty) + \infty$ and equivalent subtractions.

- (d) The standard has alternative functions for powers:
 - i. The standard pow function and the integer exponent pown function define 0^0 , 1^∞ , and ∞^0 as 1.
 - ii. The powr¹ function defines all three indeterminate forms as invalid operations and so returns NaN.
- 3. Real operations with complex results, for example:
 - (a) The square root of a negative number.
 - (b) The logarithm of a negative number.
 - (c) pow(x, y) for finite $x < 0$ and finite non-integer y .
- 4. Attempts to evaluate an operation outside its domain.

2.5.1.4 Exceptions and Pre-Substitutions

- Invalid operation: This exception is signaled when an operand of any function is a signaling NaN. In addition, attempts to evaluate a function outside its domain signals the invalid operation exception too. The returned result is a quiet NaN by default.
- DivideByZero: This exception is signaled when a function has a simple pole for finite operand(s). An infinity, with the correct sign, is returned by default.
- Inexact: This exception is signaled when a function returns an inexact result, i.e., different from the exact real result. Inexact means that not all its bits/digits are represented in the used precision; some least significant data is not visible in the given format.
- Overflow: This exception is signaled when a too large result is detected, that cannot be represented as a machine number. An infinity, with the correct sign, is returned by default.
- Underflow: This exception is signaled when an infinitesimally tiny non-zero result is detected, that cannot be represented in a normal form, i.e., with the hidden “1”. The returned result is called “denormalized number”.

2.5.1.5 Rounding

In the following, tie case is that when the number is exactly in the middle of two consecutive machine numbers, both exactly representable in the finite floating-point system. The types of rounding defined by the standard on a real number w are:

1) Rounding to nearest

- a. RNE(w) = Unbiased rounding to nearest (in case of a tie round to even).

¹domain of pow(x, y) is $[-\infty, +\infty] \times [-\infty, +\infty]$, domain of pown(x, n) is $[-\infty, +\infty] \times \mathbb{Z}$ (the set of all integers) while that of powr(x, y) = $e^{y \log x}$ is $[0, +\infty] \times [-\infty, +\infty]$

- b. $\text{RNA}(w)$ = Biased rounding to nearest (in case of a tie round away from zero).
- 2) Directed rounding
- a. $\text{RP}(w)$ = Round toward plus infinity. $\text{RP}(w)$ is the smallest machine number greater than or equal to w .
 - b. $\text{RM}(w)$ = Round toward minus infinity. $\text{RM}(w)$ is the largest machine number less than or equal to w .
 - c. $\text{RZ}(w)$ = Round toward zero (sometimes called chop or truncate due to the way it is implemented in sign and magnitude representation). $\text{RZ}(w)$ is equal to $\text{RM}(w)$ if $w > 0$, and to $\text{RP}(w)$ if $w \leq 0$. Hence, the rounded result is keeping the part of w that fits into the significand precision and the extra part is chopped.

2.5.2 Previous Implementations

The reason why x^y cannot be rounded according to the general definition of correct rounding explained in the previous subsection is that nobody knows how much computation it would cost to resolve “The Table-Maker’s Dilemma”. The Table-Maker’s Dilemma is the term used for the problem of always getting exactly rounded results when computing the elementary functions. Kahan in [56] wondered if $\log_{10} x$ and 10^x could be implemented so well that they are always rounded correctly? In his answer to this question he highlighted speed penalties of at least about 20% on average and perhaps 800% or more occasionally that correct rounding for these functions would impose on us. In fact, the amount of error analysis and/or exhaustive tests involved to reach such a boundary could represent more severe penalties and losses. The Table-Maker’s Dilemma problem and several developments for its solution have been discussed in [66, 102] and others. In our discussion, we concentrate on special values of the operands as they could be considered a main source of catastrophic errors such as those reviewed in Chapter 1.

For the power function floating-point implementation (pow), computing worst-case bounds for correct rounding would take extremely long time theoretically, and it is rather unrealizable practically. This claim allowed several researches that handle pow in binary floating-point representations, not on its whole definition domain, but for a subset of it. For instance, the subset that contains x^n for small integers n and $(x^{2^{-F}})^n$ of small $2^F - th$ roots of x raised to some very small integer power n in [64]. Another example is [63] in which the subset contains binary floating-point power function x^n with positive integer exponents $3 \leq n \leq 626$.

Jean-Michel Muller’s book (Elementary functions: Algorithms and implementation) [81] has been devoted to the elementary functions topic. Another book for Muller and others [82] covers the topic of elementary functions in part IV. In [82], similar topics were covered in a more concise way and the authors’ view of the current limits and perspectives has been presented. Peter Markstein’s book (IA-64 and elementary functions: speed and precision) [76] is more concerned with the elementary functions implementation in IA-64 processor at Intel. The following mentioned chapters of this book match our concern. Chapter 7 presents an overview of the different types of polynomials which can be used to approximate a desired function over a given interval within a finite tolerance. These polynomial approximations include: Taylor series, Chebyshev approximation, and Remez

approximation. Markstein proposes algorithms for both the exponential and the logarithmic functions in chapters 10 and 11, and combines them to come up with a routine for the power function in chapter 12. Discussions of the implementation method used for each function are provided detailing the approximation used for function evaluation, argument reduction, and table construction method chosen. In this work, the special operands of the power function are considered too and are handled following a C99-compatible definition [1].

An algorithm for decimal powering and the corresponding hardware architecture have been presented in [47] based on the mathematical identity:

$$x^y = 10^{y \log_{10} x}, \quad \text{for } x > 0 \quad (2.34)$$

For $x < 0$ and finite non-integer y , the invalid operation exception is signaled. A powering expression involving $x < 0$ and an odd integer y yields a negative result, while one involving $x < 0$ and an even integer y yields a positive result.

We provide further discussions of the special cases in Chapter 6. The purpose of our discussion is to cover the special cases which are usually not handled separately in the discussion of floating-point designs. Designers just utilize a previously proposed definition to decide their results following the current standards. Although in most cases designs handling the special cases are just similar to a table lookup, conformance with the standards is not always guaranteed for most implementations.

Chapter 3: Scaled Positive, Negative, and Alternating Sign Maps

The state-of-the-art of generalized maps has been reviewed in Chapter 2. In this chapter, a set of generalized maps where the conventional map is a special case is discussed for both logistic and tent maps. The proposed maps have extra degrees of freedom which provide different chaotic characteristics that increase the design flexibility and could be adapted to fit certain specifications required for many applications as previously explained. Yet, the advantage of simple mathematical relation using 1D maps only is still maintained. Requirements on the key-points of the resulting bifurcation diagram or the range of the system output could be achieved directly and simply through controlling the values of the parameters which are governed by simple relations. The simplest of which does not require any modifications from the viewpoint of implementation other than using a signed register for both the system parameter λ (or μ) and the output x . However, for a legible comprehensive analysis, we define the sign of the parameter explicitly confining its value to the set of positive reals R^+ .

Based on the maximum chaotic range of the output, the proposed maps can be classified as: positive, mostly positive, negative, and mostly negative maps. Moreover, further generalizations are proposed that allow scaling of the bifurcation diagrams. Independent, vertical, and horizontal scales of the bifurcation diagram are discussed for each generalized map as well as a new bifurcation diagram related to one of the added parameters. Mathematical analysis for each generalized map includes: bifurcation diagrams relative to all parameters, effective range of parameters, bifurcation points, as well as the maximum Lyapunov exponent (MLE) versus all parameters. As a simple application, a systematic procedure to design two-constraints logistic map is discussed and validated through four different examples.

3.1 Generalizations of One-Dimensional Maps

Properties of the conventional discrete 1D maps and their applications were extensively discussed in the literature. Recent models of stochastic behaviors represent a great challenge and strong motivation to devise a method for designing generalized maps. Requirements on the key-points of the resulting bifurcation diagram or the range of the system output could be achieved in many ways.

3.1.1 The Proposed Maps

In this section, we propose a simple and efficient way of designing logistic and tent maps generating outputs with various signs through controlling the signs of multiple parameters governed by simple relations.

3.1.1.1 Variations on Logistic Map

We define the following parameterized maps as variations on the conventional logistic map equation (2.1) in its general form given by

$$x_{n+1} = \pm \lambda x_n (a \pm b x_n); \lambda, a, b \in R^+ \quad (3.1)$$

where the proposed maps are furthermore controlled by the set of parameters (a, b, λ) such that $a, b, \lambda \in R^+$. The names of the maps have been chosen carefully to represent the maximum chaotic range of the output.

Definition 3.1.1. *Variations on the sign of the generalized logistic map represented by (3.1) can be subdivided into two types according to the maximum chaotic range of the output:*

1. Single sign maps

(a) *Positive logistic map given by:*

$$f_1(x_n) = x_{n+1} = \lambda x_n (a - b x_n). \quad (3.2)$$

(b) *Negative logistic map given by:*

$$f_3(x_n) = x_{n+1} = \lambda x_n (a + b x_n). \quad (3.3)$$

where $f_3(x) = -f_1(-x)$. The bifurcation diagram of this map is the reflected image of $f_1(x)$ about the horizontal axis (λ), yielding exactly the same range of output magnitude, but with opposite sign.

2. Alternating sign maps

(a) *Mostly positive logistic map given by:*

$$f_2(x_n) = x_{n+1} = -\lambda x_n (a - b x_n). \quad (3.4)$$

(b) *Mostly negative logistic map given by:*

$$f_4(x_n) = x_{n+1} = -\lambda x_n (a + b x_n). \quad (3.5)$$

where $f_4(x) = -f_2(-x)$. Thus, the relation between the two maps is similar to that between f_1 and f_3 .

Figure 3.1 shows the bifurcation diagrams for the four logistic maps in the case of unity scaling ($a = b = 1$), and the graphs of their equations at maximum λ . Table 3.1 illustrates the main differences between the proposed logistic maps in the case of unity scaling.

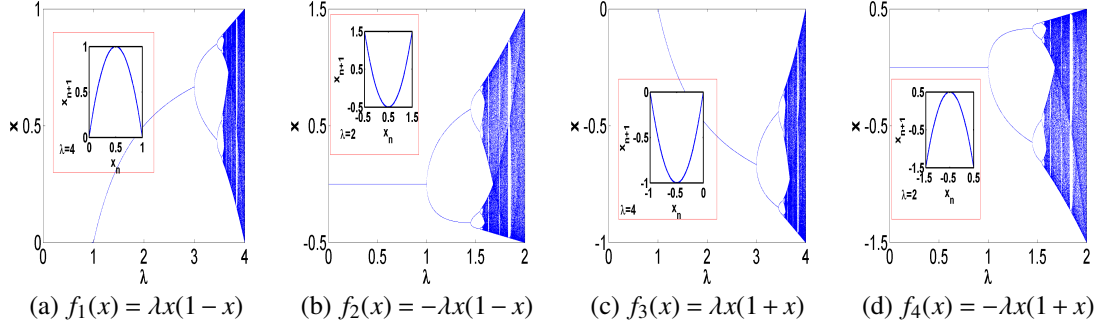


Figure 3.1: Bifurcation diagrams and graphs for the proposed unity scaling logistic maps

Table 3.1: Main differences between the proposed unity scaling logistic maps

Map	Range of λ	Range of x
$\lambda x(1-x)$	$[0, 4]$	$[0, 1]$
$-\lambda x(1-x)$	$[0, 2]$	$[-0.5, 1.5]$
$\lambda x(1+x)$	$[0, 4]$	$[-1, 0]$
$-\lambda x(1+x)$	$[0, 2]$	$[-1.5, 0.5]$

3.1.1.2 Variations on Tent Map

For generalized tent map, the sign of bx_n in the term $a \pm bx_n$ cannot be positive at the same time with the term x_n since the two straight lines x and $a + bx$ are non-intersecting and cannot construct a closed set region with bounded responses. Instead, we could control the sign of the term $\pm x_n$ along with $a \pm bx_n$ such that they always possess opposite signs. So, the set of all alternatives is given by

$$x_{n+1} = \pm \mu \min(\pm x_n, a \mp bx_n); \mu, a, b \in \mathbb{R}^+ \quad (3.6)$$

Definition 3.1.2. Variations on the sign of the generalized tent map represented by (3.6) can be subdivided into two types according to the maximum chaotic range of the output:

1. Single sign maps

(a) Positive tent map given by:

$$f_1(x_n) = x_{n+1} = \mu \min(x_n, a - bx_n). \quad (3.7)$$

(b) Negative tent map given by:

$$f_3(x_n) = x_{n+1} = -\mu \min(-x_n, a + bx_n). \quad (3.8)$$

2. Alternating sign maps

(a) Mostly positive tent map given by:

$$f_2(x_n) = x_{n+1} = -\mu \min(x_n, a - bx_n). \quad (3.9)$$

(b) Mostly negative tent map given by:

$$f_4(x_n) = x_{n+1} = \mu \min(-x_n, a + bx_n). \quad (3.10)$$

Figure 3.2 shows the bifurcation diagrams for the four tent maps in the case of unity scaling ($a = b = 1$), and the graphs of their equations at maximum μ . Table 3.2 illustrates the main differences between the proposed tent maps in the case of unity scaling.

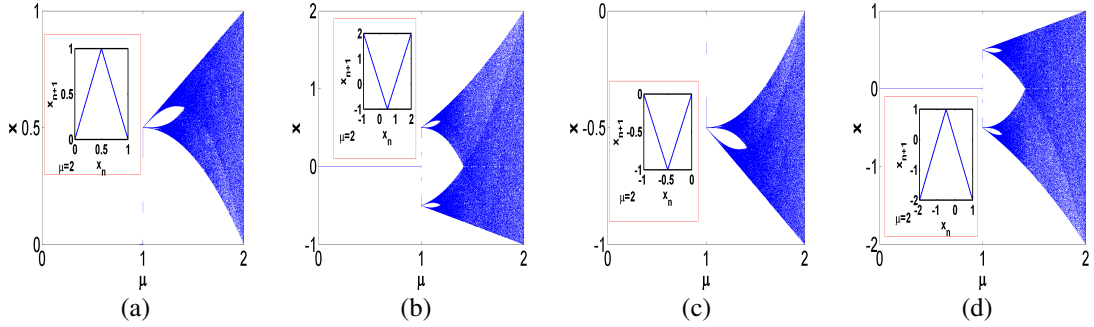


Figure 3.2: Bifurcation diagrams and graphs for the proposed unity scaling tent maps (a) $f_1(x) = \mu \min(x, 1 - x)$, (b) $f_2(x) = -\mu \min(x, 1 - x)$, (c) $f_3(x) = -\mu \min(-x, 1 + x)$, and (d) $f_4(x) = \mu \min(-x, 1 + x)$

Table 3.2: Main differences between the proposed unity scaling tent maps

Map	Range of μ	Range of x
$\mu \min(x, 1 - x)$	$[0, 2]$	$[0, 1]$
$-\mu \min(x, 1 - x)$	$[0, 2]$	$[-1, 2]$
$-\mu \min(-x, 1 + x)$	$[0, 2]$	$[-1, 0]$
$\mu \min(-x, 1 + x)$	$[0, 2]$	$[-2, 1]$

3.1.2 Alternating Sign Maps

In this subsection, we concentrate on mostly positive maps. The other two proposed maps: negative map and mostly negative map are related to positive and mostly positive maps respectively as previously mentioned. Thus, even for parameterized maps, we analyze both positive and mostly positive maps because the results for the other two maps could be deduced similarly.

3.1.2.1 Mostly Positive Logistic Map

A study of the properties of the logistic map with negative control parameter $\lambda \in [-2, 0]$, that is quite similar to our mostly positive logistic map, has been conducted in [8].

3.1.2.1.1 Parameters' Ranges Consider the recurrence $x_{n+1} = -\lambda x_n(1 - x_n)$, graphically represented by the parabola $f(x) = -\lambda x(1 - x)$; the critical point of the parabola is x_c such that

$$f'(x_c) = 0 \rightarrow x_c = 0.5 \text{ and } f(x_c) = f_{min} = -\frac{\lambda}{4}. \quad (3.11)$$

For stable steady state response, all the outputs of the recurrence relation should be limited within a closed set. This closure property could be guaranteed by setting a maximum value for λ . From the symmetry of the curve, this value can be obtained by forcing

$$|f_{min}| = x_c \rightarrow \frac{\lambda_{max}}{4} = 0.5 \rightarrow \lambda_{max} = 2. \quad (3.12)$$

A more formal way of defining the parameter's range is as follows.

Definition 3.1.3. Consider the map given by $f(x) = -\lambda x(1-x)$, the point u_1 is the one whose coordinates equal the map minimum value f_{min} , and u_2 is the non trivial fixed point or the intersection of the curve $y = f(x)$ and $y = x$. The map is stable if and only if the image of u_1 ($f(u_1)$) is less than or equal to u_2 , such that $x \in [u_1, u_2]$. This is achieved as $\lambda \in [0, 2]$.

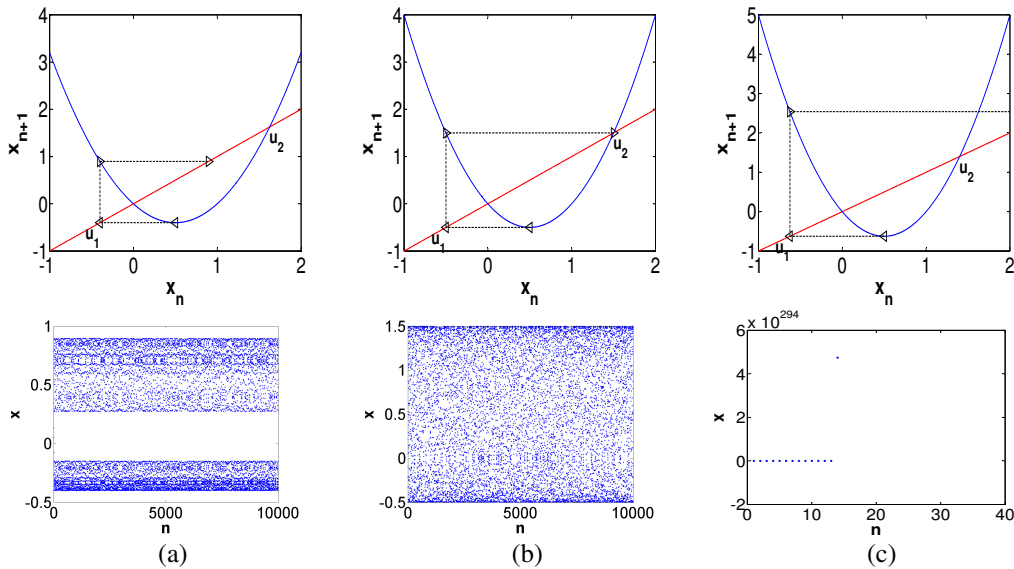


Figure 3.3: Maximum value of λ for mostly positive logistic map (a) $\lambda = 1.6$, (b) $\lambda = 2$, and (c) $\lambda = 2.5$

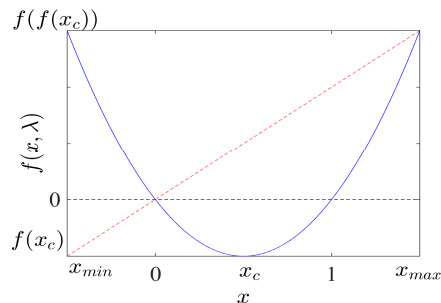


Figure 3.4: Domain and range of $f(x) = -\lambda_{max}x(1-x)$

Figure 3.3 shows the graph of the equation for different values of $\lambda = 1.6, 2$, and 2.5 , where the map exhibits bounded output if and only if $\lambda \in [0, 2]$ as expected before. For $\lambda = 2$, the values of x at u_1 and u_2 are -0.5 and 1.5 respectively, i.e., closed set responses

are yielded as $x \in [-0.5, 1.5]$. Figure 3.3 also shows the values of x carrying out 10,000 iterations of the recurrence. The time waveforms indicate the stability of the recurrence for $0 \leq \lambda \leq 2$ as it exhibits a bounded solution. On the other hand, the range $\lambda > 2$ shows complete instability of the solution as it diverges to $+\infty$ starting the fifteenth iteration.

The previous method for calculation and verification of the range of values λ and x that guarantee bounded responses has not been conducted in [8]. Instead, the value $\lambda_{max} = 2$ and the interval $x \in [-0.5, 1.5]$ were used directly in addition to a computer aided plot of the bifurcation diagram that ensures those values. However, our previous method that requires iterating several values of λ and/or an initial assumption. That's why we propose a more generic procedure of specifying the parameters' ranges that could be detailed as follows with the aid of Fig. 3.4 where a map is defined as a function whose domain (input) space and range (output) space are the same [7].

1. The lower bound on the range of the map is $f_{min} = f(x_c)$ which is the same as the lower bound on the domain x_{min} .

$$f(x_c) = -\frac{\lambda_{max}}{4}, \quad (3.13)$$

where λ_{max} here is generally different from the positive control parameter case.

2. The upper bound on the range is $f(x_{min})$, i.e., $f(f(x_c))$. So, the upper bound is equal to

$$f(f(x_c)) = (-\lambda_{max})\left(-\frac{\lambda_{max}}{4}\right)\left(1 + \frac{\lambda_{max}}{4}\right), \quad (3.14)$$

which is a function of λ_{max} .

3. Then, we equate this value to the map equation to get the corresponding solutions for the values of x : x_{min} and x_{max} respectively.

$$\frac{\lambda_{max}^2}{4} + \frac{\lambda_{max}^3}{16} = -\lambda_{max}x(1-x). \quad (3.15)$$

Rename the left hand side of the previous equation to $\lambda_{max}C$, it reduces to

$$C = -x + x^2, \quad (3.16)$$

Thus,

$$x_{min} = -\frac{\lambda_{max}}{4}, \quad (3.17a)$$

$$x_{max} = 1 + \frac{\lambda_{max}}{4}. \quad (3.17b)$$

4. The domain of the map is $D = [x_{min}, x_{max}]$, while its range is $R = [f(x_c), f(f(x_c))]$. We equate the lower and upper bounds of both intervals respectively to get λ_{max} .

$$1 + \frac{\lambda_{max}}{4} = \frac{\lambda_{max}^2}{4} + \frac{\lambda_{max}^3}{16}. \quad (3.18)$$

This equation could be solved to get three values among which one value is positive which is $\lambda_{max} = 2$.

5. Hence, we get x_{min} and x_{max} by substitution for λ_{max} as follows.

$$x_{min} = -\frac{1}{2}, \quad (3.19a)$$

$$x_{max} = \frac{3}{2}. \quad (3.19b)$$

Thus, the parameters' ranges for mostly positive logistic map are

$$(\lambda_{max}, x_{min}, x_{max}) = \left(2, -\frac{1}{2}, \frac{3}{2}\right). \quad (3.20)$$

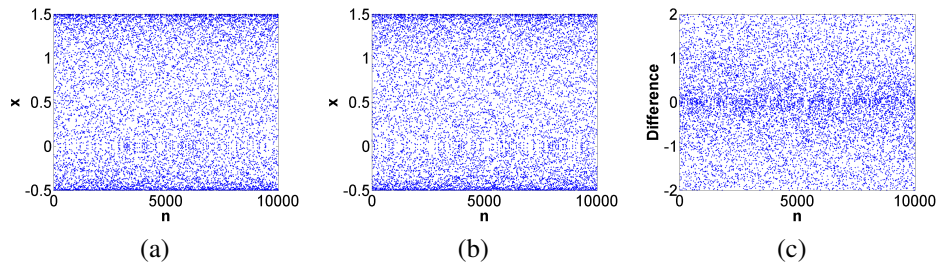


Figure 3.5: Sensitive dependence on initial point for mostly positive logistic map at $\lambda = 2$
(a) Time waveform $x_0 = 0.05$, (b) Time waveform $x_0 = 0.06$, and (c) Initial point effect

The mathematical analysis of the fixed and periodic points of mostly positive logistic map in the case of unity scaling is quite similar to that of the conventional logistic map. For $1 < \lambda < \sqrt{6} - 1$, the recurrence converges to a period-2 orbit, followed by a sequence of periodic doubling. Starting at $\lambda = \sqrt{8} - 1$, the recurrence converges to a period-3 orbit. The output continues bifurcation until λ reaches the value 2 which exhibits the maximum chaotic behavior. This indicates that mostly positive logistic map exhibits period doubling as a route to chaos with the same characteristic Feigenbaum universal constant. In addition to the bifurcation diagram shown in Fig. 3.1(b) and the graph of the equation shown in Fig. 3.3, the time waveforms starting at different initial points $x_0 = 0.05$ and $x_0 = 0.06$ are shown in Fig. 3.5. These responses demonstrate the maximum chaotic behavior of the map at $\lambda = 2$. Figure 3.5(c) shows the difference between these two responses, the large number of non zero points indicate the sensitive dependence on initial conditions which is a main property of chaotic systems.

3.1.2.1.2 Fixed Points and Their Stability Solving $x^* = f(\lambda, x^*)$ yields

$$x_1^* = 0 \text{ and } x_2^* = 1 + \frac{1}{\lambda}, \quad (3.21)$$

$$f'(x^*) = -\lambda(1 - 2x^*), \quad (3.22)$$

$$|f'(x^*)| = 1 \rightarrow \lambda_b = 1. \quad (3.23)$$

The other fixed point yields $\lambda = 3$ which is greater than λ_{max} . Hence, this solution is refused and the first bifurcation point and the corresponding function value are

$$(x_b, \lambda_b) = (0, 1). \quad (3.24)$$

3.1.2.2 Mostly Positive Tent Map

The recurrence representing mostly positive tent map is given by

$$x_{n+1} = -\mu \min(x_n, 1 - x_n), \quad (3.25a)$$

$$x_{n+1} = \begin{cases} -\mu x_n & x \leq x_k \\ -\mu(1 - x_n) & x_k < x \end{cases}, \quad (3.25b)$$

where

$$x_k = 0.5 \quad (3.26)$$

is the point of intersection of the two lines.

3.1.2.2.1 Parameters' Ranges Similar to the analysis conducted for mostly positive logistic map, we get the following parameters' ranges with the aid of Fig. 3.6.

$$(\mu_{max}, x_{min}, x_{max}) = (2, -1, 2). \quad (3.27)$$

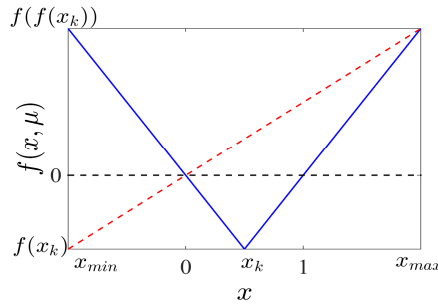


Figure 3.6: Domain and range of $f(x) = -\mu_{max} \min(x, 1 - x)$

3.1.2.2.2 Fixed Points and Their Stability Similarly,

$$x_1^* = 0 \text{ and } x_2^* = -\frac{\mu}{1-\mu}, \quad (3.28)$$

$$f'(x^*) = \begin{cases} -\mu & x^* \leq x_k \\ +\mu & x_k < x^* \end{cases}, \quad (3.29)$$

$$|f'(x^*)| = 1 \rightarrow \mu_b = 1. \quad (3.30)$$

The first bifurcation point is

$$(x_{b1}, \mu_{b1}) = (0, 1). \quad (3.31)$$

3.1.2.2.3 Periodic Points and Their Stability The composite function $f^2(x) = f(f(x))$ is given by

$$f^2(x) = \begin{cases} -\mu(1 + \mu x) & x \leq x_{k1} \\ \mu^2 x & x_{k1} \leq x \leq x_k \\ \mu^2(1 - x) & x_k \leq x \leq x_{k2} \\ -\mu(1 + \mu(1 - x)) & x_{k2} \leq x \end{cases}, \quad (3.32)$$

where $x_{k1} = -0.25$, $x_k = 0.5$, and $x_{k2} = 1.25$. To get the periodic points of period-2, equate $x_p = f^2(x_p)$

$$\begin{cases} x_p = -\frac{\mu}{1+\mu^2} & x_p \leq x_{k1} \\ x_p = 0 & x_{k1} \leq x_p \leq x_k \\ x_p = \frac{\mu^2}{1+\mu^2} & x_k \leq x_p \leq x_{k2} \\ x_p = \frac{-\mu-\mu^2}{1-\mu^2} & x_{k2} \leq x_p \end{cases}. \quad (3.33)$$

To get the bifurcation point, $|(f^2)'(x_p)| = 1$

$$(f^2)'(x) = \begin{cases} -\mu^2 & x \leq x_{k1} \\ \mu^2 & x_{k1} \leq x \leq x_k \\ -\mu^2 & x_k \leq x \leq x_{k2} \\ \mu^2 & x_{k2} \leq x \end{cases}. \quad (3.34)$$

Consequently,

$$(x_{b2}, \mu_{b2}) = \left(\left\{ -\frac{1}{2}, \frac{1}{2} \right\}, 1 \right). \quad (3.35)$$

3.2 Scaled Logistic Map: Analysis and Results

For each map of those presented in the previous section, the resulting bifurcation diagram can be designed with the desired key-points using (a, b) as scaling parameters. The parameterized cases studied are: (a, b) , such that $a, b \in R^+$, $(1, b)$, (a, a) and $(a, 1)$. The following subsections provide the analysis and results of these parameterized cases for positive and mostly positive logistic maps.

3.2.1 Independent Scaling $x_{n+1} = \pm \lambda x_n(a \pm b x_n)$

In this map, two parameters a and b are added to be capable of scaling both the horizontal and the vertical axes of the bifurcation diagram independently. The analysis presented in this subsection is used later on in section 3.4 to design any logistic map with specific characteristics.

3.2.1.1 Positive Logistic Map $f(x, \lambda, a, b) = \lambda x(a - bx)$

3.2.1.1.1 Range of λ To ensure bounded output for all iterations, the value of x should be limited to $x \in [0, a/b]$. The roots of the map, its critical point x_c , and its maximum value x_{max} are given by:

$$f(x) = 0 \text{ for } x = 0, \frac{a}{b}, \quad (3.36a)$$

$$x_c = \frac{a}{2b}, f(x_c, \lambda_{max}, a, b) = \frac{a^2 \lambda_{max}}{4b} \leq \frac{a}{b} \rightarrow \lambda_{max} \leq \frac{4}{a}. \quad (3.36b)$$

This inequality not only provides us with information on the maximum value of the parameter λ , but also that of the parameter a . Therefore, the maximum values for the parameters of the map are:

$$(\lambda_{max}, x_{max}) = \left(\frac{4}{a}, \frac{a}{b} \right), \quad (3.37a)$$

$$a \in (0, a_{max}] \text{ where } a_{max} = \frac{4}{\lambda}. \quad (3.37b)$$

3.2.1.1.2 Fixed Points and Stability Condition The fixed points are given by $x^* = f(x^*, \lambda, a, b)$, then

$$x^* = \lambda x^*(a - bx^*), \quad (3.38a)$$

$$x_1^* = 0 \text{ and } x_2^* = \frac{1}{b} \left(a - \frac{1}{\lambda} \right). \quad (3.38b)$$

The study of the stability of the fixed points is based on the calculation of the absolute value of the first derivative w.r.t x at these points $|f'(x^*, \lambda, a, b)| = \lambda(a - 2bx^*)$. They are either stable or unstable depending on whether this value is less or greater than “one” respectively. Otherwise, if the absolute value equals “one”, then it is called a bifurcation point. The derivative of this map at the fixed points is given by:

$$f'(x_1^*, \lambda, b) = \lambda \text{ and } f'(x_2^*, \lambda, b) = 2 - a\lambda. \quad (3.39)$$

Therefore, the values of λ at which the system bifurcates and their function values are

$$|f'(x^*, \lambda_b, a)| = 1 \rightarrow \lambda_b = \frac{1}{a}, \frac{3}{a}, \quad (3.40a)$$

$$f(x_1^*, \lambda_{b1}, a, b) = 0 \text{ and } f(x_2^*, \lambda_{b2}, a, b) = \frac{2a}{3b}, \quad (3.40b)$$

$$(x_b, \lambda_b) = \left\{ \left(0, \frac{1}{a} \right), \left(\frac{2a}{3b}, \frac{3}{a} \right) \right\}. \quad (3.40c)$$

Figures 3.7(a) and (b) show snapshots of the bifurcation diagrams versus the main parameter λ at different values of a and b fixed, and different values of b and a fixed respectively. The diagrams indicate that the value of λ_{max} depends on the parameter a only irrespective of b , while both parameters have an impact on the output range which is consistent with the previous analysis. The diagrams are analyzed in details in subsequent sections.

3.2.1.2 Mostly Positive Logistic Map $f(x, \lambda, a, b) = -\lambda x(a - bx)$

3.2.1.2.1 Range of λ Similarly, for mostly positive logistic map, the solution should be limited to $x \in [-a/(2b), 3a/(2b)]$.

$$x_c = \frac{a}{2b}, f(x_c, \lambda_{max}, a, b) = -\frac{a^2 \lambda_{max}}{4b} \geq -\frac{a}{2b}, \quad (3.41a)$$

$$(\lambda_{max}, x_{min}, x_{max}) = \left(\frac{2}{a}, -\frac{a}{2b}, \frac{3a}{2b} \right), \quad (3.41b)$$

$$a \in (0, a_{max}] \text{ where } a_{max} = \frac{2}{\lambda}. \quad (3.41c)$$

3.2.1.2.2 Fixed Points and Stability Condition Similarly, for mostly positive logistic map, the value of λ at which the system bifurcates and the corresponding function value are

$$(x_b, \lambda_b) = \left(0, \frac{1}{a} \right). \quad (3.42)$$

Figure 3.8 shows that the dependence of the independent scaling mostly positive logistic map on the parameters a and b could be described in a similar way to that of independent scaling positive logistic map.

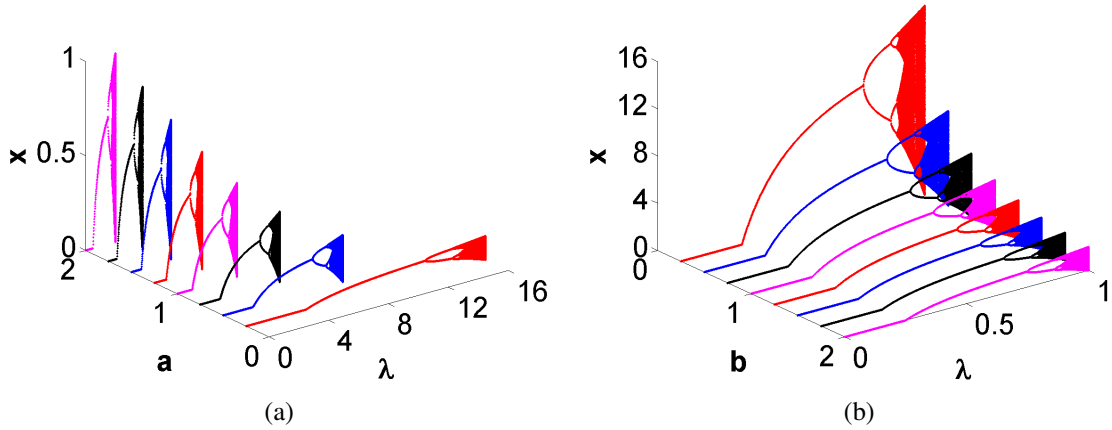


Figure 3.7: Bifurcation diagram vs. λ for independent scaling positive logistic map at (a) $b = 2$ and $a = \{0.25, 0.5, \dots, 2\}$ and (b) $a = 4$ and $b = \{0.25, 0.5, \dots, 2\}$

3.2.1.3 Maximum Lyapunov Exponent

Figure 3.9(a) shows 3D plot of MLE as a function of both λ and a at $b = 2$ for independent scaling positive logistic map. This continuous surface plot illustrates the dependence of the allowed range of λ on the value of a according to equation (3.37a). However, the value of MLE approaches the same steady state value of $\ln 2$ for maximum chaotic behavior, or at λ_{max} . Figure 3.9(b) indicates that the parameter b has no impact on the range of the main system parameter λ as λ_{max} is independent of b . Figure 3.9(c) shows the values of MLE at λ_{max} in the $a - b$ plane indicating that setting the main system parameter to λ_{max} corresponding to a achieves maximum chaotic behavior irrespective of b . Figure 3.9(d) shows this maximum chaotic output in the $a - b$ plane where generally the lower and upper bounds on the range are constrained by the values of a and b according to equation (3.37a). In order to get a wider output range, we could increase a and/or decrease b .

Figure 3.10 shows the impact of both a and b on MLE and chaotic output that could be described similar to independent scaling positive logistic map.

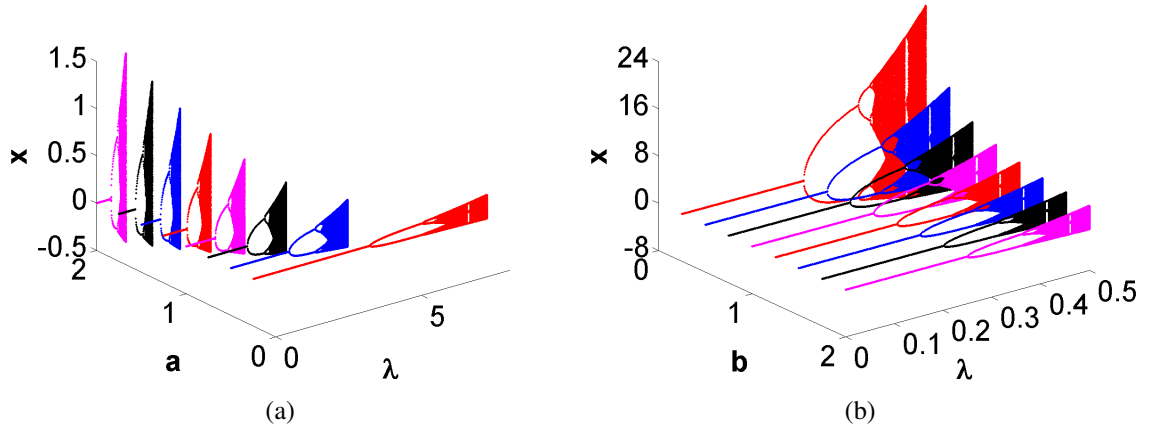


Figure 3.8: Bifurcation diagram vs. λ for independent scaling mostly positive logistic map at (a) $b = 2$ and $a = \{0.25, 0.5, \dots, 2\}$ and (b) $a = 4$ and $b = \{0.25, 0.5, \dots, 2\}$

3.2.2 Vertical Scaling $x_{n+1} = \pm\lambda x_n(1 \pm b x_n)$

In this map, an extra parameter b is added which affects the vertical scaling of the bifurcation diagram, i.e., it is a special case of the independent scaling map in which $a = 1$. This map could be used to control the vertical axis, i.e., the values of the output, not the bifurcation points. In this subsection, the range of λ , fixed points and their stability analysis, and behavior dependence on different system parameters for the proposed maps are discussed.

3.2.2.1 Positive Logistic Map $f(x, \lambda, b) = \lambda x(1 - bx)$

3.2.2.1.1 Range of λ Substituting $a = 1$ in (3.37a) yields:

$$(\lambda_{max}, x_{max}) = \left(4, \frac{1}{b}\right). \quad (3.43)$$

The function iterations $f^m(x, \lambda, b)$ at $b = 2$ are shown in Fig. 3.11, where the maximum value of the output $x_{max} = 1/b = 0.5$ and the parameter b affects the vertical axis only. The number of peaks increases as m increases and as λ approaches its maximum value $\lambda_{max} = 4$ which exhibits maximum chaotic behavior. On the other hand, for $\lambda < 3$, no peaks could be noticed even in higher iterations.

3.2.2.1.2 Fixed Points and Stability Condition Substituting $a = 1$ in (3.40c) yields:

$$(x_b, \lambda_b) = \left\{ (0, 1), \left(\frac{2}{3b}, 3 \right) \right\}. \quad (3.44)$$

3.2.2.1.3 Steady State Solutions Versus System Parameters Figure 3.12 shows the steady state solutions versus both the system parameters b and λ for vertical scaling positive logistic map. The bifurcation diagram versus λ exhibits similar behavior to the conventional case but with vertical scaling by the factor $(1/b)$ as shown in Fig. 3.12(a).

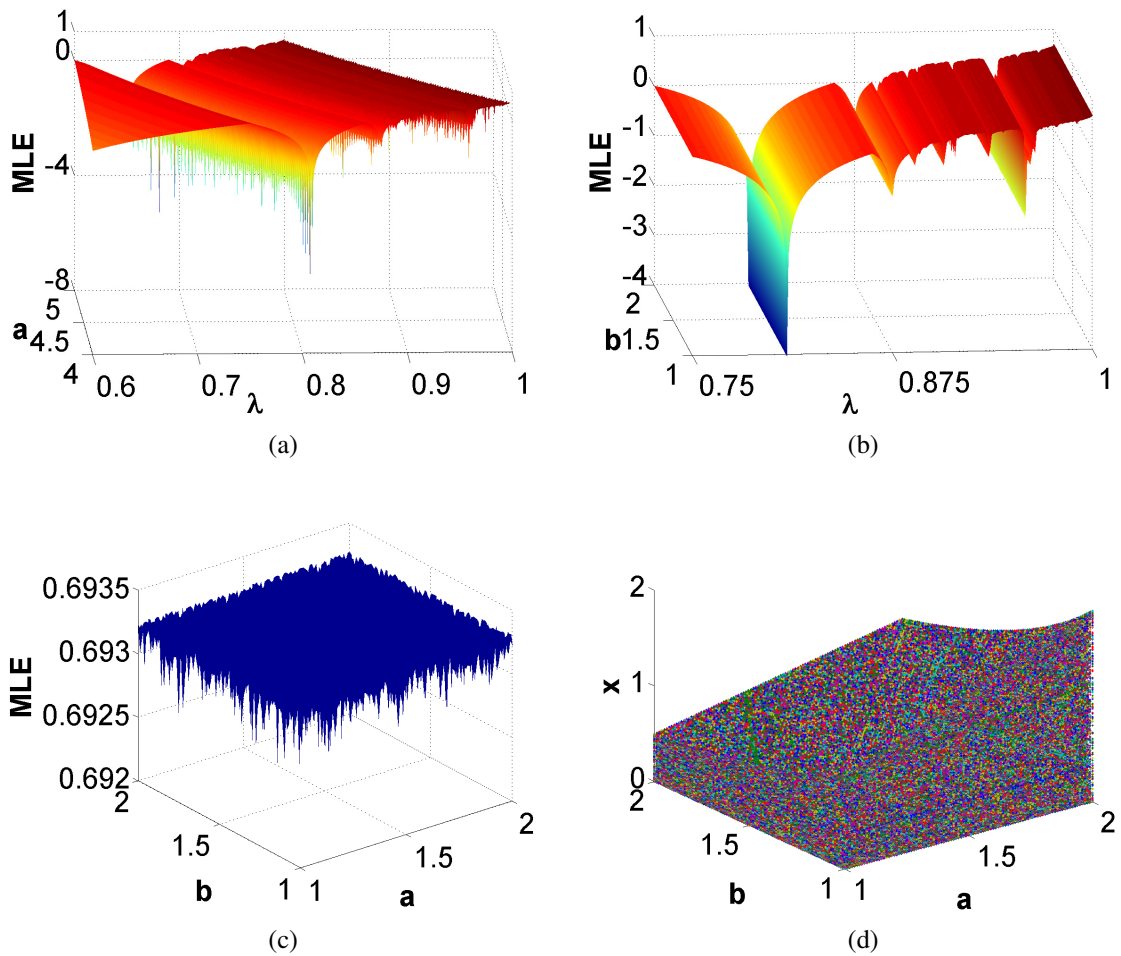


Figure 3.9: MLE of independent scaling positive logistic map as a function of (a) λ and a at $b = 2$, (b) λ and b at $a = 4$, (c) a and b at λ_{max} , and (d) Full-range chaotic output versus a and b

The nontrivial solution appears once λ exceeds the first bifurcation point $\lambda_{b1} = 1$, while the second bifurcation point is $\lambda_{b2} = 3$ with corresponding steady state value $x_{b2} = 2/(3b)$. Figure 3.12(a) also shows the cobweb plot of the map at $\lambda_{max} = 4$ and $b = 10$, which differs as the parameters change. The cobweb is a rough plot for an orbit of x , where the graph of the map function is sketched together with the diagonal line $y = x$. The plot shows that the map exhibits chaotic behavior, since the orbit has a non-periodic sequence and generates multiple outputs that cover the whole range of x .

Figure 3.12(b) shows the behavior dependence on b for a fixed value of λ . Theoretically, the only condition on b is to be positive. It should be noted that for larger values of b , extremely small initial points are needed such that they belong to the allowable range. Disregarding this note would yield results that might be falsely interpreted as a case of instability. Plotting starts at $b = 0.1$ to avoid larger values of x that decrease the clarity of the figure. The range of output results x depends on b , i.e., $x \in (0, 1/b)$ as proposed by our previous analysis. The type of solution is either fixed, period-2, multi-periods, or chaos depending on the value of λ . As λ increases, the response covers more values belonging to the allowable range until it covers the whole range $x \in (0, 1/b)$ at $\lambda_{max} = 4$.

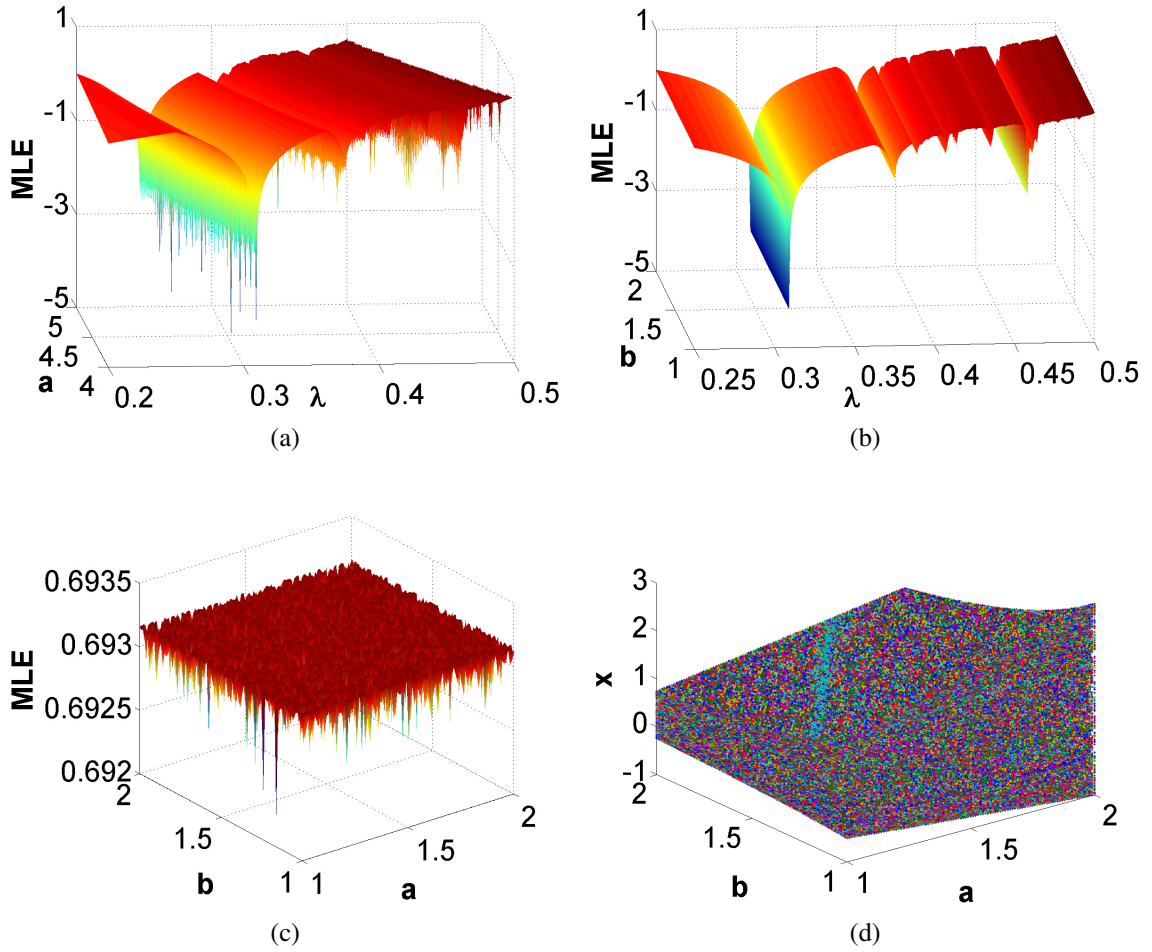


Figure 3.10: MLE of independent scaling mostly positive logistic map as a function of (a) λ and a at $b = 2$, (b) λ and b at $a = 4$, (c) a and b at λ_{max} , and (d) Full-range chaotic output versus a and b

3.2.2.2 Mostly Positive Logistic Map $f(x, \lambda, b) = -\lambda x(1 - bx)$

3.2.2.2.1 Range of λ Substituting $a = 1$ in (3.41b) yields:

$$(\lambda_{max}, x_{min}, x_{max}) = \left(2, -\frac{1}{2b}, \frac{3}{2b}\right). \quad (3.45)$$

3.2.2.2.2 Fixed Points and Stability Condition Substituting $a = 1$ in (3.42) yields:

$$(x_b, \lambda_b) = (0, 1). \quad (3.46)$$

3.2.2.2.3 Steady State Solutions Versus System Parameters Figure 3.13 shows the steady state solutions versus both the system parameters b and λ , and the cobweb plot at $\lambda_{max} = 2$ and $b = 10$ for mostly positive logistic map. The results could be interpreted and proved to be conforming to our analysis. Ten different snapshots of the bifurcation diagram versus the system parameter λ for both maps are shown in Fig. 3.14, where the idea of vertical scaling is quite clear.

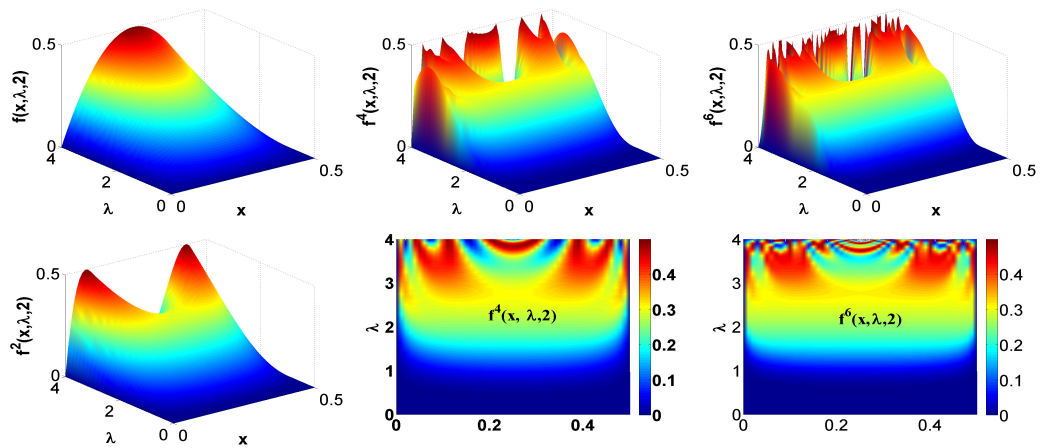


Figure 3.11: Function iterations of vertical scaling positive logistic map $f^m(x, \lambda, b)$ at $b = 2$ for $m = \{1, 2, 4, 6\}$

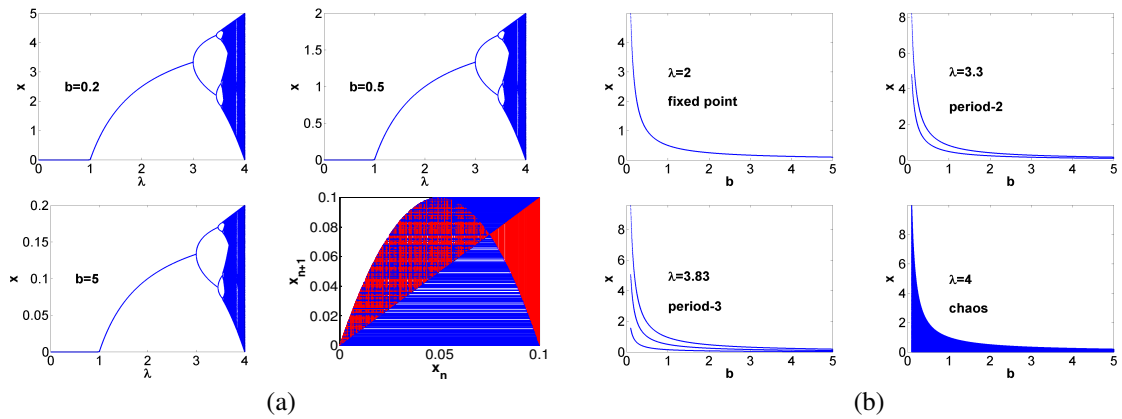


Figure 3.12: (a) Bifurcation diagram versus λ for different values of $b = \{0.2, 0.5, 5\}$ and Cobweb plot at $\lambda = 4, b = 10$, (b) Steady state solutions of x versus b for different values of $\lambda = \{2, 3.3, 3.83, 4\}$ for vertical scaling positive logistic map

3.2.2.3 Maximum Lyapunov Exponent

For both vertical scaling maps, the time evolution of MLE for different values of b roughly reaches the same steady state value. This constant value is the same as that of the conventional map $MLE = \ln 2$. Besides, all map variations exhibit the same value at λ_{max} representing maximum chaotic behavior. This value has been proved for almost all orbits of chaotic maps, starting from the tent map, and extending to the logistic map and others by conjugacy of maps [7]. Figure 3.15(a) shows 3D and contour plots of MLE as a function of both λ and b for vertical scaling positive logistic map, while Fig. 3.15(b) shows it for vertical scaling mostly positive logistic map.

A horizontal scaling map $x_{n+1} = \pm \lambda a x_n (1 \pm x_n)$ could also be designed with an extra parameter a . It affects the horizontal scaling of the bifurcation diagram, i.e., the parameter could be used to control the bifurcation points not the values of the output.

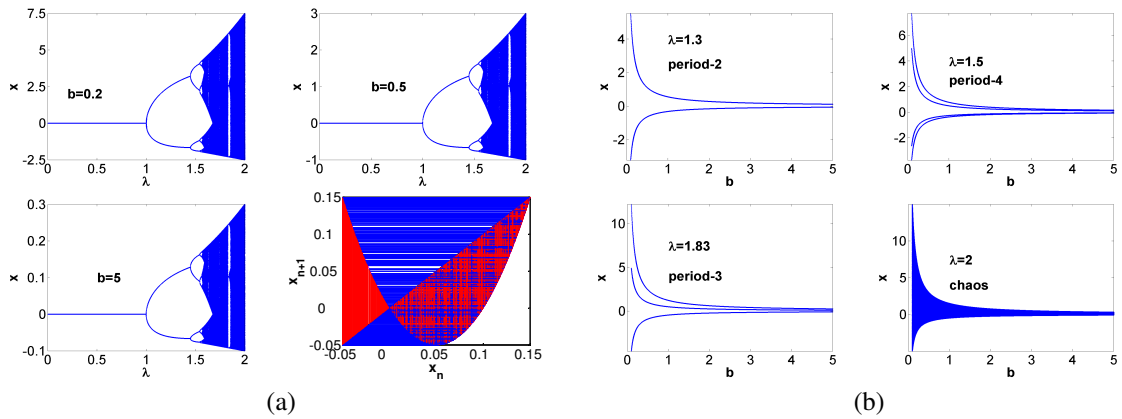


Figure 3.13: (a) Bifurcation diagram versus λ for different values of $b = \{0.2, 0.5, 5\}$ and Cobweb plot at $\lambda = 2, b = 10$, (b) Steady state solutions of x versus b for different values of $\lambda = \{1.3, 1.5, 1.83, 2\}$ for vertical scaling mostly positive logistic map

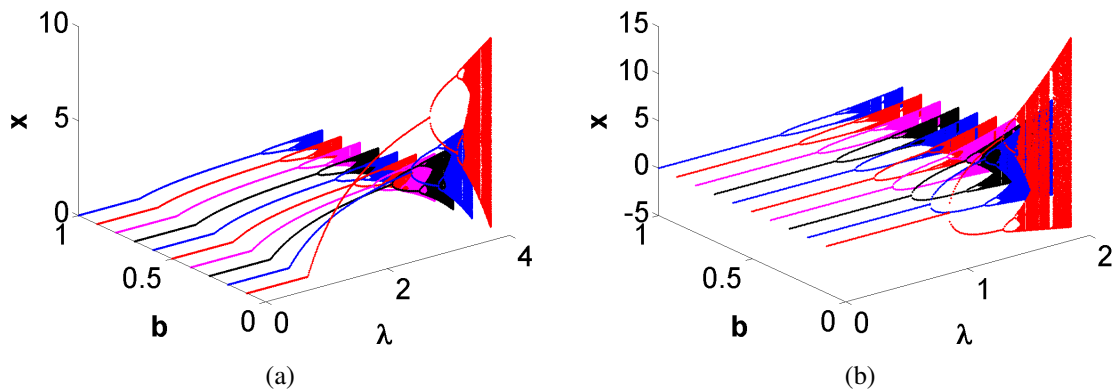


Figure 3.14: Ten different snapshots of the bifurcation diagram versus λ for vertical scaling (a) positive and (b) mostly positive maps at $b = \{0.1, 0.2, \dots, 1\}$

3.2.3 Zooming $x_{n+1} = \pm \lambda x_n (a \pm x_n)$

In this map, an extra parameter a is added which affects the zooming of the bifurcation diagram, i.e., it is a special case of the independent scaling map in which $b = 1$. Moreover, there is a new bifurcation diagram presented with respect to the system parameter a that will be generalized later on. This map is richer in analysis and could be used for controlling both axes, i.e., the bifurcation points in addition to the corresponding output values. This control can be achieved only in a dependent way, such that the area of the bifurcation diagram versus λ remains constant. In this subsection, the range of λ , fixed points and their stability analysis, and two different bifurcation diagrams are discussed.

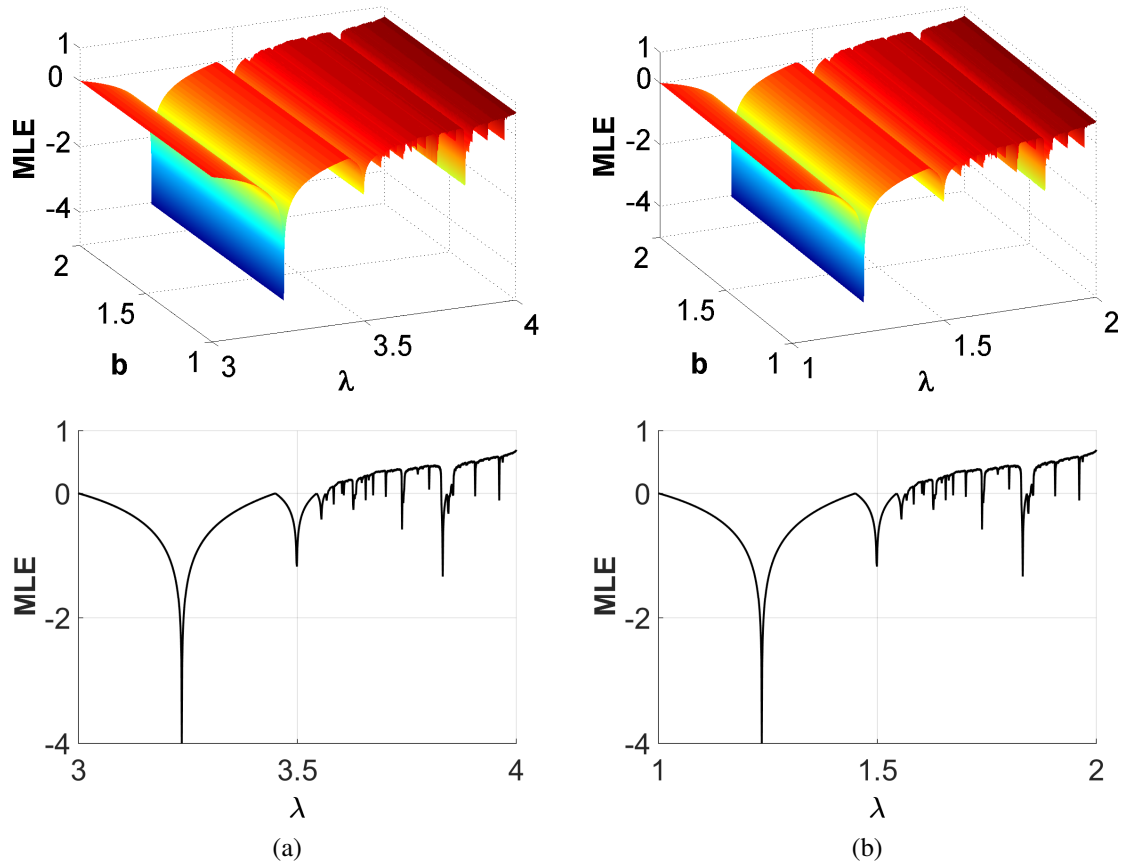


Figure 3.15: MLE as a function of both λ and b for vertical scaling (a) positive and (b) mostly positive logistic maps

3.2.3.1 Positive Logistic Map $f(x, \lambda, a) = \lambda x(a - x)$

3.2.3.1.1 Range of λ Substituting $b = 1$ in (3.37a) yields:

$$(\lambda_{max}, x_{max}) = \left(\frac{4}{a}, a \right), \quad (3.47a)$$

$$a \in (0, a_{max}] \text{ where } a_{max} = \frac{4}{\lambda}. \quad (3.47b)$$

3.2.3.1.2 Fixed Points and Stability Condition Substituting $b = 1$ in (3.40c) yields:

$$(x_b, \lambda_b) = \left\{ \left(0, \frac{1}{a} \right), \left(\frac{2a}{3}, \frac{3}{a} \right) \right\}. \quad (3.48)$$

3.2.3.1.3 Steady State Solutions Versus System Parameters Figure 3.16(a) shows the bifurcation diagram versus λ for different values of a for zooming positive logistic map. The horizontal and vertical axes corresponding to the values of the parameter λ and the solution x are scaled by $(1/a)$ and a respectively so that the total area is still the same and equals to a constant value of $a(4/a) = 4$ square units. Figure 3.16(a) also shows the

cobweb plot at $a = 4$ and $\lambda = 1$, which shows chaotic behavior where the whole range of x is covered.

Figure 3.16(b) shows the bifurcation diagram versus a for different values of λ , where the diagram has equal axes lengths which we shall call a square-axis, and its area is a function of λ . The unity area square results in the case of $\lambda = 4$ which corresponds to the maximum value of λ in the unity scaling case ($a = 1$). Recalling that $a_{max} = (4/\lambda)$ as given by (3.47b), then the values of a are scaled by $(4/\lambda)$. Accordingly, the values of x undergo the same scaling and $x \in [0, 4/\lambda]$. Thus, for positive logistic map, the horizontal and vertical axes are scaled together by the same ratio which equals $(4/\lambda)$.

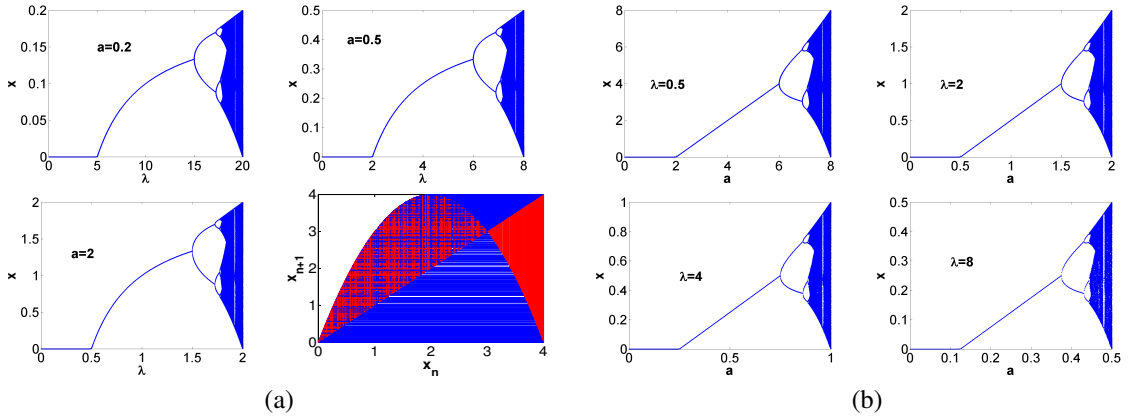


Figure 3.16: (a) Bifurcation diagram versus λ for different values of $a = \{0.2, 0.5, 2\}$ and Cobweb plot at $a = 4, \lambda = 1$, (b) Bifurcation diagram versus a for different values of $\lambda = \{0.5, 2, 4, 8\}$ for zooming positive logistic map

3.2.3.2 Mostly Positive Logistic Map $f(x, \lambda, a) = -\lambda x(a - x)$

3.2.3.2.1 Range of λ Substituting $b = 1$ in (3.41b) yields:

$$(\lambda_{max}, x_{min}, x_{max}) = \left(\frac{2}{a}, -\frac{a}{2}, \frac{3a}{2} \right), \quad (3.49a)$$

$$a \in (0, a_{max}] \text{ where } a_{max} = \frac{2}{\lambda}. \quad (3.49b)$$

3.2.3.2.2 Fixed Points and Stability Condition Substituting $b = 1$ in (3.42) yields:

$$(x_b, \lambda_b) = \left(0, \frac{1}{a} \right). \quad (3.50)$$

3.2.3.2.3 Steady State Solutions Versus System Parameters Figure 3.17(a) shows the bifurcation diagram versus λ for different values of a for zooming mostly positive logistic map. The diagrams are quite similar to those in the case of zooming positive logistic map in having constant area axes. Figure 3.17(a) also shows the cobweb plot at $a = 4$ and $\lambda = 0.5$, which shows chaotic behavior where the whole range of x is covered.

Figure 3.17(b) shows the bifurcation diagram versus a for different values of λ for mostly positive map. Similar to positive map, the horizontal and vertical axes are scaled together by the same ratio which equals $(2/\lambda)$ for mostly positive map. Yet, they don't form a square-axis due to its alternating sign nature where $x \in [-1/\lambda, 3/\lambda]$. Instead, the length of the vertical axis is double that of the horizontal one, but extending to the negative part asymmetrically. Therefore, the bifurcation diagrams with respect to λ are of equal area for both zooming maps. However, the bifurcation diagrams with respect to a are of equal axes for the positive map, and a vertical axis double the horizontal for the mostly positive map. This enriches the map characteristics and raises its degrees of freedom to be used in control applications according to the required constraints.

Figure 3.18 shows snapshots of the bifurcation diagrams versus λ at different values of a for both zooming maps. Figure 3.19 shows the other type of bifurcation diagram which is plotted versus a at different values of λ for both maps. The snapshots reflect the inverse proportionality relation between λ and a , where as a increases, the value of λ_{max} decreases and vice versa. Similarly, as λ increases, the value of a_{max} decreases and vice versa.

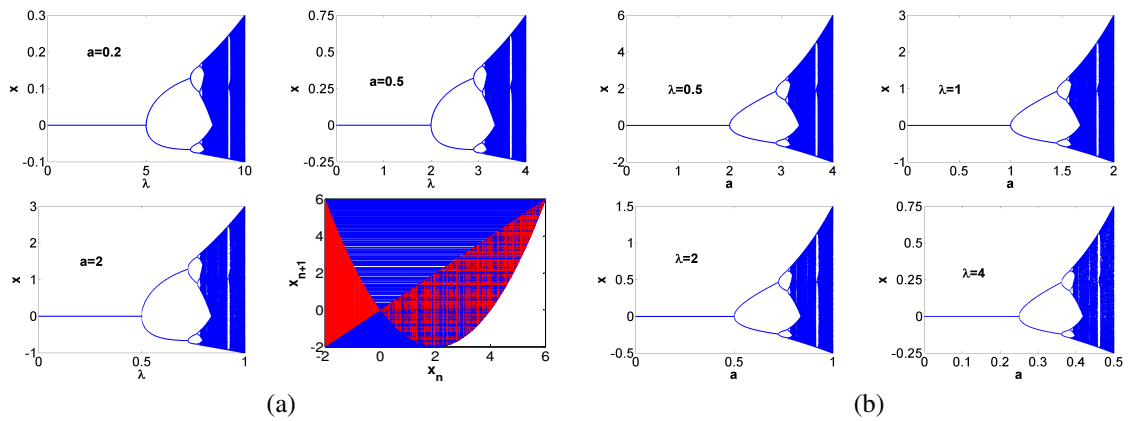


Figure 3.17: (a) Bifurcation diagram versus λ for different values of $a = \{0.2, 0.5, 2\}$ and Cobweb plot at $a = 4, \lambda = 0.5$, (b) Bifurcation diagram versus a for different values of $\lambda = \{0.5, 1, 2, 4\}$ for zooming mostly positive logistic map

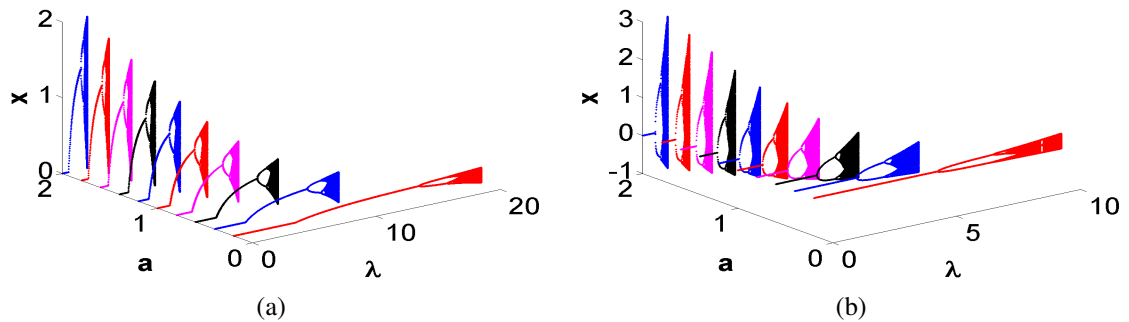


Figure 3.18: Ten different snapshots of the bifurcation diagram versus λ for zooming (a) positive map and (b) mostly positive map at $a = \{0.2, 0.4, \dots, 2\}$

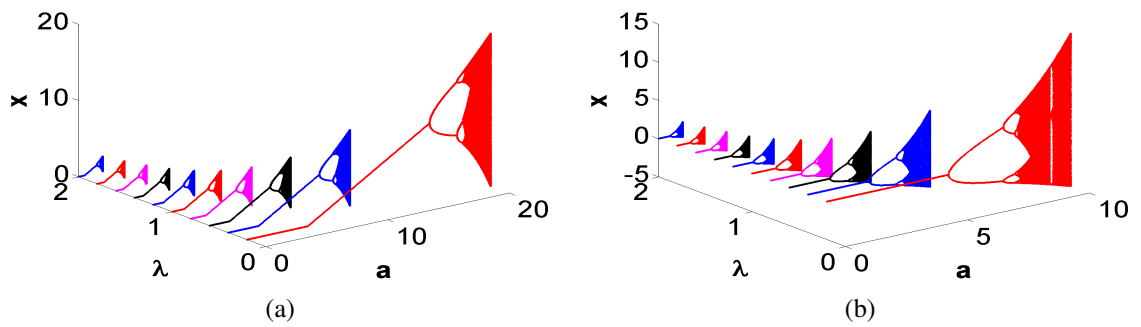


Figure 3.19: Ten different snapshots of the bifurcation diagram versus a for zooming (a) positive map and (b) mostly positive map at $\lambda = \{0.2, 0.4, \dots, 2\}$

3.2.3.3 Maximum Lyapunov Exponent

Figure 3.20(a) shows 3D plots of MLE as a function of both λ and a for zooming positive logistic map, while Fig. 3.20(b) shows them for zooming mostly positive logistic map. This continuous surface plot illustrates the dependence of the allowed range of λ on the value of a for both zooming positive, and mostly positive logistic maps. This fact is further indicated by discrete snapshots of MLE as a function of λ for different values of a . However, the value of MLE remains the same for maximum chaotic behavior, or at λ_{max} . The same steady state value $MLE = \ln 2$ is obtained which equals that of the conventional logistic map as previously explained.

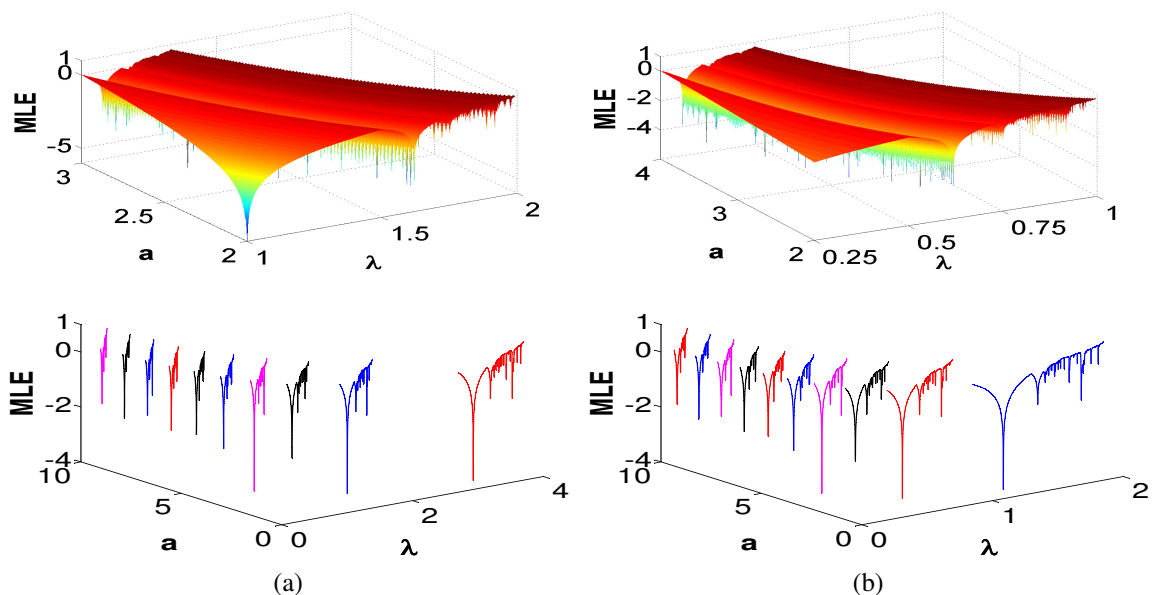


Figure 3.20: MLE as a function of both λ and a for zooming (a) positive logistic map and (b) mostly positive logistic map

3.3 General Schematic of Logistic Bifurcation Diagrams

Recalling the analyses and discussions throughout this paper and combining them with the results obtained in the last section, we can sketch a generalized schematic for the bifurcation diagrams. The key-points of the two different bifurcation diagrams versus the system parameters λ and a for both independent scaling positive, and mostly positive maps can be summarized as follows.

3.3.1 Positive Logistic Map $x_{n+1} = \lambda x_n(a - bx_n)$

3.3.1.1 Bifurcation Diagram Versus λ

The values of λ_b for the first and second bifurcation points are $(1/4)$ and $(3/4)$ of λ_{max} respectively, where $\lambda_{max} = 4/a$. Moreover, the value of the first nontrivial bifurcation point x_{b2} equals $(2/3)$ of the maximum value x_{max} , where $x_{max} = a/b$. Figure 3.21(a) shows the key-points of the bifurcation diagram versus λ in terms of a and b which conform to the results expected by our analysis.

3.3.1.2 Bifurcation Diagram Versus a

From another point of view, we can design the bifurcation diagram with respect to the parameter a where the bifurcation points and maximum values are shown in Fig. 3.21(b) in terms of the other parameters b and λ . The values of a at which bifurcation occurs a_b are $(1/4)$ and $(3/4)$ of a_{max} , where $a_{max} = 4/\lambda$. Moreover, the value of the first nontrivial bifurcation point x_{b2} equals $(1/2)$ the maximum value x_{max} , where $x_{max} = 4/(\lambda b)$.

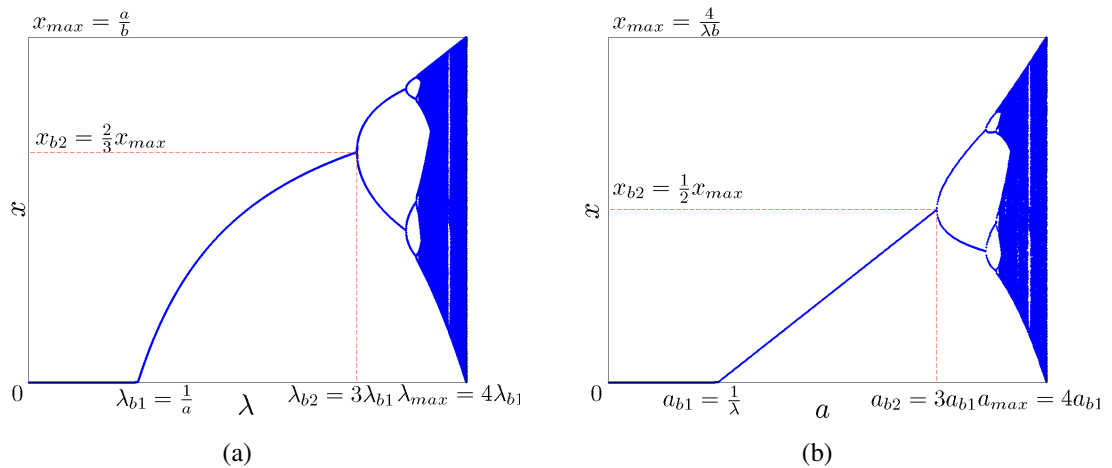


Figure 3.21: General bifurcation diagrams of independent scaling positive logistic map (a) versus λ and (b) versus a

3.3.2 Mostly Positive Logistic Map $x_{n+1} = -\lambda x_n(a - bx_n)$

3.3.2.1 Bifurcation Diagram Versus λ

Figure 3.22(a) shows the key-points of the bifurcation diagram versus λ in terms of a and b . The value of λ_b is (1/2) of λ_{max} , where $\lambda_{max} = 2/a$. The minimum and maximum values of x are $x_{min} = -a/(2b)$ and $x_{max} = (3a)/(2b)$ respectively.

3.3.2.2 Bifurcation Diagram Versus a

Figure 3.22(b) shows the key-points of the bifurcation diagram versus a in terms of b and λ . The value of a_b is (1/2) of a_{max} , where $a_{max} = 2/\lambda$. The minimum and maximum values of x are $x_{min} = -1/(\lambda b)$ and $x_{max} = 3/(\lambda b)$ respectively.

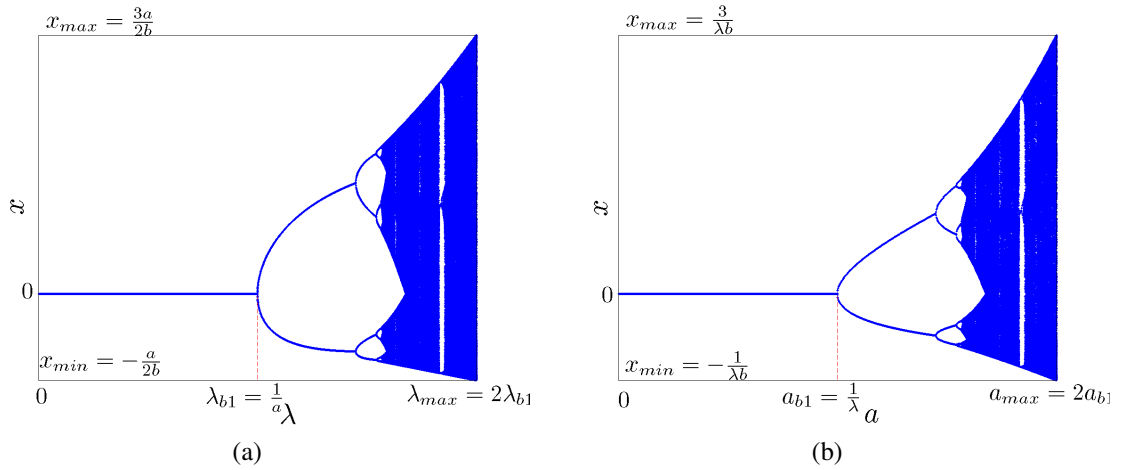


Figure 3.22: General bifurcation diagrams of independent scaling mostly positive logistic map (a) versus λ and (b) versus a

3.4 Logistic Map Design Procedure

Table 3.3 summarizes the results of independent scaling positive and mostly positive logistic maps. As previously detailed, the other two maps exhibit the same properties except for values of x that have the same magnitude, but opposite sign. Consequently, any required scaling can be achieved using the two extra parameters a and b , for bifurcation diagram versus λ . The horizontal scaling can be controlled by the parameter a , while the vertical scaling depends on the value (a/b) . To achieve horizontal scaling only while keeping the vertical axis with the values corresponding to the unity scaling case, the value of a should equal that of b . The bifurcation diagram versus a could be similarly designed and controlled by the values of λ and b . The remarkable advantage in the proposed variations on the discrete 1D logistic map, is not only about scaling or shifting, but also the increased complexity of the output that makes it more unpredictable, i.e., more convenient for applications that employ chaos.

Table 3.3: Comparison between the main aspects of the proposed generalized logistic maps

Map	$f_1(x) = \lambda x(a - bx)$ $a \in (0, \frac{4}{\lambda}], b \in R^+$	$f_2(x) = -\lambda x(a - bx)$ $a \in (0, \frac{2}{\lambda}], b \in R^+$
Roots	$x = 0, \frac{a}{b}$	
Critical points	$x_c = \frac{a}{2b}$	
Range of λ	$[0, \frac{4}{a}]$	$[0, \frac{2}{a}]$
Range of x	$[0, \frac{a}{b}]$	$[-\frac{a}{(2b)}, \frac{(3a)}{(2b)}]$
Bifurcation vs. λ	$\lambda_{b1} = \frac{1}{a}, x_{b1} = 0$	
	$\lambda_{b2} = \frac{3}{a}, x_{b2} = \frac{2a}{3b}$	-
Bifurcation vs. a	$a_{b1} = \frac{1}{\lambda}, x_{b1} = 0$	
	$a_{b2} = \frac{3}{\lambda}, x_{b2} = \frac{2}{\lambda b}$	-

3.4.1 Design Examples

The required specifications on the bifurcation diagram could be realized with respect to one of the parameters λ or a . Both realizations are considered, we denote the value of the parameter at which first, or second, bifurcation occurs as par_{b1} and par_{b2} respectively, and the maximum value of the parameter as par_{max} ; where par could be λ or a . Let us also define $extrema(x)$ such that:

$$extrema(x) = \begin{cases} x_{max}, & \text{for positive logistic map} \\ x_{min}, & \text{for negative logistic map} \end{cases} \quad (3.51)$$

The design problem involves constraints which are: the values of par_{b1} and par_{max} , in addition to par_{b2} if exist, i.e., in the case of positive or negative logistic maps, are related. For single sign maps, $par_{max} = 4par_{b1}$ and $par_{b2} = 3par_{b1}$, while for alternating sign maps $par_{max} = 2par_{b1}$. In addition, the values of x_{min} and x_{max} , in the case of alternating sign maps, are related. For the bifurcation diagram with respect to λ , $x_{max} - x_{min} = 2a/b$, while for the bifurcation diagram with respect to a , $x_{max} - x_{min} = 4/(\lambda b)$. For mostly positive map, $x_{max} = -3x_{min}$, while for mostly negative map, $x_{max} = -(1/3)x_{min}$. Table 3.4 summarizes these constraints.

Table 3.4: Constraints of the design problem

	Single sign maps	Alternating sign maps
Constraints on parameter	$par_{max} = 4par_{b1}$ $par_{b2} = 3par_{b1}$	$par_{max} = 2par_{b1}$ -
Constraints on output	λ -bifurcation a -bifurcation	$ extrema(x) = a/b$ $ extrema(x) = 4/(\lambda b)$
		$x_{max} - x_{min} = 2a/b$ $x_{max} - x_{min} = 4/(\lambda b)$

Four different cases of designing a map with certain specifications on the bifurcation diagram versus either parameter are provided. The design examples have been chosen such that they cover the cases of horizontal and independent scaling, since vertical scaling and zooming maps have been discussed thoroughly. First, the suitable map is chosen according to the required range of output: positive only, negative only, or alternating sign.

Then, the design equations are solved simultaneously to get the corresponding values of the other parameters. The bifurcation diagrams, their main key-points, and the map equation are shown in Table 3.5.

One of the applications of chaotic maps is generation of sequences of random numbers which are useful in many applications. These are called pseudo-random sequences, i.e., although the output appears to be random, it is actually generated deterministically based on a seed value. Seeds are useful when re-producible sequences are desired, e.g., debugging a simulation or encrypting/decrypting a message. Mixing the message with the chaotic sequence achieves desirable cryptographic properties of diffusion and confusion due to the high sensitivity of this sequence to changes in initial conditions. The output pattern after mixing is called the ciphertext which is an encrypted combination of the map selected by the key and the plaintext, where the key is a secret and the plaintext is the message. Moreover, successive iterations of a chaotic system reduce the statistical dependency of the ciphertext on the plaintext. Such applications require certain specifications on the bifurcation diagram in terms of the parameter's range, in addition to the output range. The encryption system used to validate our designs is explained as follows.

3.4.2 Encryption System

The encryption scheme used is based on simple xor operation between the output sequence of the map and the characters constituting the plain text as shown in Fig. 3.23. Moreover,

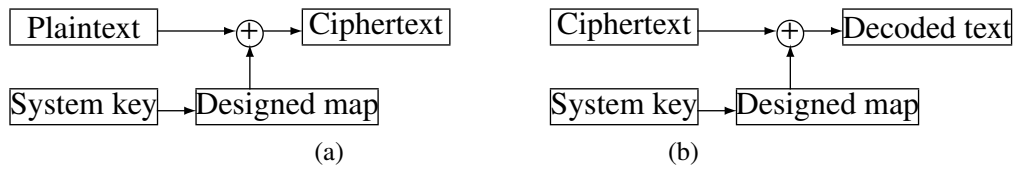


Figure 3.23: Simple text encryption system (a) Encryption scheme and (b) Decryption scheme

the extra parameters a and b in addition to the system parameter λ and the initial condition x_0 can be used to construct a more efficient encryption key. The key consists of four parameters each represented in 32 bits, i.e., a total of 128 bits as shown in Fig. 3.24. The calculation of each parameter S and initial value x_0 are given by:

$$S = S_{fix} - S_{key} \times 10^{-12} \quad (3.52a)$$

$$x_0 = \pm x_{key} \times 10^{-10} \quad (3.52b)$$

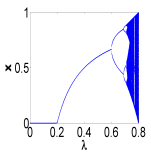
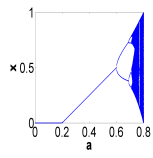
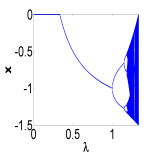
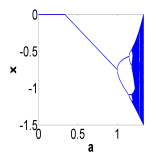
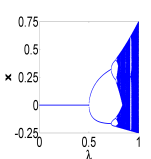
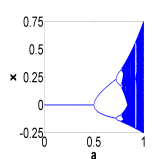
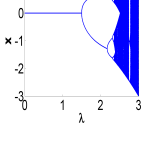
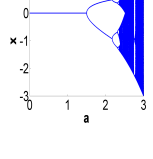
where S_{fix} is the fixed part of this parameter, S_{key} is the decimal value of the corresponding 32 bits shown in Fig. 3.24 and x_{key} is the decimal value given in the key. The scaling factors $10^{-12} \approx 2^{-40}$ and $10^{-10} \approx 2^{-34}$ are used after converting to double precision floating-point to ensure that the value of S_{key} does not affect the digits of S_{fix} . The sign of x_0 is chosen to limit the initial values within the allowable range according to the map characteristics (single or alternating sign map). In order to enhance the characteristics of the system, the double precision floating-point representation of each map output is subdivided into eight

Table 3.5: Four design cases of the logistic map

(a)

Example	Specifications	Suitable Map	λ -bifurcation Design	a -bifurcation Design
(a)	$x \geq 0, x_{max} = 1$ $par_{b1} = 0.2$	Positive	$a = 5, b = 5$	$\lambda = 5, b = \frac{4}{5}$
(b)	$x \leq 0, x_{min} = -3/2$ $par_{max} = 4/3$	Negative	$a = 3, b = 2$	$\lambda = 3, b = \frac{8}{9}$
(c)	$x_{min} = -0.25, \frac{x_{max}}{x_{min}} > 1$ $par_{max} = 1$	Mostly Positive	$a = 2, b = 4$	$\lambda = 2, b = 2$
(d)	$x_{max} = 1, \frac{x_{max}}{x_{min}} < 1$ $par_{b1} = 1.5$	Mostly Negative	$a = \frac{2}{3}, b = \frac{1}{3}$	$\lambda = \frac{2}{3}, b = \frac{3}{2}$

(b)

	Bifurcation Diagram w.r.t. λ	Bifurcation Diagram w.r.t. a
(a)	 $f(x) = \lambda x(5 - 5x)$ $\lambda_{b1} = 0.2$ $\lambda_{b2} = 0.6$ $\lambda_{max} = 0.8$ $x_{b2} = \frac{2}{3}$ $x_{max} = 1$ <p>Horizontal Scaling</p>	 $f(x) = 5x(a - \frac{4}{5}x)$ $a_{b1} = 0.2$ $a_{b2} = 0.6$ $a_{max} = 0.8$ $x_{b2} = 0.5$ $x_{max} = 1$ <p>Horizontal Scaling</p>
(b)	 $f(x) = \lambda x(3 + 2x)$ $\lambda_{b1} = \frac{1}{3}$ $\lambda_{b2} = 1$ $\lambda_{max} = \frac{4}{3}$ $x_{b2} = -1$ $x_{min} = -1.5$ <p>Independent Scaling</p>	 $f(x) = 3x(a + \frac{8}{9}x)$ $a_{b1} = \frac{1}{3}$ $a_{b2} = 1$ $a_{max} = \frac{4}{3}$ $x_{b2} = -0.75$ $x_{min} = -1.5$ <p>Independent Scaling</p>
(c)	 $f(x) = -\lambda x(2 - 4x)$ $\lambda_{b1} = 0.5$ $\lambda_{max} = 1$ $x_{min} = -0.25$ $x_{max} = 0.75$ <p>Independent Scaling</p>	 $f(x) = -2x(a - 2x)$ $a_{b1} = 0.5$ $a_{max} = 1$ $x_{min} = -0.25$ $x_{max} = 0.75$ <p>Independent Scaling</p>
(d)	 $f(x) = -\lambda x(\frac{2}{3} + \frac{1}{3}x)$ $\lambda_{b1} = 1.5$ $\lambda_{max} = 3$ $x_{min} = -3$ $x_{max} = 1$ <p>Independent Scaling</p>	 $f(x) = -\frac{2}{3}x(a + \frac{3}{2}x)$ $a_{b1} = 1.5$ $a_{max} = 3$ $x_{min} = -3$ $x_{max} = 1$ <p>Independent Scaling</p>

blocks which are xored at first, then the output is xored with a character from the plain text, and so on till the last character. Table 3.6 shows the encrypted HEX code corresponding

x_{key}	a_{key}	b_{key}	λ_{key}
32 bits	32 bits	32 bits	32 bits

Figure 3.24: System key used in the encryption scheme

to each designed map, i.e., the fixed values of the 3 parameters are set to the values given in Table 3.5 with λ_{max} or a_{max} corresponding to maximum chaotic behavior. We use the famous quote by Ghandi: “You must be the change you wish to see in the world.” as the plaintext. The encryption key is chosen as “B93E61A2A2F49CB58EA37B51C49A5E68”. Encryption schemes that offer better performance utilizing such simple maps usually use innovative combinations of them. Such combinations could be even suitable for more complicated image encryption applications, e.g., the encryption techniques presented in [93, 116].

Table 3.6: Encrypted HEX code for each design given $(\lambda_{fix}, a_{fix}, b_{fix})$

Plaintext	You must be the change you wish to see in the world.
HEX Code	596F75206D75737420626520746865206368616E676520796F75 207769736820746F2073656520696E2074686520776F726C642E
Encrypted HEX code	
Design (a) w.r.t. λ (4/a, 5, 5)	8F3B4F398DF8F5E61E20E1EC2879EFF53C1F0F04AF90FEB6292F 06CBA9A1C2B13C942A84D28CDA033A15373587775A5675008032
Design (a) w.r.t. a (5, 0.8, 4/ λ)	8FBF3C41C6BD1FD8D4C269F8A0D731CA8DB0AD0E79D6D7FE5140 9E9B62B5A32F8747504B940DF967B99E45BFAE0EA1576A6EC3EB
Design (b) w.r.t. λ (4/a, 3, 2)	0F0B2C34DCEE02A639B528B5D3D78305A3C0F4B6A7E5F0B1157E 8134C06EF6E6C1250BA656916741DB0011D86C69E5F1C1A564F6
Design (b) w.r.t. a (3, 4/3, 4/ λ)	0FEC095719D3625223CD7941B2571AFE7E2B649BE901F76D81 DC77ADBEE5450129D9FF0D0B71B963F64C2FEED57AEDF3DCC2A3
Design (c) w.r.t. λ (2/a, 2, 4)	8FEC4A6CBDFC371B4183EA5FBE09B81217CAEAF015332F7CB87 E34E3B7F1F9287AEC651E015876E9BAD40BBC222BEC2A63803C3
Design (c) w.r.t. a (2, 1, 2/ λ)	8F919CFC52131AA8E987F7D4DC75133A1E81DD58317886F57EBB 5553A8AD90B643E204628793FC095E895248EF8452E8CFD8D1D4
Design (d) w.r.t. λ (2/a, 2/3, 1/3)	0F9A0D7FB4449B8F590E88D8248AF929CEC7EDAA7C1274AB83D1 C4A36A5C24B3A8B938984DE2B6D9B26251FF0F1B6D26A8FD13FE
Design (d) w.r.t. a (2/3, 3, 2/ λ)	0F710918D640BD50302A1ADE60BC4F0BFAC536C2811122A31BB6 B6A37A7C1DB5ADCFA7F31BAE0FA9F1B9F93F2CB127B9862841CF

3.5 Scaled Tent Map: Analysis and Results

3.5.1 Independent Scaling $x_{n+1} = \pm\mu \min(x_n, a - bx_n)$

3.5.1.1 Positive Tent Map

The independent scaling positive tent map proposed in [92] preserves the linearity of the two intersecting lines providing the possibility of designing an asymmetric scalable tent

shape and is given by:

$$x_{n+1} = f(x_n, \mu, a, b) = \mu \min(x_n, a - bx_n), \quad \mu, a, b \in R^+ \quad (3.53a)$$

$$x_{n+1} = \begin{cases} \mu x_n & x_n \leq x_k \\ \mu(a - bx_n) & x_k < x_n \end{cases}, \quad (3.53b)$$

where

$$x_k = \frac{a}{1+b} \quad (3.54)$$

is the point of intersection of the two lines. We present here the general forms of the bifurcation diagrams of independent scaling positive tent map versus the main system parameter μ , in addition to the corresponding diagram for mostly positive tent map. Special scaling cases and system response versus the other parameters can also be studied similar to the case of logistic map.

3.5.1.1.1 Parameters' Ranges The solution should belong to the interval $x \in [0, a/b]$ to guarantee boundedness.

$$(\mu_{max}, x_{max}) = \left(1 + \frac{1}{b}, \frac{a}{b}\right), \quad (3.55a)$$

$$b \in (0, b_{max}] \text{ where } b_{max} = \frac{1}{\mu - 1}. \quad (3.55b)$$

3.5.1.1.2 Fixed Points and Their Stability The fixed points are given by

$$x_1^* = 0 \text{ and } x_2^* = \frac{a\mu}{1+b\mu}. \quad (3.56)$$

The value of μ at which the system bifurcates and the region of trivial fixed point ends, in addition to its corresponding function value are

$$(\mu_{b1}, x_{b1}) = (1, 0). \quad (3.57)$$

For $0 < b < 1$, a region of non-trivial fixed point appears after which the response bifurcates to a period-2 solution given by

$$(\mu_{b2}, x_{b2}) = \left(\frac{1}{b}, \frac{a}{2b}\right). \quad (3.58)$$

Figure 3.25 shows that the impact of parameters is reversed in the case of independent scaling positive tent map compared to logistic map(s). The value of μ_{max} depends only on the parameter b . For $\mu > 0$, as $b < 1$ there exists a nontrivial fixed-point solution for a range of μ before other types of solutions start. This phase does not exist for $b > 1$.

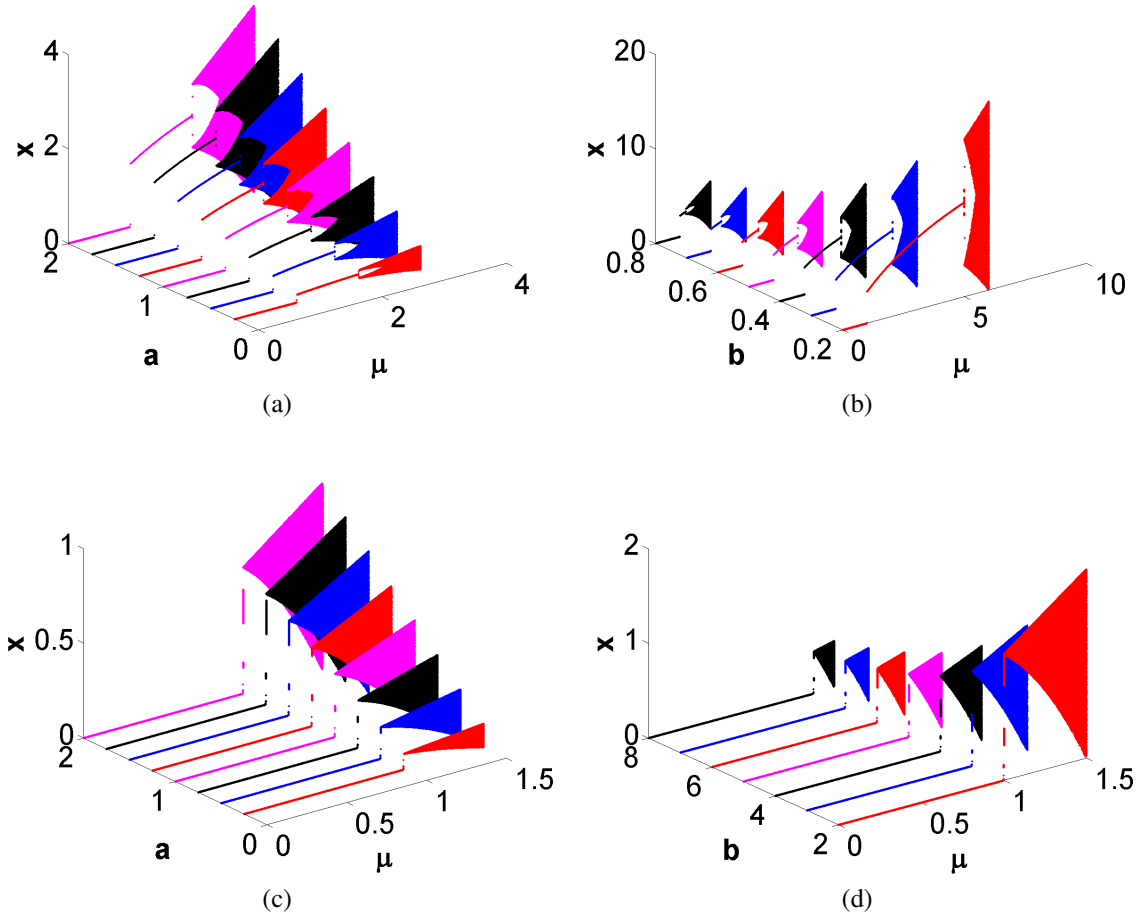


Figure 3.25: Bifurcation diagram vs. μ for independent scaling positive tent map at (a) $b = 0.5$ and $a = \{0.25, 0.5, \dots, 2\}$, (b) $a = 4$ and $b = \{0.2, 0.3, \dots, 0.8\}$, (c) $b = 2$ and $a = \{0.25, 0.5, \dots, 2\}$, and (d) $a = 4$ and $b = \{2, 3, \dots, 8\}$

3.5.1.2 Mostly Positive Tent Map

The proposed generalized mostly positive tent map is given by

$$x_{n+1} = f(x_n, \mu, a, b) = -\mu \min(x_n, a - bx_n), \quad \mu, a, b \in \mathbb{R}^+ \quad (3.59a)$$

$$x_{n+1} = \begin{cases} -\mu x_n & x_n \leq x_k \\ -\mu(a - bx_n) & x_k < x_n \end{cases}, \quad (3.59b)$$

where

$$x_k = \frac{a}{1+b} \quad (3.60)$$

is the point of intersection of the two lines.

3.5.1.2.1 Parameters' Ranges A similar analysis to that conducted for mostly positive logistic and tent maps yields the following parameters' ranges

$$(\mu_{max}, x_{min}, x_{max}) = \left(1 + \frac{1}{b}, -\frac{a}{b}, \frac{a}{b} \left(1 + \frac{1}{b} \right) \right). \quad (3.61)$$

3.5.1.2.2 Fixed Points and Their Stability

$$x_1^* = 0 \text{ and } x_2^* = -\frac{a\mu}{1-b\mu}, \quad (3.62)$$

$$f'(x^*) = \begin{cases} -\mu & x^* \leq x_k \\ +\mu b & x_k < x^* \end{cases}, \quad (3.63)$$

$$|f'(x^*)| = 1 \rightarrow \mu_b = 1, \quad (3.64)$$

$$(x_b, \mu_b) = (0, 1). \quad (3.65)$$

3.5.1.2.3 Periodic Points and Their Stability

$$f^2(x) = \begin{cases} -\mu(a + b\mu x) & x \leq x_{k1} \\ \mu^2 x & x_{k1} \leq x \leq x_k \\ \mu^2(a - bx) & x_k \leq x \leq x_{k2} \\ -\mu(a + b\mu(a - bx)) & x_{k2} \leq x \end{cases}, \quad (3.66)$$

where $x_{k1} = -\frac{ab}{(1+b)^2}$, $x_k = \frac{a}{1+b}$, and $x_{k2} = \frac{a}{b} \left(1 + \frac{1}{(1+b)^2}\right)$.

Equating $x_p = f^2(x_p)$

$$\begin{cases} x_p = -\frac{a\mu}{1+b\mu^2} & x_p \leq x_{k1} \\ x_p = 0 & x_{k1} \leq x_p \leq x_k \\ x_p = \frac{a\mu^2}{1+b\mu^2} & x_k \leq x_p \leq x_{k2} \\ x_p = \frac{-a\mu-ab\mu^2}{1-b^2\mu^2} & x_{k2} \leq x_p \end{cases}. \quad (3.67)$$

To get the bifurcation point, $|(f^2)'(x_p)| = 1$

$$(f^2)'(x) = \begin{cases} -b\mu^2 & x \leq x_{k1} \\ \mu^2 & x_{k1} \leq x \leq x_k \\ -b\mu^2 & x_k \leq x \leq x_{k2} \\ b^2\mu^2 & x_{k2} \leq x \end{cases}. \quad (3.68)$$

Consequently, $\mu_{b2} = \frac{1}{\sqrt{b}}$, the corresponding values of x_b are $x_{b2} = -\frac{a}{2\sqrt{b}}$ and $x_{b2} = \frac{a}{2b}$.

$$(x_{b2}, \mu_{b2}) = \left(\left\{ -\frac{a}{2\sqrt{b}}, \frac{a}{2b} \right\}, \frac{1}{\sqrt{b}} \right). \quad (3.69)$$

Figure 3.26 shows the impact of parameters in the case of independent scaling mostly positive tent map. The value of μ_{max} depends only on the parameter b . Yet, the value of x_{max} also depends on μ_{max} not only the ratio a/b as in the case of independent scaling positive tent map. For $\mu < 0$, as $b < 1$ there exists a period-2 solution for a range of μ before orbits of longer lengths start. This phase does not exist for $b > 1$.

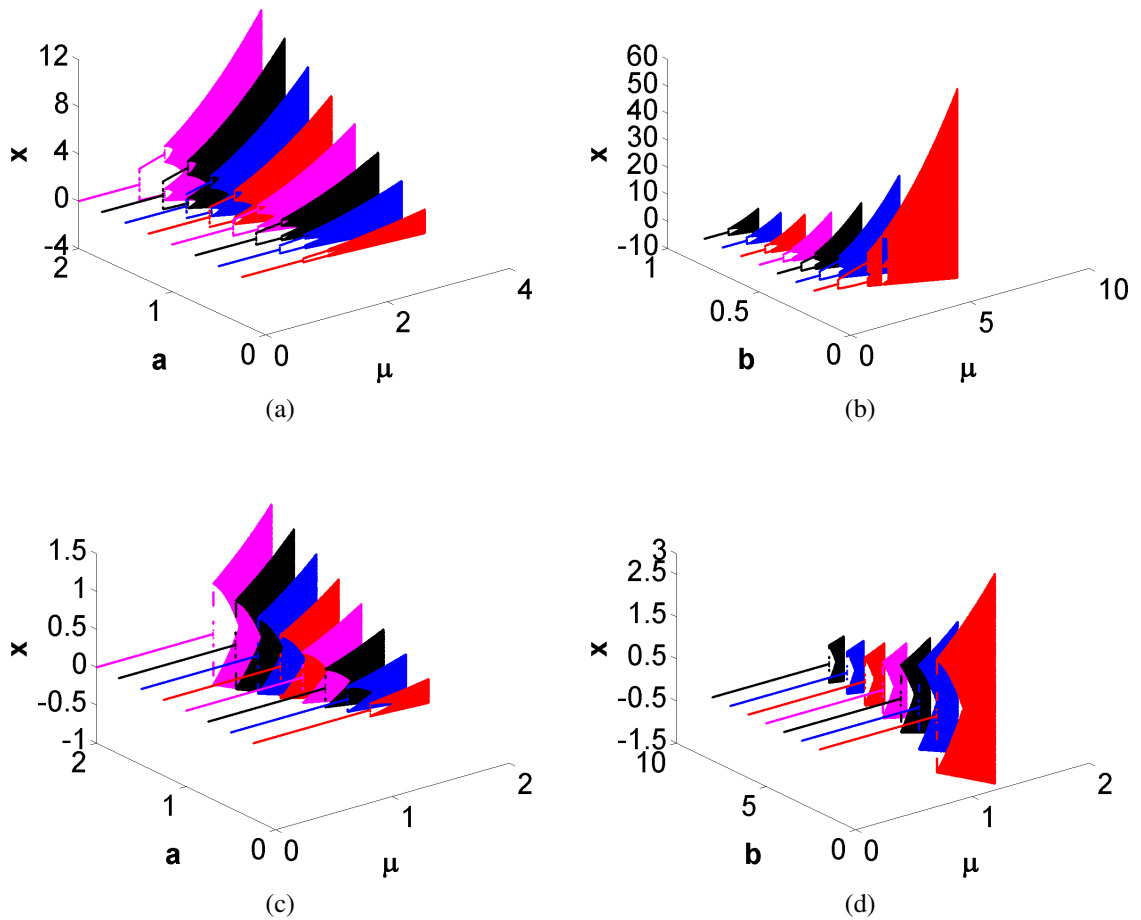


Figure 3.26: Bifurcation diagram vs. μ for independent scaling mostly positive tent map at (a) $b = 0.5$ and $a = \{0.25, 0.5, \dots, 2\}$, (b) $a = 2$ and $b = \{0.2, 0.3, \dots, 0.8\}$, (c) $b = 2$ and $a = \{0.25, 0.5, \dots, 2\}$, and (d) $a = 4$ and $b = \{2, 3, \dots, 8\}$

3.5.1.3 Maximum Lyapunov Exponent

For independent scaling positive tent map, unlike logistic map(s), the allowed range of μ depends on the value of the parameter b only irrespective of a . Figure 3.27(c) shows the values of MLE at μ_{max} in the $a-b$ plane where the largest value for MLE is obtained at $b = 1$ irrespective of the value of a , and equals $\ln 2$. Although this value slightly decreases for other values of b , it is still within the same positive range indicating chaotic behavior. Yet, the full range at maximum chaos depends on a/b similar to logistic map(s) as shown in Fig. 3.27(d).

For independent scaling mostly positive tent map, Fig. 3.28 shows similar measures. The lower bound on the full range at maximum chaos depends on a/b while the upper bound is directly proportional to both a/b and μ_{max} .

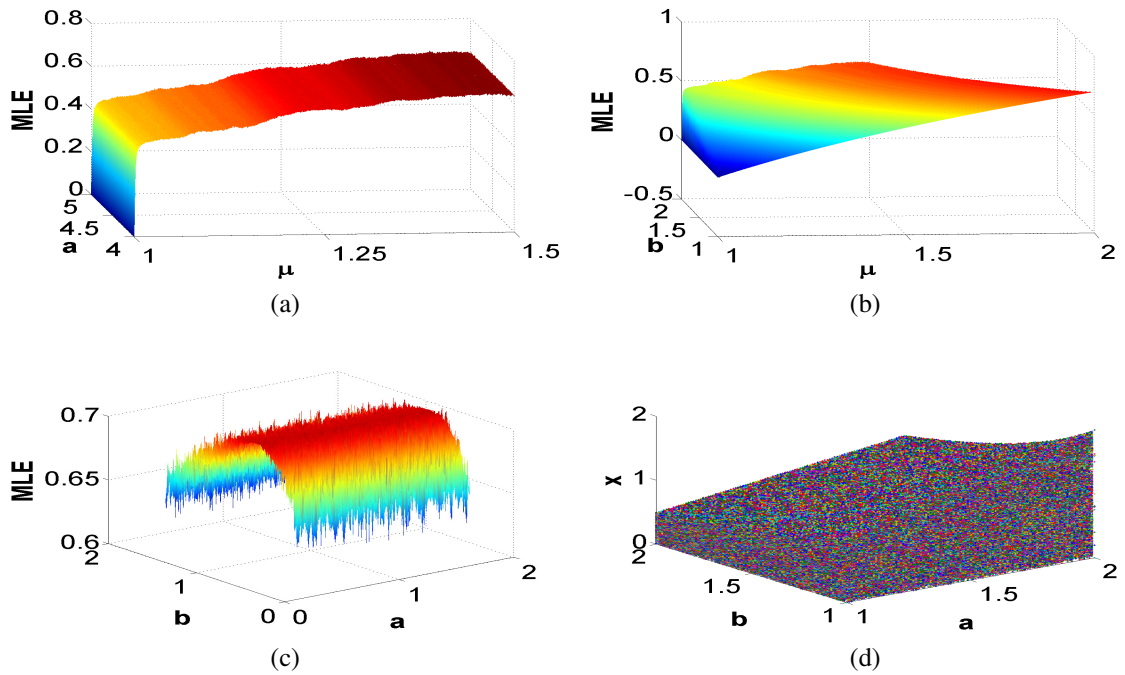


Figure 3.27: MLE of independent scaling positive tent map as a function of (a) μ and a at $b = 2$, (b) μ and b at $a = 4$, (c) a and b at μ_{max} , and (d) Full-range chaotic output versus a and b

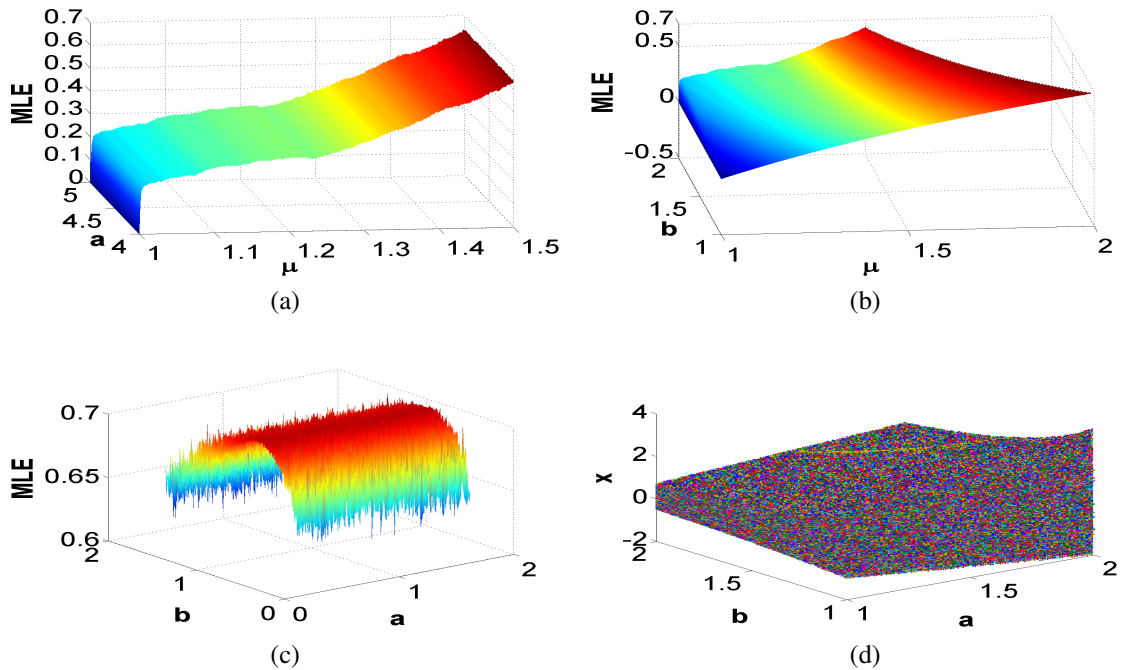


Figure 3.28: MLE of independent scaling mostly positive tent map as a function of (a) μ and a at $b = 2$, (b) μ and b at $a = 4$, (c) a and b at μ_{max} , and (d) Full-range chaotic output versus a and b

3.6 General Schematic of Tent Bifurcation Diagrams

Figure 3.29(a) shows the key-points of the bifurcation diagram versus μ in terms of a and b such that $b \leq 1$ for positive tent map which conform to the results expected by our analysis, while Fig. 3.29(b) shows them for $b > 1$. Figure 3.30 shows the general bifurcation diagrams for mostly positive tent map where $\mu_{max} = 1 + 1/b$. Setting $b = 1$ yields vertical scaling only.

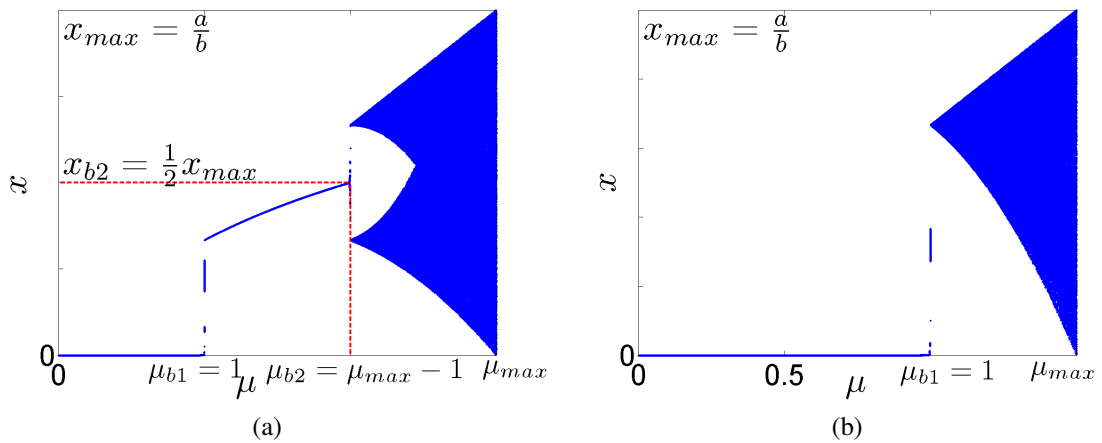


Figure 3.29: General schematic of the bifurcation diagram vs. μ of independent scaling positive tent map (a) $b \leq 1$ and (b) $b > 1$

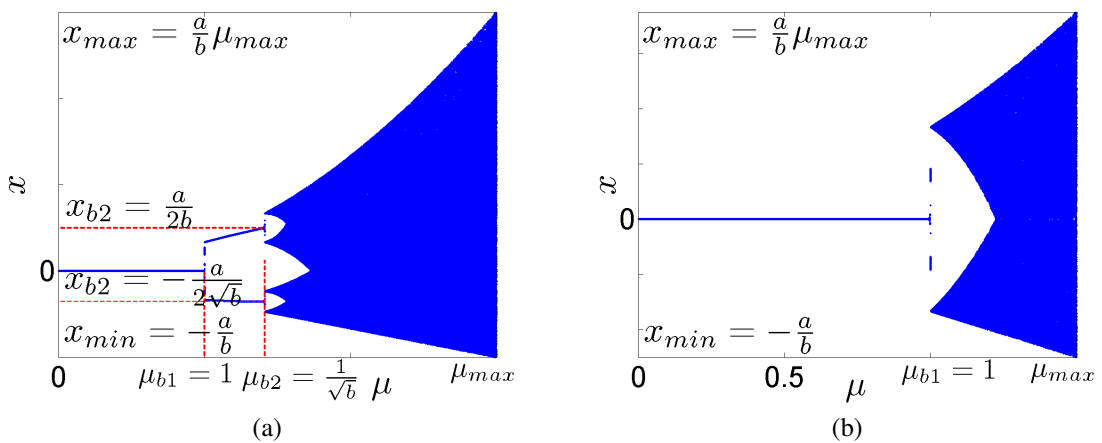


Figure 3.30: General schematic of the bifurcation diagram vs. μ of independent scaling mostly positive tent map (a) $b \leq 1$ and (b) $b > 1$

Chapter 4: General Powering Map

In the previous chapter, various generalizations were proposed on the mathematical relations representing the logistic and tent maps. Both are examples of conventional 1D discrete chaotic maps where the tent map is a piece-wise linear map, and the logistic map is a quadratic non-linear map. The generalizations allow multiple forms of bifurcation diagrams two-dimensional scaling. Special cases of one-dimensional scaling have also been illustrated. In addition, we have explored the behavior of these maps in a newly visited range of parameter which has not been investigated before; negatively controlled maps.

In this chapter, we propose a new map that could be considered a general form for 1D discrete maps employing the power function with the tent and logistic maps as special cases. It would be suitable even for maps whose iterative relations are not based on polynomials. A framework for analyzing the proposed map mathematically and predicting its behavior for various combinations of its parameters is introduced. In addition, the transition from tent map case to logistic map case is presented and explained. The possibility of generating real and imaginary bifurcations in this transition region in addition to the other regions is checked.

4.1 Motivation

The conjugacy between piecewise linear tent map and quadratic logistic map, the period doubling that both exhibit as a route to chaos, and the presence of bifurcation diagram for the newly explored range: negative parameter in both of them inspired us to propose a unifying map for 1D chaos generated by polynomials. Investigating the main differences between the equations defining both tent and logistic maps, we could notice:

- Logistic map is defined by a single relation, while tent map is defined by the minimum of two relations.
- The relation defining the logistic map is of degree-2, while those defining the tent map are of degree-1.

To get a unified relation that fits both maps, the first point could be overcome through redefining logistic map as follows:

$$x_{n+1} = \lambda \min(x_n(1-x_n), x_n(1-x_n)). \quad (4.1)$$

The second point is the key indicating that powering needs to be employed to get such unified relation. The tent map could be redefined as follows:

$$x_{n+1} = \mu \min(x_n(1-x_n)^0, x_n^0(1-x_n)). \quad (4.2)$$

Consequently, an exponent should appear for both terms x_n and $1-x_n$ to get an adaptable form suitable for both tent and logistic maps. The iterative form of the proposed map is given by

$$x_{n+1} = r \min(x_n^\alpha(1-x_n)^\beta, x_n^\beta(1-x_n)^\alpha), \quad (4.3)$$

where $\alpha, \beta \in R^+$. The main control parameter of the map r , alternatively called the system parameter, replaces λ or μ and it could have either positive or negative sign. Starting from this chapter, the sign of the main system parameter is no longer explicitly defined, and the bifurcation diagrams are plotted on the same figure for both positive and negative control parameter cases. The function plots for both r positive and negative at various values of α and β are shown in Fig. 4.1. Another way of defining the map could be as follows

$$f(x, r, \alpha, \beta) = r \min(x^\alpha (1-x)^\beta, x^\beta (1-x)^\alpha). \quad (4.4)$$

For $(\alpha, \beta) = (1, 0)$ or $(0, 1)$, the relation representing tent map is yielded as shown in Fig. 4.1(a). While for $(\alpha, \beta) = (1, 1)$, the relation representing the logistic map is yielded as shown in Fig. 4.1(b). This equation not only represents both tent and logistic maps, but it also provides the possibility of designing various new maps at different values of the powering parameters α and β , an example is shown in Fig. 4.1(c). These are called shaping parameters as they determine the shape of the map (triangular, parabolic, . . . , etc.), in addition to the corresponding bifurcation diagram. Moreover, two scaling parameters a and b could be added to maintain the scaling capability introduced in the previous chapter.

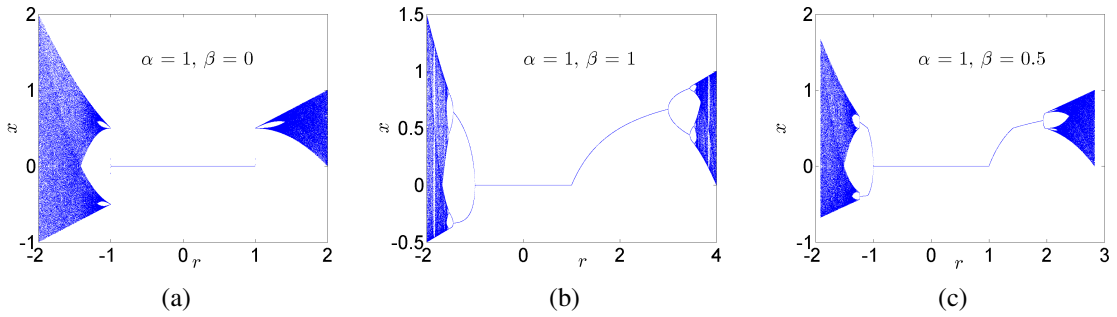


Figure 4.1: Bifurcation diagrams of general powering map for various values of α and β starting at initial point $x_0 = 0.01$ (a) $(\alpha, \beta) = (1, 0)$, (b) $(\alpha, \beta) = (1, 1)$, and (c) $(\alpha, \beta) = (1, 0.5)$

4.2 Proposed Map $x_{n+1} = r \min(x_n^\alpha (a - bx_n)^\beta, x_n^\beta (a - bx_n)^\alpha)$

4.2.1 Mathematical Analysis

A mathematical analysis for the general map with added scaling parameters a and b could be started as follows.

$$f(x, r, a, b, \alpha, \beta) = r \min(x^\alpha (a - bx)^\beta, x^\beta (a - bx)^\alpha), \quad (4.5)$$

where a, b, α , and $\beta \in R^+$.

The map is defined as selecting the minimum of the two functions, i.e., it could be plotted as two intersecting curves. To get the intersection point, we equate the equations of the two curves yielding

$$x_k = \frac{a}{b+1} \quad (4.6)$$

It is worth mentioning that in cases other than $b = 1$ another intersection point could exist in the case of $|\alpha - \beta|$ is an even number and equals $\frac{a}{b-1}$. Yet, this value lies outside the region of bounded solution in which we are interested as will be illustrated later on. It would be quite beneficial to redefine the map according to the obtained intersection point as follows:

$$\text{For } \alpha \geq \beta \quad f(x, r, a, b, \alpha, \beta) = \begin{cases} rx^\alpha (a - bx)^\beta & x \leq x_k \\ rx^\beta (a - bx)^\alpha & x_k < x \end{cases}, \quad (4.7a)$$

$$\text{For } \alpha < \beta \quad f(x, r, a, b, \alpha, \beta) = \begin{cases} rx^\beta (a - bx)^\alpha & x \leq x_k \\ rx^\alpha (a - bx)^\beta & x_k < x \end{cases}. \quad (4.7b)$$

The value $x_k = \frac{a}{b+1}$ is always true irrespective of the sign of the control parameter r or the values of α and β . However, equation (4.5) actually represents two cases according to the sign of the system parameter r . The rest of the analysis requires studying each case separately $r \geq 0$ and $r < 0$.

4.2.1.1 Positive Control Parameter Case $r \geq 0$

4.2.1.1.1 Parameters' Ranges For positive parameter case, substituting in either curves, the maximum allowable value of x_{n+1} is

$$f(x_k) = r_{max} \left(\frac{a}{b+1} \right)^{\alpha+\beta}. \quad (4.8)$$

The value of the output in the positive control parameter case is confined to the set $[0, a/b]$ to guarantee closed set responses. Hence, the maximum value of r that guarantees boundedness is

$$r_{max} = \frac{a}{b} \left(\frac{b+1}{a} \right)^{\alpha+\beta}. \quad (4.9)$$

Thus, defining

$$(r_{max}, x_{max}) = \left(\frac{a}{b} \left(\frac{b+1}{a} \right)^{\alpha+\beta}, \frac{a}{b} \right). \quad (4.10)$$

The latter equation means that the maximum output value in positive parameter case depends only on the values of the scaling parameters a and b and is independent of the shaping parameters α and β . The lower bound is always zero, while the upper bound equals a/b . Full range chaotic response could be increased through increasing a or decreasing b . In unity scaling case, the maximum allowable chaotic range, corresponding to r_{max} is $x \in [0, 1]$. On the other hand, the value of r_{max} itself depends on all parameters. It increases as the values of α or β increase, while its dependence on a and b is also related to the values of α and β .

4.2.1.1.2 Fixed Points and Their Stability The fixed points could be obtained through solving

$$\text{For } \alpha \geq \beta \quad x^* = \begin{cases} rx^{*\alpha} (a - bx^*)^\beta & x^* \leq x_k \\ rx^{*\beta} (a - bx^*)^\alpha & x_k < x^* \end{cases}, \quad (4.11a)$$

$$\text{For } \alpha < \beta \quad x^* = \begin{cases} rx^{*\beta} (a - bx^*)^\alpha & x^* \leq x_k \\ rx^{*\alpha} (a - bx^*)^\beta & x_k < x^* \end{cases}. \quad (4.11b)$$

which can be solved numerically when the values of the parameters α , β , a , and b are known to get the value of x^* at a given value of r . Hence, the bifurcation points and their stability could be studied from the first derivative equation given by

$$\text{For } \alpha \geq \beta \quad f'(x^*) = \begin{cases} r\alpha x^{*\alpha-1}(a-bx^*)^\beta - rb\beta x^{*\alpha}(a-bx^*)^{\beta-1} & x^* \leq x_k \\ r\beta x^{*\beta-1}(a-bx^*)^\alpha - rb\alpha x^{*\beta}(a-bx^*)^{\alpha-1} & x_k < x^* \end{cases}, \quad (4.12a)$$

$$\text{For } \alpha < \beta \quad f'(x^*) = \begin{cases} r\beta x^{*\beta-1}(a-bx^*)^\alpha - rb\alpha x^{*\beta}(a-bx^*)^{\alpha-1} & x^* \leq x_k \\ r\alpha x^{*\alpha-1}(a-bx^*)^\beta - rb\beta x^{*\alpha}(a-bx^*)^{\beta-1} & x_k < x^* \end{cases}, \quad (4.12b)$$

4.2.1.2 Negative Control Parameter Case $r < 0$

To determine the lower and upper bounds of the solution, we follow the same steps used for the case of logistic and tent maps with negative control parameter. Our analysis is based on the definition of the map in non-linear dynamics as a function whose domain and range are the same, where the proposed map could be plotted as shown in Fig. 4.2. The steps are detailed as follows, keeping in mind that $x_k = \frac{a}{b+1}$

1. The lower bound on the range of the map is $f(x_k)$ which is the same as the lower bound on the domain x_{min} .

$$f(x_k) = r_{min}(x_k)^{\alpha+\beta}. \quad (4.13)$$

2. The upper bound on the range is $f(x_{min})$, i.e., $f(f(x_k))$ which can be obtained by substitution in the left branch of the curve. So, the upper bound is equal to one of the following according to the values of the parameters

$$\text{For } \alpha \geq \beta \quad f(f(x_k)) = r_{min}(r_{min})^\alpha (x_k)^{\alpha(\alpha+\beta)} [a - br_{min}(x_k)^{\alpha+\beta}]^\beta, \quad (4.14a)$$

$$\text{For } \alpha < \beta \quad f(f(x_k)) = r_{min}(r_{min})^\beta (x_k)^{\beta(\alpha+\beta)} [a - br_{min}(x_k)^{\alpha+\beta}]^\alpha, \quad (4.14b)$$

which is a function of the parameters a, b, α, β in addition to r_{min} .

3. Then, we equate this value to the map equation given by (4.5) to get the corresponding solutions for the values of x : x_{min} and x_{max} .
4. The domain of the map is $D = [x_{min}, x_{max}]$, while its range is $R = [f(x_k), f(f(x_k))]$. We equate the lower and upper bounds of both intervals respectively to get r_{min} .

$$x_{min} = f(x_k), \quad (4.15a)$$

$$x_{max} = f(f(x_k)). \quad (4.15b)$$

5. Hence, we get the final values for x_{min} and x_{max} by substitution for r_{min} in their equations. Sometimes, we need to solve numerically to get their approximated values.

Fixed points and their stability could be calculated in a similar manner to that shown in the positive control parameter case.

The proposed 1D discrete map does not confine the powers α and β to be integers. They could belong to the fractional domain; in general $\alpha, \beta \in \mathbb{R}$, i.e., they could take any value belonging to the real field, but here we consider their values in \mathbb{R}^+ only. The possibility of chaos generation in these various cases of the parameters values shall be investigated later on. The analysis starts by the following map which represents the transition from the tent map to the logistic map.

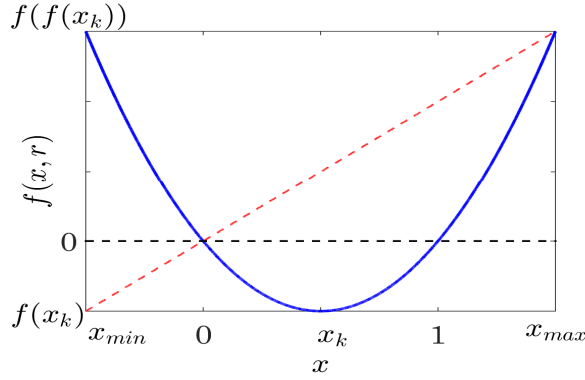


Figure 4.2: Domain and range of the proposed map with arbitrarily chosen parameters

4.3 Transition Map $(\alpha, \beta) = (1, 0) \rightarrow (1, 1)$

In this section, we go back to equation (4.4), i.e., unity scaling case. Setting either α or β to 1 while the other parameter equals zero yields the well-known equation of the conventional tent map. Similarly, setting $\alpha = \beta = 1$ yields the conventional logistic map. Assume $(\alpha, \beta) = (1, 0)$ corresponds to the tent map, the behavior as β increases from 0 to 1, till $(\alpha, \beta) = (1, 1)$ corresponding to the logistic map is investigated. The proposed 1D discrete map is given by

$$f(x, r, \beta) = r \min(x(1-x)^\beta, x^\beta(1-x)), \quad 0 < \beta < 1 \quad (4.16)$$

and could be called transition map as it represents the transition from tent map to logistic map, or alternatively called super-tent map. Figure 4.3(a) shows the curves of the map at different values of beta for $r > 0$ such that $r = r_{max}$, while Fig. 4.3(b) shows them for $r < 0$ such that $r = -1.9$. The curves show that every case for the transition map exhibits a single non-trivial fixed point. Figure 4.4(a) shows the continuous surface plot of the map equation $f(x, \beta)$ as a function of both x and β for $\beta \in [0, 1]$ at $r = (1.999)^{\beta+1}$ (a close value to r_{max}), while Fig. 4.4(b) shows it at $r = -1.9$ (a close value to r_{min}). Figure 4.4 illustrates how the behavior of the map changes gradually from tent map response to logistic map response. Figure 4.5 shows samples of the bifurcation diagrams for both positive and negative r for a set of values of the parameter β . The plots show how the bifurcation diagrams change gradually from the viewpoint of shapes and different key-points, from tent map response to logistic map response. The key-points include: r_{max} , r_{b1} , and x_{max} for $r \geq 0$ and r_{min} , r_{b1} , x_{min} , and x_{max} for $r < 0$.

4.3.1 Behavior From the Tent Map to the Logistic Map

It could come up to mind that complex values might be encountered when attempting to raise some expression, that may be negative for $r < 0$, to a fractional power. Yet, this is not the case in the transition region $(\alpha, \beta) = (1, 0) \rightarrow (1, 1)$, i.e., $\alpha = 1$ and $0 \leq \beta \leq 1$ as indicated by Fig. 4.3. We prove this property for the scaled transition map with the aid of equation 4.5 where two separate cases are discussed as follows

Proof. No complex results could be generated in the transition region $(\alpha, \beta) = (1, 0) \rightarrow (1, 1)$, i.e., $\alpha = 1$ and $0 \leq \beta \leq 1$.

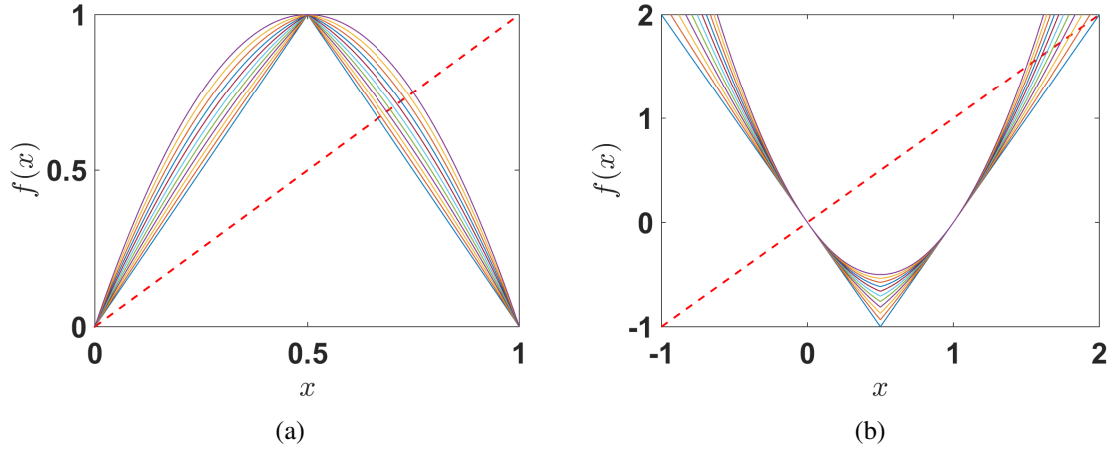


Figure 4.3: Curves of transition map $f(x, \beta)$ and their fixed points at $\beta = \{0, 0.1, \dots, 1\}$ (a) at $r = (2)^{\beta+1}$ and (b) at $r = -1.9$

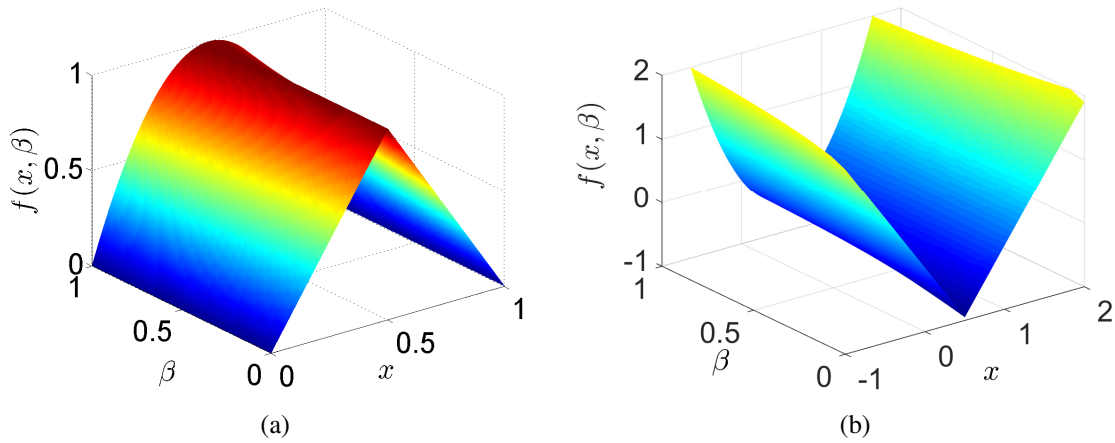


Figure 4.4: Surface plot of transition map $f(x, \beta)$ (a) at $r = (1.999)^{\beta+1}$ and (b) at $r = -1.9$

Case 1: $x_{min} \leq x_n < 0$. The minimum curve is $x_n^\alpha (a - bx_n)^\beta$ where $\alpha = 1$, i.e., the negative number is raised to the power of one.

Case 2: $\frac{a}{b} < x_n \leq x_{max}$. The minimum curve is $x_n^\beta (a - bx_n)^\alpha$ where $\alpha = 1$, i.e., the negative number is raised to the power of one. \square

Figure 4.6 shows the map and the cobweb plot at maximum chaotic behavior for various values of β and the corresponding r_{max} for $r \geq 0$ or r_{min} for $r < 0$. The plot shows that the map exhibits chaotic behavior, since the orbit has a non-periodic sequence and generates multiple outputs that cover the whole range of x . This range is the same for $r > 0$ and equals $[0, 1]$ in unity scaling case while it differs as the parameters change for $r < 0$.

4.3.2 Boundary Analysis

In this subsection, the results obtained through our general mathematical analysis are validated for the conventional tent and logistic maps in order to show their reliability.

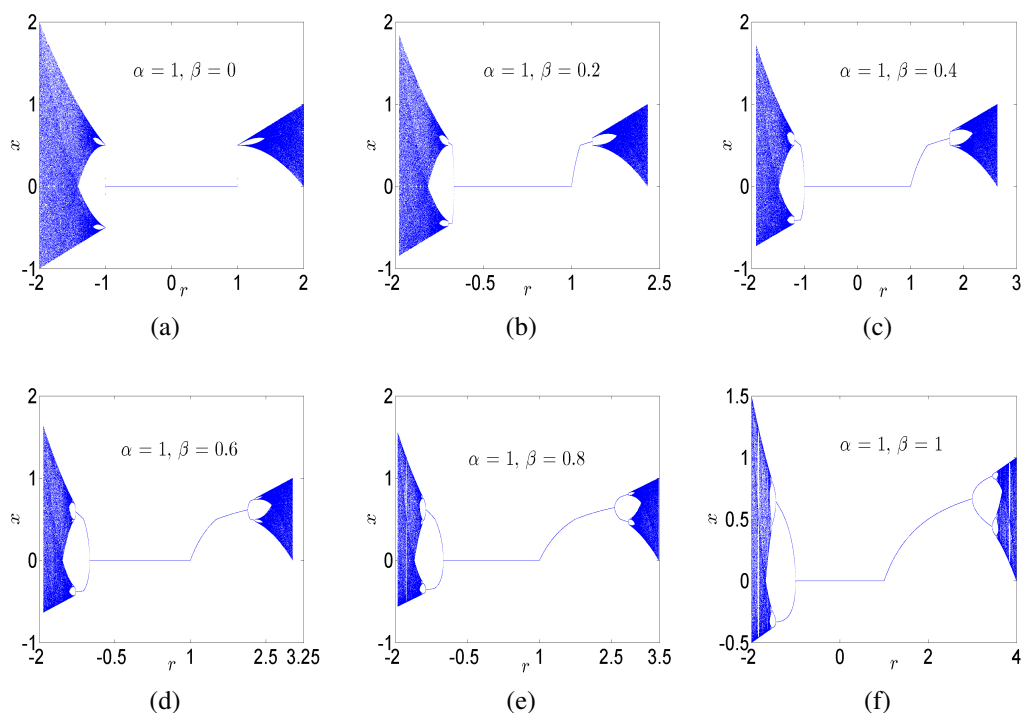


Figure 4.5: Bifurcation diagrams of the transition map for various values of β starting at initial point $x_0 = 0.01$

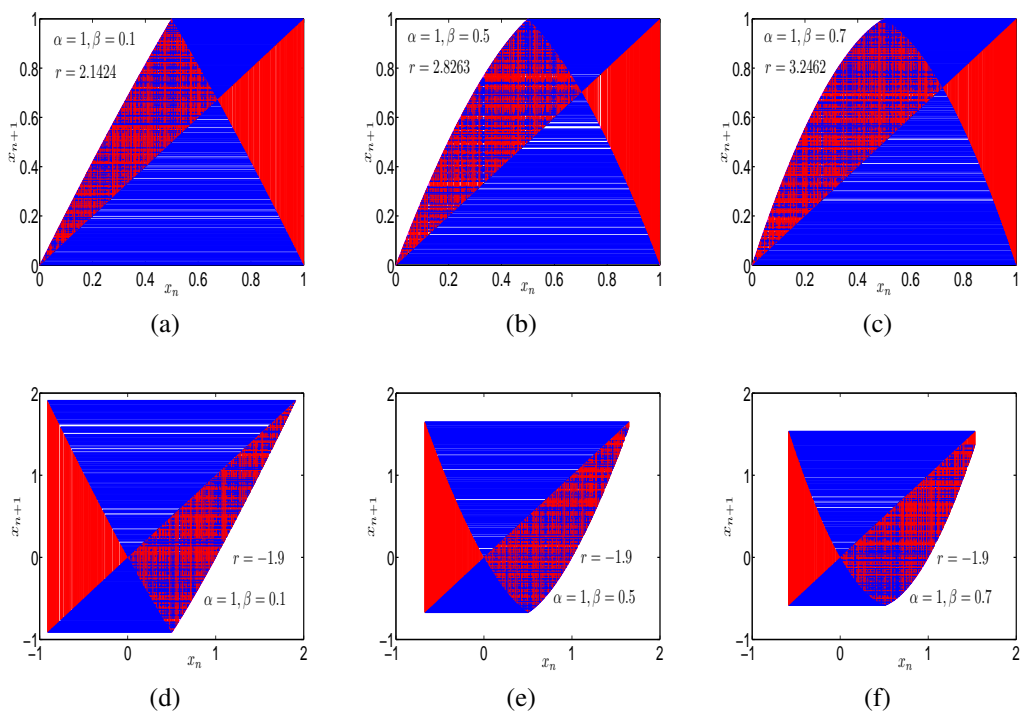


Figure 4.6: Cobweb plots of the proposed map at $\alpha = 1$ and various values of β and r starting at initial point $x_0 = 0.0005$

4.3.2.1 Tent Map

4.3.2.1.1 Positive Control Parameter Case $r \geq 0$ For the conventional tent map with positive control parameter, $(\alpha, \beta, a, b) = (1, 0, 1, 1)$. Thus, substituting in (4.10) we get

$$(r_{max}, x_{max}) = (2, 1), \quad (4.17)$$

which matches the correct values of parameters' range.

4.3.2.1.2 Negative Control Parameter Case $r < 0$ For the tent map with negative control parameter, $(\alpha, \beta, a, b) = (1, 0, 1, 1)$. Thus, from (4.6)

$$x_k = \frac{1}{2}. \quad (4.18)$$

Substituting in (4.14a) we get

$$f(f(x_k)) = r_{min}(r_{min})^1 \left(\frac{1}{2}\right)^1. \quad (4.19)$$

Solving to get x_{min} and x_{max}

$$\frac{r_{min}^2}{2} = \begin{cases} r_{min}x & x \leq x_k \\ r_{min}(1-x) & x_k < x \end{cases}. \quad (4.20)$$

Consequently,

$$x_{min} = \frac{r_{min}}{2}, \quad (4.21a)$$

$$x_{max} = 1 - \frac{r_{min}}{2}. \quad (4.21b)$$

The domain of the map $D = \left[\frac{r_{min}}{2}, 1 - \frac{r_{min}}{2}\right]$, while its range $R = \left[\frac{r_{min}}{2}, \frac{r_{min}^2}{2}\right]$. Coinciding the two intervals yields

$$1 - \frac{r_{min}}{2} = \frac{r_{min}^2}{2}. \quad (4.22)$$

Solving for r_{min} , two values are yielded. Yet, r_{min} must be negative. Thus,

$$r_{min} = -2, \quad (4.23a)$$

$$(r_{min}, x_{min}, x_{max}) = (-2, -1, 2), \quad (4.23b)$$

which matches the correct values of parameters' range.

4.3.2.2 Logistic Map

4.3.2.2.1 Positive Control Parameter Case $r \geq 0$ For the conventional logistic map with positive control parameter, $(\alpha, \beta, a, b) = (1, 1, 1, 1)$. Thus, substituting in (4.10) we get

$$(r_{max}, x_{max}) = (4, 1), \quad (4.24)$$

which matches the correct values of parameters' range.

4.3.2.2.2 Negative Control Parameter Case $r < 0$ For the logistic map with negative control parameter, $(\alpha, \beta, a, b) = (1, 1, 1, 1)$. Thus, from (4.6)

$$x_k = \frac{1}{2}. \quad (4.25)$$

The logistic map is a special case at which $\alpha = \beta$ where there is no difference between the left and right parts of the curve and it can be represented by a single parabolic equation. Substituting in either (4.14a) or (4.14b) we get

$$f(f(x_k)) = r_{min}(r_{min})^1 \left(\frac{1}{2}\right)^2 \left(1 - r_{min} \left(\frac{1}{2}\right)^2\right). \quad (4.26)$$

Solving to get x_{min} and x_{max}

$$\frac{r_{min}^2}{4} \left(1 - \frac{r_{min}}{4}\right) = r_{min}x(1-x). \quad (4.27)$$

Consequently,

$$x_{min} = \frac{r_{min}}{4}, \quad (4.28a)$$

$$x_{max} = 1 - \frac{r_{min}}{4}. \quad (4.28b)$$

The domain of the map $D = \left[\frac{r_{min}}{4}, 1 - \frac{r_{min}}{4}\right]$, while its range $R = \left[\frac{r_{min}}{4}, \frac{r_{min}^2}{4} \left(1 - \frac{r_{min}}{4}\right)\right]$. Coinciding the two intervals yields

$$1 - \frac{r_{min}}{4} = \frac{r_{min}^2}{4} - \frac{r_{min}^3}{16}. \quad (4.29)$$

Solving for r_{min} , three values are yielded. Yet, r_{min} must be negative. Thus,

$$r_{min} = -2, \quad (4.30a)$$

$$(r_{min}, x_{min}, x_{max}) = \left(-2, -\frac{1}{2}, \frac{3}{2}\right), \quad (4.30b)$$

which matches the correct values of parameters' range. The results for both tent and logistic maps through substitution in the general equations of the proposed map with the corresponding parameters values are summarized in Table 4.1. We define r_e as the ending value for r after which divergence occurs such that:

$$r_e = \begin{cases} r_{max} & r \geq 0 \\ r_{min} & r < 0 \end{cases}. \quad (4.31)$$

This analysis might give a key to discovering cases which do not exhibit a range in which the solution is bounded, where no real value for r in the allowable range could be found.

4.3.3 Key-points of Bifurcation Diagram in the Transition Region

Figure 4.7 shows the variations of various key-points of the bifurcation diagrams of the proposed map in the transition region from tent to logistic. It shows these values as functions of $\beta \in [0, 1]$. Both positive and negative control parameter cases are discussed with the aid of Fig. 4.5 and the previous analysis. Furthermore, eleven snapshots of the bifurcation diagrams are shown in Fig. 4.8 that illustrates the differences between them which are explained in detail in the following subsections.

Table 4.1: Parameters of tent and logistic maps

Parameter	Tent Map	Tent Map	Logistic Map	Logistic Map
	$r \geq 0$	$r < 0$	$r \geq 0$	$r < 0$
r_e	2	-2	4	-2
x_{min}	0	-1	0	-0.5
x_{max}	1	2	1	1.5

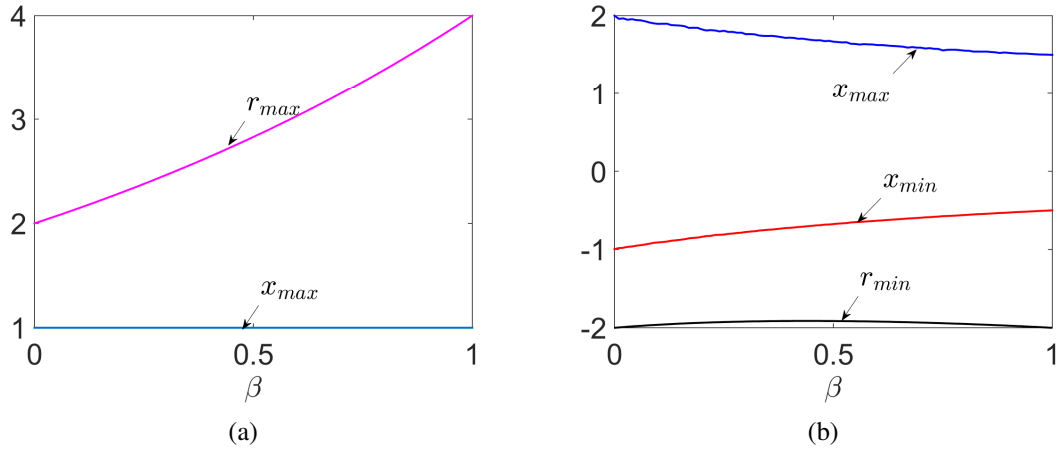


Figure 4.7: Key-points of bifurcation diagrams versus β at $\alpha = 1$ starting at $x_0 = 0.01$ (a) $r \geq 0$ and (b) $r < 0$

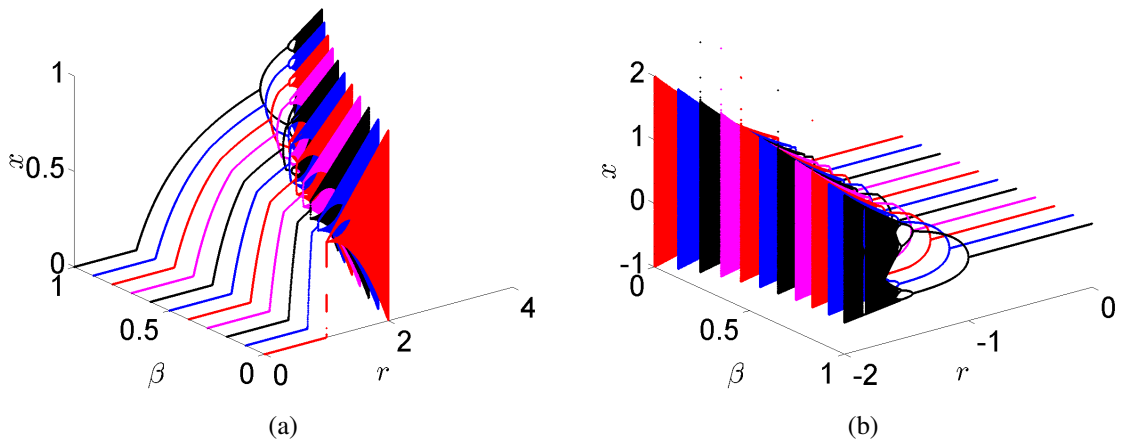


Figure 4.8: Eleven snapshots of the bifurcation diagrams of the proposed map at $\alpha = 1$ and $\beta = \{0, 0.1, \dots, 1\}$ starting at initial point $x_0 = 0.01$ (a) $r \geq 0$ and (b) $r < 0$

4.3.3.1 Positive Control Parameter Case $r \geq 0$

The value of the keypoint r_{max} of the bifurcation diagram for $r \geq 0$ as a function of $\beta \in [0, 1]$ is shown in Fig. 4.7(a). It starts from 2 in the case of tent map and increases to 4 in the case of logistic map. This could be explained by substitution with the corresponding values of

parameters $a = b = \alpha = 1$ in (4.9). Thus, r_{max} for the transition map is given by

$$r_{max} = (2)^{\beta+1} \quad (4.32)$$

which increases as β increases. The other basic key-points r_{b1} and x_{max} do not differ throughout this range of β (in addition to $x_{min} = 0$). However, the sudden change in the solution that occurred in tent map ($\beta = 0$) converts into a gradual change. Increasing the value of β , the sharp edge that exists in this range of r starts to get smoother until complete smoothness in the case of logistic map ($\beta = 1$). It should be noted that this sharp edge is neither the first nor the second bifurcation point that were obtained in our analysis in the previous chapter. The value of $r_{b1} = 1$ is fixed for all values of $\beta \in [0, 1]$, while r_{b2} value increases such that it starts to appear from 1 in the case of tent map to 3 in the case of logistic map.

4.3.3.2 Negative Control Parameter Case $r < 0$

The values of the key-points x_{min} and x_{max} of the bifurcation diagram for $r < 0$ as a function of $\beta \in [0, 1]$ are shown in Fig. 4.7(b). The value of x_{min} increases gradually as it starts from -1 in the case of tent map and increases to -0.5 in the case of logistic map. The value of x_{max} decreases gradually as it starts from 2 in the case of tent map and decreases to 1.5 in the case of logistic map. These values could be obtained numerically through the procedure discussed before to get the parameters' ranges for negative parameter case. This means that the output range shrinks as β increases throughout the interval $\beta \in [0, 1]$ at $\alpha = 1$ for $r < 0$. The value of $r_{b1} = -1$ is fixed for all values of $\beta \in [0, 1]$. For $r < 0$, r_{min} is not fixed as illustrated in Fig. 4.7(b), it slightly increases than -2 in the transition region preserving its analytical value $r_{min} = -2$ for both the tent and logistic maps.

4.3.4 Maximum Lyapunov Exponent (MLE)

For the transition map, the first derivative is given by

$$f'(x^*) = \begin{cases} r(1-x^*)^\beta - r\beta x^*(1-x^*)^{\beta-1} & x \leq x_k \\ r\beta x^{*\beta-1}(1-x^*) - rx^{*\beta} & x_k < x \end{cases} \quad (4.33)$$

The latter equation could be used to calculate MLE numerically, otherwise it can be calculated through forward difference formula that shall be introduced in Chapter 5 subsection 5.2.4 Figure 4.9(a) shows 3D plot of MLE as a function of both r and β for $r \geq 0$, while Fig. 4.9(b) shows it for $r < 0$. The surface plot shows that the value exhibited at maximum chaos (r_{max} for $r > 0$ and r_{min} for $r < 0$) for all transition maps approaches $\ln 2$ as the conventional case. This value occurs at $r_{max} = 2$ for tent map, $r_{max} = 2^\beta$ for transition map, and $r_{max} = 4$ for logistic map. On the other hand, r_{min} is equal to or slightly less than negative two.

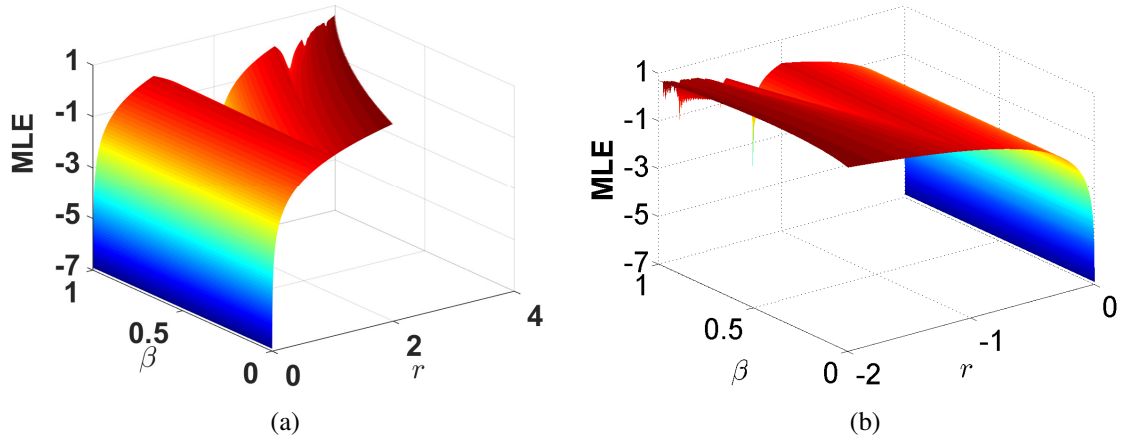


Figure 4.9: MLE as a function of both r and β for transition map (a) $r \geq 0$ and (b) $r < 0$

4.4 New Maps at Other Values for (α, β)

Several metrics have been used in the literature as indications of chaotic behavior. Among which are Lyapunov Exponent, Feigenbaum constant, Sharkovskii's Theorem, and others. The possibility of chaos generation through the proposed map for various combinations of α and β could be generally assessed through answering two main questions: is the response bounded? do bifurcations occur? For other values of β other than the interval $[0, 1]$, the case differs from the results in the previous sections where the bifurcation diagram might disappear on either side. Choosing an initial point to start with that already belongs to the bounded response interval should be considered too. In all the previous cases, the resulting value x was confined to the real field. However, other combinations of (α, β) yield complex results which can be viewed as two different bifurcation diagrams for the real and imaginary parts. The case when one of the two parameters α or β equals 1 while the other is in the range $[0, 1]$ has been discussed in the transition from tent to logistic map. The other alternatives are discussed in the following subsections for positive parameter case, where scaling parameters could be added for all cases.

4.4.1 Sub-Tent Map: $(\alpha, \beta) = (0, 0) \rightarrow (0, 1)$

This map resembles tent map where one of the shaping parameters equals zero while the other is less than one, so it is called sub-tent, i.e., below tent. Starting at $(\alpha, \beta) = (0, 0)$, the behavior as β increases from 0 to 1 till $(\alpha, \beta) = (0, 1)$ corresponding to the tent map is investigated. The proposed 1D discrete map is given by

$$f(x, r, \beta) = r \min(x^\beta, (1-x)^\beta), \quad 0 < \beta < 1 \quad (4.34)$$

In this range, not all values of the parameters yield bifurcations. In addition, the responses yielded are in general complex. Figure 4.10 shows samples of the real bifurcation diagrams for $r \geq 0$ for a set of values of the parameter β . It should be noted that the response is present for higher values of r , but exhibits monotonic and divergent behavior. Different initial points could cause the disappearance of some of the features of the bifurcation

diagram. The rate by which the responses diverge is worth consideration too. Nearly the same notes could describe the other cases discussed below.

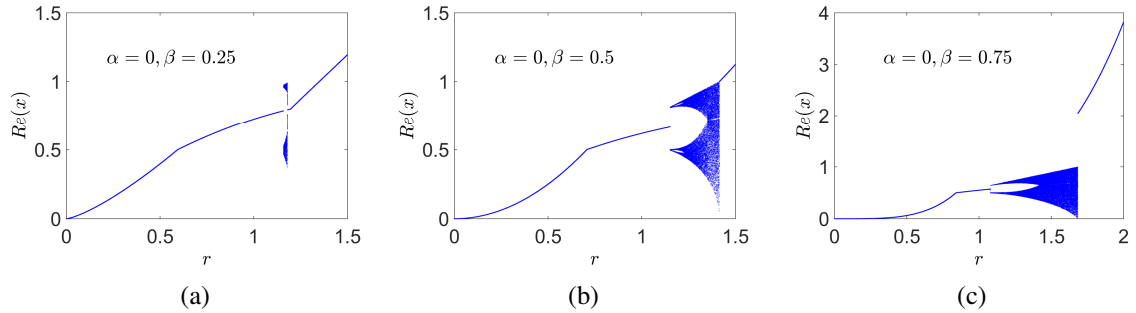


Figure 4.10: Real bifurcation diagrams of the sub-tent map for various values of β starting at initial point $x_0 = 0.05$

Figure 4.11 shows how the response moves gradually from a constant to tent map with curved surfaces in between. Figure 4.11(a) shows the curves of the sub-tent map ($\alpha = 0$) at different values of β for $r > 0$ such that $r = r_{max}$. The curves show that every case for the sub-tent map exhibits a single non-trivial fixed point. Figure 4.11(b) shows the continuous surface plot of the map equation $f(x, \beta)$ as a function of both x and β for $\beta \in [0, 1]$ at $r = (1.999)^\beta$.

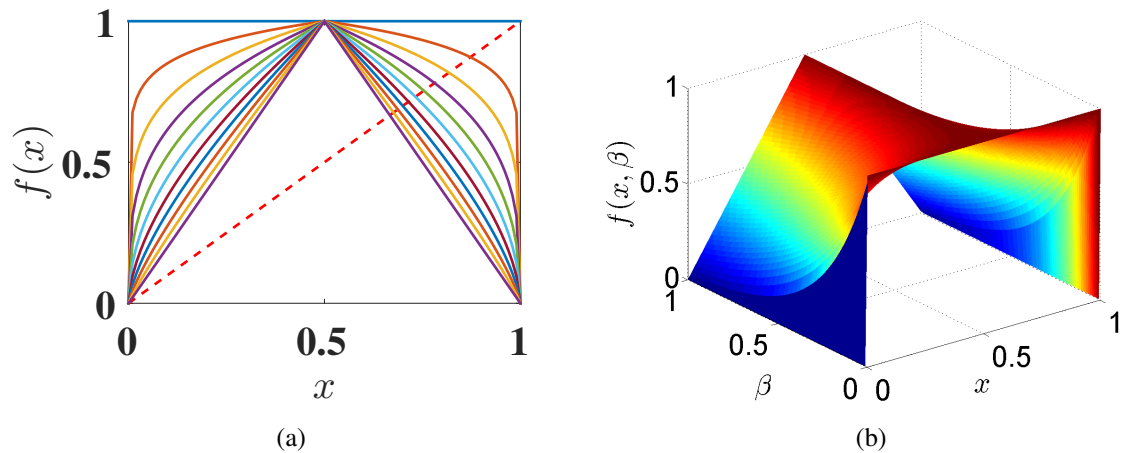


Figure 4.11: (a) Curves of sub-tent map $f(x, \beta)$ and their fixed points at $\beta = \{0.1, 0.2, \dots, 1\}$ at $r = (2)^\beta$ and (b) its surface plot at $r = (1.999)^\beta$

4.4.2 Sub-Logistic Map: $\beta = \alpha$ and $\alpha < 1$

This map resembles logistic map where both shaping parameters are equal and less than one, so it is called sub-logistic, i.e., below logistic and is given by

$$f(x, r, \alpha) = r(x(1-x))^\alpha, \quad 0 < \alpha < 1 \quad (4.35)$$

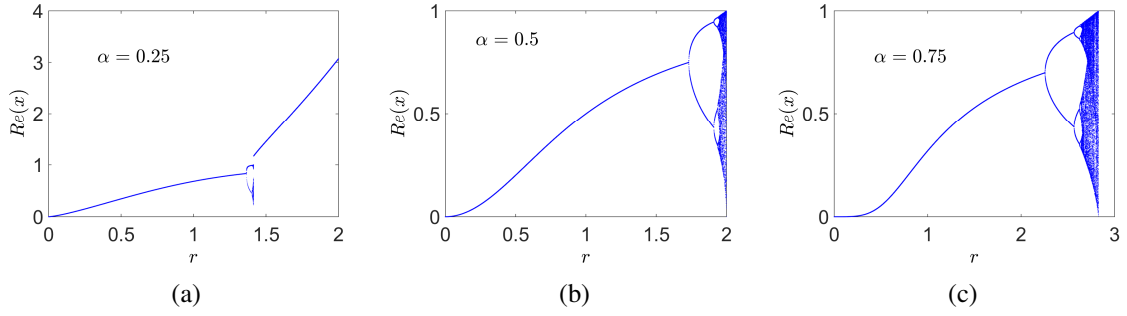


Figure 4.12: Real bifurcation diagrams of the sub-logistic map for various values of α starting at initial point $x_0 = 0.05$

Figure 4.12 shows samples of the real bifurcation diagrams for both positive and negative r for a set of values of the parameter α .

Figure 4.13(a) shows the curves of the sub-logistic map at different values of α for $r > 0$ such that $r = r_{max}$. The curves show that every case for the sub-logistic map exhibits a single non-trivial fixed point. Figure 4.13(b) shows the continuous surface plot of the map equations $f(x, \alpha)$ as a function of both x and α for $\alpha \in [0, 1]$ at $r = (1.999)^{2\alpha}$.

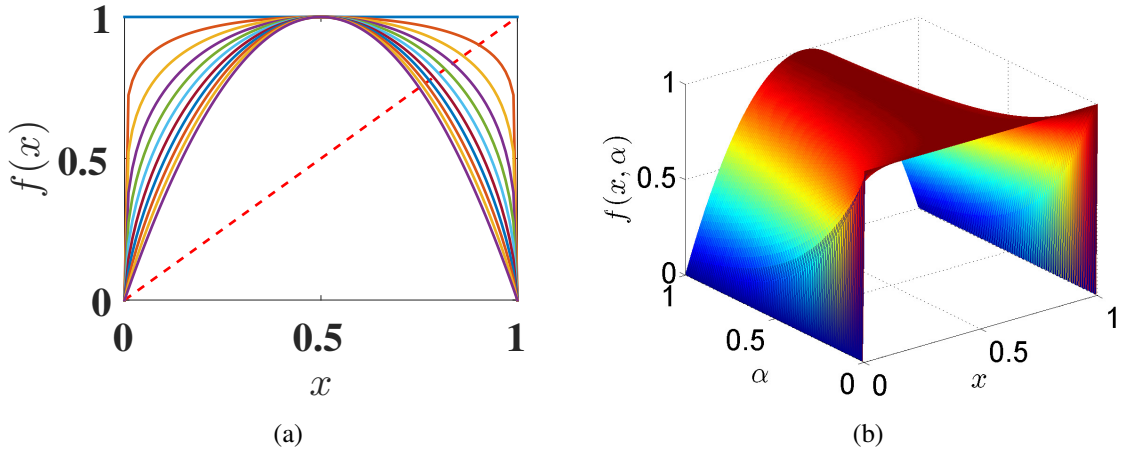


Figure 4.13: (a) Curves of the sub-logistic map $f(x, \alpha)$ and their fixed points at $\alpha = \{0.1, 0.2, \dots, 1\}$ at $r = (2)^{2\alpha}$ and (b) its surface plot at $r = (1.999)^{2\alpha}$

For $r \geq 0$, the values of the key-points of the diagrams for both sub-tent and sub-logistic maps are consistent with the expressions derived in subsection 4.2.1.1 according to Table 4.2 where the values shown in the table are calculated by substitution in (4.10).

4.4.3 Higher Order Map: $\alpha = 1$ and $\beta > 1$

The proposed map has a single parameter β which is greater than one and is given by

$$f(x, r, \beta) = r \min(x^\beta (1-x), x(1-x)^\beta), \quad \beta > 1 \quad (4.36)$$

Figure 4.14 shows samples of the bifurcation diagrams for $r > 0$ for a set of values of the parameter β .

Table 4.2: Key-points of the bifurcation diagram for sub-tent and sub-logistic maps

Map	α	β	r_{max}
Sub-tent	0	0.25	1.1892
Sub-tent	0	0.5	1.4142
Sub-tent	0	0.75	1.6818
Sub-logistic	0.25	0.25	1.4142
Sub-logistic	0.5	0.5	2
Sub-logistic	0.75	0.75	2.8284

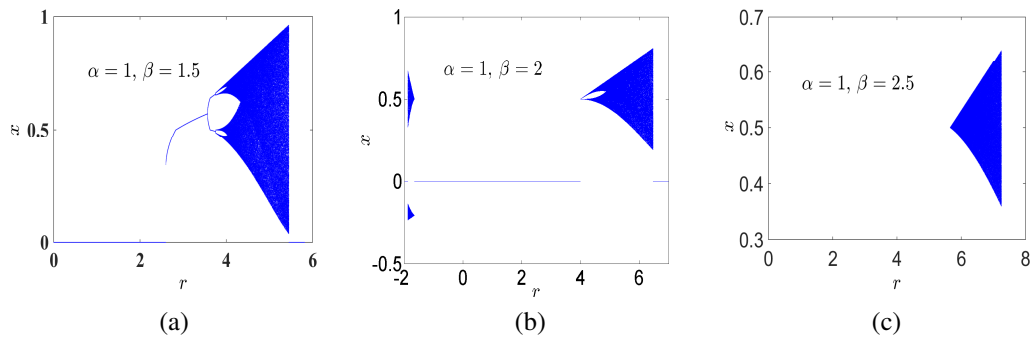


Figure 4.14: Bifurcation diagrams of higher order map for various values of β starting at initial point $x_0 = 0.5$

Figure 4.15(a) shows the curves of the higher order map at different values of β for $r > 0$ such that $r = r_{max}$. For this map, there exist two non-trivial fixed points. The appearance of either points in the bifurcation diagram is governed by their stability ranges. Figure 4.15(b) shows the continuous surface plot of the map equations $f(x, \beta)$ as a function of both x and β for $\beta \in [1, 3]$ at $r = (1.999)^{\beta+1}$.

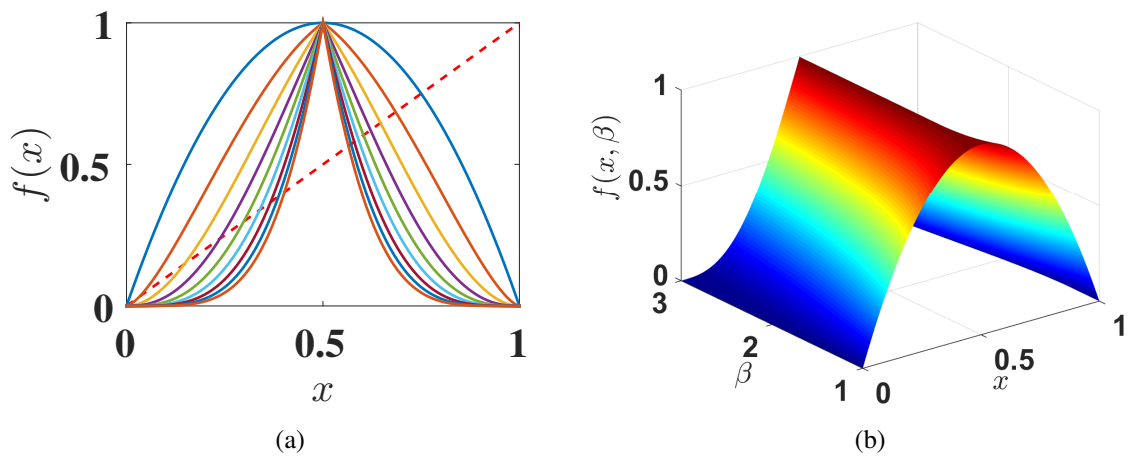


Figure 4.15: (a) Curves of higher order map $f(x, \beta)$ and their fixed points at $\beta = \{1, 1.5, \dots, 5\}$ at $r = (2)^{\beta+1}$ and (b) its surface plot at $r = (1.999)^{\beta+1}$

4.4.4 Super-Logistic Map: $\beta = \alpha$ and $\alpha > 1$

This map also resembles logistic map where both shaping parameters are equal but greater than one, so it is called super-logistic, i.e., above logistic and is given by

$$f(x, r, \alpha) = r(x(1-x))^\alpha, \quad \alpha > 1 \quad (4.37)$$

Figure 4.16 shows samples of the bifurcation diagrams for both positive and negative r for a set of values of the parameter α . Figure 4.16(a) indicates the continuity as values of parameters near the logistic map yield bifurcations in both positive and negative parameter cases. Yet, a portion of the bifurcation diagram is divided between real and imaginary parts of the response.

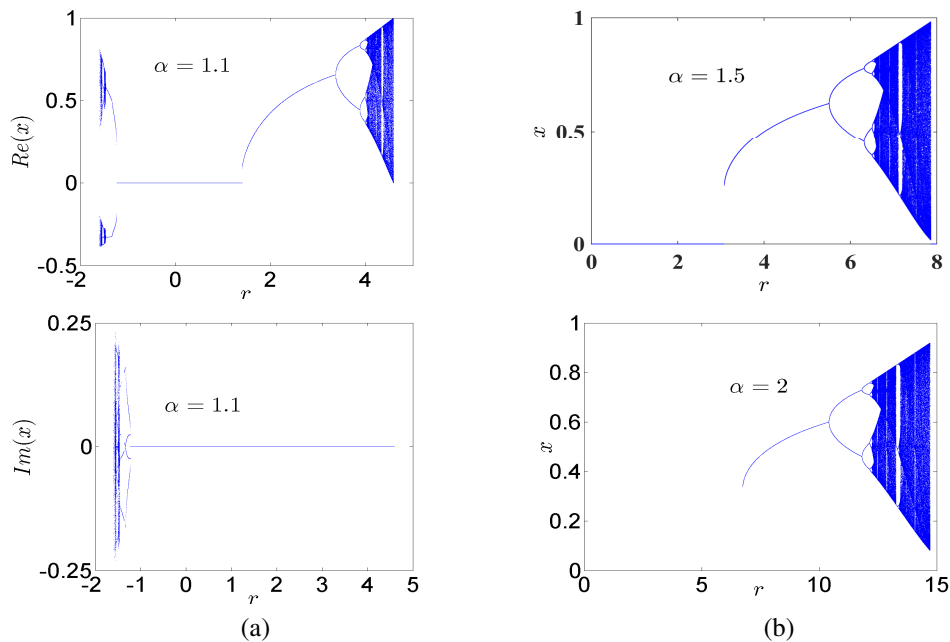


Figure 4.16: Bifurcation diagrams of super-logistic map for various values of α starting at initial point $x_0 = 0.5$

Figure 4.17(a) shows the curves of the higher order map at different values of α for $r > 0$ such that $r = r_{max}$. For this map, the appearance of either points in the bifurcation diagram is governed by their stability ranges as in the case of higher order map. Figure 4.17(b) shows the continuous surface plot of the map equations $f(x, \alpha)$ as a function of both x and α for various values of $\alpha \in [1, 3]$ at $r = (1.999)^{2\alpha}$ where the curves become narrower.

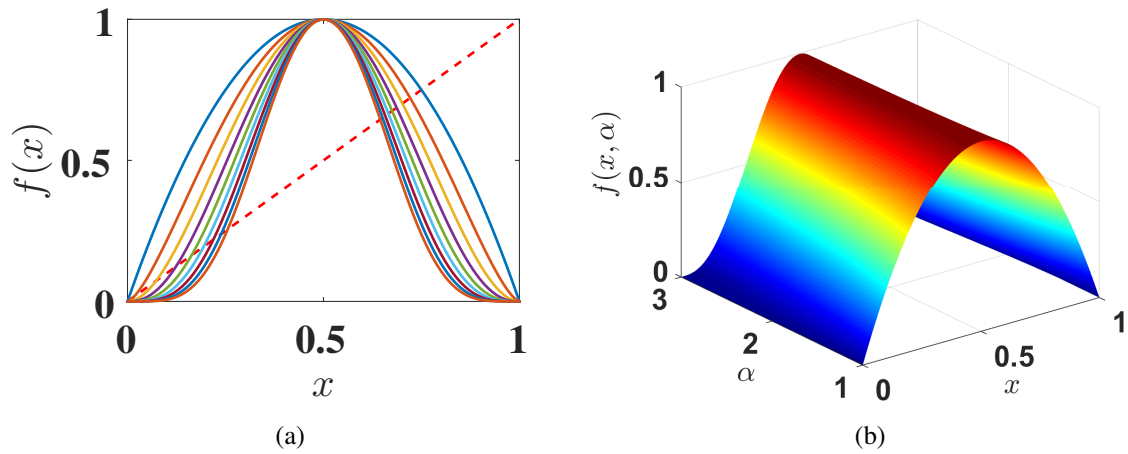


Figure 4.17: (a) Curves of super-logistic map $f(x, \alpha)$ and their fixed points at $\alpha = \{1, 1.5, \dots, 5\}$ at $r = (2)^{2\alpha}$ and (b) its surface plot at $r = (1.999)^{2\alpha}$

4.5 A New Bifurcation Diagram

Figure 4.18 shows the new bifurcation diagram plotted against the parameter β at $\alpha = 1$ and various values of r for $r > 0$. The analysis of this diagram is presented in this subsection. The diagrams behave in contrast to the bifurcation diagrams versus the parameter r , where at lower values of β chaotic response is exhibited in the full range. The range of output response x starts to shrink gradually followed by the appearance of periodic orbits that end with period-4, then period-2, and finally fixed response that disappears at a certain value of β .

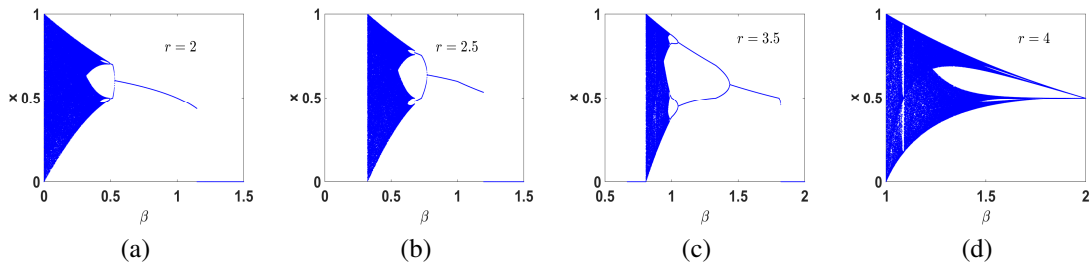


Figure 4.18: Bifurcation diagrams versus the parameter β for different values of the parameter $r > 0$ starting at initial point $x_0 = 0.5$

4.5.1 Positive Parameter $r \geq 0$

4.5.1.1 Range of β

In the analysis of the bifurcation diagram versus the parameter r , we could specify ranges of r for which bifurcations exist depending on the value of β as in (4.32). Similarly, for

the bifurcation diagram versus β , we get:

$$\beta_{min} = \frac{\ln r}{\ln 2} - 1, \quad (4.38)$$

where β_{min} is the minimum value of β for which the bifurcation diagram exists and could be plotted as shown in Fig. 4.19. Figure 4.18 indicates that as r increases, the starting point β_{min} shifts gradually according to its value given by (4.38).

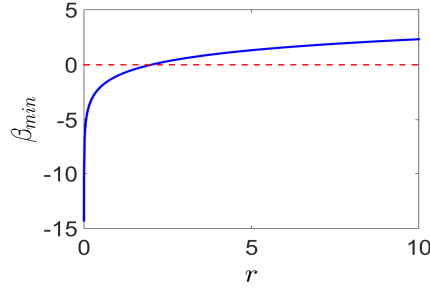


Figure 4.19: The minimum value of β for which the bifurcation diagram exists (β_{min}) as a function of the system parameter r such that $r > 0$

The upper bound of β at which the bifurcation diagram still exists represents a sudden change from the presence of a non-trivial fixed point to no response at all (or divergent response).

4.5.1.2 Corresponding Allowed Values of r

Starting at $r = 2$, bounded responses are confined to the range $\beta > 0$, the range in which we are interested, as shown in Fig. 4.18. This is consistent with the expression for calculating β_{min} given by (4.38), where for $r < 2 \rightarrow \beta_{min} < 0$. Starting at $r = 4$, the bifurcation diagram is re-shaped and translated to start at $\beta_{min} = 1$ at $r = 4$ (Fig. 4.18(d)) and this value for β_{min} increases with increasing r according to (4.38). As r increases to be greater than 5, the output responses range $[x_{min}, x_{max}]$ shrinks than the interval $[0, 1]$ gradually. Figure 4.20(a) shows snapshots for the bifurcation diagram versus the parameter β for different values of r . The snapshots indicate the shifting of the starting point β_{min} as r increases and the shrinking of the output responses starting at $r > 5$. It is worth mentioning that similar diagrams exist for $r < 2$ but the corresponding range of β includes negative values.

4.5.2 Negative Parameter $r < 0$

For $r < 0$, no bounded responses exist in the range of $\beta \in [0, 1]$ as shown in Fig. 4.20(b).

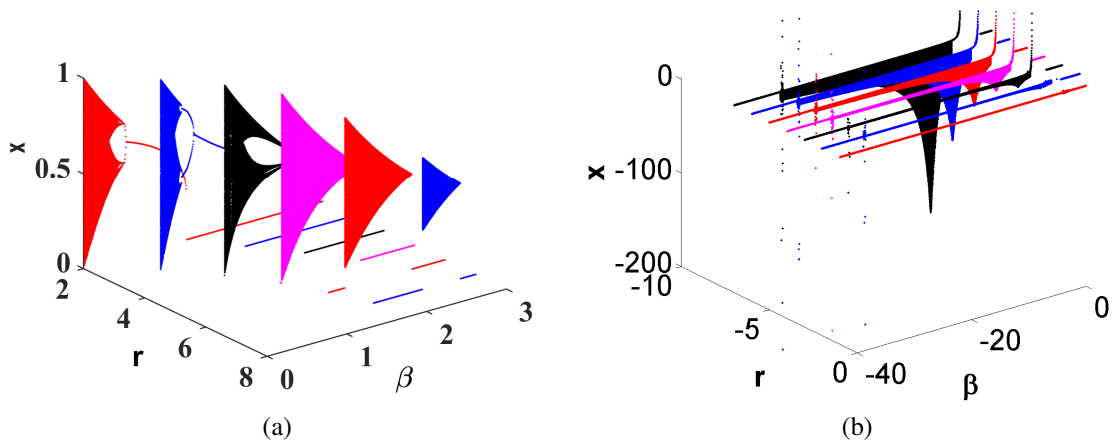


Figure 4.20: Snap shots of the bifurcation diagram versus β starting at initial point $x_0 = 0.5$ for (a) $r = \{2, 3, \dots, 7\}$ and (b) $r = \{-7, -6, \dots, -1\}$

4.6 Implementation Issues

The proposed map in its general form represents a tradeoff between the possibility of utilizing the same setup as various chaotic generators for multiple different sequences and the complexity of executing the power operation. Designs that consider the studied special cases of the map or others with the value of one of the parameters are known at design time are also encouraged. All the previous bifurcation diagrams have been plotted using MATLAB and the complex power operator (\wedge), where all operations are in the complex plane plotting both the real and imaginary parts of the final steady state response(s). Some maps exhibit real results only with zero imaginary parts, others have complex results but with chaotic responses in real parts only, and some have chaotic responses in both real and imaginary parts of the response. Would it be more convenient, at least for some cases, to use an operator or a function that performs the power operation in real arithmetic instead of complex? How much would the results differ in both fixed-point and floating-point implementations?

In the regions of divergence, calculations on MATLAB do not yield the same result when using the map equation given by (4.5) in which MATLAB selects the minimum using the built-in function $\min(\cdot)$ and (4.7) in which we compare according to the threshold intersection point x_k . Although the time waveforms and bifurcation diagrams for the real part of x are the same, the imaginary parts calculated using the two methods differ. The issue that the first method of calculation yields infinite imaginary part, while the second method yields zero imaginary part could be owed to several inconsistencies in MATLAB. Large values that could be considered infinity yielded for the real part explains the cutoff that occurs in the bifurcation diagram for fractional β . Moreover, attempting to study the previous behaviors and generate bifurcation diagrams of the proposed map at its different special cases confined to the real plane, we used the function $\text{realpow}(\cdot)$. However, an “Error” is yielded in some cases, using (4.5), that forces the program to stop. The function $\text{realpow}(\cdot)$ has been designed to report an “Error” only when attempting to use it with an expression that produces a complex result. However, in Chapter 6, we record and detail that this does not always happen. Cases that yield “Error” in calculating the responses of

the proposed map include:

- When the system parameter r is less than zero, even when a complete bifurcation diagram till chaotic behavior has been reported using (^), e.g., for the transition map. This is due to raising negative values of x to the power of non-integers.
- When the system parameter r is greater than r_{max} , the value of x is no longer confined to the interval $[0, 1]$ or $[0, a/b]$. Instead of complete divergence to $-\infty$ that occurs when using (^) keeping the results till r_{max} and plotting them, using `realpow(.)` yields an “Error” and forces the program to stop.

A comprehensive study of various implementations for the power function is included in Chapter 6. The study concentrates on negative values of the base raised to the power of non integer, and limits as either base, exponent, or both approach infinitesimal tiny or infinitely huge values. But at first fixed-point implementations of chaotic generators are considered which are widely spread as explained in Chapter 2. These finite precision implementations must have an impact on the properties of these generators that were derived assuming infinite precision (and infinite time). However, this impact has not been thoroughly studied before according to our survey in Chapter 2. Consequently, we devote the next chapter to study finite precision logistic map in two cases: the conventional case, and the negative control parameter case, or mostly positive logistic map, proposed in Chapter 3. The next two chapters explore the differences from analytical expected behavior of any mathematical problem imposed by implementation concentrating on the set of problems in which we are interested, explain the difficulties facing the implementation of the proposed map, and construct a base for determining the best way of implementing the proposed general powering map.

Chapter 5: Finite-Precision Logistic Map

Unseen behavior lying between the analytical study of chaos and its representation on digital computers deserves further exploration. The studies presented in this chapter on chaotic maps under fixed-point arithmetic can be extended to floating-point arithmetic as well, where floating-point decimals are non-uniformly distributed over the discrete domain. In this chapter, we demonstrate that executing the operations constituting the discrete 1D logistic map, for either positive or negative control parameter, individually in a sequential manner with the truncation step implemented between them would have rather catastrophic effect on the properties of the logistic map. Moreover, the order of their execution yields significantly different behaviors. The proposed finite precision logistic map is similar to digital hardware designs in which the logistic map is realized using limited precision implemented in fixed-point arithmetic. Throughout our discussion, four main parameters of the finite precision logistic map are considered, two of them are not new which are the system parameter λ and the initial point x_0 . The other two parameters are introduced by the finite precision nature of the map which are the precision p and the order of executing $f(x)$.

The effect of varying precision on several properties of the logistic map and the differences from the analytical model are discussed. These properties include the key-points of the bifurcation diagram, periodicity of the generated sequence, and MLE as a measure of the chaotic behavior of the system. Sensitive dependence on initial conditions is also considered along with studying each property. First, we discuss how the 1D logistic map can be represented in digital hardware realizations such that it can work for either positive parameter or negative parameter cases and the assumptions required to simulate this representation in software environments. Six different versions of the map are proposed based on the order of execution of the operations constituting its expression. Two of these versions are chosen primarily to conduct various experiments on them and the results obtained for the two chosen map versions are demonstrated. These results include: the bifurcation diagram, its key-points, time waveforms, periodicity of the generated sequence, and two different calculation methods for MLE.

5.1 Digital Representation of the Logistic Map

Fixed-point representation uses integer hardware operations controlled by a given convention about the location of the fractional point, e.g., 6 bits (binary) or digits (decimal) from the right. Fixed-point hardware is generally less expensive than an equivalent floating point hardware implementation. Binary fixed point is usually used for special-purpose applications on embedded processors, but decimal fixed point is more common in commercial applications.

5.1.1 Assumptions of Fixed-Point Binary Representation

Using finite precision fixed-point binary system, the evaluation of the logistic map function is carried out in a similar manner to a microprocessor instruction set, i.e., it is

subdivided into a sequence of basic operations. MATLAB fixed-point toolbox is used to simulate/mimic digital representation of the logistic map on FPGA. The ranges of x and λ obtained in Chapter 2 for conventional logistic map ($x \in [0, 1]$ and $\lambda \in [0, 4]$) and in Chapter 3 for mostly positive logistic map ($x \in [-0.5, 1.5]$ and $\lambda \in [-2, 0]$) mean that the maximum number of integer bits needed is 4 bits. The included range ($-2 \leq \lambda \leq 4$) is totally representable in 4 bits in two's complement coding. In addition, it is guaranteed that the resulting value is at most bounded by the relation $-0.5 \leq x \leq 1.5$, i.e., no extra bits are needed for handling overflow conditions.

The total number of bits or bus width, alternatively called the word length, is denoted by p for precision which is represented as p_i integer bits and p_f fractional bits such that $p = p_i + p_f$. It is worth mentioning that the number of fractional bits controls the step length that we can move by along λ values. The accuracy of λ can not exceed 2^{-p_f} as listed in Table 5.1 which does not exceed two decimal places in our discussion. The

Table 5.1: Relation between number of fractional bits and step length

p_f	maximum step length
5	$2^{-5} = 1/32$
6	$2^{-6} = 1/64$
7	$2^{-7} = 1/128$
\vdots	\vdots

range of different number of fractional bits to be studied needs specification, so precisions between 8 and 27 are selected. The suggested range is owed to the following reasons; the lower bound $p = 8$ implies having the same number of bits in both the integer and the fractional parts where both equal 4 bits. The upper bound $p = 27$ resembles the equality of the number of fractional bits $p_f = 23$ to the number of bits in the fractional part f of the single-precision binary floating-point representation which is the smallest precision among those used in most software implementations. Single-precision binary floating-point represents a number as $(-1)^s \times 1.f \times (2)^e$, where $s \in \{0, 1\}$ is the sign bit, f is the fractional part (trailing significand) represented in 23 bits with an implicit leading significand bit of 1, and e is the exponent represented in 8 bits. In addition, the following assumptions are made in simulating digital representation of the logistic map:

- The values of x , λ , and the output of each basic operation are stored in finite length registers, i.e., they are fixed-point variables.
- All operations are carried out assuming truncation (rounding towards zero).
- Numbers are represented in two's complement coding.

5.1.2 Six Different Maps in Fixed-point Arithmetic

The result of the studied function

$$f(x, \lambda) = \lambda x(1 - x) \quad (5.1)$$

can be calculated in multiple ways in fixed-point arithmetic. For instance, consider the following expressions:

- $f_1(x, \lambda) = \lambda(x(1 - x))$
- $f_2(x, \lambda) = (\lambda x)(1 - x)$
- $f_3(x, \lambda) = (\lambda(1 - x))x$

Are they equivalent? Is the associativity property maintained in a fixed-point system? Moreover, the operations could be grouped such that suboperations are performed in different order as follows:

- $f_4(x, \lambda) = (\lambda(x - x.x))$
- $f_5(x, \lambda) = (\lambda x) - (\lambda(x.x))$
- $f_6(x, \lambda) = (\lambda x) - ((\lambda x)x)$

Throughout the rest of the discussion, all results are obtained by MATLAB starting at initial point $x_0 = 0.5$, discarding the first 1,000 iterations and plotting the next 500 ones, except where stated otherwise. Figure 5.1 shows the bifurcation diagram versus the system parameter λ at $p = 9$, where it is supposed that different ranges for the control parameter λ cause the logistic map to exhibit different phases of behavior. However, it could be noticed that different orders of execution yield different bifurcation diagrams that are not the same as those expected from mathematical analysis regarding: key-points, output range, transition from a type of response to another, and density of points at ranges that are supposed to exhibit chaotic behavior. Various properties that have been considered as facts in analyzing the 1D discrete logistic map mathematically are violated in finite precision environments. Several examples are detailed below.

- Initial points with perturbations only in the least significant bit, i.e., $x_0 = 2^{-p_f}$ could cause the response to die out as if the orbit contained one of the zeros of (5.1), i.e., $x = 0$ or $x = 1$. This could be owed to getting a value in which the first p_f bits after the fractional point are zeros. Reviewing how the value at the next iteration is calculated

$$\begin{aligned} x_1 &= \lambda x_0(1 - x_0) \\ &= \lambda 2^{-p_f}(1 - 2^{-p_f}) \\ &= \lambda(2^{-p_f} - 2^{-2p_f}) \end{aligned}$$

For finite precision representation with p_f fractional bits, $(2^{-p_f} - 2^{-2p_f}) = 0$ which is not correct in the ideal infinite precision case. Moreover, any quantity less than 2^{-p_f} will be considered zero after truncation to only p_f fractional bits.

- Analytically, it would be expected that initial points with difference equals 1 between them yield the same steady state behavior (since the same orbit), i.e., the results associated with any initial point $x_0 \in (0, 1)$, for positive parameter case, are the same as those associated with $y_0 \in (0, 1)$ if $x_0 + y_0 = 1$. However, this property is not always satisfied in finite precision case due to truncation effects as previously discussed.

- Although analytically the sensitive dependence on initial conditions dominates at ranges of λ which exhibit chaotic behavior only, this property differs for finite precision systems. The bifurcation diagram over the whole range of λ might alter starting at different initial points. Figure 5.2 shows that the bifurcation diagram differs from the previous case starting at different initial point $x_0 = 0.125$.

From the bifurcation diagrams, $f_3(x)$ and $f_6(x)$ exhibit smoother maps and a behavior that seems to be more similar to that analytically expected from the logistic map than the other alternatives. Hence, the rest of the discussion will concentrate on these two versions of the logistic map, the different properties that they exhibit, and how much they conform to the behavior expected from the mathematical analysis of the map.

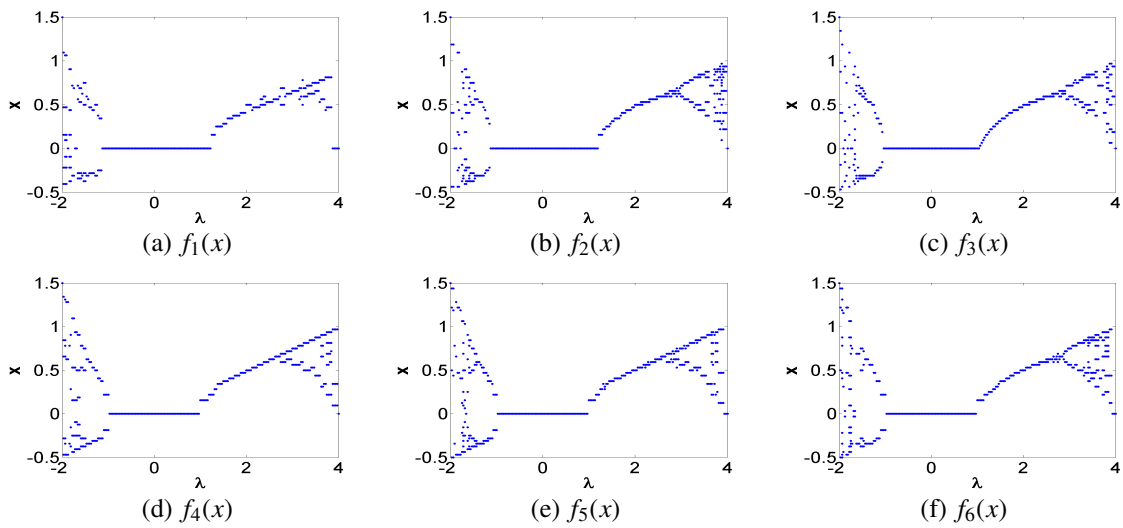


Figure 5.1: Bifurcation diagram versus the control parameter λ for the six maps starting at $x_0 = 0.5$ at $p = 9$

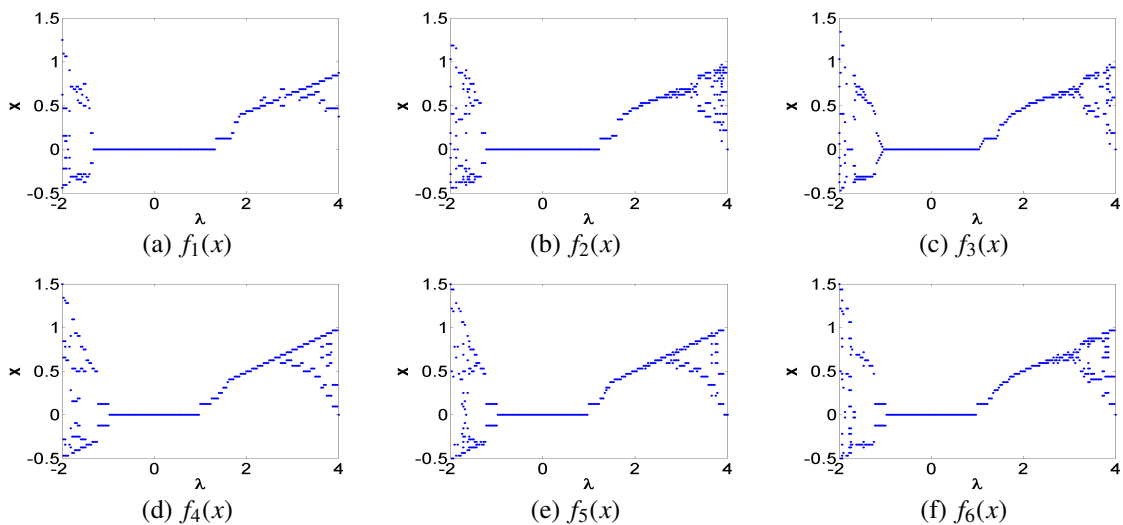


Figure 5.2: Bifurcation diagram versus the control parameter λ for the six maps starting at $x_0 = 0.125$ at $p = 9$

5.2 Properties of the Selected Maps

In order to study the effect of increasing the number of fractional bits and the impact of low precisions on the properties of logistic map, the bifurcation diagram of $f_3(x)$ versus the control parameter λ is plotted for different precisions in Fig. 5.3, while that of $f_6(x)$ is shown in Fig. 5.4, both starting at initial point $x_0 = 0.5$. The resulting diagrams reveal that the properties of the logistic map are so much affected by precision. It could be noticed that the affected properties include: the key-points of the bifurcation diagram, the number of levels or the sequence of values at a fixed value of λ , and how much chaotic is the behavior at high values of λ , near λ_{max} , or the degree of chaos.

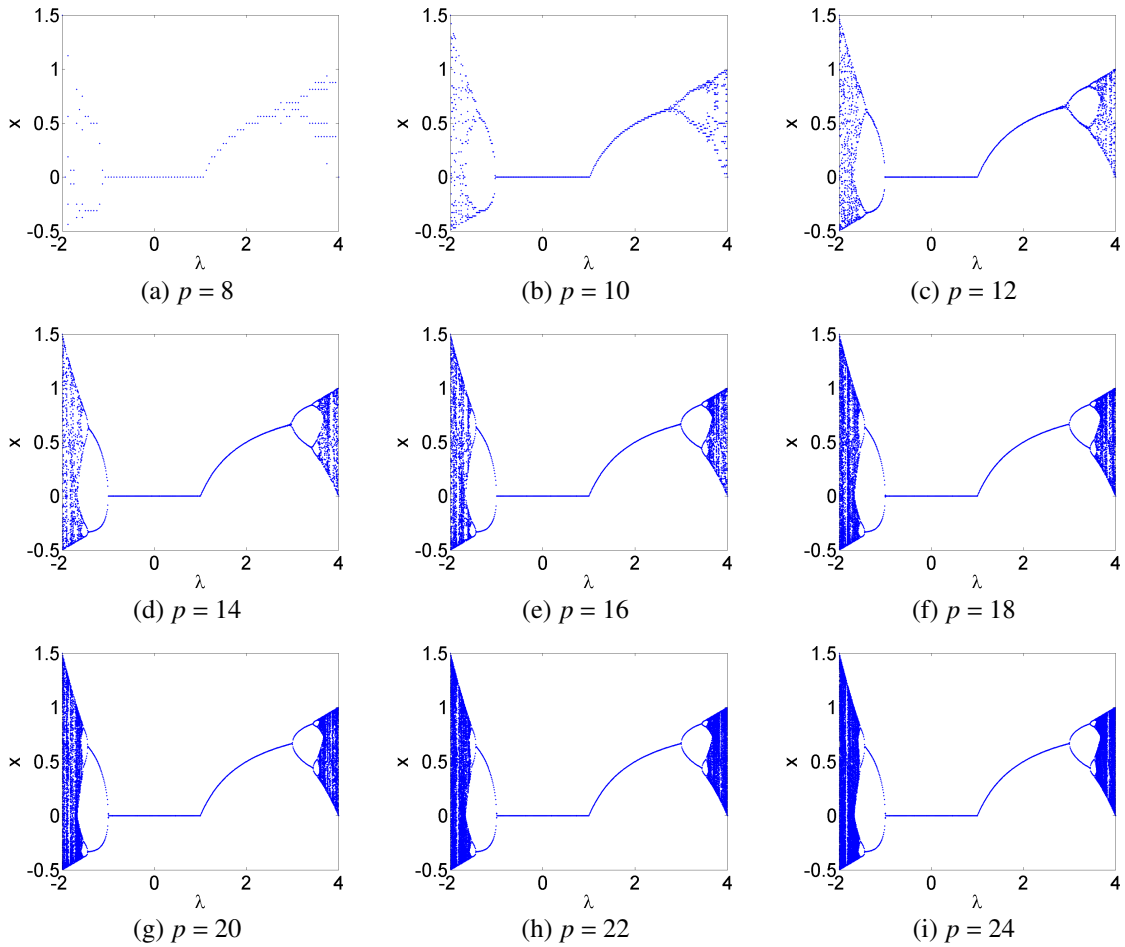


Figure 5.3: Bifurcation diagram of $f_3(x)$ versus the control parameter λ for different values of bus width starting at $x_0 = 0.5$

5.2.1 Key-points of the Bifurcation Diagram

The key-points of the bifurcation diagram play an important role as design specifications of the logistic map utilized as pseudo-random number generator (PRNG) in various applications. This is true for the conventional one with single control parameter λ or generalized maps with more parameters as those proposed in Chapter 3 and Chapter 4. We

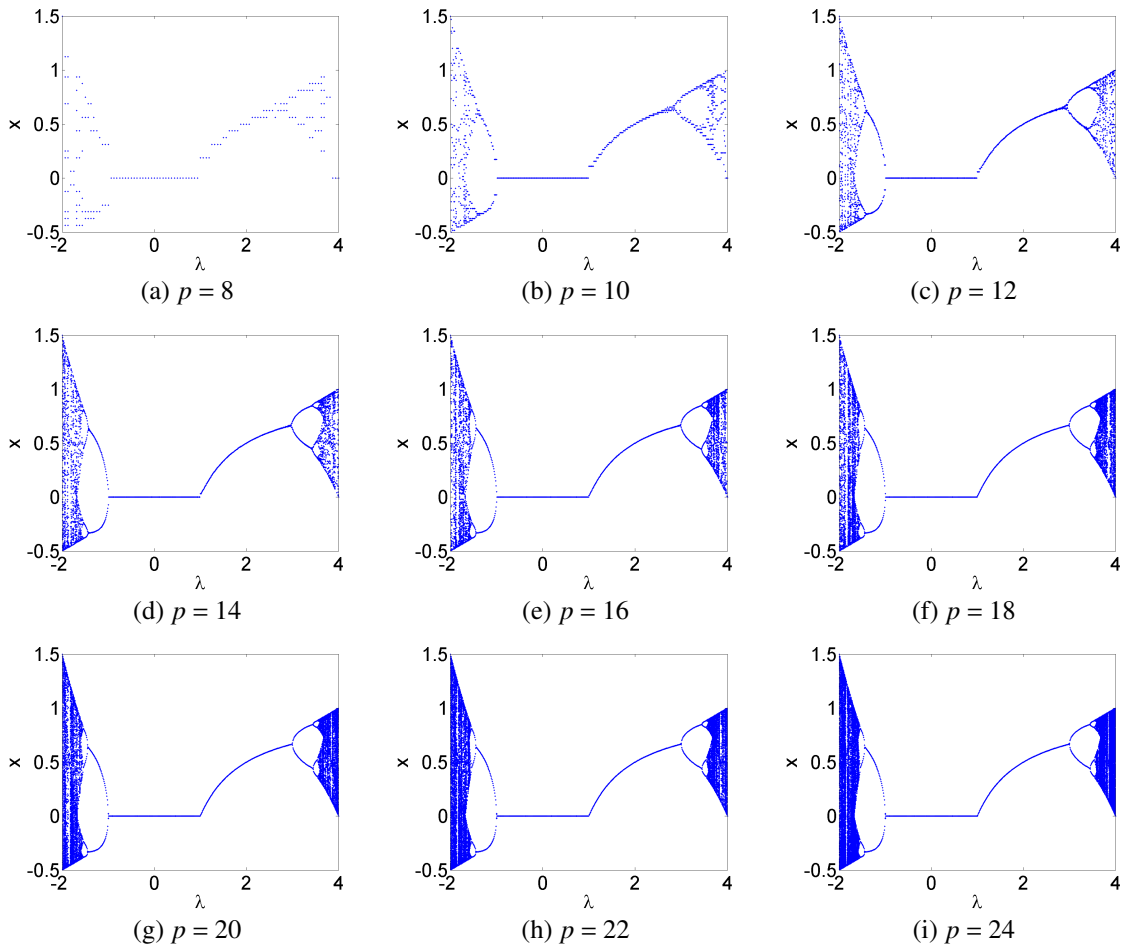


Figure 5.4: Bifurcation diagram of $f_6(x)$ versus the control parameter λ for different values of bus width starting at $x_0 = 0.5$

study the impact of finite precision implementations on the key-points of the bifurcation diagram whose analytical values have been defined in Chapter 2. The values of these key-points in finite precision case differ from the analytical expected behavior due to the truncation caused by finite precision systems. Yet, this difference decreases as p increases as discussed below.

5.2.1.1 Double-Precision Floating-Point

Using double-precision floating-point calculations, $\lambda_{b1_p} = 0.969$, $x_{max_p} = 0.999738909465984$, $\lambda_{b1_n} = -0.967$, $x_{min_n} = -0.49974730424707$, and $x_{max_n} = 1.5$. Although these results should be more accurate than those obtained using fixed-point calculations, they are still not the same as those expected from the conducted mathematical analysis. The differences between the results of double-precision floating-point calculations and the analytical expected results could be owed to their relative inaccuracy. They do not satisfy neither infinite precision nor infinite time conditions assumed analytically. The values of λ_{b1_p} and λ_{b1_n} are obtained by comparison with an optimum threshold chosen as eps which is defined in MATLAB as the spacing of floating-point numbers. Using eps with no arguments returns $2^{(-52)} \simeq 10^{(-16)}$, which corresponds to the value of this spacing in double-precision

floating-point representation. In addition, these values are calculated using accuracy 0.001, i.e., rounded to the third decimal digit after the point. As previously mentioned, floating-point arithmetic implementations of chaotic generators and their impact on their various properties need to be studied.

5.2.1.2 Fixed-Point Implementation

Using fixed-point map versions $f_3(x)$ and $f_6(x)$, let us consider the positive control parameter side at first, Figure 5.5(a) shows the values of λ_{b1_p} , obtained by comparing with eps , at different precisions ranging between $p = 8$ and $p = 26$ for various initial points ($x_0 = 0.125, 0.25, 0.375$ and 0.5). For the map $f_3(x)$, Fig. 5.5(a) shows that λ_{b1_p} starts at values higher than its analytic value “1” at low precisions, then starts to decrease gradually approaching “1”. On the other hand, $f_6(x)$ seems insensitive to/independent on precision from the viewpoint of the value of λ_{b1_p} . Although various values for initial point yield the same plot, we cannot generalize it as a fact that the value of λ_{b1_p} is independent on initial conditions. Moreover, this result should also be tested for the other map versions if they are to be used instead.

For lower and upper bounds on the output range, we calculate them at multiple initial points then consider the average of these different results. Figure 5.5(b) shows the average value $x_{max_{pavg}}$ of upper bound on responses starting at multiple initial conditions versus different precisions. For both maps, $x_{max_{pavg}}$ starts at values lower than its analytic value “1” at low precisions, then starts to increase gradually, with some fluctuations, approaching “1”.

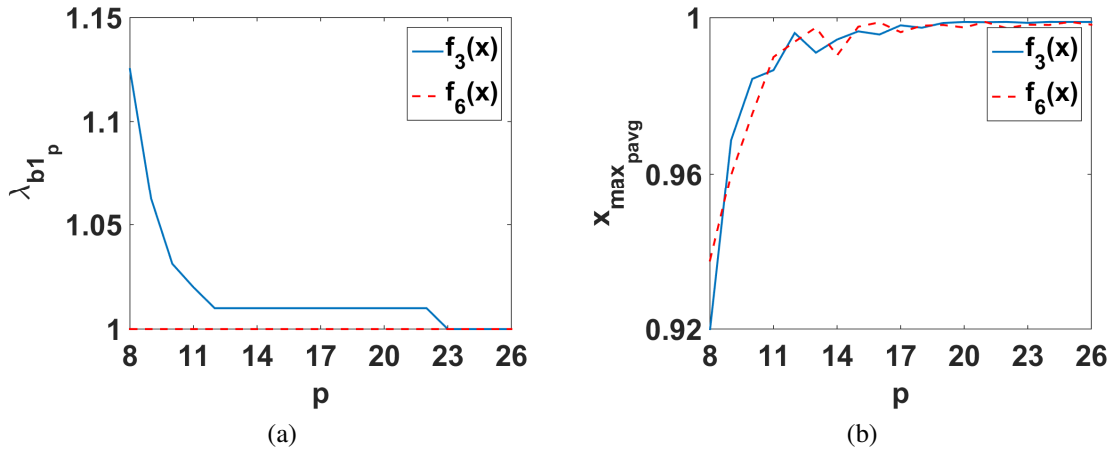


Figure 5.5: The key-points of the bifurcation diagram for both positive control parameter maps at different precisions (a) The first bifurcation point and (b) The average maximum value

Similarly, the key-points of the bifurcation diagram in the negative control parameter side are studied. Figure 5.6(a) shows the values of λ_{b1_n} where similar comments to the positive control parameter case could describe the plot but for the absolute value $|\lambda_{b1_n}|$ instead. Figure 5.6(b) shows the values of $x_{min_{navg}}$ at different precisions, whereas Fig. 5.6(c) shows the values of $x_{max_{navg}}$. The map $f_6(x)$ seems to be less sensitive to precision variation than $f_3(x)$. The average minimum value $x_{min_{navg}}$ starts at low precisions

with absolute values which are lower than 0.5, whereas $x_{max_{navg}}$ starts with values which are lower than 1.5. As precision increases, the keypoints approach their analytical values.

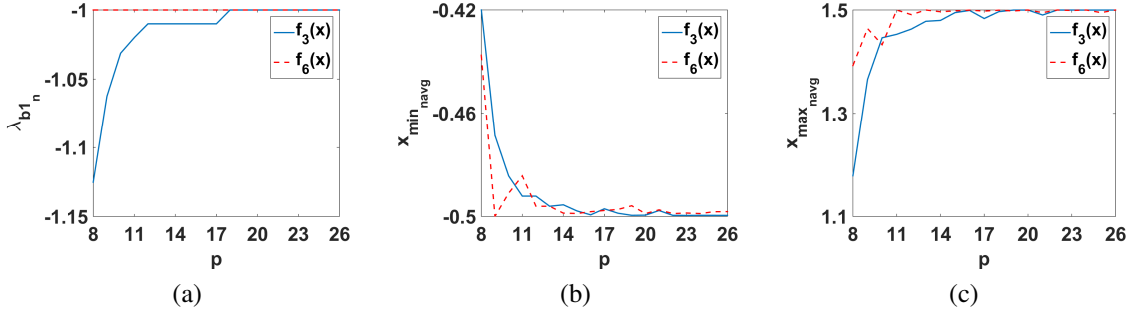


Figure 5.6: The key-points of the bifurcation diagram for both negative control parameter maps at different precisions (a) The first bifurcation point, (b) The average minimum value, and (c) The average maximum value

5.2.1.3 Sensitivity to Initial Conditions

From Figures 5.5(a) and 5.6(a), the values of λ_{b1_p} and λ_{b1_n} seem insensitive to the value of initial point x_0 . On the other hand, the effect of initial point on the values of x_{max_p} , x_{min_n} , and x_{max_n} can not be avoided in finite precision implementations especially at low precisions. The value x_{max_p} occurs at $\lambda = 4$, whereas x_{min_n} and x_{max_n} occur at $\lambda = -2$. Both values of λ exhibit maximum chaotic behavior, where sensitivity to initial conditions is a basic characteristic of the bifurcation diagram. Analytically, the “infinite” sequence generated at maximum chaotic behavior has a lower and upper bounds which “must” be reached. Yet, in finite precision implementations, the length of the generated sequence is limited by both finite precision and time. Thus, it is not guaranteed whether its lower and upper bounds shall coincide with the analytical values or not. The dependence of these bounds on initial points is unavoidable in finite precision, especially at low precisions.

5.2.1.4 Precision Threshold

From Figures 5.5 and 5.6, it is clear that the values of the key-points derived through mathematical analysis are the asymptotes that finite precision values approach as $p \rightarrow \infty$. The thresholds for used precision could be selected as acceptable threshold or safer threshold. An acceptable precision threshold could be set as the value of p at which the key-point starts to exhibit fixed value. For instance, $p = 14$ for $f_3(x)$ and $p = 12$ for $f_6(x)$. Moreover, the value of p at which the key-point approaches its analytical value more could be considered as a safer threshold. For instance, $p = 23$ for $f_3(x)$ and $p = 20$ for $f_6(x)$.

5.2.2 Time Waveforms

Another important perspective that should be studied in the behavior of the logistic map is the time waveforms in the ranges of λ that should exhibit chaotic behavior. Mathematically speaking, the steady state response should be in the form of a new value generated at each

discrete time instant such that no periodicity can be recognized. In this section, the time waveforms at value of λ that are supposed to be chaotic are analyzed. The effect of varying the used precision and the initial point at which the orbit starts are explained.

5.2.2.1 Precision and Initial Point Effects at $\lambda = 3.9375$

Figure 5.7 shows the time waveforms at $\lambda = 3.9375$ of the map $f_3(x)$ with positive control parameter starting at different initial conditions for different precisions. Figure 5.8 shows the time waveforms of the map $f_6(x)$ at the same λ . The reason behind choosing this value is the interest in this range near $\lambda = 4$ as it exhibits the widest chaotic response which is rich in applications as previously discussed. This specific value corresponds to an exactly representable fixed-point number with four fractional bits; corresponding to $p = 8$ the narrowest precision that is examined in our study. It should be noted that a value like this for λ did not appear in the bifurcation diagrams as it was previously explained that the accuracy, of the value of λ , does not exceed two decimal places. The waveforms show that the versions of the map implemented in finite precision exhibit periodic behavior for this value of λ which is supposed to be chaotic. The practical meaning of this periodic behavior is that the sequence generated by the map is not efficiently random, instead it is a finite length periodic sequence.

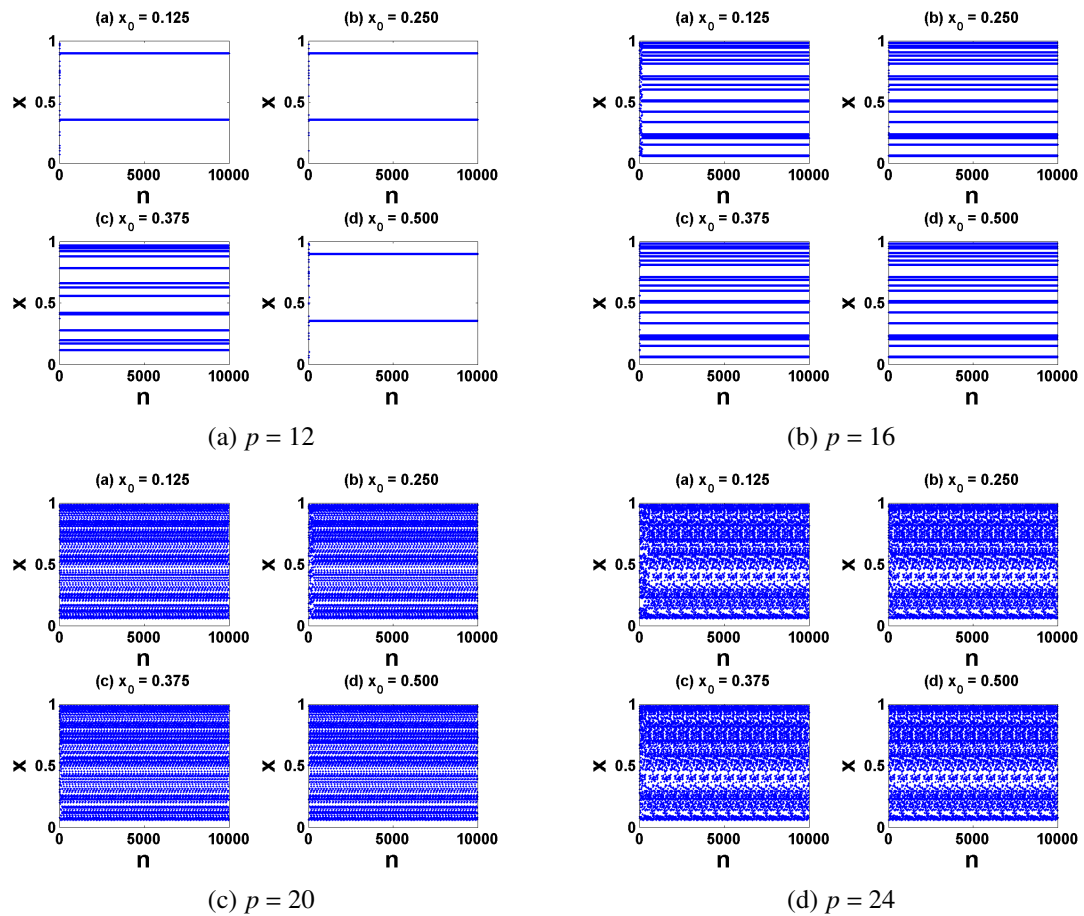


Figure 5.7: Time waveforms of $f_3(x)$ at $\lambda = 3.9375$ starting at different initial conditions and various precisions

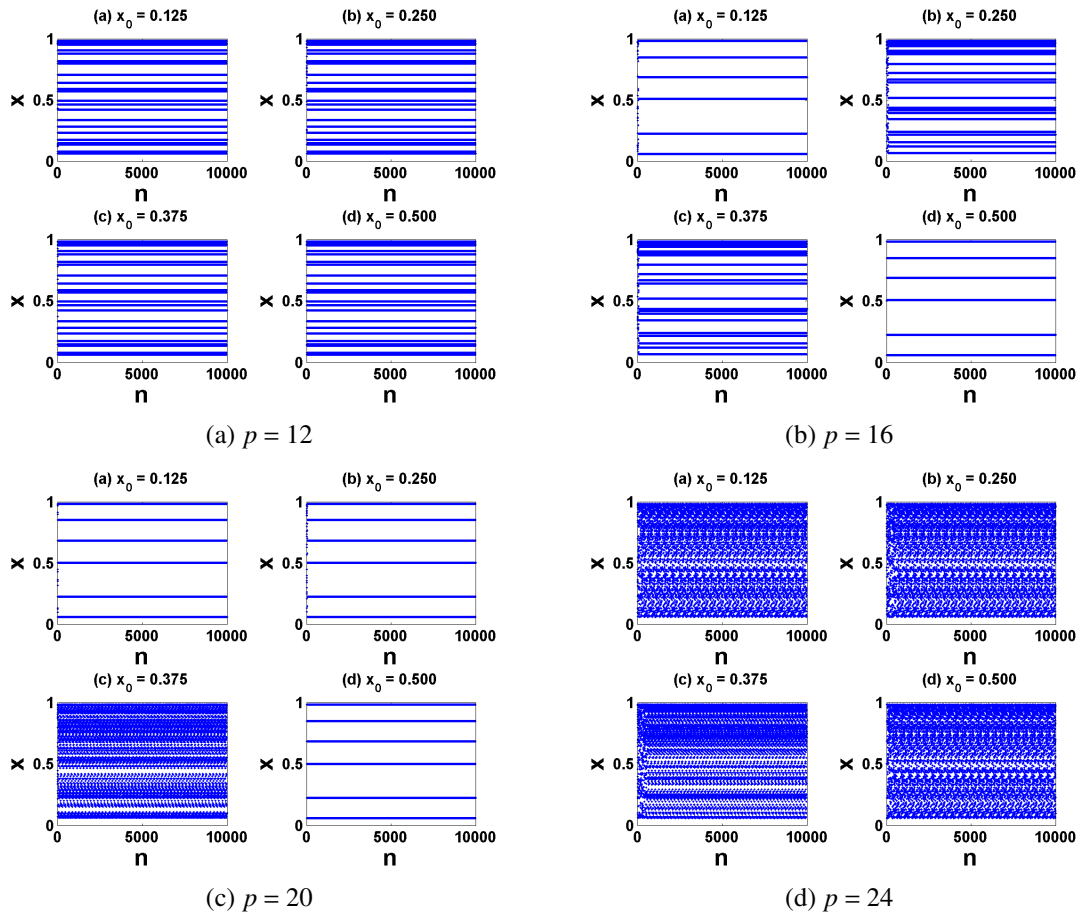


Figure 5.8: Time waveforms of $f_6(x)$ at $\lambda = 3.9375$ starting at different initial conditions and various precisions

In the following discussion, we attempt to study the reason behind the strange result obtained in Fig. 5.8(c). In addition, other values of λ that are expected to yield chaotic behavior from the mathematical analysis in infinite precision are not chaotic at all, instead they yield periodic responses in the steady state. The cobweb plot shown in Fig. 5.9 is a rough plot for the orbit of x starting at different initial points x_0 , where the graph of the map function is sketched together with the diagonal line $y = x$. Although the four cobweb plots seem different, the steady state, colored in red and blue, in case of $x_0 = 0.125$ or 0.25 is the same as the plot in case of $x_0 = 0.5$ which fluctuates between six different values, i.e., period-6. The output states are as follows (these are the decimal equivalents of the binary sequences represented in the used fixed-point representation at $p = 20$, i.e., 16 fractional bits): $0.5 \rightarrow 0.984375 \rightarrow 0.0605621337890625 \rightarrow 0.2240142822265625 \rightarrow 0.6844635009765625 \rightarrow 0.85040283203125 \rightarrow 0.5009307861328125 \rightarrow 0.984375 \dots$. Any orbit including one of these six values, consequently the others, (for this order, p , and λ) will converge to a sequence that consists only of them. On the other hand, the case $x_0 = 0.375$ exhibits chaotic behavior, since the orbit has a non-periodic sequence. As previously mentioned, assuming infinite precision such a type of solution could be mathematically identified through solving $f^6(x_p) = x_p$. This equation along with the stability analysis of the periodic point; $|(f^6)'(x_p)| = 1$ would yield value(s) for λ at which

this solution starts to appear and the corresponding values for x . According to [111], the value $\lambda = 3.9375$ is really close to one of the real roots of the yielded polynomial.

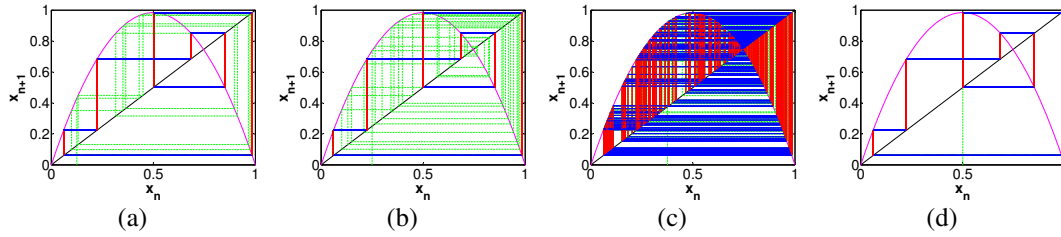


Figure 5.9: Cobweb plots of $f_6(x)$ starting at different initial conditions (a) $x_0 = 0.125$, (b) $x_0 = 0.25$, (c) $x_0 = 0.375$, and (d) $x_0 = 0.5$

The cases at which no chaotic behavior is obtained depend on several factors, e.g., the values of λ , p , and x_0 . Any combination of these factors might drive us away from the expected chaotic behavior yielding a periodic solution instead, which may sometimes be of a significantly short period. General notes can describe the time waveforms at values of the parameter that are analytically close to maximum chaotic response such as: very low precisions exhibit undesirable periodic behavior for almost all initial conditions, and high precisions exhibit chaotic behavior for some or most of the initial conditions. However, the phenomena of deviation from chaotic behavior does not appear in a continuous manner along with varying precision in the intermediate range. The solution could be chaotic at a certain precision p , then become periodic at the next precision $p + 1$. The same note could be used to describe the case fixing the used precision and varying the initial point.

5.2.2.2 Precision and Initial Point Effects at $\lambda = 3.984375$

For further illustration, Figure 5.10 shows the time waveforms of $f_3(x)$ for $\lambda = 3.984375$ which is closer to $\lambda_{max} = 4$, while Fig. 5.11 shows them for $f_6(x)$. This value is representable in at least six fractional bits corresponding to $p = 10$. Although the neighborhood of this value for λ exhibits chaotic behavior and does not contain near values that generate periodic sequences, Fig. 5.10 shows how some combinations of p , and x_0 could also yield faulty periodic response. Similar tracking of the calculations as that conducted for the previous case could be used to illustrate how the recurrence settles to a specific periodic sequence instead of the chaotic behavior expected from mathematical analysis.

5.2.2.3 Precision and Initial Point Effects at $\lambda = -1.9375$ and $\lambda = -1.984375$

Figures 5.12 and 5.13 show the time waveforms at $\lambda = -1.9375$ using $f_3(x)$ and $f_6(x)$ respectively starting at different initial points. The figures show how some combinations of p , and x_0 could also yield faulty periodic response. Similar tracking of the calculations could be performed to illustrate how the recurrence settles to a specific periodic sequence instead of the chaotic behavior expected from mathematical analysis. Figures 5.14 and 5.15 could be described similarly and are included for further illustration on the impact of finitude on behavior of both positive and mostly logistic maps. They could lose their characteristic long periods at various combinations of p , λ , and x_0 . The possibility of exhibiting short periods and getting these faulty undesired responses decreases a lot along with increasing the used precision.

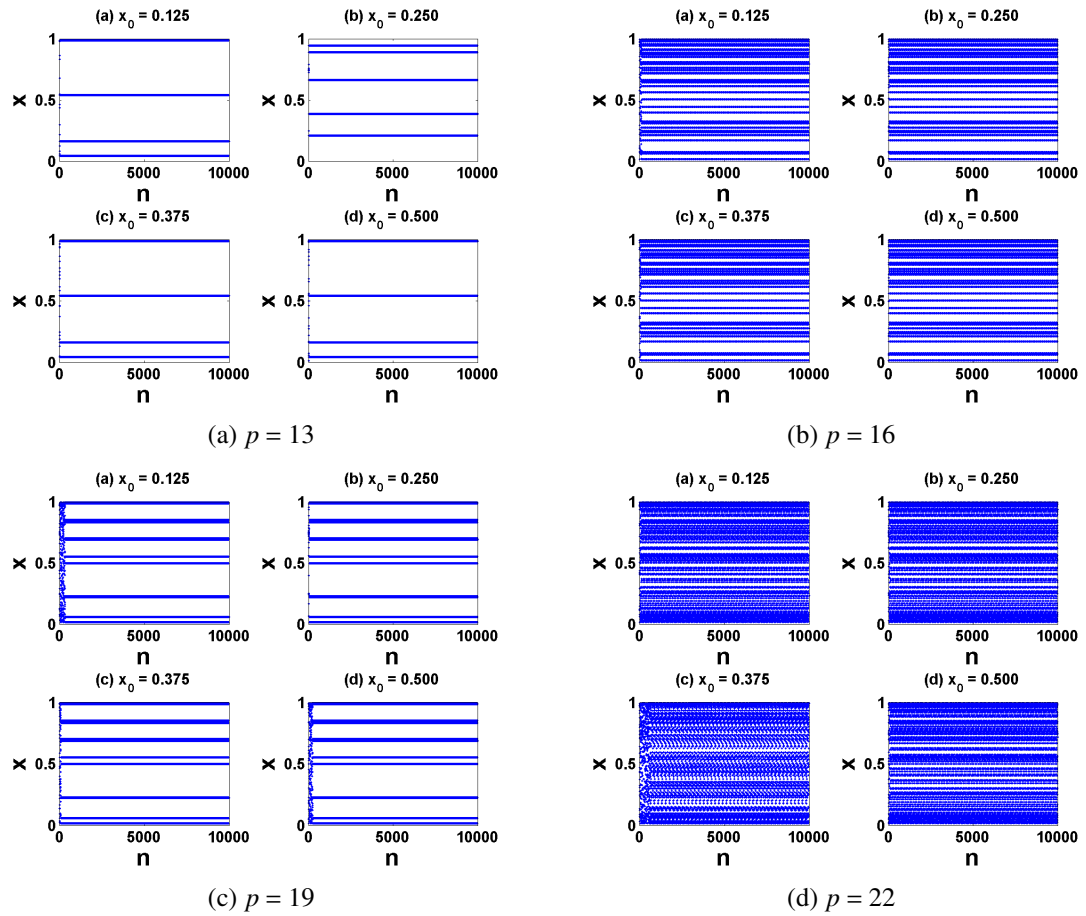


Figure 5.10: Time waveforms of $f_3(x)$ at $\lambda = 3.984375$ starting at different initial conditions and various precisions

5.2.2.4 Precision Threshold

From the above discussion and further figures exhaustively covering the whole studied range of precisions for different initial points, we could define the minimum threshold precision, according to the time waveforms, as the precision at which the response is chaotic irrespective of the initial point. This threshold precision could, for instance, be set to $p = 21$ for $f_3(x)$ and $p = 23$ for $f_6(x)$ as acceptable thresholds but not very safe. The requirements on time waveforms would lead us to prefer adhering to the safer precision thresholds decided in the previous section handling key-points of the bifurcation diagram instead of going for the acceptable threshold only.

5.2.3 Periodicity of the Generated Sequence

The sequences corresponding to different parameters (p , $f(x)$, λ , and x_0) which are generated by the digitally implemented logistic map with finite precision do not follow an identified, continuous manner as detailed in the previous subsection. It would be quite useful to point out which combinations of the parameters yield responses other than those expected through mathematical analysis. Reaching such combinations could be described as an attempt to solve the inverse problem where the value of x_{i+1} is known

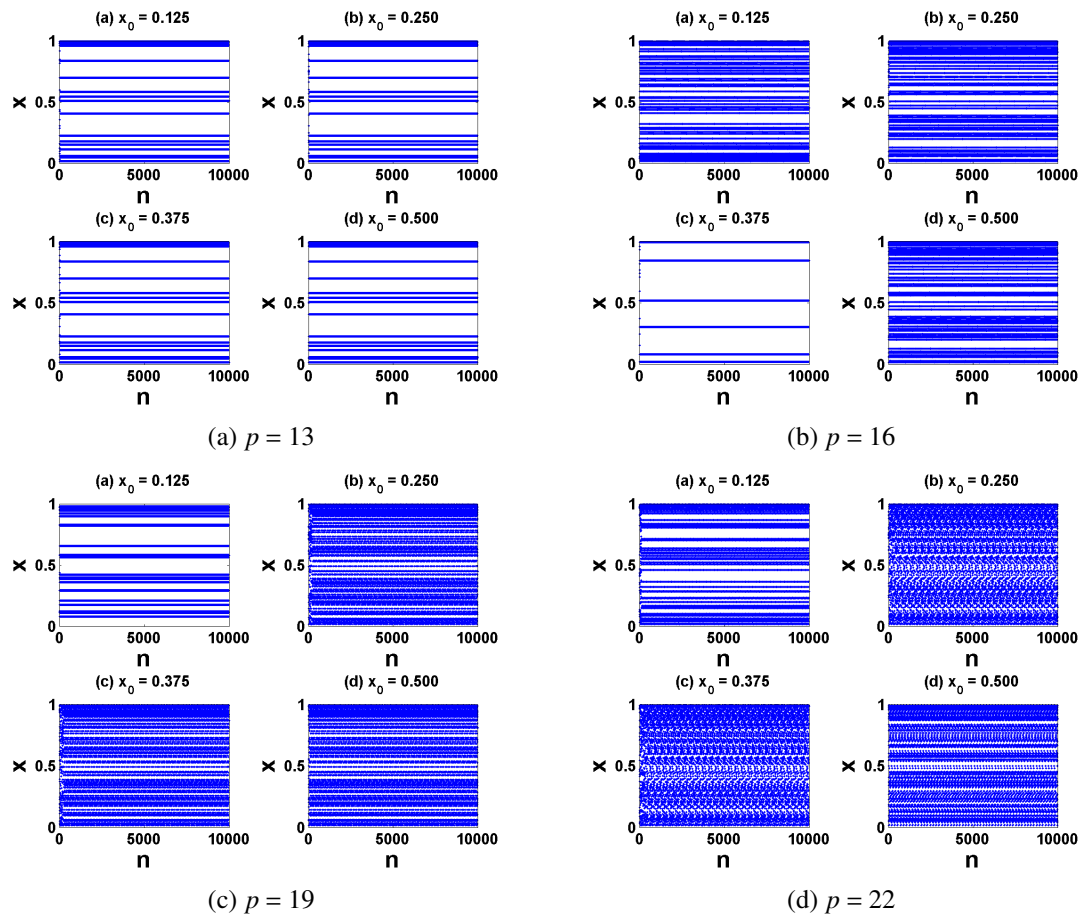


Figure 5.11: Time waveforms of $f_6(x)$ at $\lambda = 3.984375$ starting at different initial conditions and various precisions

and the question is: what is the value of x_i that had yielded it?, for given values of p and λ , as well as an order of execution $f(x)$. The answer to such a question would not be straight forward, because the register holding the successive values of x has a finite length. Consequently, the real number, possibly irrational, yielded by solving the inverse problem should be mapped to finite precision fixed-point arithmetic. However, the non-linearity of the relation representing the logistic map makes it hard to decide whether the mapped value should be lower or higher than the analytical solution. The number of steps above or below in the used precision cannot be easily decided either.

The type of the response in the steady state of the recurrence is obtained in terms of the length of the period formed by the successive solutions, i.e., a sequence of k unique values is described as “period- k ”. The factors that affect k include: the order of execution of the operations (f), the value of the control parameter λ , the initial point at which the recurrence starts x_0 , in addition to the precision p . For high precisions, parallel programming will probably be considered to overcome long runtime. However, the following primarily results have been obtained for rather low precisions.

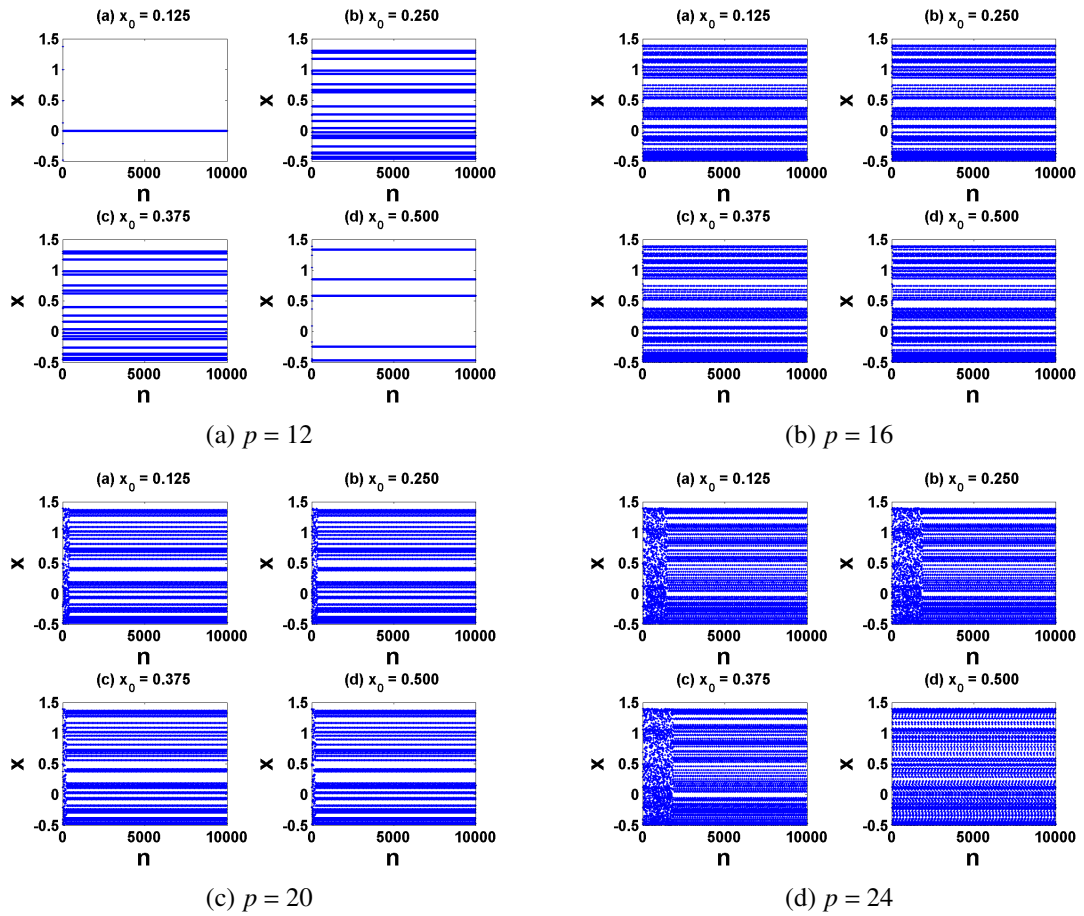


Figure 5.12: Time waveforms of $f_3(x)$ at $\lambda = -1.9375$ starting at different initial conditions and various precisions

5.2.3.1 Positive Logistic Map (Positive Control Parameter Case)

Figure 5.16(a) shows the maximum period, or maximum k , obtained with different orders of execution plotted versus precision. It seems that $f_3(x) = (\lambda(1-x))x$ yields relatively higher periods, yet some other orders are not bad. The best are $f_2(x)$, $f_3(x)$, and $f_6(x)$ as we had decided before from the bifurcation diagrams in section 5.1. It could be noticed that higher precisions provide more levels among which the solution(s) could take their values allowing higher values for k .

5.2.3.2 Mostly Positive Logistic Map (Negative Control Parameter Case)

Figure 5.16(b) shows how the logistic map with negative control parameter exhibits longer periods than that with positive control parameter and at lower precisions. Moreover, as precision increases, nearly all orders generate sequences with an acceptable maximum period, i.e., the dependence of the results on the order of execution is no longer that much important at higher precisions. This indicates the merits of the proposed logistic map with negative control parameter over the conventional logistic map with positive control parameter.

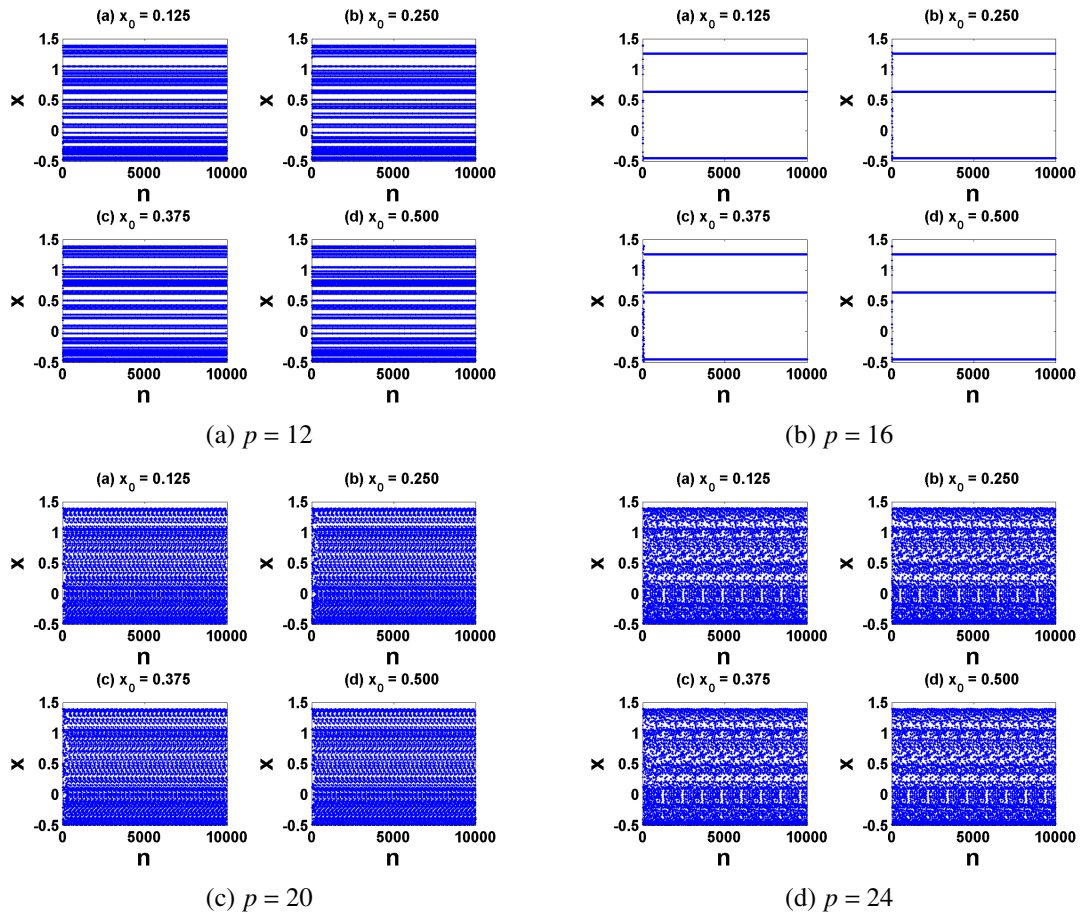


Figure 5.13: Time waveforms of $f_6(x)$ at $\lambda = -1.9375$ starting at different initial conditions and various precisions

5.2.4 Maximum Lyapunov Exponent

As previously mentioned in Chapter 2, MLE is an indication whether the system exhibits chaotic behavior or not. Chaotic orbits are characterized by their sensitive dependence on initial conditions, i.e., how orbits starting at close initial conditions differ as the iteration index $n \rightarrow \infty$. Analytically, the largest value for MLE is obtained at $\lambda_{max} = 4$, which corresponds to maximum chaotic behavior. A chaotic orbit is a synonym for aperiodicity, i.e., period $\rightarrow \infty$ under the impractical assumptions $n \rightarrow \infty$ and $p \rightarrow \infty$. On the other hand, finite precision systems and finite time simulations do not allow this ideal case to happen. Instead, both the number of iterations and the allowed set of different numbers that could be generated are limited. Two methods for calculating MLE are explained in the next two subsections.

5.2.4.1 Analytical Derivative Formula

Finite time calculations could be used to evaluate MLE with the previously proposed formula

$$MLE = \lim_{n \rightarrow \infty} \left(\frac{1}{n} \sum_{i=0}^{n-1} \ln |f'(x_i)| \right) \quad (5.2)$$

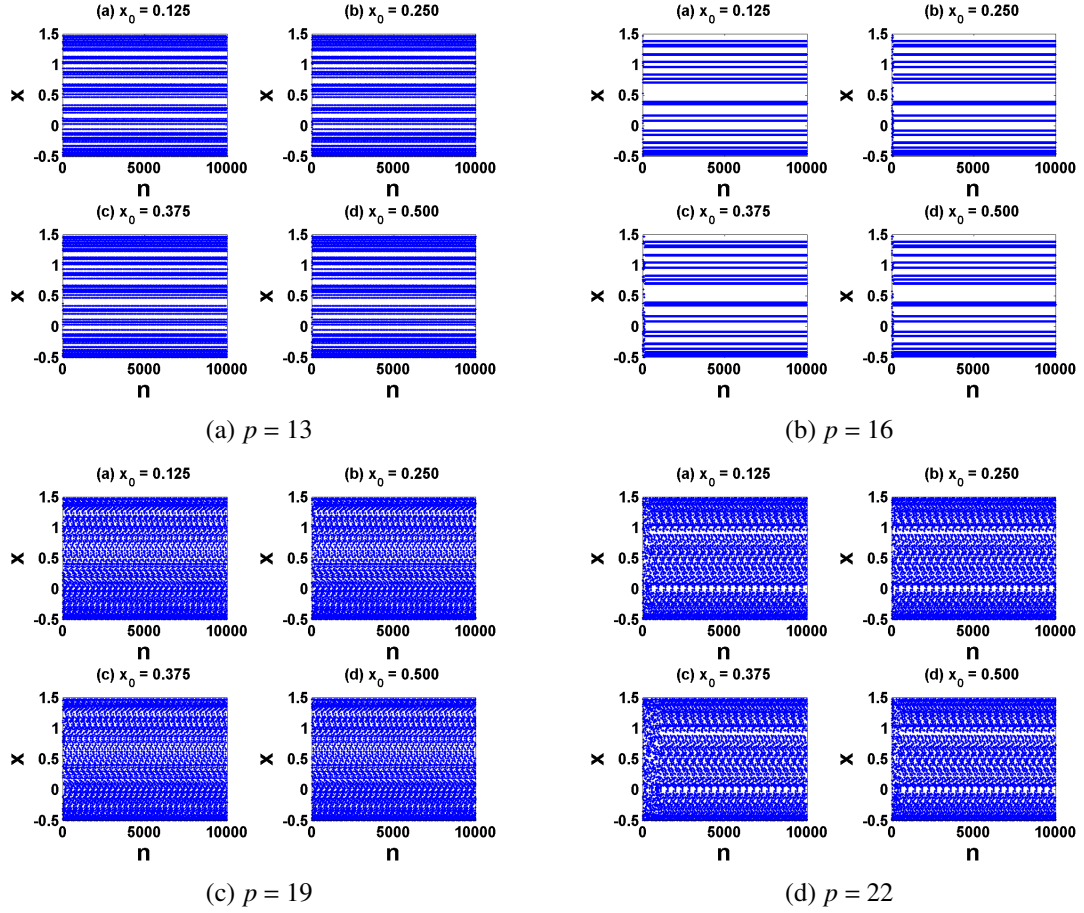


Figure 5.14: Time waveforms of $f_3(x)$ at $\lambda = -1.984375$ starting at different initial conditions and various precisions

choosing n large enough for MLE value to reach its steady state. In the analytical derivative formula, we calculate MLE as follows

$$MLE = \frac{1}{n} \sum_{i=0}^{n-1} \ln |\lambda(1 - 2x_i)| \quad (5.3)$$

such that $n = 100,000$.

5.2.4.2 Numerical Approximation of First Derivative

Is the previous way used to evaluate $f'(x_i) = \lambda(1 - 2x_i)$ not totally correct with finite precisions? Shall we instead use the numerical approximations for first derivative which are more compatible with a discrete map than continuous differentiation rules? These questions are investigated in the following discussion. Several numerical approximations were tried in our simulations: forward difference formula, backward difference formula, central difference formula, and five point method. All of them yielded quite similar results, yet the rest of the discussion handles forward difference formula for simplicity.

Consider the forward difference definition of the first derivative

$$f'(x_i) = \frac{f(x_i + \Delta) - f(x_i)}{\Delta}$$

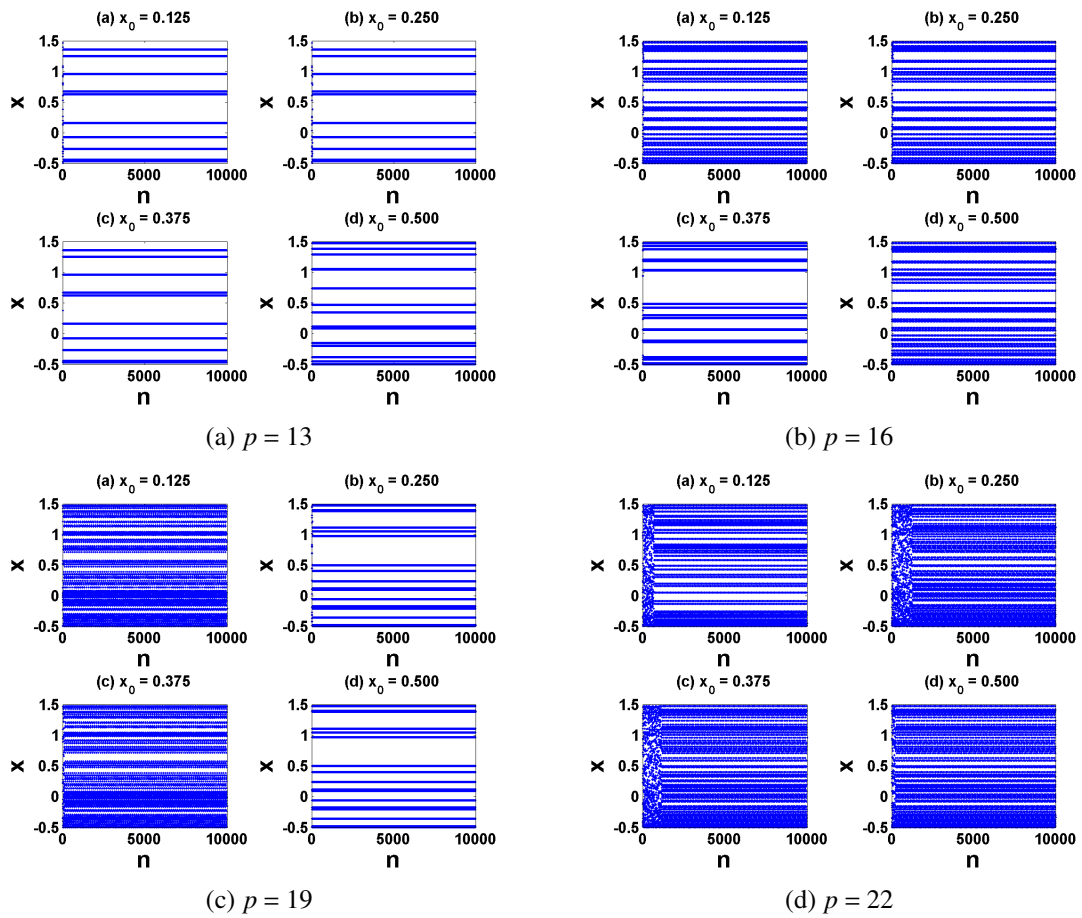


Figure 5.15: Time waveforms of $f_6(x)$ at $\lambda = -1.984375$ starting at different initial conditions and various precisions

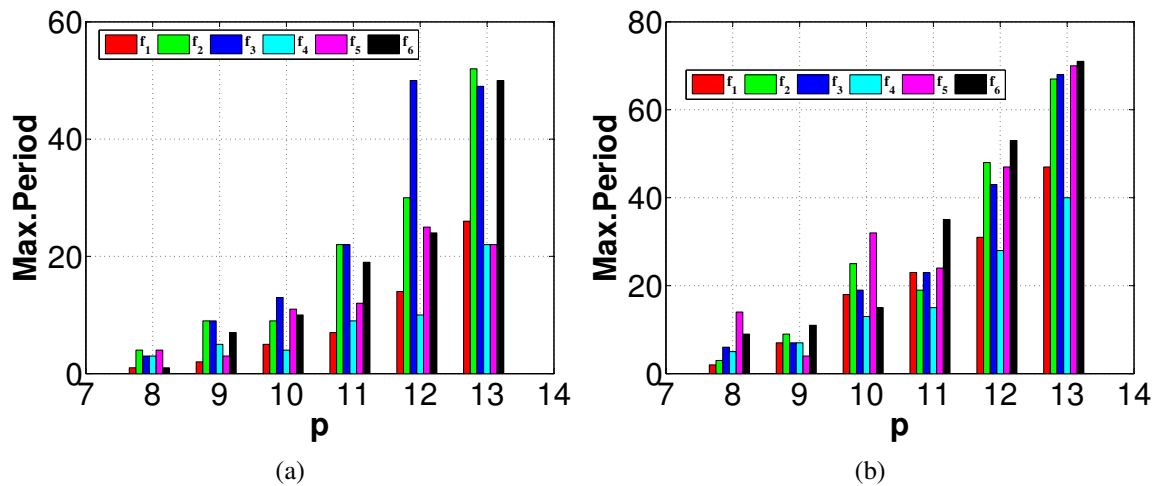


Figure 5.16: Maximum period obtained with different orders of execution plotted versus precision $p = 8 \rightarrow 13$ for (a) $\lambda > 0$ and (b) $\lambda < 0$

The logarithm for the absolute value of this quantity is

$$\ln \left| \frac{f(x_i + \Delta) - f(x_i)}{\Delta} \right|$$

since $\Delta > 0$, it yields

$$\ln|f(x_i + \Delta) - f(x_i)| - \ln \Delta$$

Substituting in the MLE equation

$$\begin{aligned} \text{MLE} &= \frac{1}{n} \sum_{i=0}^{n-1} [\ln|f(x_i + \Delta) - f(x_i)| - \ln \Delta] \\ &= \frac{1}{n} \sum_{i=0}^{n-1} \ln|f(x_i + \Delta) - f(x_i)| - \frac{1}{n} n \ln \Delta \\ &= \left[\frac{1}{n} \sum_{i=0}^{n-1} \ln|f(x_i + \Delta) - f(x_i)| \right] - \ln \Delta \end{aligned}$$

where Δ theoretically cannot be less than 2^{-p_f} which is the minimum representable value. However, for practical issues and to avoid overflow, Δ is set to quite higher value. Thus,

$$MLE = \frac{1}{n} \sum_{i=0}^{n-1} \ln \frac{|f(x_i + \Delta) - f(x_i)|}{\Delta} \quad (5.4)$$

and

$$MLE = \left[\frac{1}{n} \sum_{i=0}^{n-1} \ln|f(x_i + \Delta) - f(x_i)| \right] - \ln \Delta \quad (5.5)$$

both theoretically seem to be equivalent correct ways of evaluating MLE. In fact, they yield equal results in double-precision. It is worth mentioning that both methods were tried in double-precision floating-point precisions for $\lambda = 3.9375$ and both yielded $\text{MLE} = 0.525998$ after 100,000 iterations and $\text{MLE} = 0.525981$ with $\Delta = 2^{(-35)}$ in the second method. On the other hand, $\lambda = 3.96875$ yielded $\text{MLE} = 0.583946$ and $\lambda = 3.984375$ yielded $\text{MLE} = 0.62966$. This value for Δ has been carefully chosen in order that it does not cause overflow. However, results obtained in limited precisions do not match this theoretical hypothesis that the two methods are equivalent. Some of the obtained results do not match and others are so strange that they make us doubt the validity of both approaches for calculating MLE in finite precision fixed-point arithmetic and the reliability of the numbers computed through them in deciding how much chaotic a system is. Yet, the problems in calculating MLE using both methods disappear completely at quite higher precisions and dominate only at relatively low precisions less than $p = 22$. A safe precision threshold from the viewpoint of MLE calculation could be set to $p \geq 22$ as illustrated by Fig. 5.17 and 5.18.

We conclude that at rather low precisions, MLE cannot be used as a standalone indicator of chaotic behavior without considering the corresponding sequence length. The value of MLE calculated through either of the two methods could be falsely positive while the output sequence is clearly periodic. This result come in accordance with the discussion presented in [69]. The procedure presented in this chapter could be repeated for generalized versions and other maps according to the allowed ranges of different parameters in order to study the impact of finite precision fixed-point implementation on their properties. It could also be extended to floating-point arithmetic implementations in both real and complex plane.

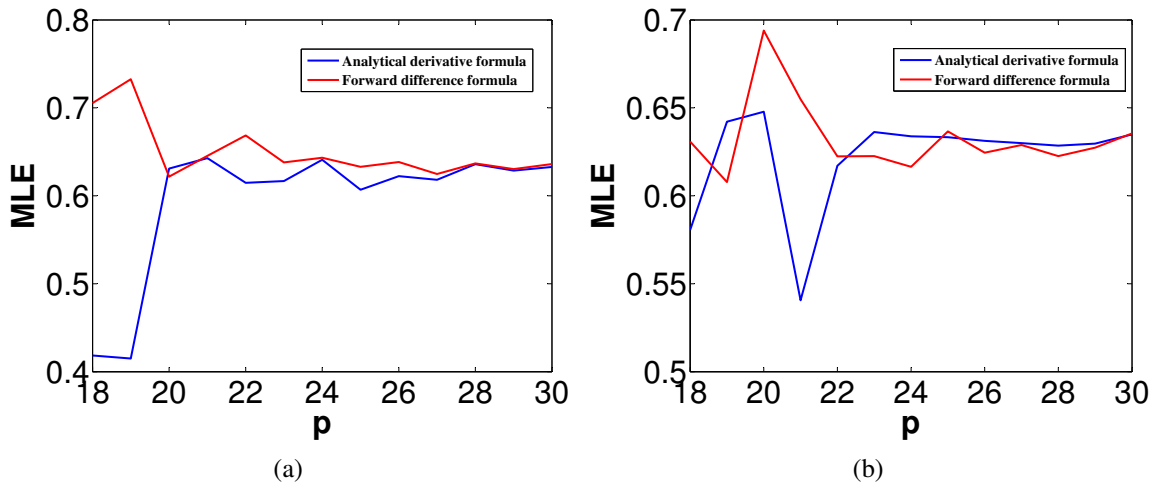


Figure 5.17: MLE evaluation in two different methods using $f_3(x)$ starting at $x_0 = 0.125$ at (a) $\lambda = 3.984375$ and (b) $\lambda = -1.984375$

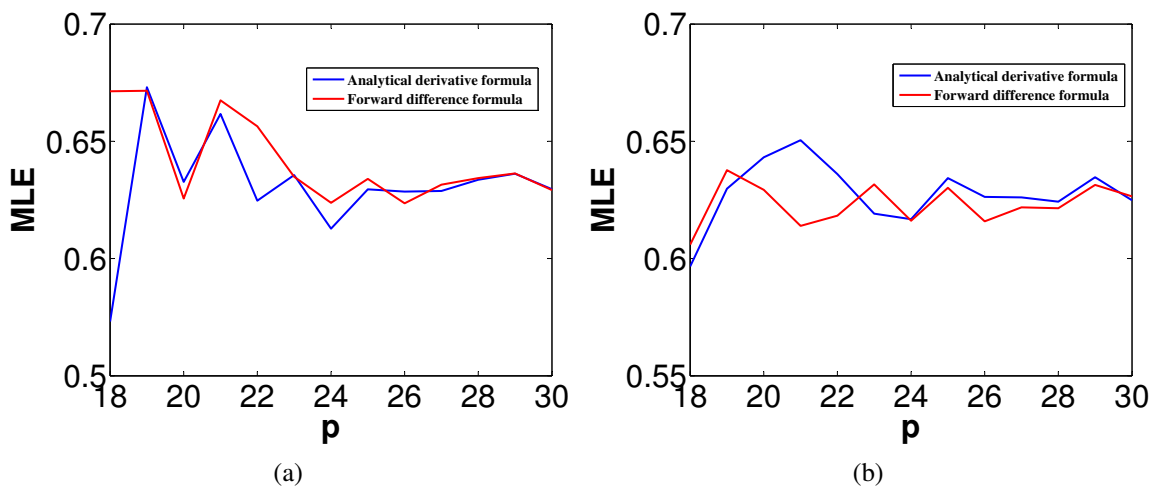


Figure 5.18: MLE evaluation in two different methods using $f_6(x)$ starting at $x_0 = 0.125$ at (a) $\lambda = 3.984375$ and (b) $\lambda = -1.984375$

Chapter 6: Elementary Functions: The Power Operation $z = x^y$

Besides the basic arithmetic operations, IEEE Std 754-2008 specifies additional operations which are recommended for all supported arithmetic formats as discussed in Chapter 2. In a specific programming environment, these recommended operations might be represented by operators or by functions whose names might differ from those in the standard. In this chapter, we first highlight the remarkable importance of the power function in many applications, besides our proposed general powering map. For normal and subnormal floating-point numbers, reasons for the complexity of evaluating pow have been listed in Chapter 2. Yet, when one or both of the arguments of pow is a special value (of which there are numerous combinations), defining and constructing results that both mathematically make sense and conform to various standards is a challenge. Many indeterminate cases arise from passing these special values as arguments for pow. In our study, we are concerned with the special values of the operands of the power function $z = x^y$. The most important special values are ± 0 , ± 1 , \pm infinity, and NaN. Our study is organized as follows.

- Studying how the IEEE Standard for Floating-Point Arithmetic 754-2008, C99 and C11 standards define the correct results of this operation when dealing with the special values of the operands covering nine different single variable power functions which are: $(+1)^y$, $(+0)^y$, $(+\infty)^y$, $(\text{NaN})^y$, $(-1)^y$, $(-0)^y$, $(-\infty)^y$, $(x)^{(+\infty)}$, and $(x)^{(-\infty)}$. By $(\pm\infty)^y$ and $(x)^{\pm\infty}$, we mean $\lim_{n \rightarrow \pm\infty} n^y$ and $\lim_{n \rightarrow \pm\infty} x^n$ respectively, yet we use the former for simplicity.
- Proposing a mathematically justified definition for the correct results of the power function over the real field on the occurrence of these special values as operands for the function. The definition attempts to satisfy the continuity of the studied function as we go from plane to plane (complex, real, integer and natural) without violating other mathematical properties. The missing expressions in the standards for complex power function are covered mathematically as well.
- Testing different software implementations of the binary floating-point power function for how they deal with these special values. Our tests include gcc compiler version 4.5.2 along with math library and MathCW library, MATLAB versions 6.5, R2012a, R2012b, and R2014b, Octave version 3.6.2, and Mathematica versions 8 and 9.
- Testing different software implementations of the decimal floating-point power function for how they deal with these special values. Our tests include gcc compiler version 4.5.2 along with DecNumber, Intel, and MathCW decimal libraries.
- Classifying the behavior of different programming languages from the point of view of how much they conform to the current standards and whether they include any form of inconsistency among them or incompatibility between different versions of the same software.

6.1 When do we Need to Perform the Calculation x^y ?

Equations that model the behavior of systems that are of significant importance to engineers frequently involve the exponentiation operation x^y . Real world applications that need to calculate a certain value using an equation involving computing x^y are numerous. Exponentiation is used extensively in many fields with applications such as:

- Finance and economics: compound interest.
- Biology: bacterial decay and population growth.
- Chemistry: chemical reaction kinetics and radioactive decay.
- Electromagnetics and physics: wave behavior.
- Computer science: public-key cryptography.
- Our own example of utilizing it in a chaotic map proposed in Chapter 4.

6.2 Definitions Proposed by IEEE 754-2008, C99, and C11 Standards

Language standards such as C99, C11, as well as the IEEE Standard for Floating-Point Arithmetic 754 (IEEE Std 754-2008) specify the expected behavior of floating-point arithmetic in computer programming environments and the handling of special values and exception conditions. This section shows the correct results defined for the various expressions formed by different combinations of the special values of the operands for the power function. These standards are approximately consistent except for some points which are not clearly declared in the IEEE standard.

6.2.1 IEEE 754-2008 Standard for Floating-Point Arithmetic (IEEE Std 754-2008)

IEEE Std 754-2008 published in August 2008 [2] includes nearly all of the original IEEE Std 754-1985 standard. Tables 6.1 and 6.2 show the results for $\text{pow}(x,y)$ for the special values of the operands x and y as defined in IEEE Std 754-2008. Those results show that the standard does not propose a clear definition for the cells with ND (ND stands for Not Defined cases in IEEE Std 754-2008) where the base $x = \pm\infty$ except for the case $(\pm\infty)^{(\pm 0)} = +1$ and where the exponent $y = \pm\infty$ except for the cases $(\pm 0)^{(+\infty)} = +0$, $(\pm 0)^{(-\infty)} = +\infty$, and $(\pm 1)^{(\pm\infty)} = +1$. This can be overcome by considering $\text{pow}(+\infty, y) = \text{pow}(+0, -y)$ and $\text{pow}(-\infty, y) = \text{pow}(-0, -y)$ as the function $(+0)^y$ is the reciprocal of $(+\infty)^y$. Similarly, the function $(-0)^y$ is the reciprocal of $(-\infty)^y$. This would give the same results as those defined in C99 and C11 standards as discussed in the following subsection.

The results show that the power function with the base equal to negative one is considered a special case only with the exponent equal to ± 0 , $\pm\infty$, non integer, or NaN. Other integer values for the exponent are not covered in the special cases section of

Table 6.1: Results for x^y for $x = +\infty, -\infty, +0, -0, +1, -1$ and NaN (QNaN) as defined in IEEE Std 754-2008, the result is QNaN for SNaN operands

$y \backslash x$	+0	-0	odd int > 0	even int > 0	non int > 0	odd int < 0	even int < 0	non int < 0	$+\infty$	$-\infty$	NaN
$+\infty$	+1	+1	ND	ND	ND	ND	ND	ND	ND	ND	NaN
$-\infty$	+1	+1	ND	ND	ND	ND	ND	ND	ND	ND	NaN
+0	+1	+1	+0	+0	+0	$+\infty$	$+\infty$	$+\infty$	+0	$+\infty$	NaN
-0	+1	+1	-0	+0	+0	$-\infty$	$+\infty$	$+\infty$	+0	$+\infty$	NaN
+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1
-1	+1	+1	-1	+1	NaN	-1	+1	NaN	+1	+1	NaN
NaN	+1	+1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Table 6.2: Results for x^y for $y = +\infty$ and $-\infty$ as defined in IEEE Std 754-2008 where $\text{NaN}^{(\pm\infty)} = \text{NaN}$

$y \backslash x$	$x < -1$	$-1 < x < 0$	$0 < x < 1$	$1 < x$	+0	-0	+1	-1	$+\infty$	$-\infty$
$+\infty$	ND	ND	ND	ND	+0	+0	+1	+1	ND	ND
$-\infty$	ND	ND	ND	ND	$+\infty$	$+\infty$	+1	+1	ND	ND

the standard. These other integer exponents are treated normally as negative one raised to the power of even integer which yields positive one and negative one raised to the power of odd integer which yields negative one. It should be noted that the standard specifies two types of NaN: signaling NaN (SNaN) and quiet NaN (QNaN) as detailed in Chapter 2. An example where SNaN may arise in some environments is the presence of uninitialized variable on the right hand side of an assignment. The standard further specifies that attempts to evaluate a function outside its domain shall return a QNaN and signal the invalid operation exception. Hence, for any SNaN operands, according to IEEE Std 754-2008, the operation $\text{pow}(x, y)$ is invalid and the result is a QNaN. In all our tables and for the rest of our discussion, we only consider QNaN operands and present the corresponding operation results. It might be assumed that $\text{pow}(+1, \text{NaN})$ and $\text{pow}(\text{NaN}, \pm 0)$ should produce a NaN. However, many math libraries return +1 for $\text{pow}(+1, y)$ for any real number y , and even if y is $\pm\infty$. Similarly, they produce +1 for $\text{pow}(x, \pm 0)$ even when x is ± 0 or $\pm\infty$. Another considerable case is $\text{pow}(x, y)$ for finite $x < 0$ and finite non-integer y , shall an implementation allow this case? What result shall it return then? These cases are discussed in detail in section 6.3.

According to IEEE Std 754-2008, the domain of $\text{pow}(x, y)$ is $[-\infty, +\infty] \times [-\infty, +\infty]$. Two additional power functions are defined in the standard that provide a more strict and limited domain alternatives. The first is $\text{powr}(x, y)$ which restricts the base x to be positive and returns a NaN when NaN operands or indeterminate expressions occur. The second is $\text{pown}(x, n)$ where the exponent n must be an integer. We are concerned with $\text{pow}(x, y)$ as it represents the widest domain among the three alternatives and it is provided

by most of the implementations unlike the two other functions which are not implemented in several libraries. Another considerable note is that a single mathematical expression may result in different results according to different definitions of various functions in IEEE Std 754-2008, e.g., $\sqrt{-0}$ results in different values according to the chosen function to perform this operation, $\text{sqrt}(-0)$, $(-0)^{0.5}$ and $(-0)^{\frac{1}{2}}$ are mathematically equivalent while the standard defines them as follows:

1. $\text{sqrt}(-0) = -0$
2. $\text{pow}(-0, y) = +0$, for $y > 0$ and not an odd integer, i.e., $\text{pow}(-0, 0.5) = +0$
3. $\text{rootn}(-0, 2) = (-0)^{\frac{1}{2}} = +0$

Interested readers may find more information about the peculiarities of the IEEE standard in [54, 55, 58] and potential bugs in $\text{pow}(x, y)$ in [65].

6.2.2 C99 and C11 Standards

C99 standard for the C language [1] and C11 [3] are versions of the International standard concerned with and specifying form and interpretation of programs in the programming language C. This specification does not define the behavior of signaling NaNs, where it generally uses the term NaN to denote quiet NaNs. Functions with NaN operand(s) return a NaN result with no floating-point exception raised except where stated otherwise. Tables 6.3 and 6.4 show the results for $\text{pow}(x, y)$ for the special values of the operands x and y as defined in both C99 and C11 standards. It is worth mentioning that $\text{pow}(\pm 0, -\infty)$ was added to the special cases in C11 and defined to be equal to $+\infty$. However, that was implicitly included in $\text{pow}(x, -\infty) = +\infty$, for $|x| < 1$. Both versions propose the same results as those in IEEE Std 754-2008 and add definitions for most of the cases which were not defined in it [107].

6.3 Mathematical Discussion

Analyzing the possible mathematical definitions upon which the power function could be defined, we first thought of defining the real power function $\text{pow}_r(x, y)$ for simpler implementation as suggested in Chapter 4. The real power is the operation which the standards define. However, even when we still assume real operands, complex values may result from expressions in which the base is negative and the exponent is a non integer. Thus, the need for the complex power function $\text{pow}_c(x, y)$ arose. A main reason for using real numbers is that they contain all limits, i.e., in mathematical terminology the reals are complete. Completeness implies that there are no gaps nor missing points in the real number line. IEEE 754's floating-point numbers approximate the familiar field of real numbers R algebraically completed by defining the signed zero ± 0 and the adjunction of $\pm\infty$ and NaN. NaNs mainly result when NaN operands or indeterminate expressions occur.

Indeterminate mathematical expressions are those which are not definitively or precisely determined. Examples on such expressions involve certain forms of limits are

Table 6.3: Results for x^y for $x = +\infty, -\infty, +0, -0, +1, -1$ and NaN as defined in C99 and C11 standards

$y \backslash x$	+0	-0	odd int > 0	even int > 0	non int > 0	odd int < 0	even int < 0	non int < 0	$+\infty$	$-\infty$	NaN
$+\infty$	+1	+1	$+\infty$	$+\infty$	$+\infty$	+0	+0	+0	$+\infty$	+0	NaN
$-\infty$	+1	+1	$-\infty$	$+\infty$	$+\infty$	-0	+0	+0	$+\infty$	+0	NaN
+0	+1	+1	+0	+0	+0	$+\infty$	$+\infty$	$+\infty$	+0	$+\infty$	NaN
-0	+1	+1	-0	+0	+0	$-\infty$	$+\infty$	$+\infty$	+0	$+\infty$	NaN
+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1	+1
-1	+1	+1	-1	+1	NaN	-1	+1	NaN	+1	+1	NaN
NaN	+1	+1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Table 6.4: Results for x^y for $y = +\infty$ and $-\infty$ as defined in C99 and C11 standards where $\text{NaN}^{(\pm\infty)} = \text{NaN}$

$y \backslash x$	$x < -1$	$-1 < x < 0$	$0 < x < 1$	$1 < x$	+0	-0	+1	-1	$+\infty$	$-\infty$
$+\infty$	$+\infty$	+0	+0	$+\infty$	+0	+0	+1	+1	$+\infty$	$+\infty$
$-\infty$	+0	$+\infty$	$+\infty$	+0	$+\infty$	$+\infty$	+1	+1	+0	+0

said to be indeterminate when the limiting behavior of individual parts of the expression is not sufficient to determine the overall limit. For example, $\lim_{x \rightarrow 0} \frac{f(x)}{g(x)}$ where $\lim_{x \rightarrow 0} f(x) = \lim_{x \rightarrow 0} g(x) = 0$, represents an indeterminate limit of the form $0/0$. The value of the overall limit could vary according to the limiting behavior of the combination of the two functions, e.g., $\lim_{x \rightarrow 0} \frac{x}{x^2} = \infty$, $\lim_{x \rightarrow 0} \frac{x}{x} = 1$, while $\lim_{x \rightarrow 0} \frac{x^2}{x} = 0$. There are seven indeterminate forms involving the special values 0, 1, and ∞ :

$$\frac{0}{0}, 0 \times \infty, \frac{\infty}{\infty}, \infty - \infty, 0^0, \infty^0, 1^\infty$$

Mathematically speaking, we can define each of the studied functions as follows.

6.3.1 The Base $x = \text{Positive One}$ ($+1^y$)

For the case when the base $x = \text{positive one}$, the result is +1 for any finite exponent. For an infinite exponent the result can be considered +1 for discrete inputs (Base = +1 and Exponent = $+\infty$ or $-\infty$) but other values may occur if originating from limits, e.g.,

$$\lim_{n \rightarrow +\infty} \left(1 + \frac{1}{n}\right)^n = e \quad (6.1)$$

$$\lim_{n \rightarrow +\infty} \left(1 + \frac{b}{n}\right)^n = e^b \quad (6.2)$$

Another interesting point is the argument of whether we take the limit from right or left, e.g., $\lim_{x \rightarrow (+1)^+} (x)^{+\infty} = +\infty$, while $\lim_{x \rightarrow (+1)^-} (x)^{+\infty} = +0$.

The number positive one is the multiplicative identity. If we consider the power operation as an extension of the multiplication operation, the result is positive one regardless how many times one is multiplied by itself. Thus, the result of the power operation when the base = +1 is +1 regardless of the exponent even if it is a NaN (QNaN), i.e., $(+1^y) = +1 \forall y$ when considered as discrete inputs. Limits may yield different results as shown above, but no way of such backward tracing or presubstitution for values originating from limits exist in the state-of-the-art of programming to the best of our knowledge. This definition for $(+1^y)$ matches the definition given by the standards in table 6.1 and 6.3.

6.3.2 The Base $x = \text{Positive Zero } (+0^y)$

For the case when the base $x = \text{positive zero}$, the result is +0 when the exponent y is positive finite ($0 < y < +\infty$). The result is $+\infty$ when the exponent is negative finite ($-\infty < y < 0$). The other possibilities to study are infinite values of y , $y = \pm 0$, and $y = \text{NaN}$. In general, a NaN operand should yield a NaN result except in the cases discussed in subsection 6.3.4.

For infinite exponent, $(+0)^{+\infty} = +0$ which could be proved by induction. Another proof depends on $\lim_{n \rightarrow +\infty} (\frac{1}{n})^n \rightarrow +0$. Let

$$z = \lim_{n \rightarrow +\infty} \left(\frac{1}{n}\right)^n.$$

The logarithm for both sides yields

$$\ln z = \ln \lim_{n \rightarrow +\infty} \left(\frac{1}{n}\right)^n = \lim_{n \rightarrow +\infty} \ln \left(\frac{1}{n}\right)^n = \lim_{n \rightarrow +\infty} -n \ln n,$$

where both n and $\ln n$ are monotonically increasing functions which approach infinity as n tends to infinity. The product of these two functions is thus infinity which is then multiplied by negative one to yield negative infinity. Therefore, $\ln z = -\infty$. Hence, $z = e^{-\infty} = +0$

The case $y = -\infty$ yields $(+0)^{-\infty} = +\infty$, where $\lim_{n \rightarrow +\infty} \left(\frac{1}{n}\right)^{-n} \rightarrow +\infty$ following the same limiting procedure as above.

Lastly, for the $y = 0$ case $(+0)^{+0}$ can be considered +1 for direct (discrete) inputs. However, if the values for the base $x = +0$ and the exponent $y = +0$ originate from limits, the expression $(+0)^{(+0)}$ yields different values:

$$\lim_{n \rightarrow +\infty} \left(\frac{1}{n}\right)^{1/n} = +1 \quad (6.3)$$

$$\lim_{n \rightarrow +\infty} \left(\frac{1}{e^n}\right)^{1/n} = \frac{1}{e} \quad (6.4)$$

$$\lim_{n \rightarrow +\infty} \left(\frac{1}{a^n}\right)^{1/n} = \frac{1}{a}, |a| > 1 \quad (6.5)$$

$$\lim_{n \rightarrow +\infty} \left(\frac{1}{a^{n^2}}\right)^{1/n} = +0, |a| > 1 \quad (6.6)$$

The value of $(+0)^{(-0)}$ can be studied in a similar manner. For such expressions, limit operations are performed to justify particular choices for tiny, or huge arguments. However, programmers have frequently disagreed on how those limits can be taken. The limiting result depends on whether x or y approach its limit more rapidly. The increase in the value of $(+0)^{+0}$ that originates from limits when the base $(+0)$ comes from $e^{-\infty}$ or $a^{-\infty}$

(for $a > 1$) is due to the rapid decrease of e^{-n} and a^{-n} more than $1/n$ as n approaches infinity. Thus, $\lim_{n \rightarrow +\infty} (\frac{1}{e^n})^{1/n}$ and $\lim_{n \rightarrow +\infty} (\frac{1}{a^n})^{1/n}$ (for $a > 1$) are positive and less than one. Similarly, $\lim_{n \rightarrow +\infty} (\frac{1}{a^n})^{1/n}$ (for $a < -1$) is negative and greater than negative one. To detect bounds of $(+0)^{+0}$, let us consider $a = +\infty$ and dealing with the limit $\lim_{n \rightarrow +\infty} (\frac{1}{a^n})^{1/n} = \lim_{n \rightarrow +\infty} ((+0)^n)^{1/n} = +0$. Similarly, $a = -\infty$ and dealing with the limit $\lim_{n \rightarrow +\infty} (\frac{1}{a^n})^{1/n} = \lim_{n \rightarrow +\infty} ((-0)^n)^{1/n} = -0$. Therefore, according to the origination of operands in the expression $(0)^0$ and their limiting behavior, we could get results z such that $|z| \leq 1$.

The overall result of $(+0)^y$ is

$$(+0)^y = \begin{cases} +0, & 0 < y \text{ (including } y = +\infty) \\ +\infty, & y < 0 \text{ (including } y = -\infty) \\ \text{by definition (chosen as } +1 \text{ for discrete inputs),} & y = \pm 0 \\ \text{NaN,} & y = \text{NaN} \end{cases} \quad (6.7)$$

This summary matches the definition given by the standards in table 6.1 and 6.3.

6.3.3 The Base $x = \text{Positive Infinity}$ $(+\infty)^y$

For the case when the base $x = \text{positive infinity}$, the result is $+\infty$ when the exponent y is positive finite ($0 < y < +\infty$). The result is $+0$ when the exponent is negative finite ($-\infty < y < 0$). The other possibilities to study are infinite values of y , $y = \pm 0$, and $y = \text{NaN}$.

For infinite exponent, $(+\infty)^{+\infty} = +\infty$, where $\lim_{n \rightarrow +\infty} n^n \rightarrow +\infty$ while $(+\infty)^{-\infty} = +0$, where $\lim_{n \rightarrow +\infty} n^{-n} \rightarrow +0$ following the same limiting procedure done for $(+0)^{+\infty}$.

For the $y = 0$ case, $(+\infty)^{+0}$ can be considered $+1$ for direct (discrete) inputs. However, if the values for the base $x = +\infty$ and the exponent $y = +0$ originate from limits, the expression $(+\infty)^{+0}$ yields different values:

$$\lim_{n \rightarrow +\infty} n^{1/n} = +1 \quad (6.8)$$

$$\lim_{n \rightarrow +\infty} (e^n)^{1/n} = e^{+1} = e \quad (6.9)$$

$$\lim_{n \rightarrow +\infty} (a^n)^{1/n} = a^{+1} = a, |a| > 1 \quad (6.10)$$

$$\lim_{n \rightarrow +\infty} (a^{n^2})^{1/n} = +\infty, |a| > 1 \quad (6.11)$$

The value of $(+\infty)^{(-0)}$ can be studied in a similar manner. Similar to the case $(+0)^{+0}$, the increase in the value of $(+\infty)^{+0}$ that originates from limits when the base $= +\infty$ comes from $e^{+\infty}$ or $a^{+\infty}$ (for $a > 1$) is due to the rapid increase of e^n and a^n more than n as n approaches infinity. Thus, $\lim_{n \rightarrow +\infty} (e^n)^{1/n}$ and $\lim_{n \rightarrow +\infty} (a^n)^{1/n}$ (for $a > 1$) are greater than positive one. Similarly, $\lim_{n \rightarrow +\infty} (a^n)^{1/n}$ (for $a < -1$) is less than negative one. To detect bounds of $(+\infty)^{+0}$, let us consider $a = +\infty$ and dealing with the limit $\lim_{n \rightarrow +\infty} (a^n)^{1/n} = \lim_{n \rightarrow +\infty} ((+\infty)^n)^{1/n} = +\infty$. Similarly, $a = -\infty$ and dealing with the limit $\lim_{n \rightarrow +\infty} (a^n)^{1/n} = \lim_{n \rightarrow +\infty} ((-\infty)^n)^{1/n} = -\infty$. Therefore, according to the origination of operands in the expression $(+\infty)^0$ and their limiting behavior, we could get results z such that $|z| \geq 1$.

The overall result of $(+\infty)^y$ is

$$(+\infty)^y = \begin{cases} +\infty, & 0 < y \text{ (including } y = +\infty) \\ +0, & y < 0 \text{ (including } y = -\infty) \\ \text{by definition (+1 for discrete inputs),} & y = \pm 0 \\ \text{NaN,} & y = \text{NaN} \end{cases} \quad (6.12)$$

This summary matches the definition given by the standards in table 6.1 and 6.3. It also satisfies the mathematical reciprocity relation between $+0$ and $+\infty$ where we expect $(+\infty)^y = 1/(+0)^y$.

6.3.4 The Base $x = \text{NaN}$ (NaN^y)

As stated earlier, we only discuss quiet NaN operands which result from indeterminate expressions. Any expression involving a NaN operand results in a NaN except for $(+1)^{\text{NaN}} = +1$ (as discussed earlier) and $\text{NaN}^{(\pm 0)}$ explained here. An indeterminate expression, e.g., a limit in the form $\frac{0}{0}$ has a value; a solution exists but its value is unknown to us. We may need to perform further steps to evaluate its limit such as applying l'Hôpital's rule to get rid of this NaN. Another example is the form $\infty - \infty$ which may come from previous operations overflowing due to narrow precision, in this case we can get rid of this NaN by using a wider precision. Thus, for QNaN case, $\text{NaN}^{(\pm 0)}$ can be dealt with as an indeterminate number raised to the power of ± 0 which results in $+1$.

Considering a QNaN resulting from an indeterminate number,

$$\text{NaN}^{(\pm 0)} = +1 \quad (6.13)$$

$$\text{NaN}^{(y)} = \text{NaN}, y \neq \pm 0. \quad (6.14)$$

This definition matches the definition given by the standards in table 6.1 and 6.3.

6.3.5 The Base $x = \text{Negative One}$ (-1^y)

For the case when the base $x = \text{negative one}$, the result is $+1$ for zero exponent and even integer exponent. The result is -1 for odd integer exponent and is complex for a non integer exponent. An example for this last condition is $(-1)^{(0.5)}$ for the case of positive non integer exponent and $(-1)^{(-0.5)}$ for the case of negative non integer exponent. We may propose several definitions for the result in the real plane.

Definition 6.3.1. *Considering $(-1)^{0.5}$ to be the square root of a negative number which is not defined in the real plane, we may give the result as NaN. Thus $(-1)^{(-0.5)} = \frac{1}{(-1)^{0.5}} = \text{NaN}$ as it is a division operation that involves a NaN operand.*

The problem with definition 6.3.1 is that a NaN result causes other NaN results to appear in subsequent operations which leads to meaningless final outcomes in algorithms. Furthermore, many exceptions in the following operations may occur, e.g., when comparing with this NaN result.

Definition 6.3.2. *Considering $x^y = |x|^y$, for any non-integer y . Hence $(-1)^{0.5} = (+1)^{0.5} = +1$ and $(-1)^{(-0.5)} = (+1)^{(-0.5)} = +1$.*

Obviously, this definition is wrong mathematically because, in the real plane, $(-1)^{1/3} = -1$ while $1^{1/3} = 1$, hence $(-1)^{1/3} \neq 1^{1/3}$.

Definition 6.3.3. Considering $x^y = (\text{sign}x)|x|^y$, we get $(-1)^{0.5} = -(+1)^{0.5} = -1$ and $(-1)^{(-0.5)} = -(+1)^{(-0.5)} = -1$.

This definition is also wrong mathematically because if we raise the right hand side of $(-1)^{0.5} = -1$ to the second power we get $(-1)^2 = +1$ not -1 as expected from the left hand side, and the equality does not hold.

Definition 6.3.4. Considering $(-1)^y = (-1)^{m/n}$, we may find the simplest form of m/n and define

$$(-1)^{m/n} = \begin{cases} +1, & m \text{ even and } n \text{ odd} \\ -1, & \text{both } m \text{ and } n \text{ odd} \\ \text{NaN}, & n \text{ even} \end{cases}$$

The problem with this definition is its discontinuity and the slightly higher complexity than other definitions in order to find the simplest form m/n before making the decision. It is important to note that odd roots of -1 such as $(-1)^{1/3}$ have a solution in the real plane. So, for a rational number in its simplest form m/n where m and n do not share common factors, the value of $(-1)^{m/n}$ has a solution in the real plane for odd n and does not have a solution for even n . The set of rational numbers is known to be infinitely dense (between any two numbers belonging to the set there exist infinitely many numbers which belong to the same set). This density makes $(-1)^{m/n}$ have a solution in infinitely many points and stay undefined in infinitely many other points. This definition appears to be the most sound mathematically within the real plane but the authors do not know of any current computer system that applies it.

Definition 6.3.5. The expression $(-1)^y$ for y non integer has a sequence in producing various complex results in the form $+1e^{i\theta}$, i.e., magnitude = $+1$ and various phase θ where $0 \leq \theta < 2\pi$.

Considering $(-1)^y = (-1)^{m/n}$, we may find the simplest form of m/n and define

$$\begin{aligned} (-1)^{m/n} &= (e^{i\pi})^{\frac{m}{n}} \\ &= e^{i(\pi+2k\pi)(\frac{m}{n})}, \quad k = 0, 1, 2, \dots, n-1 \end{aligned}$$

We only describe rational $y = m/n$ because all the floating-point numbers are in fact rational numbers. For binary floating-point numbers, the denominator n is always even. An even root of -1 is not defined in the real numbers and only has a complex solution. For example, if we concentrate on $(-1)^{\frac{y}{2}}$ where y is a positive odd integer

$$(-1)^{\frac{y}{2}} = \begin{cases} +i, & y = 1, 5, \dots \\ -i, & y = 3, 7, \dots \end{cases}$$

while it yields their complex conjugates for $(-1)^{\frac{-y}{2}}$

$$(-1)^{\frac{-y}{2}} = \begin{cases} -i, & y = 1, 5, \dots \\ +i, & y = 3, 7, \dots \end{cases}$$

For decimal floating-point numbers, $y = m/n$ in its simplest form may have an odd denominator. For example, in the case of $y = 0.6 = 3/5$ there is a real solution for $(-1)^{3/5}$ which is -1 . Obviously this definition provides multiple solutions. In some cases, none of them may be real. Hence, this definition is not suitable for the real power functions defined in the standards that do not allow complex results and are confined to the real plane.

Definition 6.3.6. *Defining the real power $\text{pow}_r(x,y)$ as the real component of the complex $\text{pow}_c(x,y)$.*

The problem with this definition is that different expressions may have the same real part but different imaginary parts. For example, $(-1)^{1/2} = +i$ and $(-1)^{3/2} = -i$, it is unreasonable to assign zero for both although they are different. The use of this definition in all cases of negative base and non integer exponent makes the problem even worse.

The first definition (Definition 6.3.1, $(-1)^{0.5} = \text{NaN}$) is consistent with the standards and with the historical mathematical definitions which state that even roots of negative numbers are undefined. The NaN result here is used to propagate an error resulting from the invalid operation $\text{pow}_r(x,y)$ for finite $x < 0$ and finite non-integer $y/2$ which produces an invalid complex result not acceptable for the real power function. However, the first definition produces the mathematically wrong NaN result for odd roots.

For future systems and standards, it might be useful to consider definition 6.3.4 for the real power function and definition 6.3.5 for the complex power function, probably with a way to return only one single solution out of the set of complex solutions.

For an infinite exponent, assuming integer, if originating from limits $\lim_{n \rightarrow +\infty} (-1)^n$ does not have a unique result because $\lim_{n \rightarrow +\infty} (-1)^{2n} = +1$ while $\lim_{n \rightarrow +\infty} (-1)^{2n+1} = -1$. For direct (discrete) inputs, we shall consider infinity as an unknown large even integer as it can be expressed as a multiple of two. Thus, $(-1)^{+\infty} = +1$ and $(-1)^{-\infty} = 1/(-1)^{+\infty} = +1$.

The overall result of $(-1)^y$ is (using definition 6.3.4)

$$(-1)^y = \begin{cases} +1, & y = \pm 0, \pm\infty, \text{ or even integer} \\ -1, & y \text{ odd integer} \\ +1, & y = m/n \text{ with } m \text{ even and } n \text{ odd} \\ -1, & y = m/n \text{ with both } m \text{ and } n \text{ odd} \\ \text{NaN}, & y = m/n \text{ with } n \text{ even} \\ \text{NaN}, & y = \text{NaN} \end{cases} \quad (6.15)$$

This summary matches the definition given by the standards in table 6.1 and 6.3 except for the $y = m/n$ cases where it provides more details.

6.3.6 The Base $x = \text{Negative Zero } (-0^y)$

For the case when the base $x = \text{negative zero}$, the result is $+0$ when the exponent is a positive even integer, -0 when it is a positive odd integer, $+\infty$ when it is a negative even integer, and $-\infty$ when it is a negative odd integer (excluding $y = 0$). The other possibilities to study are infinite values of y , $y = \pm 0$, and y non integer. Given the definitions of $(+0)^y$ and $(-1)^y$, the cases for $(-0)^y$ are easily derived.

For infinite exponent, $(-0)^{+\infty} = (-1)^{+\infty} \times (+0)^{+\infty} = +1 \times +0 = +0$. Similarly, $(-0)^{-\infty} = (-1)^{-\infty} \times (+0)^{-\infty} = +1 \times +\infty = +\infty$.

For zero exponent, $(-0)^{+0} = (-1)^{+0} \times (+0)^{+0} = +1 \times (+0)^{+0} = (+0)^{+0}$ which means that $(-0)^{+0}$ can take the same values as $(+0)^{+0}$ according to how the base 0 and the exponent 0 originated and the same for $(-0)^{-0}$.

For non integer $y = m/n$, the value $(-0)^{(m/n)} = (-1)^{m/n} \times (+0)^{m/n}$ yields

$$(-0)^{(m/n)} = \begin{cases} +0, & 0 < m/n, m \text{ even and } n \text{ odd} \\ -0, & 0 < m/n, \text{ both } m \text{ and } n \text{ odd} \\ +\infty, & m/n < 0, m \text{ even and } n \text{ odd} \\ -\infty, & m/n < 0, \text{ both } m \text{ and } n \text{ odd} \\ \text{NaN}, & n \text{ even} \end{cases}$$

The overall result of $(-0)^y$ is

$$(-0)^y = \begin{cases} +0, & 0 < y \text{ and } y \text{ an even integer or } y = +\infty \\ -0, & 0 < y \text{ and } y \text{ an odd integer} \\ +\infty, & y < 0 \text{ and } y \text{ an even integer or } y = -\infty \\ -\infty, & y < 0 \text{ and } y \text{ an odd integer} \\ \text{by definition (+1 for discrete inputs),} & y = \pm 0 \\ +0, & y = m/n, 0 < y, m \text{ even and } n \text{ odd} \\ -0, & y = m/n, 0 < y, \text{ both } m \text{ and } n \text{ odd} \\ +\infty, & y = m/n, y < 0, m \text{ even and } n \text{ odd} \\ -\infty, & y = m/n, y < 0, \text{ both } m \text{ and } n \text{ odd} \\ \text{NaN}, & y = m/n, n \text{ even} \\ \text{NaN}, & y = \text{NaN} \end{cases} \quad (6.16)$$

This summary matches the definition given by the standards in table 6.1 and 6.3 except for the $y = m/n$ cases where it provides more details.

6.3.7 The Base $x = \text{Negative Infinity } (-\infty^y)$

Similarly, the case $(-\infty^y) = (-1)^y \times (+\infty)^y$. Hence, the overall result of $(-\infty)^y$ is

$$(-\infty)^y = \begin{cases} +\infty, & 0 < y \text{ and } y \text{ an even integer or } y = +\infty \\ -\infty, & 0 < y \text{ and } y \text{ an odd integer} \\ +0, & y < 0 \text{ and } y \text{ an even integer or } y = -\infty \\ -0, & y < 0 \text{ and } y \text{ an odd integer} \\ \text{by definition (+1 for discrete inputs),} & y = \pm 0 \\ +\infty, & y = m/n, 0 < y, m \text{ even and } n \text{ odd} \\ -\infty, & y = m/n, 0 < y, \text{ both } m \text{ and } n \text{ odd} \\ +0, & y = m/n, y < 0, m \text{ even and } n \text{ odd} \\ -0, & y = m/n, y < 0, \text{ both } m \text{ and } n \text{ odd} \\ \text{NaN}, & y = m/n, n \text{ even} \\ \text{NaN}, & y = \text{NaN} \end{cases} \quad (6.17)$$

This summary matches the definition given by the standards in table 6.1 and 6.3 except for the $y = m/n$ cases where it provides more details. It is obvious that $(-\infty)^y = 1/(-0)^y$, i.e., the reciprocity between -0 and $-\infty$ remains valid under this definition. The definitions of the current standard, which do not provide the details for $y = m/n$, fail to maintain the mathematical relation $(-\infty^y) = (-1)^y \times (+\infty)^y$ for a non-integer y .

6.3.8 The Cases of $y = \pm\infty$

The last two single variable power functions to study are those with the exponent $y = \pm\infty$. These two functions do not need detailed discussion as their results were discussed within the previous cases just explained. They match the definition given by the standards in table 6.2 and 6.4 such that

$$x^{+\infty} = \begin{cases} +0, & 0 \leq |x| < 1 \\ +1, & 1 = |x| \\ +\infty, & 1 < |x| \text{ (including } x = \pm\infty) \\ \text{NaN}, & x = \text{NaN} \end{cases} \quad (6.18)$$

$$x^{-\infty} = \begin{cases} +\infty, & 0 \leq |x| < 1 \\ +1, & 1 = |x| \\ +0, & 1 < |x| \text{ (including } x = \pm\infty) \\ \text{NaN}, & x = \text{NaN} \end{cases} \quad (6.19)$$

Tables 6.5 and 6.6 show the results of the floating point power operation defined over the real field $\text{pow}_r(x,y)$ for the special values as proposed by our mathematical discussion where BD stands for “by definition”. Different definitions may arise from operands originating from limits, the tables provide the defined value = +1 used for discrete inputs.

Table 6.5: Results for x^y for $x = +\infty, -\infty, +0, -0, +1, -1$ and NaN as proposed by our mathematical discussion, the result is QNaN for SNaN operands

y \ x	+0	-0	odd int > 0	even int > 0	non int > 0	odd int < 0	even int < 0	non int < 0	$+\infty$	$-\infty$	NaN
$+\infty$	BD= +1	BD= +1	$+\infty$	$+\infty$	$+\infty$	+0	+0	+0	$+\infty$	+0	NaN
$-\infty$	BD= +1	BD= +1	$-\infty$	$+\infty$	$-\infty,$ $+\infty,$ NaN	-0	+0	-0, +0, NaN	$+\infty$	+0	NaN
+0	BD= +1	BD= +1	+0	+0	+0	$+\infty$	$+\infty$	$+\infty$	+0	$+\infty$	NaN
-0	BD= +1	BD= +1	-0	+0	-0, +0, NaN	$-\infty$	$+\infty$	$-\infty,$ $+\infty,$ NaN	+0	$+\infty$	NaN
+1	+1	+1	+1	+1	+1	+1	+1	+1	BD= +1	BD= +1	+1
-1	+1	+1	-1	+1	-1, +1, NaN	-1	+1	-1, +1, NaN	BD= +1	BD= +1	NaN
NaN	+1	+1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

These cases were discussed in the hope that future implementations introduce a clear method for the choice of the assigned value by definition, or they may find a way to include

Table 6.6: Results for x^y for $y = +\infty$ and $-\infty$ as proposed by our mathematical discussion

$\begin{matrix} x \\ y \end{matrix}$	$x < -1$	$-1 < x < 0$	$0 < x < 1$	$1 < x$	± 0	+1	-1	$\pm\infty$	NaN
$+\infty$	$+\infty$	+0	+0	$+\infty$	+0	BD= +1	BD= +1	$+\infty$	NaN
$-\infty$	+0	$+\infty$	$+\infty$	+0	$+\infty$	BD= +1	BD= +1	+0	NaN

them in the process of evaluating the function. Moreover, a broader, more comprehensive and mathematically sound definition is proposed for the case $(-1)^y$ for y non integer. It was consequently extended to the cases $(-0)^y$ and $(-\infty)^y$. This definition was proposed in the hope that future standards and implementations may consider it.

6.3.9 Complex Power Function

For the floating-point power operation defined over the complex field $\text{pow}_c(x, y)$, the results are the same as those of $\text{pow}_r(x, y)$ for all the special values except the cases $(-1)^y$, $(-0)^y$, $(-\infty)^y$, where y non integer, for which definition 6.3.5 is recommended, this definition specifies the result of $(-1)^y$. Consequently, $(-0)^y = +0 \times (-1)^y$ and $(-\infty)^y = +\infty \times (-1)^y$. However, the standards define them differently because they confine the function to the real plane. Complex results are allowable in our case. Hence, we discuss these expressions separately.

By non integer, it is meant that $|y| \in [0, 1]$. If an expression $(-1)^y$ occurs such that y neither belong to the set of integer numbers nor the interval $[0, 1]$, it is treated differently. In this case, y should be split into $y_{int} + y_{frac}$ where y_{int} stands for the integer part of y and y_{frac} stands for the fractional part of y . Hence,

$$\begin{aligned} (-1)^y &= (-1)^{y_{int} + y_{frac}} \\ \therefore (-1)^y &= (-1)^{y_{int}} \times (-1)^{y_{frac}} \end{aligned}$$

Attempting to evaluate $(-1)^y$ is equivalent to solving the polynomial of degree n of the form

$$z^n + 1 = 0$$

which has n solutions according to the fundamental theorem of algebra. Hence,

$$\begin{aligned} z^n &= -1 \\ \therefore z &= (-1)^{(1/n)} \end{aligned}$$

In floating-point arithmetic, y non integer is a rational number in its simplest form m/n where m and n do not share common factors. The n^{th} root(s) of -1 are found first. Then, each root is raised to power m .

Definition 6.3.7. Considering $(-1)^{m/n}$, we may define

$$\begin{aligned} (-1)^{m/n} &= (e^{i\pi})^{\frac{m}{n}} \\ &= e^{i(\pi+2k\pi)(\frac{m}{n})}, \quad k = 0, 1, 2, \dots, n-1 \end{aligned}$$

Definition 6.3.8. Considering $(-0)^{m/n} = (-1)^{m/n} \times +0$, we may define

$$\begin{aligned} (-0)^{m/n} &= +0(e^{i\pi})^{\frac{m}{n}} \\ &= +0e^{i(\pi+2k\pi)(\frac{m}{n})}, \quad k = 0, 1, 2, \dots, n-1 \end{aligned}$$

Definition 6.3.9. Considering $(-\infty)^y = (-1)^{m/n} \times +\infty$, we may define

$$\begin{aligned} (-\infty)^{m/n} &= +\infty(e^{i\pi})^{\frac{m}{n}} \\ &= +\infty e^{i(\pi+2k\pi)(\frac{m}{n})}, \quad k = 0, 1, 2, \dots, n-1 \end{aligned}$$

Definitions 6.3.7, 6.3.8 and 6.3.9 present the results of $(-1)^{(m/n)}$, $(-0)^{(m/n)}$ and $(-\infty)^{(m/n)}$ for $m/n > 0$. For $m/n < 0$, the results are the reciprocals of those presented in the definitions such that $\frac{1}{re^{i\theta}} = \frac{1}{r}e^{-i\theta}$.

Definition 6.3.8 could be arguable because it might be said that all four zeros $\pm 0 \pm i0$ are arithmetically equal. Yet, we choose to preserve the phase of a complex zero to satisfy the mathematical reciprocity relation between -0 and $-\infty$ where we expect $(-\infty)^y = 1/(-0)^y$. For example, according to our definition 6.3.8 the expression $(-0)^{1/2} = +0(e^{i\pi})^{\frac{1}{2}}$ has two solutions, which are

at $k = 0$

$$(-0)^{1/2} = +0e^{i(\pi)(\frac{1}{2})} = +0e^{i\pi/2} = +i0$$

at $k = 1$

$$(-0)^{1/2} = +0e^{i(\pi+2\pi)(\frac{1}{2})} = +0e^{i3\pi/2} = -i0$$

Its reciprocal is

$$\frac{1}{(-0)^{1/2}}$$

and it has two solutions too, which are

$$\frac{1}{+0e^{i\pi/2}} = +\infty e^{-i\pi/2} = +\infty e^{i3\pi/2} = -i\infty$$

$$\frac{1}{+0e^{i3\pi/2}} = +\infty e^{-i3\pi/2} = +\infty e^{i\pi/2} = +i\infty$$

that should equal to the two solutions of $(-\infty)^{1/2}$. According to definition 6.3.9

$$(-\infty)^{1/2} = +\infty(e^{i\pi})^{\frac{1}{2}}$$

has two solutions, which are

at $k = 0$

$$(-\infty)^{1/2} = +\infty e^{i(\pi)(\frac{1}{2})} = +\infty e^{i\pi/2} = +i\infty$$

at $k = 1$

$$(-\infty)^{1/2} = +\infty e^{i(\pi+2\pi)(\frac{1}{2})} = +\infty e^{i3\pi/2} = -i\infty$$

which means that the two solutions of each of the two equivalent expressions $(-\infty)^{1/2}$ and $\frac{1}{(-0)^{1/2}}$ are the same.

On the other hand, someone who may propose that $(-0)^{1/2} = +0$ would handle $(-\infty)^{1/2}$ in two different ways. Either define a result similar to ours and get two complex results, hence reciprocity property is no longer maintained; or evaluate $(-\infty)^{1/2}$ using reciprocity to be $+\infty$. The latter definition abusively disregards the phase of the infinite result; consequently ignoring the signs of the real and imaginary parts of this complex infinity. Whether all complex infinities should be arithmetically equal is a topological question [54]. Obviously the previous definitions provide multiple solutions for each expression. These multiple solutions arise from the n solutions of the polynomial

$$z^n + 1 = 0$$

Software implementations that allow answers belonging to the set of complex numbers only provide one of these multiple solutions according to their own peculiarities. Testing these implementations would open a wide discussion especially in the absence of standards that specify their correct results. This requires developing a robust mathematical definition of correct results before testing phase. For the rest of this chapter, we stick with the definitions given by the current standards in our comparisons and classification for the different programming languages.

6.4 Results of Different Programming Languages

This section demonstrates the results of the studied programming languages, emphasizing which expressions yield results inconsistent with the standards. It is worth mentioning that MATLAB honors the sign of zero and interprets it if further calculations are done, although it is not displayed in the ordinary (standard) output. Yet, Mathematica considers 0 to be neither positive nor negative. We tried to cover the most widely used software libraries in our study. Other implementations may have their own peculiarities, e.g., [25] covers some of the special cases we studied such that they are tabulated with no detailed analysis. However, the case of $(-\infty)^y$ for y non integer is not covered.

6.4.1 Binary Real Power

Tables 6.7 and 6.8 show which binary real power software implementations yield inconsistent results with the standard and for which expressions. Blank cells indicate that this expression yields the same result using various studied implementations and all conform to the standards.

6.4.1.1 Gcc Compiler Version 4.5.2 Along With Math Library

Gcc compiler version 4.5.2 was released in December 2010. It was tested on operating system Ubuntu 11.04-Desktop-i386 with kernel 2.6.38-8-generic using the math library. It conforms to the standards except for $(-1)^{\text{noninteger}} = -\text{NaN}$ instead of NaN.

6.4.1.2 Gcc Compiler Version 4.5.2 Along With MathCW Library (Binary Format)

The MathCW Mathematical Function Library, released in 2010, [12] is a large portable numerical library in C that provides a C99 math-function repertoire with many additional

functions, support for extended data types, support for additional I/O formats, and support for decimal, as well as binary, floating-point arithmetic [11]. The results of the MathCW library (binary format) conform to the standards.

6.4.1.3 MATLAB 6.5 Using the Function “realpow”

MATLAB 6.5, released in July 2002, provides a power function defined over the real field called (realpow) in addition to the ordinary power operator (^) which allows complex results. As shown in tables 6.7 and 6.8, the inconsistencies in the case of MATLAB 6.5 produce either NaN or “Error”. It is expected that an “Error” is reported when using realpow with an expression that produces a complex result. However, some of the expressions that yield “Error” when using realpow produce real values with the ordinary power operator (^) which allows complex results. These expressions are $(-\infty)^{(\pm\infty)}$, $(-\infty)^{\text{NaN}}$, $(-1)^{(\pm\infty)}$, $(-1)^{\text{NaN}}$, $(\text{NaN})^{(\pm\infty)}$, $(\text{NaN})^{\text{NaN}}$ and $(x)^{(\pm\infty)}$ for x negative. Moreover, their results (using (^)) conform to the standards except for $(-1)^{(\pm\infty)}$. It should be noted that, according to this conflict in its results, MATLAB 6.5 does not satisfy the continuity of the power function as we go from the complex plane to the real plane.

6.4.1.4 MATLAB R2012a, R2012b, and R2014b Using the Function “realpow”

The results of realpow in case of MATLAB R2012a, R2012b, and R2014b released in March 2012, September 2012, and October 2014 respectively, are the same as those of the ordinary power operator except for the expression $(-1)^y$, for y non integer. Having the same results for both the complex and the real power functions satisfies the continuity of the studied function as we go from the complex plane to the real plane. The case $(-1)^y$, for y non integer gives an error when using realpow, because it has a complex result which is not allowed for realpow. The significance of definition 6.3.4 in subsection 6.3.5 appears here: not all non integer powers of -1 have only complex results but one of the multiple solutions could be purely real. The fact that newer versions do not give the same results as MATLAB 6.5 overcomes some forms of inconsistency that exist in MATLAB 6.5 using the function realpow. Yet, $\text{NaN}^{(\pm 0)}$ yield NaN using newer versions instead of $+1$ while MATLAB 6.5 conforms to the standards. This is a form of incompatibility where a question arises here why does an old version agree (in evaluating a certain expression) with the standards more than the newer versions?

6.4.1.5 Octave 3.6.2 Using the Function “realpow”

Octave 3.6.2, released in May 2012, was tested on operating system Debian 6 with linux kernel 2.6.32-5-686. The function realpow was added in Octave version 3.2.0, released in June 2009, and it reports an error if any return value is complex. The expressions that yield “Error” when using realpow produce complex values with the ordinary power operator (^) which allows complex results. Inconsistencies produce either NaN or “Error” as in the case of MATLAB except for $(-\infty)^y$, for y negative non integer. On the other hand, when Octave 3.6.2 was tested on operating system Windows XP, the results of the expressions $(+1)^{(\pm\infty)}$, $(+1)^{\text{NaN}}$ and $\text{NaN}^{(\pm 0)}$ differ such that they yield NaN instead of $+1$.

On operating system Windows 7, the same expressions yield NaN except $(+1)^{(+\infty)}$ which yields +1.

6.4.1.6 Mathematica 8 and 9

Both versions 8 and 9 of Mathematica, released in November 2010 and 2012 respectively, do not provide a power function defined over the real field only. Thus, we compare their results with the standards for the expressions that yield real results only. In Mathematica [5], Indeterminate (Built-in Mathematica Symbol) is a symbol that represents a numerical quantity whose magnitude cannot be determined. Mathematica returns the symbol Indeterminate upon encountering indeterminate expressions such as those explained in section 6.3 in evaluating an expression. Whenever an indeterminate result should be returned from arithmetic computation, Mathematica prints a warning message. If Indeterminate is used as operand in an arithmetic computation, the corresponding result is also Indeterminate. The symbol Indeterminate plays a role in Mathematica similar to “NaN” in the IEEE Floating-Point Standard. In Mathematica, NaN (Computer Arithmetic Package Symbol) is the symbol used by the functions in the Computer Arithmetic Package to represent a non representable number. It is clear that Mathematica, in contradiction to the previous studied software implementations, differentiates between indeterminate expressions (originating from limits) and expressions involving NaN operands. Complex Infinity (Built-in Mathematica Symbol) represents a quantity with infinite magnitude, but undetermined complex phase.

Mathematica yields similar results to those in definition 6.3.5 in subsection 6.3.5 for expressions with negative base and non integer exponent. For example, $(-1)^y = +1e^{i\theta}$, where y non integer and consequently $(-\infty)^y$, where y positive non integer which is treated as $(-\infty)^y = (-1)^y \times (+\infty)^y = +\infty e^{i\theta}$, such that $0 \leq \theta < 2\pi$. Mathematica yields a single result for such an expression among the multiple solutions according to definition 6.3.5. The expressions $(\pm\infty)^{(\pm 0)}$, $(\pm 0)^{(\pm 0)}$ and $(\pm 1)^{(\pm\infty)}$ are indeterminate values for Mathematica, i.e., this tool does not provide an algebraically closed system.

6.4.2 Decimal Real Power

As previously detailed in Chapter 2, almost all major programming languages used for commercial applications support decimal arithmetic either directly or through libraries. Because of the increasing emphasis on decimal arithmetic, we include a study of the decimal floating-point $\text{pow}(x, y)$ for DecNumber, Intel and MathCW decimal libraries.

6.4.2.1 Gcc Compiler Version 4.5.2 Along With DecNumber Library (Version 3.68)

The DecNumber library [4] implements the General Decimal Arithmetic Specification in ANSI C which defines a decimal arithmetic that is supposed to match the decimal arithmetic in the IEEE 754 Standard for Floating-Point Arithmetic. However, the following expressions yield NaN in inconsistency with the standards; $(-\infty)^{(\pm\infty)}$, $(\pm 0)^{(\pm 0)}$, $(-1)^{(\pm\infty)}$, $(\text{NaN})^{(\pm 0)}$, $(+1)^{\text{NaN}}$, $(+1)^y$, for y non integer, $(-\infty)^y$, for y non integer and $(x)^{(\pm\infty)}$ for x negative (excluding -0).

Table 6.7: Results for x^y for $x = +\infty, -\infty, +0, -0, +1, -1$ and NaN for different binary real power implementations, two more inconsistencies for Mathematica are: $(+\infty)^{(\pm 0)}$ yielding Indeterminate, $(+\infty)^{+\infty}$ and $(\pm 0)^{-\infty}$ yielding Complex Infinity

(a)

$\begin{matrix} y \\ \backslash \\ x \end{matrix}$	± 0	non int > 0	non int < 0	int < 0
$-\infty$	Indeterminate (Mathematica)	Error (MATLAB6.5, Octave3.6.2), Complex result (Mathematica)	Error (MATLAB6.5), $+0, -0.5 \leq y$ $-0,$ $-1 \leq y < -0.5$ (Octave3.6.2)	
± 0	Indeterminate (Mathematica)		Complex Infinity (Mathematica)	Complex Infinity (Mathematica)
-1		Error (MATLAB6.5, R2012, R2014, Octave3.6.2), Complex result (Mathematica)	Error (MATLAB6.5, R2012, R2014, Octave3.6.2), Complex result (Mathematica)	
NaN	NaN (MATLABR2012, R2014, Mathematica)	Error (MATLAB6.5)	Error (MATLAB6.5)	

(b)

$\begin{matrix} y \\ \backslash \\ x \end{matrix}$	$+\infty$	$-\infty$	NaN
$-\infty$	Error (MATLAB6.5, Octave3.6.2), Complex Infinity (Mathematica)	Error (MATLAB6.5, Octave3.6.2)	Error (MATLAB6.5, Octave3.6.2)
$+1$	NaN (MATLAB6.5, R2012, R2014), Indeterminate (Mathematica)	NaN (MATLAB6.5, R2012, R2014), Indeterminate (Mathematica)	NaN (MATLAB6.5, R2012, R2014, Mathematica)
-1	NaN (MATLABR2012, R2014), Error (MATLAB6.5, Octave3.6.2), Indeterminate (Mathematica)	NaN (MATLABR2012, R2014), Error (MATLAB6.5, Octave3.6.2), Indeterminate (Mathematica)	Error (MATLAB6.5, Octave3.6.2)
NaN	Error (MATLAB6.5)	Error (MATLAB6.5)	Error (MATLAB6.5)

6.4.2.2 Gcc Compiler Version 4.5.2 Along With Intel Decimal Library (Release 2.0 Update 1)

Intel decimal library [20] is a software implementation of the IEEE 754-2008 Decimal Floating-Point Arithmetic specification. The expression $(-\infty)^y$, for y non integer, yields NaN which is inconsistent with the standards.

6.4.2.3 Gcc Compiler Version 4.5.2 Along With MathCW Library (Decimal Format)

MathCW library (decimal format) is consistent with the standards for the power function for the special values of the operands.

Table 6.8: Results for x^y for $y = +\infty$ and $-\infty$ for different binary real power implementations

(a)

$\begin{matrix} x \\ y \end{matrix}$	$x < -1$	$-1 < x < 0$	$+1$
$+\infty$	Error (MATLAB6.5, Octave3.6.2), Complex Infinity (Mathematica)	Error (MATLAB6.5, Octave3.6.2)	NaN (MATLAB6.5)
$-\infty$	Error (MATLAB6.5, Octave3.6.2)	Error (MATLAB6.5, Octave3.6.2), Complex Infinity (Mathematica)	NaN (MATLAB6.5)

(b)

$\begin{matrix} x \\ y \end{matrix}$	-1	$-\infty$	NaN
$+\infty$	Error (MATLAB6.5, Octave3.6.2)	Error (MATLAB6.5, Octave3.6.2)	Error (MATLAB6.5)
$-\infty$	Error (MATLAB6.5, Octave3.6.2)	Error (MATLAB6.5, Octave3.6.2)	Error (MATLAB6.5)

6.5 Summary of the Results of the Studied Implementations Versus Current Standards

Tables 6.9, 6.10 and 6.11 provide a summary for the results of the studied implementations of real power function and forms of inconsistency among them as well as incompatibility between different versions. For binary floating-point, there are two tables. Table 6.9 provides the results of the studied programming languages. Table 6.10 compares these results versus the results defined by the standards. For decimal floating-point, Table 6.11 compares the results of the studied libraries versus the results defined by the standards. “Stds” is an abbreviation for the standards. Blank cells indicate that the implementation conforms to the standards. The \neq marks indicate that some form of inconsistency or incompatibility occurs. “Error” results in MATLAB realpow when using realpow with an expression that produces a complex result. However, some of these expressions produced real results in MATLAB 6.5 with the ordinary power operator. Similarly, for Octave, realpow reports an error if any return value is complex.

Table 6.9: Summary for the binary real power function results

Expression	MATLAB		Octave 3.6.2	Mathematica
	6.5	R2012a,b R2014b		
$(\pm\infty)^{(\pm 0)}$				Indeterminate
$(+\infty)^{(+\infty)}$				Complex Infinity
$(-\infty)^{(+ve \text{ non int})}$	Error		Error	Complex result
$(-\infty)^{(-ve \text{ non int})}$	Error		± 0	
$(-\infty)^{(+\infty)}$	Error		Error	Complex Infinity
$(-\infty)^{(-\infty)}$	Error		Error	
$(-\infty)^{(\text{NaN})}$	Error		Error	
$(\pm 0)^{(\pm 0)}$				Indeterminate
$(\pm 0)^y, -\infty \leq y < -0$				Complex Infinity
$(+1)^{(\pm\infty)}$	NaN	NaN	NaN	Indeterminate
$(+1)^{(\text{NaN})}$	NaN	NaN	NaN	NaN
$(-1)^{\text{non int}}$	Error	Error	Error	Complex result
$(-1)^{(\pm\infty)}$	Error		Error	Indeterminate
$(-1)^{(\text{NaN})}$	Error		Error	
$(\text{NaN})^{\text{non int}}$	Error			
$(\text{NaN})^{(\pm 0)}$	+1	NaN	NaN	NaN
$(\text{NaN})^{(\pm\infty)}$	Error			
$(\text{NaN})^{(\text{NaN})}$	Error			
$(x)^{(+\infty)}, x < -1$	Error		Error	Complex Infinity
$(x)^{(+\infty)}, -1 < x < 0$	Error		Error	
$(x)^{(-\infty)}, x < -1$	Error		Error	
$(x)^{(-\infty)}, -1 < x < 0$	Error		Error	Complex Infinity

Table 6.10: Summary for the binary real power function forms of inconsistency and incompatibility

Expression	Stds.	Inconsistency			Incompatibility MATLAB
		MATLAB	Octave 3.6.2	Mathematica	
$(\pm\infty)^{(\pm 0)}$	+1			≠	
$(+\infty)^{(+\infty)}$	$+\infty$			≠	
$(-\infty)^y$, y +ve non int	$+\infty$	≠(6.5)	≠	≠	≠
$(-\infty)^y$, y -ve non int	+0	≠(6.5)	≠		≠
$(-\infty)^{(+\infty)}$	$+\infty$	≠(6.5)	≠	≠	≠
$(-\infty)^{(-\infty)}$	+0	≠(6.5)	≠		≠
$(-\infty)^{\text{NaN}}$	NaN	≠(6.5)	≠		≠
$(\pm 0)^{(\pm 0)}$	+1			≠	
$(\pm 0)^y$, $-\infty \leq y < -0$	$+\infty$ ¹			≠	
$(+1)^{(\pm\infty)}$	+1	≠	≠	≠	
$(+1)^{\text{NaN}}$	+1	≠	≠	≠	
$(-1)^{\text{non int}}$	NaN	≠	≠	≠	
$(-1)^{(\pm\infty)}$	+1	≠(6.5)	≠	≠	≠
$(-1)^{\text{NaN}}$	NaN	≠(6.5)	≠		≠
$(\text{NaN})^{\text{non int}}$	NaN	≠(6.5)			≠
$(\text{NaN})^{(\pm 0)}$	+1	≠ (R2012a,b) (R2014b)	≠	≠	≠
$(\text{NaN})^{(\pm\infty)}$	NaN	≠(6.5)			≠
$(\text{NaN})^{\text{NaN}}$	NaN	≠(6.5)			≠
$(x)^{(+\infty)}$, $x < -1$	$+\infty$	≠(6.5)	≠	≠	≠
$(x)^{(+\infty)}$, $-1 < x < 0$	+0	≠(6.5)	≠		≠
$(x)^{(-\infty)}$, $x < -1$	+0	≠(6.5)	≠		≠
$(x)^{(-\infty)}$, $-1 < x < 0$	$+\infty$	≠(6.5)	≠	≠	≠

¹Standards define $(\pm 0)^y = +\infty$, for $-\infty \leq y < -0$ except $(-0)^{\text{(-ve odd int)}} = -\infty$

Table 6.11: The decimal power function results and forms of inconsistency and incompatibility, where MathCW is completely consistent with the standards

Expression	DecNumber	Intel	Stds.	Inconsistency	
				DecNumber	Intel
$(-\infty)^{(+\infty)}$	NaN		$+\infty$	\neq	
$(-\infty)^{(-\infty)}$	NaN		$+0$	\neq	
$(\pm 0)^{(\pm 0)}$	NaN		$+1$	\neq	
$(-1)^{(\pm\infty)}$	NaN		$+1$	\neq	
$(\text{NaN})^{(\pm 0)}$	NaN		$+1$	\neq	
$(+1)^{\text{NaN}}$	NaN		$+1$	\neq	
$(+1)^{\text{non int}}$	NaN		$+1$	\neq	
$(-\infty)^{(\text{+ve non int})}$	NaN	NaN	$+\infty$	\neq	\neq
$(-\infty)^{(\text{-ve non int})}$	NaN	NaN	$+0$	\neq	\neq
$(x)^{(+\infty)}, x < -1$	NaN		$+\infty$	\neq	
$(x)^{(+\infty)}, -1 < x < 0$	NaN		$+0$	\neq	
$(x)^{(-\infty)}, x < -1$	NaN		$+0$	\neq	
$(x)^{(-\infty)}, -1 < x < 0$	NaN		$+\infty$	\neq	

Chapter 7: Conclusions and Suggestions for Future Work

Generalized cases of the 1D chaotic logistic and tent maps were proposed where the control parameter can be either positive or negative, providing a detailed analysis of the properties of each map in the two different cases. Moreover, the possible variations on the relation representing the map according to the signs of different control parameters and the resulting output ranges were presented. These variations differ in their characteristics and do not confine the output to a restricted range of fractions between 0 and 1. Thus, allowing more flexibility that fits the most recent applications such as quantitative financial modeling, traffic, weather forecasting, and others. The generalized form of these variations for the logistic map is given by:

$$x_{n+1} = \pm \lambda x_n (a \pm b x_n); \quad \lambda, a, b \in R^+ \quad (7.1)$$

while for the tent map, it is given by

$$x_{n+1} = \pm \mu \min(\pm x_n, a \mp b x_n); \quad \mu, a, b \in R^+ \quad (7.2)$$

Based on the maximum chaotic range of the output, the proposed maps can be classified as: positive map, mostly positive map, negative map, and mostly negative map. These maps differ in the ranges of both the system parameter (λ or μ) and x , the symmetry, and the key-points of the bifurcation diagram. The generalized maps were analyzed and a study on how to design any of them to suit certain specifications was conducted, proposing four scaling cases. For the logistic map, the parameters (a, b) may take one of four cases: (a, b) , $a, b \in R^+$ called the independent scaling case, $(1, b)$ called the vertical scaling case, (a, a) called the horizontal scaling, and $(a, 1)$ called the zooming case. Similar scaling cases with slight differences could be defined for the tent map. The choice of these names depends on the effect of the added parameter(s) on the bifurcation diagram. The proposed parameterized versions were analyzed for each map from the viewpoint of iteration effect, ranges of λ and x , the fixed points, the bifurcation diagrams, and MLE with respect to all system parameters. New bifurcation diagrams versus the new parameters were introduced and the general schematic of the bifurcation diagram versus λ was plotted as a function of the other system parameters, as well as the general schematic of the newly proposed bifurcation diagrams. Different design examples were presented to verify the provided design procedure according to the general schematic. The designed maps were validated for usage in encryption applications through a simple text encryption scheme.

A chaotic map has been proposed that could be considered a general form for 1D discrete maps employing the power function with the tent and logistic maps as special cases, in addition to other various newly proposed cases. It would be suitable even for maps whose iterative relations employ fractional powers instead of being based on polynomials. The completely new general map with arbitrary powers that could implement various 1D maps with different behavior utilizing the same resources is given by:

$$x_{n+1} = r \min\left(x_n^\alpha (a - b x_n)^\beta, x_n^\beta (a - b x_n)^\alpha\right), \quad a, b, \alpha, \beta \in R^+ \quad (7.3)$$

where α and β are shaping parameters that specify the shape of the resulting map response and bifurcation diagrams versus different parameters. A framework for analyzing the

proposed map mathematically and predicting its behavior for various combinations of its parameters was introduced. In addition, the transition from tent map case to logistic map case was presented and explained in detail. More special cases are presented with names chosen according to their relation to the conventional maps.

Fixed-point hardware is generally less expensive than an equivalent floating-point hardware implementation. A design guide of a computationally efficient pseudo-random number generator (PRNG) with finite precision is needed for practical considerations of chaos based communication and cryptography. Thus, a simulation for a digitally implemented 1D logistic map was proposed that could operate in both modes: positive control parameter and negative control parameter is suggested based on MATLAB fixed-point toolbox. The proposed finite precision logistic map exhibits different properties from the analytical properties that were reviewed for positive control parameter case in Chapter 2 and mathematically derived for negative control parameter case in Chapter 3. New factors must be taken into consideration besides the system parameter λ which are: the used precision, equivalently the buswidth p , the order of execution of operations chosen to implement the map $f(x)$, and the initial point at which the iterations start x_0 . A slight perturbation in any of these factors could yield massively different responses with varied properties. A criteria for choosing the precision and specifying the number of integer bits was set. An exhaustive simulation for the properties of the map over the available precisions was carried out in the range $p = 8$ to $p = 27$. The properties include: the bifurcation diagram, its key-points, time waveforms, periodicity of the generated sequence, and MLE. These properties were compared with those obtained through mathematically analyzing the map over the infinite real field. It has been decided that the precision threshold should exceed $p = 20$ by a good enough safety margin that could guarantee properties of the finite precision logistic map that are quite acceptable, given the cost savings offered by such a choice. The impact of finite precision in floating-point arithmetic implementations was also tackled briefly.

Back to the proposed general powering map and its implementation, whether in floating-point or fixed-point. Multiple differences from the standards were found in the various software implementations of the floating-point power operation in both binary and decimal formats. We analyzed the results defined by the standards mathematically and proposed a broader, more comprehensive and mathematically sound definition for the case $(-1)^y$. The definition consequently extended to the cases $(-0)^y$ and $(-\infty)^y$. We then pointed out the deviations from the current standards for several implementations classifying them into inconsistency with the standards and incompatibility among different versions of the same software implementation. We hope that our study helps future implementations to become consistent with the standards and that our mathematical proposal forms the base for future standards and implementations.

We hope that this study helps in implementing the proposed maps without losing much of their analytically studied properties. Future work may take further steps towards analyzing and implementing powering maps, exploring 2D maps, in addition to the continuous domain. The best way to implement the powering map at the parameters' values required for a specific application needs to be decided. In addition, the possibility of treating it as a 2D map in some cases could be considered such as those with a complex valued response. Moreover, the performance of the proposed maps could be compared to discretization of higher order chaotic systems such as Lorenz and Rössler attractors.

References

- [1] “C99 standard for C programming language”, ISO/IEC 9899: 1999. Information technology-Programming languages-C, <http://www.open-std.org/jtc1/sc22/wg14/www/docs/n1256.pdf>, 1999.
- [2] “IEEE standard for floating-point arithmetic”, IEEE std 754-2008, 2008.
- [3] “C11 standard for C programming language”, ISO/IEC 9899: 2011. Information technology-Programming languages-C, <http://www.open-std.org/jtc1/sc22/wg14/www/docs/n1570.pdf>, 2011.
- [4] “DecNumber library version 3.68”. webpage, <http://speleotrove.com/decimal/decnumber.html>, 2012. created on February 2012, last visited on December 2014.
- [5] “Wolfram Mathematica”. webpage, <http://www.wolfram.com/mathematica/>, 2013. last visited on December 2014.
- [6] ADDABBO, T., ALIOTO, M., FORT, A., ROCCHI, S., AND VIGNOLI, V. The digital tent map: performance analysis and optimized design as a low-complexity source of pseudorandom bits. *IEEE Transactions on Instrumentation and Measurement* 55, 5 (2006), 1451–1458.
- [7] ALLIGOOD, K. T., SAUER, T. D., AND YORKE, J. A. *Chaos: An introduction to dynamical systems*. Springer, 1996.
- [8] AMBADAN, J. T., AND JOSEPH, K. B. Asymmetrical mirror bifurcations in logistic map with a discontinuity at zero. In *National conference on nonlinear systems and dynamics, NCNSD* (2006).
- [9] BARAKAT, M. L., RADWAN, A. G., AND SALAMA, K. N. Hardware realization of chaos based block cipher for image encryption. In *International Conference on Microelectronics (ICM)* (2011), IEEE, pp. 1–5.
- [10] BASALTO, N., BELLOTTI, R., DE CARLO, F., FACCHI, P., AND PASCAZIO, S. Clustering stock market companies via chaotic map synchronization. *Physica A: Statistical Mechanics and its Applications* 345, 1 (2005), 196–206.
- [11] BEEBE, N. H. A new math library. *International Journal of Quantum Chemistry* 109, 13 (2009), 3008–3025.

- [12] BEEBE, N. H. The MathCW mathematical function library. webpage, <http://ftp.math.utah.edu/pub/mathcw/>, 2010. created on August 2010, last visited on December 2014.
- [13] BORCHERDS, P. H., AND McCAULEY, G. P. The digital tent map and the trapezoidal map. *Chaos, Solitons & Fractals* 3, 4 (1993), 451–466.
- [14] BOWEN, R., AND CHAZOTTES, J.-R. *Equilibrium states and the ergodic theory of Anosov diffeomorphisms*, vol. 470. Springer, 1975.
- [15] BRESTEN, C. L., AND JUNG, J.-H. A study on the numerical convergence of the discrete logistic map. *Communications in Nonlinear Science and Numerical Simulation* 14, 7 (2009), 3076–3088.
- [16] BURGIN, M., AND MEISSNER, G. Negative probabilities in financial modeling. *Wilmott* 2012, 58 (2012), 60–65.
- [17] CHEN, B., AND WORNELL, G. W. Analog error-correcting codes based on chaotic dynamical systems. *IEEE Transactions on Communications* 46, 7 (1998), 881–890.
- [18] CHEVILLARD, S., JOLDEŞ, M., AND LAUTER, C. Sollya: An environment for the development of numerical codes. In *Mathematical Software - ICMS 2010* (2010), vol. 6327 of *Lecture Notes in Computer Science*, Springer, pp. 28–31.
- [19] CORLESS, R. M. What good are numerical simulations of chaotic dynamical systems? *Computers & Mathematics with Applications* 28, 10 (1994), 107–121.
- [20] CORNEA, M. Intel decimal floating-point math library. webpage, <http://software.intel.com/en-us/articles/intel-decimal-floating-point-mathlibrary>, 2011. created on October 2011, last visited on December 2014.
- [21] COX, J. C., ROSS, S. A., AND RUBINSTEIN, M. Option pricing: A simplified approach. *Journal of financial Economics* 7, 3 (1979), 229–263.
- [22] CUOMO, K. M., AND OPPENHEIM, A. V. Circuit implementation of synchronized chaos with applications to communications. *Physical review letters* 71, 1 (1993), 65–68.
- [23] CURTRIGHT, T., AND ZACHOS, C. Negative probability and uncertainty relations. *Modern Physics Letters A* 16, 37 (2001), 2381–2385.
- [24] DANGOISSE, D., GLORIEUX, P., AND HENNEQUIN, D. Chaos in a CO₂ laser with modulated parameters: experiments and numerical simulations. *Physical Review A* 36, 10 (1987), 4775.
- [25] DE DINECHIN, F., ECHEVERRIA, P., LÓPEZ-VALLEJO, M., AND PASCA, B. Floating-point exponentiation units for reconfigurable computing. *ACM Transactions on Reconfigurable Technology and Systems (TRETs)* 6, 1 (2013).
- [26] DI BERNARDO, M., AND VASCA, F. Discrete-time maps for the analysis of bifurcations and chaos in DC/DC converters. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications* 47, 2 (2000), 130–143.

- [27] DIRAC, P. Bakerian lecture. the physical interpretation of quantum mechanics. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences* (1942), 1–40.
- [28] DIRAC, P. Quantum electrodynamics. *Commun. Dublin Inst. Adv. Stud., A 1* (1943).
- [29] DORNBUSCH, A., AND DE GYVEZ, J. P. Chaotic generation of PN sequences: a VLSI implementation. In *Proceedings of the 1999 IEEE International Symposium on Circuits and Systems ISCAS'99*. (1999), vol. 5, IEEE, pp. 454–457.
- [30] DUFFIE, D., AND SINGLETON, K. J. Modeling term structures of defaultable bonds. *Review of Financial studies* 12, 4 (1999), 687–720.
- [31] EDELMAN, A. The mathematics of the pentium division bug. *SIAM Rev.* 39, 1 (1997), 54–67.
- [32] ELHADJ, Z., AND SPOTT, J. The effect of modulating a parameter in the logistic map. *Chaos: An Interdisciplinary Journal of Nonlinear Science* 18, 2 (2008).
- [33] FAHMY, H. A. H., ELDEEB, T., HASSAN, M. Y., FAROUK, Y., AND EISSA, R. R. Decimal floating point for future processors. In *International Conference on Microelectronics (ICM), 2010* (2010), IEEE, pp. 443–446.
- [34] FAHMY, H. A. H., AND FLYNN, M. The case for a redundant format in floating point arithmetic. In *16th IEEE Symposium on Computer Arithmetic* (2003), IEEE, pp. 95–102.
- [35] FAHMY, H. A. H., LIDDICOAT, A. A., AND FLYNN, M. J. Improving the effectiveness of floating point arithmetic. In *Thirty-Fifth Asilomar Conference on Signals, Systems, and Computers, Asilomar, California, USA* (2001), vol. 1, pp. 875–879.
- [36] FAHMY, H. A. H., RAAFAT, R., ABDEL-MAJEED, A. M., SAMY, R., ELDEEB, T., AND FAROUK, Y. Energy and delay improvement via decimal floating point units. In *Proceedings of the 19th IEEE Symposium on Computer Arithmetic, Portland, Oregon, USA* (2009), pp. 221–224.
- [37] FEIGENBAUM, M. J. Quantitative universality for a class of nonlinear transformations. *Journal of statistical physics* 19, 1 (1978), 25–52.
- [38] FERRIE, C., AND EMERSON, J. Frame representations of quantum mechanics and the necessity of negativity in quasi-probability representations. *Journal of Physics A: Mathematical and Theoretical* 41, 35 (2008).
- [39] FEYNMAN, R. P. Negative probability. *Quantum implications: essays in honour of David Bohm* (1987), 235–248.
- [40] FLORES, B. C., SOLIS, E. A., AND THOMAS, G. Chaotic signals for wideband radar imaging. In *AeroSense 2002* (2002), International Society for Optics and Photonics, pp. 100–111.
- [41] GELL-MANN, M., AND HARTLE, J. B. Decoherent histories quantum mechanics with one real fine-grained history. *Physical Review A* 85, 6 (2012).

- [42] GU, J., AND CHEN, S. Nonlinear analysis on traffic flow based on catastrophe and chaos theory. *Discrete Dynamics in Nature and Society* 2014. Article ID 535167, 2014.
- [43] GUPTA, N. D., AND GHOSH, S. A report on the wilson cloud chamber and its applications in physics. *Reviews of Modern Physics* 18, 2 (1946).
- [44] HABUTSU, T., NISHIO, Y., SASASE, I., AND MORI, S. A secret key cryptosystem by iterating a chaotic map. In *Advances in Cryptology EUROCRYPT 91* (1991), Springer, pp. 127–140.
- [45] HAN, Y. D., HWANG, W. Y., AND KOH, I. G. Explicit solutions for negative-probability measures for all entangled states. *Physics Letters A* 221, 5 (1996), 283–286.
- [46] HANROT, G., LEFÈVRE, V., PLISSIER, P., THÉVENY, P., AND ZIMMERMANN., P. MPFR. webpage, <http://www.mpfr.org/>, 2004.
- [47] HASSAN, M. Y., ELDEEB, T., AND FAHMY, H. A. H. Algorithm and architecture for on-line decimal powering computation. In *Conference Record of the Forty Fourth Asilomar Conference on Signals, Systems and Computers (ASILOMAR), 2010* (2010), IEEE, pp. 1158–1162.
- [48] HAUG, E. G. Why so negative to negative probabilities? *Wilmott Magazine* (2004), 34–38.
- [49] HE, Y., ZHOU, J., LI, C., YANG, J., AND LI, Q. A precise chaotic particle swarm optimization algorithm based on improved tent map. In *Fourth International Conference on Natural Computation, 2008. ICNC'08.* (2008), vol. 7, IEEE, pp. 569–573.
- [50] HERNANDEZ, E. D. M., LEE, G., AND FARHAT, N. H. Analog realization of arbitrary one-dimensional maps. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications* 50, 12 (2003), 1538–1547.
- [51] HOFMANN, H. F. How to simulate a universal quantum computer using negative probabilities. *Journal of Physics A: Mathematical and Theoretical* 42, 27 (2009), 275–304.
- [52] HSIEH, D. A. Chaos and nonlinear dynamics: application to financial markets. *The journal of finance* 46, 5 (1991), 1839–1877.
- [53] JARROW, R. A., AND TURNBULL, S. M. Pricing derivatives on financial securities subject to credit risk. *The journal of finance* 50, 1 (1995), 53–85.
- [54] KAHAN, W. Branch cuts for complex elementary functions. *The State of the Art in Numerical Analysis, M. J. D. Powell and A. Iserles, Eds., Oxford University Press, NY* (1987).
- [55] KAHAN, W. Lecture notes on the status of ieee standard 754 for binary floating-point arithmetic, <http://www.eecs.berkeley.edu/~wkahan/ieee754status/IEEE754.PDF>, 1996.

- [56] KAHAN, W. A logarithm too clever by half. *World-Wide Web document* <https://www.cs.berkeley.edu/~wkahan/LOG10HAF.TXT> (2004).
- [57] KAHAN, W. How futile are mindless assessments of roundoff in floating-point computation?, 2006. <https://www.cs.berkeley.edu/~wkahan/Mindless.pdf>.
- [58] KAHAN, W. Needed remedies for the undebuggability of large-scale floating-point computations in science and engineering. Computer Science Dept. Colloquia at the University of California, Berkely, <https://www.cs.berkeley.edu/~wkahan/NeedDebug.pdf>, 2009.
- [59] KANSO, A., AND SMAOUI, N. Logistic chaotic maps for binary numbers generations. *Chaos, Solitons & Fractals* 40, 5 (2009), 2557–2568.
- [60] KHRENNIKOV, A. *Interpretations of probability*. Walter de Gruyter, 2009.
- [61] KOCAREV, L., AND JAKIMOSKI, G. Logistic map as a block encryption algorithm. *Physics Letters A* 289, 4 (2001), 199–206.
- [62] KOLMOGOROV, A. N. *Foundations of the Theory of Probability*. Chelsea Publishing Co., 1950.
- [63] KORNERUP, P., LAUTER, C., LEFÈVRE, V., LOUVET, N., AND MULLER, J.-M. Computing correctly rounded integer powers in floating-point arithmetic. *ACM Transactions on Mathematical Software (TOMS)* 37, 1 (2010).
- [64] LAUTER, C. Q., AND LEFÈVRE, V. An efficient rounding boundary test for $\text{pow}(x, y)$ in double precision. *IEEE Transactions on Computers* 58, 2 (2009), 197–207.
- [65] LEFÈVRE, V. about bug number 182574: libc6: The $\text{pow}()$ function gives incorrect results on special cases (nan, inf...). e-mail discussion, <http://lists.debian.org/debian-glibc/2003/03/msg00202.html>, 2003.
- [66] LEFÈVRE, V., MULLER, J.-M., AND TISSERAND, A. Toward correctly rounded transcendentals. *IEEE Transactions on Computers* 47, 11 (1998), 1235–1243.
- [67] LEVESON, N. G. Role of software in spacecraft accidents. *Journal of spacecraft and Rockets* 41, 4 (2004), 564–575.
- [68] LEVINSOHN, E. A., MENDOZA, S. A., AND PEACOCK-LÓPEZ, E. Switching induced complex dynamics in an extended logistic map. *Chaos, Solitons & Fractals* 45, 4 (2012), 426–432.
- [69] LI, S., CHEN, G., AND MOU, X. On the dynamical degradation of digital piecewise linear chaotic maps. *International Journal of Bifurcation and Chaos* 15, 10 (2005), 3119–3151.
- [70] LLOYD, A. L. The coupled logistic map: a simple model for the effects of spatial heterogeneity on population dynamics. *Journal of Theoretical Biology* 173, 3 (1995), 217–230.

- [71] LORENZ, E. N. Deterministic nonperiodic flow. *Journal of the atmospheric sciences* 20, 2 (1963), 130–141.
- [72] MANDAL, S., AND BANERJEE, S. Analysis and CMOS implementation of a chaos-based communication system. *IEEE Transactions on Circuits and Systems I: Regular Papers* 51, 9 (2004), 1708–1722.
- [73] MANDELBROT, B. B. *The fractal geometry of nature*, vol. 173. Macmillan, 1983.
- [74] MANSINGKA, A., AFFAN ZIDAN, M., BARAKAT, M., RADWAN, A. G., AND SALAMA, K. N. Fully digital jerk-based chaotic oscillators for high throughput pseudo-random number generators up to 8.77 Gbits/s. *Microelectronics Journal* 44, 9 (2013), 744–752.
- [75] MANSINGKA, A. S., RADWAN, A. G., ZIDAN, M. A., AND SALAMA, K. N. Analysis of bus width and delay on a fully digital signum nonlinearity chaotic oscillator. In *IEEE 54th International Midwest Symposium on Circuits and Systems (MWSCAS)* (2011), IEEE, pp. 1–4.
- [76] MARKSTEIN, P. *IA-64 and elementary functions: speed and precision*. Prentice Hall, 2000.
- [77] MATTHEWS, R. On the derivation of a “chaotic” encryption algorithm. *Cryptologia* 13, 1 (1989), 29–42.
- [78] MAY, R. M. Simple mathematical models with very complicated dynamics. *Nature* 261, 5560 (1976), 459–467.
- [79] MONNIAUX, D. The pitfalls of verifying floating-point computations. *ACM Transactions on Programming Languages and Systems (TOPLAS)* 30, 3 (2008).
- [80] MÜCKENHEIM, W., LUDWIG, G., DEWDNEY, C., HOLLAND, P. R., KYPRIANIDIS, A., VIGIER, J. P., PETRONI, N. C., BARTLETT, M. S., AND JAYNES, E. T. A review of extended probabilities. *Physics Reports* 133, 6 (1986), 337–401.
- [81] MULLER, J.-M. *Elementary functions: Algorithms and implementation*. Springer, 2006.
- [82] MULLER, J.-M., BRISEBARRE, N., DE DINECHIN, F., JEANNEROD, C.-P., LEFÈVRE, V., MELQUIOND, G., REVOL, N., STEHLÉ, D., TORRES, S., ET AL. *Handbook of floating-point arithmetic*. Springer, 2010.
- [83] MULLER, J. M., DE DINECHIN, F., AND LAUTER, C. CRlibm. webpage, <http://lipforge.ens-lyon.fr/www/crlibm/>, 2004.
- [84] NEJATI, H., BEIRAMI, A., AND MASSOUD, Y. A realizable modified tent map for true random number generation. In *51st Midwest Symposium on Circuits and Systems MWSCAS* (2008), IEEE, pp. 621–624.
- [85] NELSON, P. basic calculator (bc). webpage, https://www.gnu.org/software/bc/manual/html_mono/bc.html, 2001.

- [86] ORRELL, D., SMITH, L., BARKMEIJER, J., AND PALMER, T. N. Model error in weather forecasting. *Nonlinear processes in geophysics* 8, 6 (1999), 357–371.
- [87] PAREEK, N. K., PATIDAR, V., AND SUD, K. K. Image encryption using chaotic logistic map. *Image and Vision Computing* 24, 9 (2006), 926–934.
- [88] PATIDAR, V., SUD, K. K., AND PAREEK, N. K. A pseudo random bit generator based on chaotic logistic map and its statistical testing. *Informatica (Slovenia)* 33, 4 (2009), 441–452.
- [89] PERSOHN, K. J., AND POVINELLI, R. J. Analyzing logistic map pseudorandom number generators for periodicity induced by finite precision floating-point representation. *Chaos, Solitons & Fractals* 45, 3 (2012), 238–245.
- [90] PHATAK, S. C., AND RAO, S. S. Logistic map: A possible random-number generator. *Physical review E* 51, 4 (1995).
- [91] RADWAN, A. G. On some generalized discrete logistic maps. *Journal of Advanced Research* 4, 2 (2013), 163–171.
- [92] RADWAN, A. G., AND ABD-EL-HAFIZ, S. K. Image encryption using generalized tent map. In *IEEE 20th International Conference on Electronics, Circuits, and Systems (ICECS)* (2013), IEEE, pp. 653–656.
- [93] RADWAN, A. G., ABD-EL-HAFIZ, S. K., AND ABDELHALEEM, S. H. An image encryption system based on generalized discrete maps. In *IEEE International Conference on Electronics, Circuits and Systems (ICECS), 21st* (2014), IEEE, pp. 283–286.
- [94] RADWAN, A. G., MANSINGKA, A. S., ZIDAN, M. A., AND SALAMA, K. N. On the short-term predictability of fully digital chaotic oscillators for pseudo-random number generation. In *IEEE 20th International Conference on Electronics, Circuits, and Systems (ICECS), 2013* (2013), IEEE, pp. 373–376.
- [95] RADWAN, A. G., SOLIMAN, A. M., AND EL-SEDEEK, A. MOS realization of the modified Lorenz chaotic system. *Chaos, Solitons & Fractals* 21, 3 (2004), 553–561.
- [96] RADWAN, A. G., SOLIMAN, A. M., AND ELWAKIL, A. S. 1-D digitally-controlled multiscroll chaos generator. *International Journal of Bifurcation and Chaos* 17, 01 (2007), 227–242.
- [97] ROTHMAN, T., AND SUDARSHAN, E. Hidden variables or positive probabilities? *International journal of theoretical physics* 40, 8 (2001), 1525–1543.
- [98] RUAN, H., YAZ, E. E., ZHAI, T., AND YAZ, Y. I. A generalization of tent map and its use in EKF based chaotic parameter modulation/demodulation. In *43rd IEEE Conference on Decision and Control (CDC), 2004* (2004), vol. 2, IEEE, pp. 2071–2075.
- [99] RUELLE, D. *Chaotic evolution and strange attractors*, vol. 1. Cambridge University Press, 1989.

- [100] SAYED, W. S., RADWAN, A. G., AND FAHMY, H. A. H. Design of positive, negative, and alternating sign generalized logistic maps. *Discrete Dynamics in Nature and Society* 2015. Article ID 586783, 2015.
- [101] SCHÖLL, E. *Nonlinear spatio-temporal dynamics and chaos in semiconductors*, vol. 10. Cambridge University Press, 2001.
- [102] SCHULTE, M., AND SWARTZLANDER, E. Exact rounding of certain elementary functions. In *11th Symposium on Computer Arithmetic* (1993), IEEE, pp. 138–145.
- [103] SCULLY, M. O., WALTHER, H., AND SCHLEICH, W. Feynman’s approach to negative probability in quantum mechanics. *Physical Review A* 49, 3 (1994).
- [104] SINGH, N., AND SINHA, A. Chaos-based secure communication system using logistic map. *Optics and Lasers in Engineering* 48, 3 (2010), 398–404.
- [105] STROGATZ, S. H. *Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering*. Westview press, 2014.
- [106] SUNEEL, M. Electronic circuit realization of the logistic map. *Sadhana* 31, 1 (2006), 69–78.
- [107] THOMAS, J. IEEE 754 support in C99. presentation, <http://grouper.ieee.org/groups/754/meeting-materials/2001-07-18-c99.pdf>, 2001.
- [108] TREVISAN, A., AND PALATELLA, L. Chaos and weather forecasting: the role of the unstable subspace in predictability and state estimation problems. *International Journal of Bifurcation and Chaos* 21, 12 (2011), 3389–3415.
- [109] VÁZQUEZ-MEDINA, R., DÍAZ-MÉNDEZ, A., DEL RÍO-CORREA, J. L., AND LÓPEZ-HERNÁNDEZ, J. Design of chaotic analog noise generators with logistic map and MOS QT circuits. *Chaos, Solitons & Fractals* 40, 4 (2009), 1779–1793.
- [110] VERHULST, P. F. Notice sur la loi que la population suit dans son accroissement. correspondance mathématique et physique publiée par A. *Quetelet* 10 (1838), 113–121.
- [111] WEISSTEIN, E. W. Logistic map. From MathWorld—A Wolfram Web Resource. <http://mathworld.wolfram.com/LogisticMap.html>. last visited on December 2014.
- [112] WIGGINS, S., AND GOLUBITSKY, M. *Introduction to applied nonlinear dynamical systems and chaos*, vol. 2. Springer, 2003.
- [113] WIGNER, E. On the quantum correction for thermodynamic equilibrium. *Physical Review* 40, 5 (1932).
- [114] YI, X. Hash function based on chaotic tent maps. *IEEE Transactions on Circuits and Systems II: Express Briefs* 52, 6 (2005), 354–357.

- [115] YI, X., TAN, C. H., AND SIEW, C. K. A new block cipher based on chaotic tent maps. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications* 49, 12 (2002), 1826–1829.
- [116] ZHU, Z.-L., ZHANG, W., WONG, K.-W., AND YU, H. A chaos-based symmetric image encryption scheme using a bit-level permutation. *Information Sciences* 181, 6 (2011), 1171–1186.
- [117] ZIDAN, M. A., RADWAN, A. G., AND SALAMA, K. N. Random number generation based on digital differential chaos. In *IEEE 54th International Midwest Symposium on Circuits and Systems (MWSCAS), 2011* (2011), IEEE, pp. 1–4.
- [118] ZIDAN, M. A., RADWAN, A. G., AND SALAMA, K. N. Controllable V-shape multiscroll butterfly attractor: system and circuit implementation. *International Journal of Bifurcation and Chaos* 22, 06 (2012).

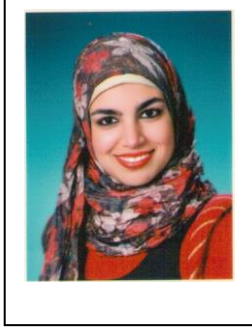
المخلص

تستخدم خرائط الفوضى و الدوال الأولية بكثرة فى العمليات الحسابية العلمية و فى العديد من التطبيقات المنتشرة على نطاق واسع. على سبيل المثال، يتم استخدام خرائط الفوضى المنفصلة أحادية البعد فى النمذجة و توليد الأعداد شبه العشوائية، كما تتطلب الحسابات المالية دالة القوى (الأس) $z=x^y$ بشدة. علاوة على ذلك، فإننا نستخدم دالة القوى فى بعض من خرائط الفوضى الجديدة المقترحة. و لكن الحساسية المفرطة لهذه الدوال تجعلها أكثر عرضة للأخطاء من العديد من العلاقات الرياضية المنفذة رقمياً حالياً. نهدف فى هذه الرسالة إلى سد الفجوة بين التحليل الرياضى للصور التقليدية و العامة لهاتين المجموعتين من المشاكل من ناحية، و تنفيذهما على الأنظمة و البرامج العددية من ناحية أخرى.

أولاً، نقترح تنويعات على إشارات المعاملات فى خرائط الفوضى المنفصلة أحادية البعد الأكثر شهرة: الخريطة اللوجستية و خريطة الخيمة. تم اقتراح أربعة أشكال مختلفة لكل من الخريطة اللوجستية و خريطة الخيمة تسمح لنطاق الناتج أن يكون له إشارة واحدة أو يتراوح بالتناوب بين إشارتين مستخدمين نفس العلاقة تقريباً بإشارات متنوعة للمعاملات. تسمح النطاقات الجديدة بنمذجة ظواهر إضافية و تعطى نواتج لها نطاقات أوسع مع تسلسل أطول. ثم نضيف معاملات مقياسية للتدرج (a,b) و التى يمكن استخدامها فى التحكم فى نطاقات النواتج وواحد من أهم خصائص خرائط الفوضى: مخطط التشعب. كما نناقش أيضاً حالات خاصة من تقنية التدرج المقترحة. ثم نقدم خريطة أسية عامة لها معاملات تشكيل (α,β) تعمل كأسس اختيارية تضيف القدرة على التحكم فى شكل استجابة الخريطة و مخطط التشعب. يمكن ضبط الأسس للحصول على خريطة الخيمة و الخريطة اللوجستية و خرائط فوضى جديدة فيما بينهما و التى أسميناها "الخريطة الانتقالية". كما نقوم بدراسة نطاقات أخرى للمعاملات الأسية.

من ناحية التنفيذ، نقوم بتحليل سلوك الخريطة اللوجستية، فى حالة المعامل الموجب و السالب، فى حالات التنفيذ محدود الدقة والتأكيد على تأثير المحدودية على الخصائص الديناميكية للخريطة مع تغيير الدقة المستخدمة. نهجنا فى هذه الدراسة جديد لأنه يستخدم أداة العلامة الكسرية الثابتة (Fixed-point toolbox) من MATLAB لمحاكاة التنفيذ على معدات FPGA التى تقوم بتنفيذ العمليات منفردة بطريقة متسلسلة الواحدة تلو الأخرى مع تنفيذ خطوة الاقتران بينهم، و نتناول ترتيب تنفيذ العمليات و تأثيره لأول مرة على حد علمنا. نوصى وفقاً لذلك بحد أدنى للدقة بالإضافة إلى الإشارة إلى مزايا خريطتنا المقترحة ذات المعامل السالب عن مثيلتها التقليدية.

أما عن دالة القوى، نقترح تعريفاً لنتائجها بالنسبة للقيم الخاصة من المتغيرات مصحوباً بالمبررات الرياضية و نختبر كيف تتناولها و تتعامل معها المعايير وتطبيقات البرمجيات المختلفة. نقدم التضارب بين التطبيقات و المعايير و نناقش عدم التوافق بين الإصدارات المختلفة من البرنامج نفسه. يمكن أن تساعد هذه الدراسة فى إنتاج تطبيقات قابلة للتكرار، أى إعطاء نفس النتيجة فى مختلف طرق تنفيذ البرامج المبنية على نفس اللغة. وبالإضافة إلى ذلك، فإن الجمع بين هذه الدراسة و الجزء السابق يمكن أن يوفر إطاراً لتحليل آثار التنفيذ محدود الدقة على سلوك الخريطة الأسية العامة المقترحة، بعد أن نقرر أفضل طريقة لتنفيذها.



وفاء صابر عبد الحليم سيد

1991\3\20

مصرية

2013\10\1

2015\--\--

الرياضيات و الفيزيكا الهندسية

ماجستير العلوم

مهندسة:

تاريخ الميلاد:

الجنسية:

تاريخ التسجيل:

تاريخ المنح:

القسم:

الدرجة:

المشرفون:

أ.د. عبد اللطيف الصديق حسين

أ.د. حسام على حسن فهمي

أ.م.د. أحمد جمعة أحمد رضوان

الممتحنون:

أ.د. ناصر حسن سويلم (الممتحن الخارجي)

كلية العلوم-جامعة القاهرة

أ.د. سلوى كمال عبد الحفيظ (الممتحن الداخلي)

أ.د. عبد اللطيف الصديق حسين (المشرف الرئيسي)

أ.د. حسام على حسن فهمي (عضو)

أ.م.د. أحمد جمعة أحمد رضوان (عضو)

عنوان الرسالة:

خرائط الفوضى العامة و الدوال الأولية بين التحليل و التنفيذ

الكلمات الدالة:

الخريطة اللوجستية ، خريطة الخيمة ، التنفيذ الرقمي، دقة محدودة ، دالة القوى

ملخص الرسالة:

يهدف في هذه الرسالة إلى سد الفجوة بين التحليل الرياضي لخرائط الفوضى العامة المنفصلة أحادية البعد و تنفيذها على الأنظمة و البرامج العددية. نقترح عدة تنويعات و تعميمات على الخريطة اللوجستية و خريطة الخيمة و نوظف دالة القوى في خريطة عامة يمكن أن تعطى كلاً منهما و خرائط أخرى جديدة. نقدم خرائط لها معامل تحكم سالب يسمح بنواتج أوسع نطاقاً تتراوح بين إشارتين و يمكن التحكم فيها عن طريق معاملات التدرج. علاوة على ذلك، نحلل الخريطة الأسية العامة المقترحة رياضياً و نتحقق من النتائج للحالات الحدودية. نقدم خريطة انتقالية ذات استجابات وسيطة بين خريطة الخيمة و الخريطة اللوجستية. ندرس الخريطة اللوجستية محدودة الدقة و نشرح تأثير المحدودية على خواصها المختلفة. كما نختبر التطبيقات المبنية على النقطة العائمة لدالة القوى بالنسبة للقيم الخاصة من المتغيرات.

خرائط الفوضى العامة و الدوال الأولية بين التحليل و التنفيذ

اعداد

وفاء صابر عبد الحليم سيد

رسالة مقدمة إلى كلية الهندسة - جامعة القاهرة
كجزء من متطلبات الحصول على درجة ماجستير العلوم
في
الرياضيات الهندسية

يعتمد من لجنة الممتحنين:

المشرف الرئيسي الاستاذ الدكتور: عبد اللطيف الصديق حسين

عضو الاستاذ الدكتور: حسام على حسن فهمى

عضو الاستاذ الدكتور: أحمد جمعة أحمد رضوان

الممتحن الداخلي الاستاذ الدكتور: سلوى كمال عبد الحفيظ

الممتحن الخارجي الاستاذ الدكتور: ناصر حسن سويلم
كلية العلوم - جامعة القاهرة

كلية الهندسة - جامعة القاهرة
الجيزة - جمهورية مصر العربية
2015

خرائط الفوضى العامة و الدوال الأولية بين التحليل و التنفيذ

اعداد

وفاء صابر عبد الحليم سيد

رسالة مقدمة إلى كلية الهندسة - جامعة القاهرة
كجزء من متطلبات الحصول على درجة ماجستير العلوم
في
الرياضيات الهندسية

تحت اشراف

أ.د. حسام على حسن فهمى
أستاذ

قسم الالكترونيات و الاتصالات الكهربائية
كلية الهندسة - جامعة القاهرة

أ.د. عبد اللطيف الصديق حسين
أستاذ

قسم الرياضيات و الفيزيكا الهندسية
كلية الهندسة - جامعة القاهرة

أ.م.د. أحمد جمعة أحمد رضوان
أستاذ مساعد

قسم الرياضيات و الفيزيكا الهندسية
كلية الهندسة - جامعة القاهرة

كلية الهندسة - جامعة القاهرة
الجيزة - جمهورية مصر العربية

2015



خرائط الفوضى العامة و الدوال الأولية بين التحليل و التنفيذ

اعداد

وفاء صابر عبد الحليم سيد

رسالة مقدمة إلى كلية الهندسة - جامعة القاهرة
كجزء من متطلبات الحصول على درجة ماجستير العلوم
في
الرياضيات الهندسية

كلية الهندسة - جامعة القاهرة
الجيزة - جمهورية مصر العربية
2015