OPEN ACCESS
Indo-JC
Indonesia Journal of Computing
ISSN 2460-9056
socj.telkomuniversity.ac.id/indojc

Ind. Journal on Computing
Vol. 6, Issue. 1, April 2021. pp. 41-52
doi:10.34818/indojc.2021.6.1.546

Implementation of $f(x) = 3(x^3 - x^2 - x) + 2$ as CSPRNG Chaos-Based Random Number Generator

Maria Rosalina Yopeng [1], Alz Danny Wowor [2]

Faculty of Information Technology, Universitas Kristen Satya Wacana
Jl. Dr. O. Notohamidjojo No 1-10, Salatiga 50714,

[1] 672013801@student.uksw.edu, [2] alzdanny.wowor@uksw.edu

Abstract

This research implemented the cubic function $f(x) = 3(x^3 - x^2 - x) + 2$ using a Fixed-Point Iteration to produce several iteration functions that can be used as random number generator. The test results obtain six iteration functions, and based on graphic visualization with Scatter plot and randomness test with mono bit test, bit block, and run test, the results only obtain two iteration functions namely $x^2 - 1 + 2/(3x)$ and $f(x) = 1 + 1/x - 2/(3x^2)$ which can produce CSPRNG Chaos-based random number. Encryption testing shows that both functions can generate keys that make plaintext and ciphertext statistically unrelated, so the $f(x) = 1 + 1/x - 2/(3x^2)$ function can be used as a CSPNRG chaos-based random number generator function.

Keywords: $f(x) = 3(x^3 - x^2 - x) + 2$, Fixed Point Iteration, CSPRNG Chaos.

Abstrak

Penelitian ini mengimplementasi fungsi Kubik $f(x) = 3(x^3 - x^2 - x) + 2$ menggunakan Fixed Point Iteration untuk menghasilkan beberapa fungsi iterasi yang dapat dijadikan sebagai pembangkit bilangan acak. Hasil pengujian diperoleh enam fungsi iterasi, dan berdasarkan visualisasi grafik dengan Scatter plot dan uji keacakan dengan uji mono bit, blok bit, dan uji run diperoleh hanya dua fungsi iterasi yaitu $x^2 - 1 + 2/(3x)$ dan $f(x) = 1 + 1/x - 2/(3x^2)$ yang dapat menghasilkan bilangan acak berbasis CSPRNG Chaos. Pengujian enkripsi menunjukkan kedua fungsi pembangkit dapat menghasilkan kunci yang membuat plainteks dan cipherteks tidak berhubungan secara statistik, sehingga fungsi $f(x) = 1 + 1/x - 2/(3x^2)$ dapat digunakan sebagai fungsi pembangkit bilangan acak berbasis CSPNRG chaos.

Kata Kunci: $f(x) = 3(x^3 - x^2 - x) + 2$, Fixed Point Iteration, CSPRNG Chaos.

I. Introduction

KEY is an important information in cryptography or information (data) security, which key must be confidential and cannot be known by unauthorized parties. Apart from being secret, key also needs to have several advantages, for example having a random nature and its periodic entries are longer than entries taken as key. These important things are the basis for cryptographers to design key generation algorithms. Many methods are used to generate good keys, one of which is to use a random number generator function based on the Cryptography Secure Pseudo Random Number Generator Chaos (CSPRNG chaos). This method usually uses a function as a generator and the output of each iteration is a collection of random numbers.

The logistical function $f(x) = rx(1 - x)$ or in the iterative form $x_{i+1} = rx_i(1 - x_i)$, is an example of a generator function that has a CSPRNG Chaos basis. This function can generate random numbers with long periodic entries, besides changing the value of x0 as an initialization will have a big effect on all

Received on January, 2021. Accepted on Mei, 2021

iteration values. This is what makes the logistic function widely used as a key generator in cryptographic algorithms [1] [2].

Algebraically, logistic function is a form of the second degree polynomial, and the research [3] has developed grade-1, degree-2, and degree-3 polynomials which have the possibility of being used as generator functions. By specifically looking at degree-3 function, the research [3] uses the function $f(x) = x^3 + 6x^2 + 19x - 20$ as a generator function.

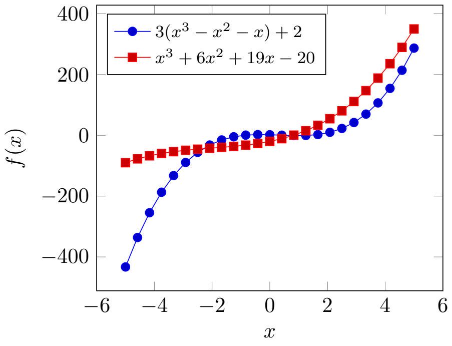

Fig. 1: *Comparison of Two Quadratic Functions*

This research is searching for other degree-3 polynomial functions to serve as the generator function. A trial-and-error process is used with a coefficient simulation $a; b; c; d$ from $ax^3 + bx^2 + cx + d$ in the interval $[0, 10]$, the function $f(x) = (x^3 + x^2 + x) + 2$ is obtained which it can generate several iteration functions and can be used as a function generator. On the other hand, visually the function $f(x) = (x^3 + x^2 + x) + 2$ has a different curve with the function $f(x) = x^3 + 6x^2 + 19x - 20$. This can be seen in the slope of the function given in Figure 1. Thus, the two functions will result significantly different values.

This research uses a Fixed Point Iteration (FPI) to change a function into several functions of iteration which can then be used as a CSPRNG Chaos-based random number generator function. Statistical tests such as Run Test, Monobit, Blockbit are carried out to ensure that each generated sequence of numbers meets the property of randomness.

II. Related Research

A. Previous Research

Previous researches are needed to see the connection with the research topic that can be used as a reference for continuation or comparison. Here are some studies related to this research topic.

The research entitled "Regeneration of Polynomial Function in the Design of Chaos CSPRNG-Based Algorithms" is the first literature review. This research is looking for polynomial function that can be used as a random number generator. Polynomial functions of degree-1, degree-2, and degree-3 are used as generating functions. The obtained results: the polynomial function can be used as a generator function, however, a good selection of constants and coefficients on the function is needed, and algebraic manipulation dexterity based on FPI is also needed, so that iterative functions can be very important in generating CSPNRG chaos [3].

The second literature review entitled "Regeneration of Quadratic Functions as Key Generators Based on the Fixed-Point Iteration Method in Cryptography" discusses the search for other quadratic functions that can be used as key generators by regenerating using the fixed-point iteration method to be an iteration function. From the regeneration search results of the quadratic function, it is known that if a quadratic function has two different real number roots, it has a great chance of being used as a generating function [4].

The third literature review is the research “Regeneration of $f(x) = x^2 - 9x - 99$ Function as CSRPNG Chaos-Based Random Number Generator". This research developed the study [1], specifically looking for other quadratic functions that can be used as random number generators. This is used as a comparison and also a reference in finding other polynomial functions, but the difference is that this study uses a cubic function as the candidate for CSPRNG Chaos-based random number generator function [5].

The research with the title “Finding an Interval Solution for Coefficients and Constants $f(x) = x - (x^2 - 3)/176$ as a Random Number Generating Function" becomes the fourth literature study. Research [6] continues the research [3] by selecting the one of function squares that do not produce random numbers, which the research tries to find the constants and coefficients that can be used to generate random numbers.

B. Research Authenticity

The current research is different from previous research. Research [3] is used as a reference, in particular using the fixed-point iteration method as a method for converting a function into an iterative function. The difference lies in the use of different generator function and additional test methods for generating functions. Research [3] only uses correlation tests and graphic visualization. This research adds a random number testing method in the form of run test method, mono bit, and bit block.

Research [4] also does the same thing as research [3], but it only focuses on the quadratic function. The research found several new quadratic functions which can be developed into a CSPRNG Chaos-based random number generator function. Research [5] and research [6] try to develop a special function of degree-2 which can also be used as a random number generating function.

III. RESEARCH METHODS

There are several stages of research method of this research as shown in Figure 2. Each stage is the flow of the research plan that leads the research problems to the research objectives.

The research stages can be explained as follows: The first stage is problem identification, which at this stage, an analysis of the existing problem is carried out and an initial mind frame is created. Problem identification is carried out to see cryptographic problems that will become a reference for research. A literature study is carried out to obtain literatures and theories related to CSPRNG Chaos-based random number generator using polynomial functions. The second stage is the research planning which contains each process that is required in generating random numbers.

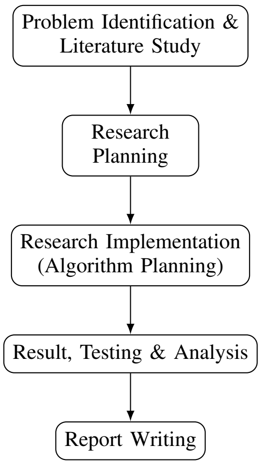
Problem Identification & Literature Study
Research Planning
Research Implementation (Algorithm Planning)
Result, Testing & Analysis
Report Writing

Fig. 2: *Schematic of the Research Flow*

In the third stage, an algorithm design is made, starting with algebraic manipulation of the selected polynomial function to look for a function that can be a random number generator. The fourth stage is the testing of the research results, especially statistical testing using methods of Run Test, Monobyte, Blockbyte. The fifth stage is report writing to explains the results of the research.

IV. Results and Discussions

A. Research Design

The 3-degree polynomial function $f(x) = 3x^3 - 3x^2 - 3x + 2$ is chosen to perform the search for random generator function as shown in Figure 2. The selected function will go through an algebraic manipulation process using the FPI method that will produce several iteration functions to obtain output in the form of numbers that can be used as keys. Each sequence of the generated numbers using the FPI method will go to the stage of statistical testing.

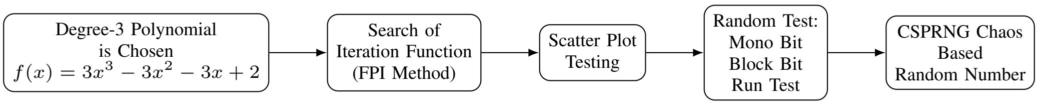

Fig. 3: General Schematic of Random Number Search

Each output from FPI experiences a chaotic condition when viewed visually based on the distribution of numbers using a scatter plot at Cartesian coordinates, where iteration as the abscissa and ordinate is the results of iteration. A function is called a generator if the scatter plot graph is broken and does not facilitate guessing. On the other hand, if function of each iteration result appears to form a pattern, easy to guess and not broken, the function has failed as a generator function.

B. Iteration Function Search

The process of finding an iterative function is carried out by paying attention to the variables of the function, then a fixed-point iteration process is carried out to obtain a candidate for the generator function. Function $f(x) = 3x^3 - 3x^2 - 3x + 2$ has a complete coefficient for each degree of the polynomial. The search for the first iterative function will be carried out starting from the 1st degree variable. Based on TIP method, the ratio $f(x) = 0$, is known so that the algebraic manipulation process can be carried out as shown in Equation 1.

$$\begin{aligned} 3x^3 - 3x^2 - 3x + 2 &= 0 \\ 3x &= 3x^3 - 3x^2 + 2 \\ x &= x^3 - x^2 + 2/3 \end{aligned} \tag{1}$$

in iteration form:

$$x_{i+1} = x_i^3 - x_i^2 + 2/3 \tag{2}$$

The next iteration function is generated based on degree-2 and degree-3 variables. By using TIP method and carrying out the same process as in Equation 1, two iteration functions of the degree-2 variable and three iteration functions of the degree-3 variable are obtained. Complete results are given in Table I.

TABLE I: Iteration Function Search

Function	**Polynomic Degree**	**Fixed-Point Iteration**	**Iteration Function**
function 1	degree-1	$x = x^3 - x^2 + \frac{2}{3}$	$x_{i+1} = x_i^3 - x_i^2 + \frac{2}{3}$
function 2	degree-2	$x = x^2 - 1 + \frac{2}{3x}$	$x_{i+1} = x_i^2 - 1 + \frac{2}{3x_i}$
function 3	degree-2	$x = \sqrt{x^3 - x + \frac{2}{3}}$	$x_{i+1} = \sqrt{x_i^3 - x_i + \frac{2}{3}}$
function 4	degree-3	$x = 1 + \frac{1}{x} - \frac{2}{3x^2}$	$x_{i+1} = 1 + \frac{1}{x_i} - \frac{2}{3x_i^2}$
function 5	degree-3	$x = \sqrt{x + 1 - \frac{2}{3x}}$	$x_{i+1} = \sqrt{x_i + 1 - \frac{2}{3x_i}}$
function 6	degree-3	$x = \sqrt[3]{\frac{3x^2+3x-2}{3}}$	$x_{i+1} = \sqrt[3]{\frac{3x_i^2+3x_i-2}{3}}$

The determination of generator function is carried out based on the geometric expression using Scater plot [1]. The initial value x_0 is used as the initial value to get the results of the next iteration. Therefore, each iteration function in Table I will use the initialization $x_0 = 0.024988457987$ as the initial test [1], then a search for the value of x_0 is carried out to find other random numbers.

C. *Function 1;* $x_{i+1} = x_i^3 - x_i^2 + \frac{2}{3}$

The iteration results are in 15 digits so it is possible to retrieve five data to be used as keys. Taking numbers in Mantissa based on the 1st, 2nd, and 3rd sequences is called data 1, while the taking in the 4th, 5th, and 6th sequences is called data 2, and so on for data 3, data 4 and up to data 5 for retrieval on 13th, 14th, and 15th sequences. Table 2 shows how to retrieve the integer from Mantissa, in the iteration function $x_{i+1} = x_i^3 - x_i^2 + \frac{2}{3}$.

The search for the iteration process uses Function 1, with initialization $x_0 = 0.024988457987$, the first seven iterations are obtained as shown in Table II. Five sets of numbers are taken which are consecutively designated as num-1, num-2, num-3, num-4, and num-5. Each set of number is viewed by column, which is a candidate for random number. and which can be used as a key in cryptography if it meets visualization tests and some statistical tests.

TABLE II: Results of the iteration $x_{i+1} = x_i^3 - x_i^2 + \frac{2}{3}$

i	x_i	**num-1**	**num-2**	**num-3**	**num-4**	**num-5**
1	-0.872333333333333	0.87	233	333	333	333
2	-0.758114300481148	0.75	811	430	048	148
3	-0.343787186463378	0.34	378	718	646	378
4	0.507844956868619	507	844	956	868	619
5	0.539736681928056	539	736	681	928	056
6	0.532584742505241	532	584	742	505	241
7	0.534085961116021	534	085	961	116	021
⋮	⋮	⋮	⋮	⋮	⋮	⋮

Visualization is made using a Cartesian diagram as the simplest test and carried out to determine whether the function $x_{i+1} = x_i^3 - x_i^2 + \frac{2}{3}$ with the x_0 initialization will be able to produce chaotic output or not. The test results using $x_0 = 0.024988457987$ give results that are not chaotic, and even produce a pattern in the form of a straight line. Similar results are also obtained for numbers-1, numbers-2, numbers-3, numbers-4, numbers-5 forming a straight-line pattern, although some of the iterations at the beginning form a random number. This requires searching for another x_0 value, which has the possibility of producing a random number.

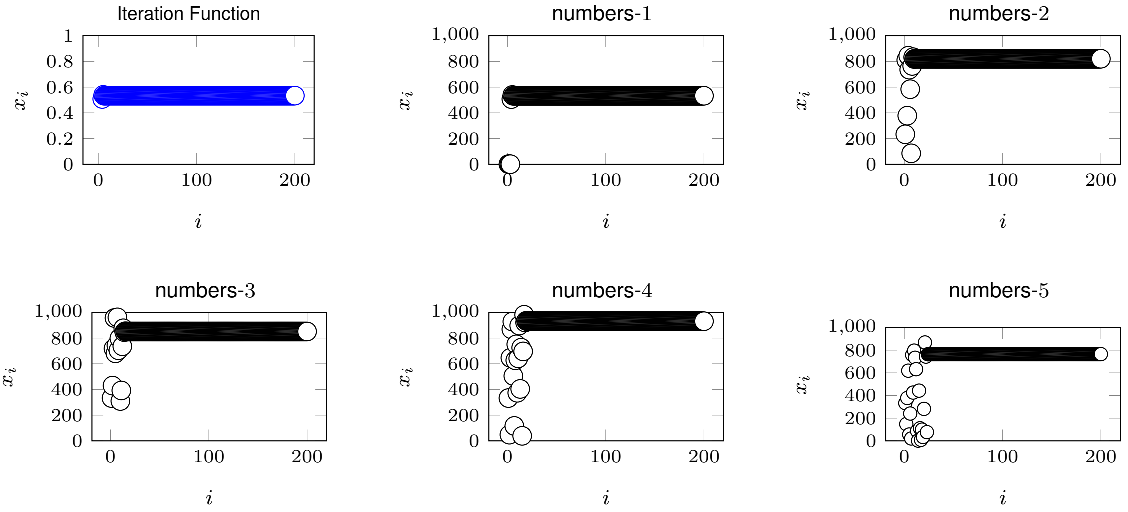

Fig. 4: Results of the First 200 Iterations of the Function $x_{i+1} = x_i^3 - x_i^2 + \frac{2}{3}$

The search for different x_0 initialization value needs to be done to determine the domain interval in function 1. The graph in Figure 4 is the comparison between the value of x_0 and the number of different values generated in the first 200 iterations. The test is carried out using the degree of accuracy of 10^{-8}, which the function $x_{i+1} = x_i^3 - x_i^2 + \frac{2}{3}$ can only be used if $-0,90848291 \leq x_0 \leq 1,37465999$.

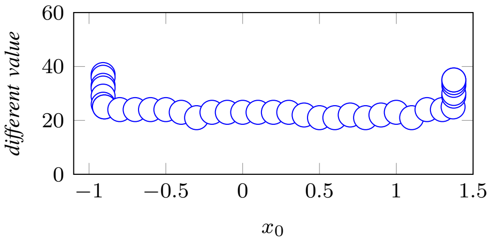

Fig. 5: Input of x_0 to the function $x_{i+1} = x_i^3 - x_i^2 + \frac{2}{3}$

The highest number of different numbers produced by the function $x_{i+1} = x_i^3 - x_i^2 + \frac{2}{3}$ only 37 numbers, when $x_0 = -0.90848291$. For other values of x_0 in the interval $-0.90848291 \leq x_0 \leq 1.37465999$ results in different values lesser than 37. Thus, the function $x_{i+1} = x_i^3 - x_i^2 + \frac{2}{3}$ cannot be used as a random number generator function.

D. Function 2; $x_{n+1} = x_{i-1}^2 - 1 + \frac{2}{3x_{i-1}}$

The iteration results of the function $x_i = x_{i-1}^2 - 1 + \frac{2}{3x_{i-1}}$ can produce two graphs. This is a good starting condition because cutting the Mantissa will produce some random numbers. Numbers-1 and numbers-2 only produce a few random numbers, although they still produce patterned numbers. Numbers-3, numbers-4, and numbers-5 succeed in generating random numbers based on CSPRNG Chaos for the first 200 iterations.

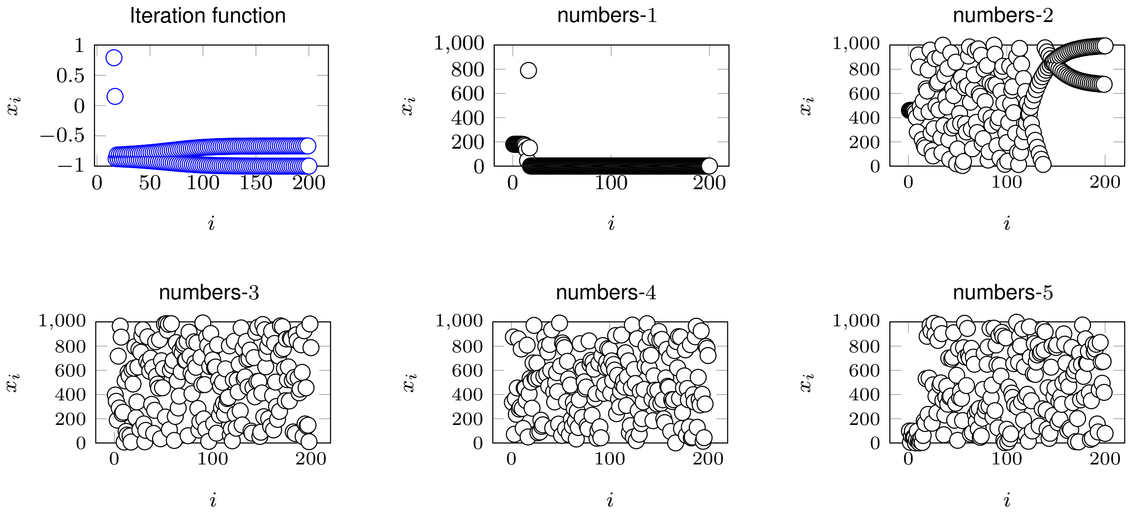

Fig. 6: First 200 Iterations of Scatter Plot for Function $x_i = x_{i-1}^2 - 1 + \frac{2}{3x_{i-1}}$

The domain interval for function 2 is shown in Figure 7, using the degree of accuracy of 10^{-3}, the interval is $1.84712708. \leq x_0 \leq 1.17999999$. The first 200 iterations can generate 200 different numbers. Thus $x_i = x_{i-1}^2 - 1 + \frac{2}{3x_{i-1}}$ can be used as a CSPRNG Chaos-based random number generator function.

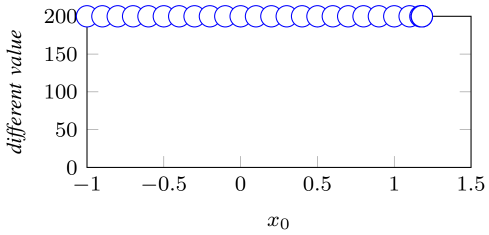

Fig. 7: Input x_0 in function $x_i = x_{i-1}^2 - 1 + \frac{2}{3x_{i-1}}$

E. Function 3; $x_{i+1} = \sqrt{x_i^3 - x_i + \frac{2}{3}}$

The iteration result a pattern forming a straight line. The randomness test is needed to see statistically satisfactory randomness. The same results are obtained for numbers-1, numbers-2, numbers-3, numbers-4, and numbers-5. The complete results for the first 200 iterations are shown in Figure 8.

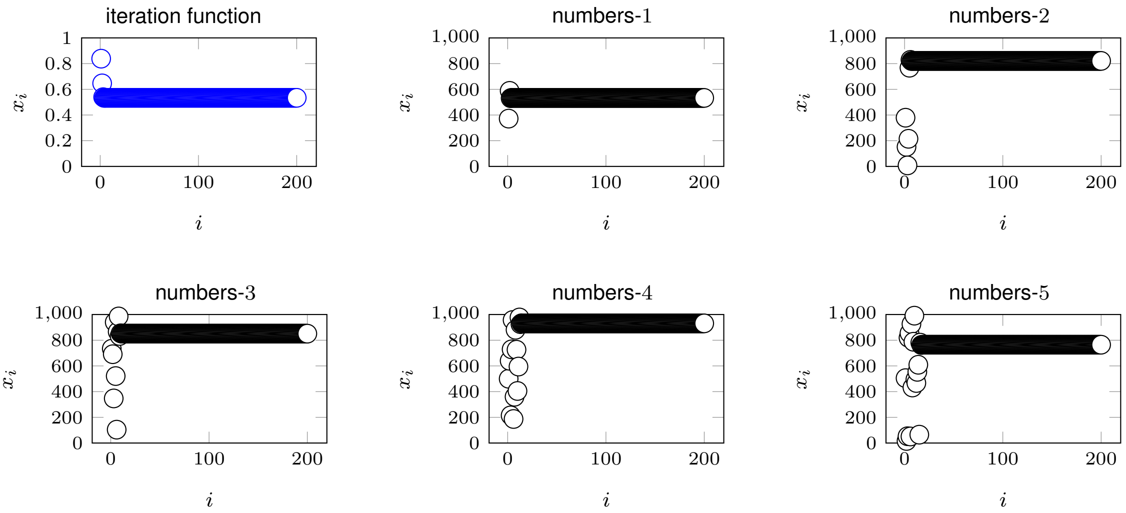

Fig. 8: Scatter Plot 200 for the First Iteration of Function $x_{i+1} = \sqrt{x_i^3 - x_i + \frac{2}{3}}$

The search for different values of x_0 initialization obtains a domain interval $-1,23999999 \leq x_0 \leq 1,34766105$ which can produce several different numbers. The graph in Figure 9 is a comparison between the value of x_0 and the number of different values generated in the first 200 iterations. By using the degree of 10^{-3} accuracy, the highest difference value is 24 when $x_0 = 1.34766105$.

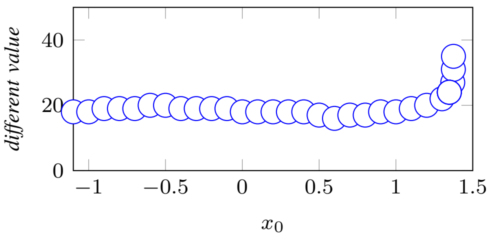

Fig. 9: Input x_0 to the Function $x_{i+1} = \sqrt{x_i^3 - x_i + \frac{2}{3}}$

F. Function 4; $x_{n+1} = 1 + \frac{1}{x} - \frac{2}{3x^2}$

The iteration results of the function $x = 1 + \frac{1}{x} - \frac{2}{3x^2}$ for the first 200 iterations are shown in Figure 10. The results for numbers-4 and number-5 can produce random numbers based on CSPNRG Chaos. Meanwhile number-1, number-2, and number-3 have not succeeded in generating 200 random numbers. So that $x = 1 + \frac{1}{x} - \frac{2}{3x^2}$ can be used as a generator function.

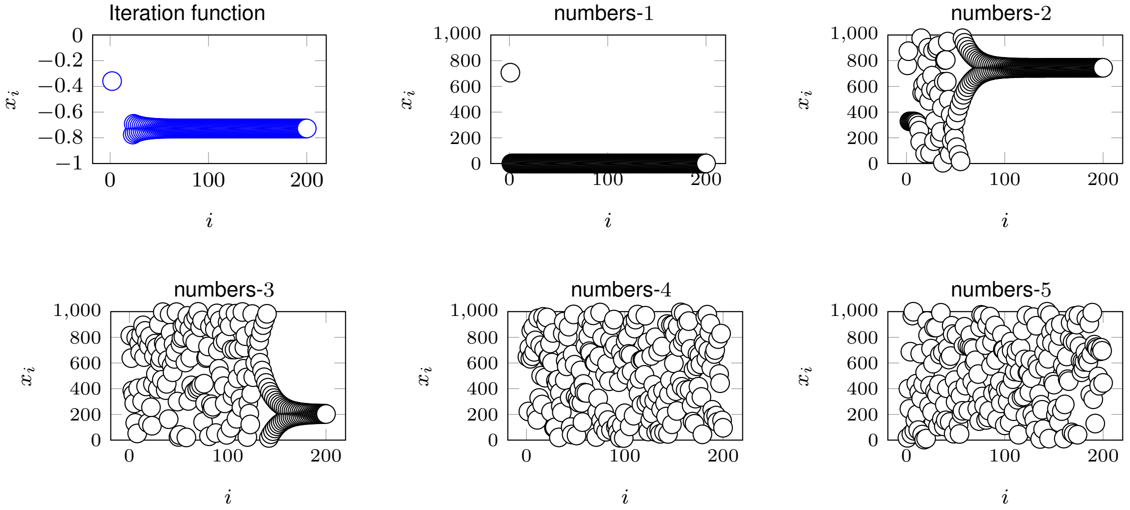

Fig. 10: Results of the First 200 Iterations of the Function $x_{n+1} = 1 + \frac{1}{x} - \frac{2}{3x^2}$

A degree of 10^{-8}, accuracy is used, and the domain search for x_0 initialization interval can produce a solution is $-1,87328412 \leq x \leq 0,84486170$. Figure 11 shows the number of different values generated based on x_0 input in the first 200 iterations.

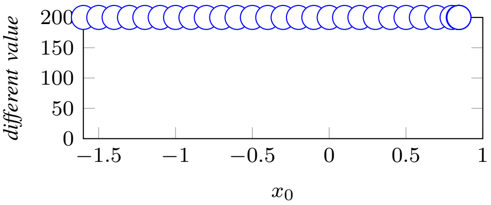

Fig. 11: Input x_0 in function $x_{n+1} = 1 + \frac{1}{x} - \frac{2}{3x^2}$

Thus, the function $x_{n+1} = 1 + \frac{1}{x} - \frac{2}{3x^2}$ can generate random numbers based on CSPRNG Chaos, especially in using Mantissa for number-4 and number-5.

G. Function 5; $x_{n+1} = \sqrt{x + 1 - \frac{2}{3x}}$

Function $x = \sqrt{x + 1 - \frac{2}{3x}}$ on $x_0$is one of the results of algebraic manipulation based on the 3-degree polynomial of $f(x) = 3x^3 - 3x^2 - 3x + 2$, and becomes the second function in the form of the square root. The results in the first seven iterations are shown in Table III.

TABLE III: Data 7 First Iterations of Function $x_{n+1} = \sqrt{x + 1 - \frac{2}{3x}}$

i	x_i	**num-1**	**num-2**	**num-3**	**num-4**	**num-5**
1	0.000057735025461546000	0	57	735	25	461
2	1.000009622457950000000	0	9	622	457	95
3	1.154701927260590000000	154	701	927	260	59
4	1.176818015846200000000	176	818	15	846	2
5	1.179946046202420000000	179	946	46	202	42
6	1.180387795910370000000	180	387	795	910	37
7	1.180450167790570000000	180	450	167	790	57
⋮	⋮	⋮	⋮	⋮	⋮	⋮

The process for the first 200 iterations of function $x_{n+1} = \sqrt{x + 1 - \frac{2}{3x}}$ is shown in Figure 12. The iteration function and each extract of the Mantissa that uses the truncation function in the number-i, where $i = 1, \cdots, 5$ also do not create a random number.

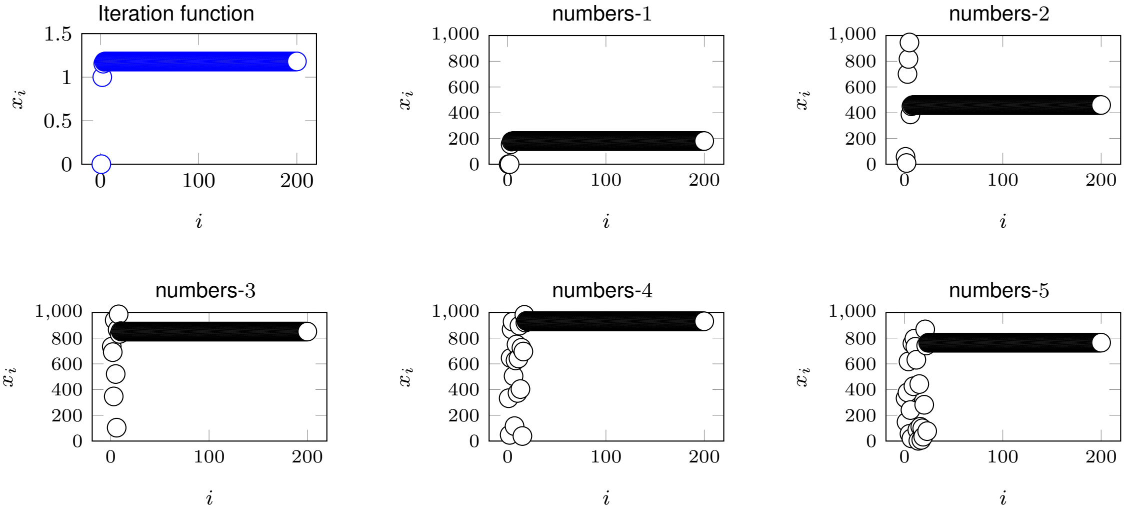

Fig. 12: Results of the First 200 Iterations of the Function $x_{n+1} = \sqrt{x + 1 - \frac{2}{3x}}$

It is necessary to look for different initialization of x_0 value to determine the function domain interval. The graph in Figure 13 shows a comparison between the value of x_0 with a number of different values generated in the first 200 iterations. The obtained number of different values are only 39 for $x_0 = 1,37466105$. Thus, it can be concluded that the function $x_{n+1} = \sqrt{x + 1 - \frac{2}{3x}}$ cannot be used as a CSPRNG Chaos-based random number generator function.

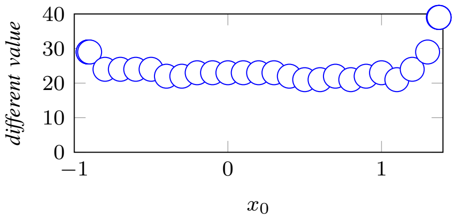

Fig. 13: Input x_0 in function $x_{n+1} = \sqrt{x + 1 - \frac{2}{3x}}$

H. Function 6; $x_{i+1} = \sqrt[3]{\frac{3x_i^2+3x_i-2}{3}}$

Function 6 is the only function that has a power root of 3. The results of the first seven iterations are shown in Table IV, which the iteration results are increasing too quickly, the acquisition of Mantissa values becomes uncontrollable, and the random number search from Mantissa cannot be performed. So that the function $x_{i+1} = \sqrt[3]{\frac{3x_i^2+3x_i-2}{3}}$ cannot be used as a random number generator function.

TABLE IV: Data of the First 7 Iterations of the Function $x_{i+1} = \sqrt[3]{\frac{3x_i^2+3x_i-2}{3}}$

i	x_i	**num-1**	**num-2**	**num-3**	**num-4**	**num-5**
1	$-1,86672254870127$	0,86	672	254	870	127
2	$3,467069026684220$	467	069	026	684	220
3	$45,07618743873060$	-	-	-	-	-
4	$116436076,112202 \times 10^{7}$	-	-	-	-	-
5	$496265422834181 \times 10^{19}$	-	-	-	-	-
6	$163764525704569 \times 10^{122}$	-	-	-	-	-
7	$804564596377273 \times 10^{258}$	-	-	-	-	-
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

I. Randomness Testing

The iteration function that visually creates chaos are function 2; $x_{i+1} = x_i^2 - 1 + \frac{2}{3x_i}$ and function 4; $x_{i+1} = 1 + \frac{1}{x_i} - \frac{2}{3x_i^2}$. This section determines whether each function can generate random numbers, so that it can then be used as a random number generator function. As a result, both functions will be tested for random numbers using mono-bit test, bit block, and also run test.

The value of randomness test $\alpha = 1\%$ is used for the three test methods. Thus, if p-value < 0,01, the number is declared as not random, and if p-value $\geq 0,01$, it is declared as random. Determination of test results for a set of numbers is determined by two or three tests that contain p-value $\geq 0,01$. The test results for the two functions are shown in Table V and Table VI, respectively.

TABLE V: Randomness Test of the Function $x_{i+1} = x_i^2 - 1 + \frac{2}{3x_i}$

No	**Tested Data**	*p-value*			**Testing Result**
		Mono Bit	**Block Bit**	**Run-Test**	
1	Numbers-1	0,0041	0,0072	0,0083	not random
2	Numbers-2	0,0007	0.0077	0,0042	not random
3	Numbers-3	0,1204	0,0022	0.4998	random
4	Numbers-4	0,4950	0,1217	0.4969	random
5	Numbers-5	0,1626	0,2680	0.4954	random

The statistical test results for the function $x_{i+1} = x_i^2 - 1 + \frac{2}{3x_i}$ are similar to the results of the visualization test using the Scatter Plot, because only numbers-1 and numbers-2 that do not produce random number. In numbers-3, the p-value for the bit block test is unreachable, but the other two tests are reachable, so it is categorized as random. While numbers-3, numbers-4, and numbers-5 in the three tests, the p-value is obtained that meet the randomness category.

TABLE VI: Randomness Test of the Function $x_{i+1} = 1 + \frac{1}{x_i} - \frac{2}{3x_i^2}$

No	**Tested Data**	*p-value*			**Testing Result**
		Mono Bit	**Block Bit**	**Run-Test**	
1	Numbers-1	0,0042	0,0073	0,0064	not random
2	Numbers-2	0,0014	0,0045	0,0082	not random
3	Numbers-3	0,0072	0,0144	0,0012	not random
4	Numbers-4	0,7684	0,0273	0.5039	random
5	Numbers-5	0,8875	0,1013	0.4971	random

The visualization test for the function $x_{i+1} = 1 + \frac{1}{x_i} - \frac{2}{3x_i^2}$ is shown in Figure 10, which it is clear that in the first 200 iterations, numbers-1, numbers-2, and numbers-3 still have a straight-line pattern even though the graphs vary widely. In the initial iteration, the function succeeds in producing some number that appear random, but then form a straight line. A slightly different result occurs for numbers 3, which the bit block test shows that the p-value is in the random category, even though the two other test methods give different results. This result occurs because in the first 200 iterations, about 100 numbers appear to have formed chaos. It makes the bit block test detects the numbers-3 as random.

The obtained results for numbers-4 and numbers-5 for the three tests produce a value greater than the p-value that meets the randomness requirement. Thus, the function $x_{i+1} = 1 + \frac{1}{x_i} - \frac{2}{3x_i^2}$ only produces two sets of number which can be used as a CSPRNG chaos-based random number generator.

J. Encryption Testing

The encryption testing is carried out to see numbers-3, numbers-4, and numbers-5 in the function $x_{i+1} = x_i^2 - 1 + \frac{2}{3x_i}$ and numberss-4 and numbers-5 in the function $x_{i+1} = 1 + \frac{1}{x_i} - \frac{2}{3x_i^2}$ as keys using Equation 3, which plaintext E_k is the encryption process which it uses plaintext (P), key (K), dan ciphertext (C).

$$E_k : \; P + K = C \; (mod\ 256) \tag{3}$$

Three different plaintexts are used, the hope is that they can represent the variations of plaintext that the user might use. The first plaintext is “fti uksw” where this sentence represents sn ordinary plaintext which only consists of a combination of letters or alphabets. The second plaintext is “$aL4tIgA” which is a combination of number, letter and symbol, and the last plaintext is a plaintext which has the same letter “zyyyyyyy”.

Table VII shows the complete results, where the numbers-3, numbers-4, and numbers-5 in the function $x_{i+1} = x_i^2 - 1 + \frac{2}{3x_i}$ which are called as key-1, key-2, and key-3, respectively. While, numbers-4 and numbers-5 in the function $x_{i+1} = 1 + \frac{1}{x_i} - \frac{2}{3x_i^2}$ are key-4 and key-5.

TABLE VII: Correlation Test Results

No	**i^{th} key**	**Tested Plaintext**			**Mean**	**Level of Relationship**
		fti uksw	**$aL4tIgA**	**zyyyyyyy**		
1	key-1	0,0349	0,1944	0,0585	0,0585	very low
2	key-2	0,0492	0,2453	0,0801	0,0801	very low
3	key-3	0,2661	0,0322	0,2602	0,2602	low
4	key-4	0,0632	0,1210	0,1793	0,1793	very low
5	key-5	0,0920	0,0118	0,2911	0,2911	low

Correlation test is used as the method to see how well the keys used can disguise plaintexts into ciphertexts. Each plaintext test is seen as the independent variable (x), and the ciphertext is seen as the dependent variable (y). Each key will be tested three times, and the mean determines the level of correlation.

All key tests get good results, which on the average, plaintexts are at very low and low levels. Thus, each generated key can make plaintext and ciphertext that is not statistically related. Therefore, each key that is generated can keep the information secret on a plaintext. As a result, the generated key using the function $f(x) = 3(x^3 - x^2 - x) + 2$ can produce five groups of numbers that can be used as keys in cryptography.

V. Conclusion

Function $f(x) = 3(x^3 - x^2 - x) + 2$ can produce six iteration functions, and only two, namely $x^2 - 1 + \frac{2}{3x}$ and $1 + \frac{1}{x} - \frac{2}{3x^2}$ which can be a CSPRNG Chaos-based random number generator function. Statistical testing using mono bit, block bit, and run test shows that two functions can generate five random numbers that can be used as keys in cryptography.

The encryption test shows the ability of the generated key from the function $f(x) = 3(x^3 - x^2 - x) + 2$ which the results of the correlation test are in "very low" and "low" categories. This shows that the use of a key with a polynomial function as a generator can make plaintexts and ciphertexts that are not statistically related.

CSPNRG chaos-based random number can not only be generated using the Lorentz function, but also regenerated using other polynomial functions such as $f(x) = 3(x^3 - x^2 - x) + 2$, and can also be used as a generator function. It only needs the right x_0 initialization simulation, the right selection of coefficients and constants and also the capability to use the process of algebraic manipulation in the fixed-point iteration method.

References

[1] Holt, Nathan, *Chaotic Cryptography: Applications of Chaos Theory to Cryptography*, Rochester Institute of Technology, 2017.

[2] Wang, L., & Cheng, H., *Pseudo-Random Number Generator Based on Logistic Chaotic System*, Entropy Journal, 2019.

[3] Wowor, Alz Danny, *Regenerasi Fungsi Polinomial Dalam Rancangan Algoritma Berbasis CSPRNG Chaos Sebagai Pembangkit Kunci pada Kriptografi Block Cipher*. Limits Journal, 14 : 4, 2017.

[4] Suling, P.M.C., & Wowor, A.D., *Regenerasi Fungsi Kuadrat sebagai Pembangkit Kunci Berbasis Metode Iterasi Titik Tetap (Fixed Point) pada Kriptografi*. UKSW: Skripsi S1 Teknik Informatika, 2017.

[5] Lihananto, D., & Wowor, A.D., *Regenerasi Fungsi* $x^2 - 9x - 99$ *dalam Pembangkit Bilangan Acak Berbasis CSPRNG Chaos*. UKSW: Skripsi S1 Teknik Informatika, 2020.

[6] Balamu, M., & Wowor, A.D., *Pencarian Interval Solusi Untuk Koefisien dan Konstanta* $f(x) = x - (x^2 - 3)/176$ *Sebagai Fungsi Pembangkit Bilangan Acak*. UKSW: Skripsi S1 Teknik Informatika, 2019.

[7] Rukhin, A., Soto, J., Nechvatal, J., Smid, M., Barker, E., Leigh, S., Levenson, M., Vangel, M., Banks, D., Heckert, A., Dray, J., & Vo, S., *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*, Gaithersburg: National Institute of Standards and Technology, 2010.

[8] Steward, J., *Calculus; Early Transcendentals*, Belmont: Brooks/Cole, 2015.

[9] Chapra, S. & Canale, R., *Numerical Methods for Engineers*, Sixth Edition, New York: Mc Graw Hill, 2010.

[10] Munir, R., *Kriptografi*, Bandung: Informatika, 2019.

[11] Montgomery, D.C. & Runger, G.C., *Applied Statistics and Probability for Engineers*, Third Edition, New York: John Wiley & Sons, 2003.