Przykładem funkcji zachowującej się chaotycznie jest funkcja zwana odwzorowaniem logistycznym.

Jest to funkcja odwzorowująca przedział jednostkowy w siebie: $f : [0, 1] \to [0, 1]$. Opisuje się ją wzorem :

$$\begin{cases} x \mapsto kx(1 - x) & 0 < k < 4 \end{cases}$$

Funkcja ta przypomina trochę równanie rekurencyjne/ ciąg - aby otrzymać nastepny element korzystamy z wartości poprzedniego elementu. Okazuje się jednak, że kolejne wartości tej funkcji są bardzo wrażliwe na zadane warunki początkowe (czyli parametr k) i w zależności od nawet najmniejszej zmiany parametru mogą się znacznie od siebie różnić. Czyli wybierając k różniące się nawet o mały ułamek, możemy otrzymać skrajne wartości tej funkcji.

W zadaniu rozpatrujemy parametr $k\epsilon[2, 4]$.

W przedziale (2,3) funkcja zachowuje się dość podobnie i ma granicę w $(1 - \frac{1}{k})$.

Gdy parametr k przekracza wartość 3, punkt $(1 - \frac{1}{k})$ traci dotychczasowy charakter atraktora. Pojawiają się dwa nowe punkty przyciągania - atraktor staje się dwupunktowy - funkcja odwzorowania zaczyna zbliżać się coraz bardziej na przemian do dwóch różnych punktów. Rozdwojenie to nazywane jest bifurkacją.

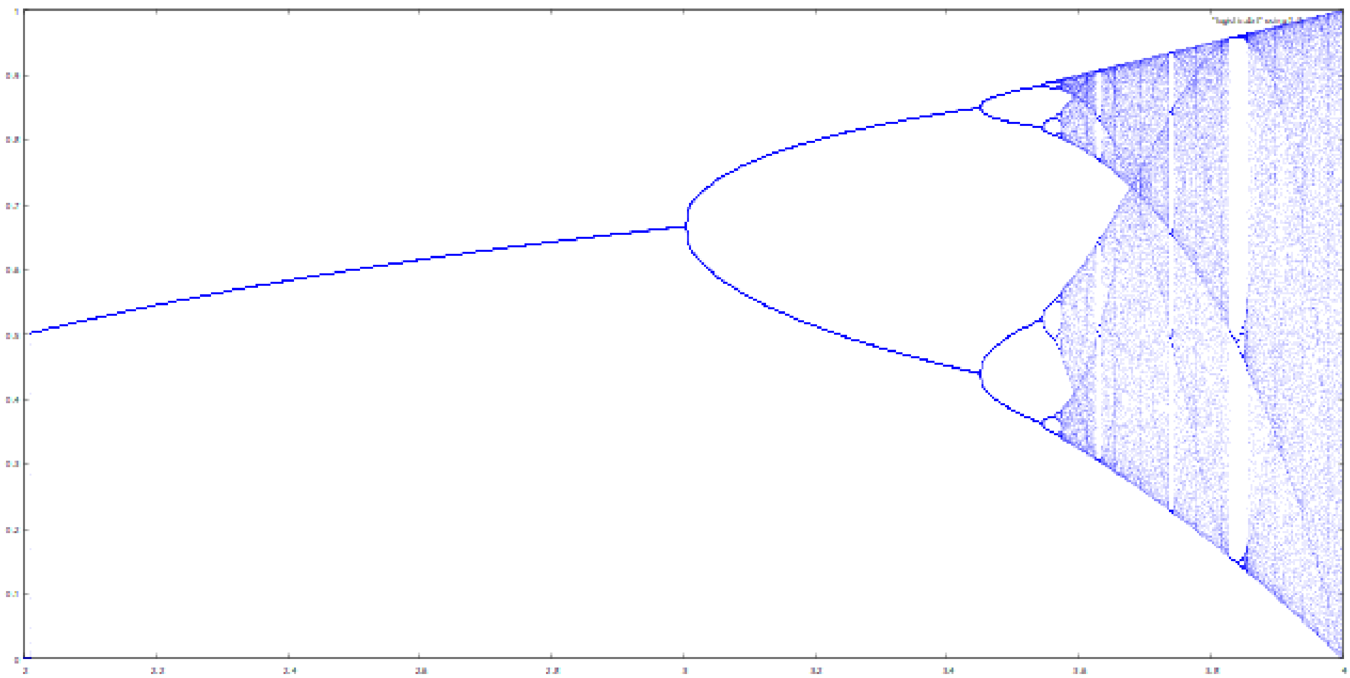
Przy dalszym zwiększaniu parametru k zachodzą dalsze komplikacje charakteru ciągów x_n i kolejne rozdwojenia punktów atraktora. Przy tym są one coraz częstsze – jeśli przez kn oznaczyć wartości parametru k, odpowiadające kolejnym bifurkacjom, to różnice pomiędzy wyrazami ciągu (k_n) stają się coraz mniejsze.

Co więcej, przyrosty te maleją w sposób wyraźnie uporządkowany, stosunek kolejnych różnic jest zbiezny do pewnej stałej, zwanej stałą Feigenbauma. $\delta = 4,6692$, cała funkcja odwzorowania też jest zbieżna, do punktu 3,5699. Gdy parametr k osiąga tę wartość, liczba punktów atraktora rośnie a sam atraktor staje się fraktalny.

Iteracje odwzorowania stają się chaotyczne. Dla parametru k=4 punkty atraktora pojawiają się w cały przedziale.

W moim programie nadaję w iteracji nowe wartości funkcji odwzorowania logistycznego. Za każdym razem odrobinę zmieniam parametr k. Wykonuję tak około 600 iteracji, dla każdej z nich poszukuję ponad 100 wartości punktów odwzorowania. Następnie zapisuję je do pliku i wykonuję wykres w gnuplot w zależności od (k,x).

Diagram bifurkacyjny odwzorowania logistycznego:



Wykres załączam również osobno w rozszerzeniu .svg.

Kod programu :

```c
#include <stdio.h>
#include <stdlib.h>


int main(){
FILE *plik;
        if((plik=fopen("logistic.dat","w"))==NULL) return -1;
    double k=2;
    int n=0;
    double x;

    double function(double x){
        return k*x*(1-x);
     }
     while(k<=4){
         n=0;
         while(n<150){
             x=function(x);
     fprintf(plik,"%.8f %.8f \n",k,x);
             n++;
         }
       k+=0.003;
     }
fclose(plik);
     return 0;
}
```