

# Rozwiązanie zadania N10

Krzysztof Waniak

Narysować zbiór:  $\{x_n : n > 100\}$  (atraktor) w zależności od parametru  $k \in [2,4]$  dla odwzorowania logistycznego.

Odwzorowanie logistyczne matematycznie dane jest wzorem:

$$x_{n+1} = k * x_n * (1 - x_n) \text{ gdzie: } x_0 \in (0,1)$$

Kod programu:

```
#include<stdio.h>
#include<ctype.h>           /* zawiera F_OK itp.    */
#include<unistd.h>          /* zawiera funkcje access(), usleep() */
#define wyp(a) printf(#a "\n")
#define wyp2(a) printf(#a)
#define wypisz(a) printf("%lf",a)
#define karetka printf("\n")
#define karetka2 printf("\n\n")
#define space printf(" ")

/* Sprawdzanie, czy wejściowy plik nie istnieje, jeśli nie istnieje zwraca
wartosc "TRUE" */
int nieistnieje(const char* nazwa)
{
    return access(nazwa, F_OK);
}

double bifurkacje(double x, double k)
{
    return k*x*(1 - x);
}

int main(void)
{
    FILE *fwynik;
    char plik_b[30];
    double x = 0.64;
    double pl;
    int i, d;

    wyp(Podaj nazwe pliku wyjsciowego);
    scanf("%s", &plik_b[0]);
    while(!nieistnieje(plik_b))
    {
        karetka;
        wyp(Taka nazwa pliku juz istnieje);
        wyp(Wprowadz inna nazwe pliku);
        karetka;
        scanf("%s", &plik_b[0]);
    }
    karetka;

    fwynik = fopen(plik_b, "w");
    for(pl = 2; pl <= 4; pl += 0.04)
    {
```

```

        for (i = 0; i < 100; i++)
        {
            x = bifurkacje(x, p1);
        }

        for(i = 0; i < 500; i++)
        {
            x = bifurkacje(x, p1);
            fprintf(fwynik, "%2.2f %2.12f\n", p1, x);
            printf("%2.2f %2.12f\n", p1, x);
        }
    }
    fclose(fwynik);

    karetka2;
    d=0;
    wyp2(Wynik obliczony i zapisany jako:);
    space;
    while(plik_b[d]!='\0')    printf("%c",plik_b[d++]);
    karetka2;
    /*system("pause");*/
    return 0;
}

```

Wynik działania programu zapisany do wskazanego pliku. W archiwum przykładowy wynik programu zapisany do pliku wynik.txt:

Wykres na podstawie wynik.txt

