

Rozwiązanie zadania N14

Krzysztof Waniak

Zastosować wzór interpolacyjny Lagrange'a dla jednorodnie rozmieszczonych $N = 20$ i $N = 40$ punktów w przedziale $[-1; 1]$ dla funkcji

$$f_1(x) = \frac{1}{1 + 30x^2},$$
$$f_2(x) = \exp(-20x^2).$$

Narysować wykresy tych funkcji oraz ich interpolantów (również dla punktów z poza siatki interpolacji). Przeanalizować wyniki.

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 func1(double x)
{
    return 1 / (1 + 30 * (x * x));
}

double func2(double x)
{
    return exp(-20 * (x * x));
}

double lagrange(double tab[21][2], double x)
{
    double y = 0, a, b;
    int j,k;
    for (j = 0; j < 21; j++)
    {
        a = 1;
        b = 1;
        for (k = 0; k < 21; k++)
        {
            if (j != k)
            {
                a *= (x - tab[k][0]);
            }
        }
    }
}
```

```

    }
    }
    for (k = 0; k < 21; k++)
    {
        if (j != k)
        {
            b *= (tab[j][0] - tab[k][0]);
        }
    }
    y += tab[j][1] * (a / b);
}
return y;
}

int main(void)
{
    FILE *fwynik1;
    FILE *fwynik2;
    FILE *fwynik3;
    FILE *fwynik4;
    FILE *fpunkty1;
    FILE *fpunkty2;
    char plik_a[30];
    char plik_b[30];
    char plik_1[30];
    char plik_2[30];
    char plik_3[30];
    char plik_4[30];
    int d;
    char p1;
    int p2;
    int j = 0;
    double x,i;
    double tabFunc1[21][2];
    double tabFunc2[21][2];

    for(i = -1; i <= 1; i += 0.1)
    {
        tabFunc1[j][0] = 0.0;
        tabFunc1[j][1] = 0.0;
        tabFunc2[j][0] = 0.0;
        tabFunc2[j][1] = 0.0;
        j++;
    }

    wyp(1 + enter - wydruk punktow funkcji);
    wyp(inny dowolny przycisk + enter by przejsc do wydruku
interpolantow);
    scanf("%s", &p1);
    karetka;
    p2 = p1 - '0';

    if(p2 == 1)
    {
        wyp(Podaj nazwe pliku wyjsciowego dla punktow pierwszej funkcji:);
        scanf("%s", &plik_a[0]);
        while(!nieistnieje(plik_a))
        {
            karetka;
            wyp(Taka nazwa pliku juz istnieje);
            wyp(Wprowadz inna nazwe pliku);

```

```

        karetk;
        scanf("%s", &plik_a[0]);
    }
    karetk;
    wyp(Podaj nazwe pliku wyjsciowego dla punktow drugiej funkcji:);
    scanf("%s", &plik_b[0]);
    while(!nieistnieje(plik_b))
    {
        karetk;
        wyp(Taka nazwa pliku juz istnieje);
        wyp(Wprowadz inna nazwe pliku);
        karetk;
        scanf("%s", &plik_b[0]);
    }
    karetk;
    fpunkty1 = fopen(plik_a, "w");
    fpunkty2 = fopen(plik_b, "w");
}

j=0;
for(i = -1; i <= 1; i += 0.1)
{
    tabFunc1[j][0] = i;
    tabFunc1[j][1] = func1(i);
    tabFunc2[j][0] = i;
    tabFunc2[j][1] = func2(i);
    if(p2 == 1)
    {
        fprintf(fpunkty1, "%1.2f %f \n", i, func1(i));
        fprintf(fpunkty2, "%1.2f %f \n", i, func2(i));
    }
    j++;
}

wyp(Podaj nazwe pliku wyjsciowego dla pierwszej funkcji (20
punktow):);
scanf("%s", &plik_1[0]);
while(!nieistnieje(plik_1))
{
    karetk;
    wyp(Taka nazwa pliku juz istnieje);
    wyp(Wprowadz inna nazwe pliku);
    karetk;
    scanf("%s", &plik_1[0]);
}
karetk;
wyp(Podaj nazwe pliku wyjsciowego dla drugiej funkcji (20 punktow):);
scanf("%s", &plik_2[0]);
while(!nieistnieje(plik_2))
{
    karetk;
    wyp(Taka nazwa pliku juz istnieje);
    wyp(Wprowadz inna nazwe pliku);
    karetk;
    scanf("%s", &plik_2[0]);
}
karetk;
wyp(Podaj nazwe pliku wyjsciowego dla pierwszej funkcji (40

```

```

punktow):);
scanf("%s", &plik_3[0]);
while(!nieistnieje(plik_3))
{
    karetk;
    wyp(Taka nazwa pliku juz istnieje);
    wyp(Wprowadz inna nazwe pliku);
    karetk;
    scanf("%s", &plik_3[0]);
}
karetk;
wyp(Podaj nazwe pliku wyjsciowego dla drugiej funkcji (40 punktow):);
scanf("%s", &plik_4[0]);
while(!nieistnieje(plik_4))
{
    karetk;
    wyp(Taka nazwa pliku juz istnieje);
    wyp(Wprowadz inna nazwe pliku);
    karetk;
    scanf("%s", &plik_4[0]);
}
karetk;

fwynik1 = fopen(plik_1, "w");
fwynik2 = fopen(plik_2, "w");
fwynik3 = fopen(plik_3, "w");
fwynik4 = fopen(plik_4, "w");

for (x = -1; x < 1; x += 0.1)
{
    fprintf(fwynik1,"%3.2f %2.12f \n", x ,lagrange(tabFunc1, x));
    /*printf("%3.2f = %2.12f \n", x ,lagrange(tabFunc1, x));
    karetk;*/
    fprintf(fwynik2,"%3.2f %2.12f \n", x ,lagrange(tabFunc2, x));
    /*printf("%3.2f = %2.12f \n", x ,lagrange(tabFunc2, x));
    karetk;*/
}

for (x = -1; x < 1; x += 0.05)
{
    fprintf(fwynik3,"%3.2f %2.12f \n", x ,lagrange(tabFunc1, x));
    /*printf("%3.2f = %2.12f \n", x ,lagrange(tabFunc1, x));
    karetk;*/
    fprintf(fwynik4,"%3.2f %2.12f \n", x ,lagrange(tabFunc2, x));
    /*printf("%3.2f = %2.12f \n", x ,lagrange(tabFunc2, x));
    karetk;*/
}

fclose(fwynik1);
fclose(fwynik2);
fclose(fwynik3);
fclose(fwynik4);
fclose(fpunkty1);
fclose(fpunkty2);

karetk2;
d=0;
wyp2(Wynik obliczony i zapisany jako:);
space;
while(plik_1[d]!='\0') printf("%c",plik_1[d++]);
karetk;

```

```

d=0;
wyp2(Wynik obliczony i zapisany jako:);
space;
while(plik_2[d]!='\0') printf("%c",plik_2[d++]);
karetka;
d=0;
wyp2(Wynik obliczony i zapisany jako:);
space;
while(plik_3[d]!='\0') printf("%c",plik_3[d++]);
karetka;
d=0;
wyp2(Wynik obliczony i zapisany jako:);
space;
while(plik_4[d]!='\0') printf("%c",plik_4[d++]);
karetka2;

if(p2 == 1)
{
    d=0;
    wyp2(Punkty pierwszej funkcji wyliczone i zapisane jako:);
    space;
    while(plik_a[d]!='\0') printf("%c",plik_a[d++]);
    karetka;
    d=0;
    wyp2(Punkty drugiej funkcji wyliczone i zapisane jako:);
    space;
    while(plik_b[d]!='\0') printf("%c",plik_b[d++]);
    karetka;
}
karetka;

/*system("pause");*/
return 0;
}

```

Wynik działania programu zapisany do pliku wynik.txt: