

Dokumentacja Zadanie 1.4

Treść:

wejście: przedział $[a,b]$, liczba naturalna n , pewna funkcja $f(x)$

zadanie:

- Stworzyć siatkę równoległą n elementów
- podać w postaci ogólnej mnożniki Lagrange'a
- podać w postaci ogólnej wielomian Lagrange'a interpolujący wartości $f(x_0), \dots, f(x_{n-1})$
- Narysować wielomian $f(x)$ i wielomian interpolacyjny na jednym wykresie

Interpolacja Lagrange'a – jest to jedna z wielu metod dzięki której możemy wyznaczyć wielomian interpolacyjny. Polega ona na tym że za pomocą ustalonego wzoru

$$l_i(x) = \prod_{0 \leq j \leq n, j \neq i} \frac{x - x_j}{x_i - x_j} = \frac{x - x_1}{x_i - x_1} \dots \frac{x - x_{i-1}}{x_i - x_{i-1}} \frac{x - x_{i+1}}{x_i - x_{i+1}} \dots \frac{x - x_n}{x_i - x_n}$$

należy wyliczyć mnożniki. Ich ilość jest równa ilości argumentów użytych do interpolacji. Sam wielomian będzie sumą iloczynów mnożników oraz wartości funkcji użytych przy interpolacji:

$$P(x) = \sum_{i=1}^n y_i l_i(x),$$

Algorytm wyliczający mnożniki Lagrange'a:

arg[] - tablica argumentów

n - ilość argumentów

Li=1 - współczynnik

i - adres argumentu w tablicy odpowiadającego aktualnie wyznaczanemu współczynnikowi

j=0;

while(j<n)

{

if(j!=i)

{

```

        Li=Li*((x-arg[j])/(arg[i]-arg[j]));
    }
    j++;
}

```

Przykładowe rozwiązanie

$$\begin{aligned}
 x_0=2, x_1=3, x_2=5 \quad y_0=3, y_1=2, y_2=6 \\
 l_0(x) &= \frac{(x-3)(x-5)}{(2-3)(2-5)} = \frac{1}{3}(x^2 - 8x + 15) \\
 l_1(x) &= \frac{(x-2)(x-5)}{(3-2)(3-5)} = -\frac{1}{2}(x^2 - 7x + 10) \\
 l_2(x) &= \frac{(x-3)(x-2)}{(5-2)(5-3)} = \frac{1}{6}(x^2 - 5x + 6) \\
 P(x) &= x^2 - 8x + 15 - x^2 + 7x - 10 + x^2 - 5x + 6 = x^2 - 6x + 11
 \end{aligned}$$

Rozwiązujemy to w ten sposób, że przy liczeniu danego mnożnika w liczniku od x odejmujemy po kolei wartości argumentów za wyjątkiem tego którego aktualnie mnożnik przyjmuje. Po wyliczeniu wszystkich mnożników przystępujemy do wyliczenia wielomianu - dodajemy po kolei wartości, które otrzymaliśmy po wymnożeniu mnożników, wcześniej owe wartości mnożąc przez y odpowiadającemu indeksowi mnożnikowi.

Opis działania programu

Program został napisany w języku C.

Użyte struktury:

- tabwsp i tabwspft: tabele współczynników przy potęgach x funkcji wejściowej oraz współczynników przy składnikach zawierających funkcje trygonometryczne.
- tabX oraz tabY: tabele przechowujące wyznaczone argumenty do interpolacji oraz odpowiadające im wartości funkcji
- tabL: tabela przechowująca wyliczone mnożniki Lagrange'a (w formie struktury wielomian, zawierającej kolejne współczynniki wielomianu oraz jego stopień)

Użyte metody:

- funkcja potęgująca
- funkcja mnożąca wielomiany

Wejście:

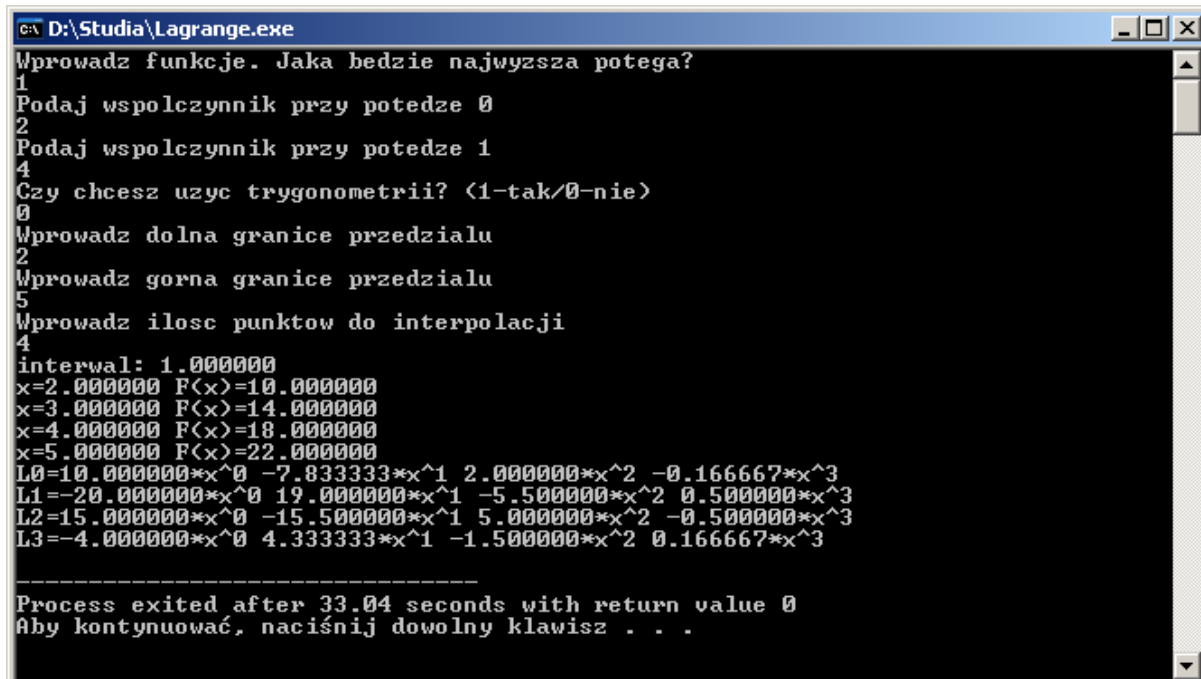
- stopień wielomianu początkowego
- kolejne współczynniki przy potęgach tego wielomianu (od najniższej do najwyższej)
- zapytanie o użycie trygonometrii (w przypadku wyrażenia zgody wprowadzenie współczynników stojących przy sinus, cosinus, tangens, cotangens)
- granice przedziału interpolacji oraz ilość punktów jaka ma wystąpić na tym przedziale (w przypadku wprowadzenia górnej granicy przedziału niższej niż dolna użytkownik jest proszony o wprowadzenie jej ponownie)

Wyjście:

- interwał, czyli różnica między poszczególnymi punktami interpolacji
- mnożniki Lagrange'a w formie kolejnych współczynników przy potęgach x

-wielomian interpolacyjny wypisany do pliku wielomianP.txt umieszczonego w folderze aplikacji

W celu wyrysowania wielomianu $f(x)$ i wielomianu interpolacyjnego należy wprowadzić wartość, którą otrzymaliśmy w wielomianP.txt do programu matematycznego, np. do Wolfram Alhpa (<http://www.wolframalpha.com/>)



```

D:\Studia\Lagrange.exe
Wprowadz funkcje. Jaka bedzie najwyzsza potega?
1
Podaj wspolczynnik przy potedze 0
2
Podaj wspolczynnik przy potedze 1
4
Czy chcesz uzyc trygonometrii? <1-tak/0-nie>
0
Wprowadz dolna granice przedzialu
2
Wprowadz gorna granice przedzialu
5
Wprowadz ilosc punktow do interpolacji
4
interwal: 1.000000
x=2.000000 F(x)=10.000000
x=3.000000 F(x)=14.000000
x=4.000000 F(x)=18.000000
x=5.000000 F(x)=22.000000
L0=10.000000*x^0 -7.833333*x^1 2.000000*x^2 -0.166667*x^3
L1=-20.000000*x^0 19.000000*x^1 -5.500000*x^2 0.500000*x^3
L2=15.000000*x^0 -15.500000*x^1 5.000000*x^2 -0.500000*x^3
L3=-4.000000*x^0 4.333333*x^1 -1.500000*x^2 0.166667*x^3

-----
Process exited after 33.04 seconds with return value 0
Aby kontynuować, naciśnij dowolny klawisz . . .

```

$$P(x)=(2.000000)*x^0+(4.000008)*x^1+(0.000000)*x^2+(-0.000000)*x^3$$

Kod programu:

```

#include <stdio.h>
#include <math.h>

int potega;
float *tabwspft;
float tabwspft[4];

float przedzialA;
float przedzialB;
int iloscP;

float *tabX;
float *tabY;

typedef struct wielomian
{
    float *wsp;//tablica wspolczynnkow
    int D;//dlugosc
} W;
```

```
W *tabL;
```

```
W mnoz(W x, W y)
```

```
{
    int i,j;

    W zwrot;
    zwrot.D=x.D+y.D;
    zwrot.wsp=(float*)malloc( (zwrot.D+1)*sizeof(float));

    for (i=0;i<=zwrot.D;i++)//zerowanie wspolczynn timer wielomianu
    {
        zwrot.wsp[i]=0;
    }
    for (i=0;i<=x.D;i++)
    {
        for(j=0;j<=y.D;j++)
        {
            zwrot.wsp[i+j]=zwrot.wsp[i+j]+x.wsp[i]*y.wsp[j];
        }
    }

    return zwrot;
}
```

```
W dod(W x, W y)
```

```
{
    W zwrot;
    if(x.D>=y.D)
    {
        zwrot=x;
    }
    else
    {
        zwrot=y;
    }

    int i, g;
    if(x.D>=y.D)
    {
        g=y.D;
    }
    else
    {
        g=x.D;
    }

    for (i=0;i<=g;i++)
    {
        zwrot.wsp[i]=x.wsp[i]+y.wsp[i];
    }
}
```

```

    }

    return zwrot;
}

float pot(float b, float p)
{
    float w=1;
    int i;
    for(i=0;i<p;i++)
    {
        w=w*b;
    }
    return w;
}

int main()
{
    int i;
    int j;
    float w;

    //Wprowadzanie funkcji
    tabwspft[0]=0;
    tabwspft[1]=0;
    tabwspft[2]=0;
    tabwspft[3]=0;

    printf("Wprowadz funkcje. Jaka bedzie najwyzsza potega?\n");
    scanf("%d", &potega);
    tabwspft=(float*)malloc(potega*sizeof(float));
    for (i=0;i<=potega;i++)
    {
        printf("Podaj wspolczynnik przy potedze %d\n",i);
        scanf("%f", &w);
        tabwspft[i]=w;
    }
    printf("Czy chcesz uzyc trygonometrii? (1-tak/0-nie)\n");
    int odp;
    scanf("%d",&odp);
    if (odp==1)
    {
        printf("Wprowadz wspolczynnik przy sin(x)\n");
        scanf("%f", &w);
        tabwspft[0]=w;

        printf("Wprowadz wspolczynnik przy cos(x)\n");
        scanf("%f", &w);
        tabwspft[1]=w;
    }
}

```

```

    printf("Wprowadz wspolczynnik przy tg(x)\n");
    scanf("%f", &w);
    tabwspft[2]=w;

    printf("Wprowadz wspolczynnik przy ctg(x)\n");
    scanf("%f", &w);
    tabwspft[3]=w;
}

//END

//Wprowadzanie przedzialu i ilosci punktow
printf("Wprowadz dolna granice przedzialu\n");
scanf("%f", &w);
przedzialA=w;
printf("Wprowadz gorna granice przedzialu\n");
scanf("%f", &w);
while(w<=przedzialA)
{
    printf("Za mala gorna granica! Wprowadz jeszcze raz\n");
    scanf("%f", &w);
}
przedzialB=w;
printf("Wprowadz ilosc punktow do interpolacji\n");
scanf("%d", &iloscP);
//END

//Ustalanie argumentow i wartosci funkcji
tabX=(float*)malloc(iloscP*sizeof(float));
tabY=(float*)malloc(iloscP*sizeof(float));

tabX[0]=przedzialA;
tabX[iloscP-1]=przedzialB;
float il=(float)iloscP;
float interwal=(przedzialB-przedzialA)/(il-1);

for(i=0;i<iloscP-1;i++)
{
    tabX[i]=przedzialA+i*interwal;
    if(tabX[i]==0)
    {
        tabX[i]=tabX[i]+0.0000001;
    }
}

float sumaY;
for(i=0;i<iloscP;i++)//dla kazdego X
{
    sumaY=0;

```

```

        for(j=0;j<=potega;j++)//pozbieraj w sume wartosc Y
        {
            sumaY=sumaY+(pot(tabX[i],j))*(tabwspft[j]);
        }
        tabY[i]=sumaY+
        tabwspft[0]*(float)sin((double)tabX[i])+
        tabwspft[1]*(float)cos((double)tabX[i])+
        tabwspft[2]*(float)tan((double)tabX[i])+
        tabwspft[3]*(float)cos((double)tabX[i]) / (float)sin((double)tabX[i] );
    }
//END

//TEST
printf("interwal: %f\n", interwal);
for(i=0;i<iloscP;i++)
{
    printf("x=%f F(x)=%f\n", tabX[i], tabY[i]);
}
//TEST

//Obliczanie wspolczynnika Lagrange'a
tabL=(W*)malloc(iloscP*sizeof(W));

for(i=0;i<iloscP;i++)
{
    W baza;
    baza.D=0;
    baza.wsp=(float*)malloc( (baza.D+1)*sizeof(float));
    baza.wsp[0]=1;

    j=0;
    while(j<iloscP)
    {
        if(j!=i)
        {
            W czynnik;
            W iloczyn;
            czynnik.D=1;
            czynnik.wsp=(float*)malloc( (baza.D+1)*sizeof(float));

            float mianownik=tabX[i]-tabX[j];
            czynnik.wsp[1]=1/mianownik;
            czynnik.wsp[0]=(-tabX[j])/mianownik;

            iloczyn=mnoz(czynnik, baza);
            baza = iloczyn;
        }
        j++;
    }
    tabL[i]=baza;
}

```

```
}  
//END
```

```
//Wypisanie współczynników Lagrange'a i przemnożenie ich przez Y  
for(i=0;i<iloscP;i++)  
{  
    printf("L%d=",i);  
    for(j=0;j<=tabL[i].D;j++)  
    {  
        printf("%f*x^%d ",tabL[i].wsp[j],j);  
        tabL[i].wsp[j]=tabL[i].wsp[j]*tabY[i];  
    }  
    printf("\n");  
}  
//END
```

```
//Dodanie współczynników w wielomian interpolacyjny  
W P;  
P.D=iloscP-1;  
P.wsp=(float*)malloc( (P.D+1)*sizeof(float));
```

```
for(i=0;i<iloscP;i++)  
{  
    float sumaP=0;  
    for(j=0;j<iloscP;j++)  
    {  
        sumaP=sumaP+tabL[j].wsp[i];  
    }  
    P.wsp[i]=sumaP;  
}  
//Wypisanie wielomianu interpolacyjnego do pliku  
FILE *plik;  
plik=fopen("wielomianP.txt","w");  
fprintf(plik,"P(x)=");  
for(i=0;i<P.D;i++)  
{  
    fprintf(plik,"(%f)*x^%d+",P.wsp[i],i);  
}  
i=P.D;  
fprintf(plik,"(%f)*x^%d",P.wsp[i],i);  
  
fclose(plik);  
//END  
getchar();  
return 0;  
}
```