

Zadanie 11:

11 O. Posługując się wzorem trapezów i metodą Romberga, oblicz całkę

$$I = \int_0^{\infty} \sin\left(\pi \frac{1+\sqrt{x}}{1+x^2}\right) e^{-x} dx \quad (8)$$

z dokładnością do 10^{-7} .

Wskazówka:

$$I = \underbrace{\int_0^A \sin\left(\pi \frac{1+\sqrt{x}}{1+x^2}\right) e^{-x} dx}_{I_1} + \underbrace{\int_A^{\infty} \sin\left(\pi \frac{1+\sqrt{x}}{1+x^2}\right) e^{-x} dx}_{I_{\text{ogon}}} \quad (9)$$

przy czym

$$|I_{\text{ogon}}| \leq \int_A^{\infty} \left| \sin\left(\pi \frac{1+\sqrt{x}}{1+x^2}\right) \right| e^{-x} dx \leq \int_A^{\infty} e^{-x} dx = e^{-A}. \quad (10)$$

Znajdź A takie, że $e^{-A} < 10^{-7}$, a następnie znajdź numerycznie wartość I_1 z odpowiednią dokładnością.

Kod w języku C++:

```
#include<iostream>
#include<cmath>
#include<vector>
#include<iomanip>

using namespace std;
long double trapezoidArea(long double, long double, int);
long double calcFunction(long double);

int main(){

    long double tolerance = pow(10, -7);

    int i = -1;
    long double rest=0;

    while(true){
        rest=std::pow(M_E,i);
        if(rest < tolerance)
            break;
        i--;
    }
    cout << "A= " << i << endl;

    //gorna granica calki
    long double upper=-i;
```

```

//dolna granica calki
long double lower = 0;
//limit iteracji
long double maxIters = 32;

std::vector<long double> rombergs; //obliczanie przyblizen do obecnego wiersza
std::vector<long double> prev_rombergs; //trzyma poprzednie
przyblizenia(poprzedni wiersz)

//ustawianie precyzji
cout<<std::fixed<<std::setprecision(7);

//oblicz pierwszy wyraz
rombergs.push_back(trapezoidArea(lower, upper, 0));
cout<<"k="<<0<<" "<<rombergs[0]<<endl;
cout<<"W funkcji "<<calcFunction(0.001)<<endl;
cout<<"podzial na 10000000 "<<trapezoidArea(0,17,10)<<endl;

int tempK;
long double temp;
int k;

for(k=1; k<maxIters-1; k++){

    //wcześniej obliczony wiersz staje sie prev_rombergs-kopiowanie
    prev_rombergs=rombergs;
    //czyszcimy obecny wektor
    rombergs.clear();

    if(k<10)
        cout<<"k= "<<k<<" ";
    else
        cout <<"k="<<k<<" ";

    tempK=k;

    for(int n=0; n<=k; n++){
        if(n==0){
            rombergs.push_back(trapezoidArea(lower,upper,k));
            cout<<rombergs[0]<<" ";
            tempK--;
        }
        else{
            temp=(std::pow(4,n)*rombergs[n-1]-prev_rombergs[n-1])/(std::pow(4,n)-
1);

            cout<<temp<<" ";
            rombergs.push_back(temp);
            tempK--;
        }
    }
    cout<<endl;

    if(std::abs(rombergs[k]-prev_rombergs[k-1]) < tolerance){
        cout<<"Pole calki w przedziale ["<<lower<<","<<upper<<"]
wynosi"<<rombergs[k]<<" +- "<<tolerance<<endl;
        break;
    }
}
}

```

```

    long double result=rombergs[k]+rest;
    cout<<"Pole laczne calki od [0,inf) wynosi "<<result<<" +- "<<tolerance<<endl;

    return 0;
}

//korzysta z zlozonej metody trapezow by podzielic calke na k przedzialow
long double trapezoidArea(long double low, long double high, int k){

    //ilosc punktow podzialu
    long double N=pow(2,k);
    long double area=0;

    //przedzial pomiedzy argumentami funkcji
    long double h=(high-low)/N;

    for(long double i=low; i<=high; i=i+h){

        if(i==low || i==high)
            area+=calcFunction(i)/2;
        else
            area+=calcFunction(i);
    }
    return area*h;
}

//oblicza wartosc funkcji podcalkowej dla danego x
long double calcFunction(long double x){
    return sin(M_PI*(1+sqrt(x))/(1+pow(x,2)))*std::pow(M_E,-x);
}

```

Wynik działania programu:

```

A= 17
k=0 0.000000
W funkcji -0.000000
podzial na 10000000 -0.2150856

k= 1 0.0002891 0.0003854
k= 2 0.0294521 0.0391731 0.0417589
k= 3 0.2657806 0.3445568 0.3649158 0.3700452
k= 4 0.2180085 0.2031511 0.1937240 0.1810067 0.1903046
k= 5 -0.0284941 -0.1109283 -0.1318670 -0.1370351 -0.1383215 -0.1386427
k= 6 -0.1370753 -0.1732690 -0.1774251 -0.1781482 -0.1783095 -0.1783485 -0.1783582
k= 7 -0.1871272 -0.2038111 -0.2058472 -0.2062984 -0.2064088 -0.2064362 -0.2064431 -0.2064448
k= 8 -0.2063470 -0.2127536 -0.2133497 -0.2134688 -0.2134969 -0.2135039 -0.2135056 -0.2135060 -0.2135061
k= 9 -0.2133668 -0.2157068 -0.2159037 -0.2159442 -0.2159539 -0.2159563 -0.2159569 -0.2159571 -0.2159571
k=10 -0.2158856 -0.2167252 -0.2167931 -0.2168073 -0.2168106 -0.2168115 -0.2168117 -0.2168117 -0.2168118 -0.2168118
k=11 -0.2167825 -0.2170815 -0.2171052 -0.2171102 -0.2171113 -0.2171116 -0.2171117 -0.2171117 -0.2171117 -0.2171117
k=12 -0.2171007 -0.2172068 -0.2172151 -0.2172169 -0.2172173 -0.2172174 -0.2172174 -0.2172174 -0.2172174 -0.2172174
k=13 -0.2172134 -0.2172510 -0.2172539 -0.2172545 -0.2172547 -0.2172547 -0.2172547 -0.2172547 -0.2172547 -0.2172547
k=14 -0.2172533 -0.2172666 -0.2172676 -0.2172678 -0.2172679 -0.2172679 -0.2172679 -0.2172679 -0.2172679 -0.2172679
k=15 -0.2172674 -0.2172721 -0.2172724 -0.2172725 -0.2172725 -0.2172725 -0.2172725 -0.2172725 -0.2172725 -0.2172725
k=16 -0.2172724 -0.2172740 -0.2172742 -0.2172742 -0.2172742 -0.2172742 -0.2172742 -0.2172742 -0.2172742 -0.2172742
k=17 -0.2172741 -0.2172747 -0.2172748 -0.2172748 -0.2172748 -0.2172748 -0.2172748 -0.2172748 -0.2172748 -0.2172748
k=18 -0.2172747 -0.2172750 -0.2172750 -0.2172750 -0.2172750 -0.2172750 -0.2172750 -0.2172750 -0.2172750 -0.2172750
k=19 -0.2172750 -0.2172750 -0.2172750 -0.2172750 -0.2172750 -0.2172750 -0.2172750 -0.2172750 -0.2172750 -0.2172750

Pole calki w przedziale [0.0000000,17.0000000] wynosi-0.2172750 +- 0.0000001
Pole laczne calki od [0,inf) wynosi -0.2172750 +- 0.0000001

```

Opis rozwiązania:

Metoda Romberga to iteracyjna metoda obliczania coraz bliższego przybliżenia całki, która korzysta z ekstrapolacji Richardsona i złożonej metody trapezów. Nie zawsze ona działa, liczba iteracji może być nieskończona a całka rozbieżna. W naszym przykładzie całka zmierza od zera do nieskończoności więc nie możemy jej bezpośrednio obliczyć numerycznie. Całkę należy rozbić na dwa przedziały:

$$\int_0^{\infty} f(x)dx = \int_0^A f(x) + \int_A^{\infty} f(x)$$

W zależności od dokładności możemy zauważyć, że całka nie będzie nigdy większa niż e^{-x} .

$$\int_A^{\infty} \left| \sin\left(\frac{1+\sqrt{x}}{1+x^2}\right) \right| dx < \int_A^{\infty} e^{-x} dx$$

Możemy wyszacować analitycznie i przybliżyć całkę e^{-x} , która będzie mniejsza niż zadana dokładność. Dzięki temu będzie możliwe określenie przedziału $[0, A]$, w którym obliczanie całki będzie miało znaczący wpływ na wynik końcowy. Dla dokładności 10^{-7} A wyniosło 17

Omówienie wyników

Obliczone A to granica całki, od której dalszy wynik nie wpływa znacząco na całociowe pole

$$A = 17$$

Co udowadnia:

```
Pole całki w przedziale [0.0000000,17.0000000] wynosi -0.2172750 +- 0.0000001
Pole łączne całki od [0,inf) wynosi -0.2172750 +- 0.0000001
```

Metoda Romberga osiąga oczekiwaną znacznie szybciej niż gdybyśmy próbowali korzystać bezpośrednio z metody trapezów.