Brandon Swatek - zadanie numeryczne nr 7

Zadanie numeryczne

7. Posługując się wzorem trapezów i metodą Romberga, oblicz całkę:

$$I = \int_0^\infty \sin\left(\pi \frac{1+\sqrt{x}}{1+x^2}\right) e^{-x} \mathrm{d}x \tag{1}$$

z dokładnością do 10^{-7}.

Wskazówka:

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

przy czym

$$|I_{\mathrm{ogon}}| \leqslant \int_A^\infty \left|\sin\left(\pi \frac{1+\sqrt{x}}{1+x^2}\right)\right| e^{-x} \mathrm{d}x \leqslant \int_A^\infty e^{-x} \mathrm{d}x = e^{-A} \tag{3}$$

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

Zadanie rozwiązuje wykorzystując wzór trapezów i metodę Romberga. Oryginalna całka jest dzielona na sumę dwóch całek I1 i I2. Całka I1 jest liczona z użyciem wzoru trapezów i metody Romberga.

In [1]:
```
import numpy as np
import math
from scipy import integrate


def trapeze_method(f, a, b, n):
    h = (b-a)/n
    x = a
    fx = f(x)
    for i in range(1, n):
        x += h
        fx += 2*f(x)
    return (fx + f(b)) * h * 0.5


def romberg(f, a, b, eps, nmax):
    Q = np.zeros((nmax, nmax), float)
    for i in range(0, nmax):
        n = 2**i
        Q[i, 0] = trapeze_method(f, a, b, n)
        for k in range(0, i):
            Q[i, k+1] = (4**(k+1) * Q[i, k] - Q[i-1, k]) / (4**(k+1)-1)
            if i > 0 and abs(Q[i, k+1] - Q[i, k]) < eps:
                break
        print(Q[i, 0:i+1])
    return Q
```

Całka I2 jest dobrana w taki sposób aby jej wynik był mniejszy niż zadana dokładność. Całkę można policzyć na przykład korzystając z metody scipy.integrate.quad dostępnej w Pythonie.

Przy wyborze parametru b = 17 otrzymujemy wynik o zadowalającej dokładności.

In [2]:
```
def fun(x):
    return np.sin(math.pi * ((1 + math.sqrt(x)) / (1 + pow(x, 2)))) * m
ath.exp(-x)

def fun2(x):
    return math.exp(-x)

a = 0.0
b = 17.0
c = np.inf
eps = 1.0e-17
nmax = 18

A = romberg(fun, a, b, eps, nmax)
I1 = A[nmax-1, nmax-1]
err, I2 = integrate.quad(fun2, b, c)
```

```
[1.95197823e-08]
[0.00028908 0.00038543]
[0.02945206 0.03917306 0.0417589 ]
[0.26578065 0.34455684 0.36491576 0.37004523]
[0.21880847 0.20315108 0.19372403 0.1910067  0.19030459]
[-0.02849412 -0.11092832 -0.13186695 -0.13703506 -0.1383215  -0.1386427
4]
[-0.13707531 -0.17326903 -0.17742508 -0.17814822 -0.17830945 -0.1783485
4
 -0.17835824]
[-0.18712715 -0.2038111  -0.20584724 -0.20629838 -0.20640878 -0.2064362
4
 -0.2064431  -0.20644482]
[-0.20634696 -0.21275356 -0.21334972 -0.21346881 -0.21349693 -0.2135038
6
 -0.21350559 -0.21350602 -0.21350612]
[-0.21336684 -0.2157068  -0.21590369 -0.21594423 -0.21595393 -0.2159563
3
 -0.21595693 -0.21595708 -0.21595712 -0.21595713]
[-0.21588564 -0.21672524 -0.21679314 -0.21680726 -0.21681064 -0.2168114
8
 -0.21681169 -0.21681174 -0.21681175 -0.21681176 -0.21681176]
[-0.2167825  -0.21708146 -0.2171052  -0.21711016 -0.21711135 -0.2171116
4
 -0.21711171 -0.21711173 -0.21711174 -0.21711174 -0.21711174 -0.2171117
4]
[-0.2171007  -0.21720676 -0.21721511 -0.21721686 -0.21721728 -0.2172173
8
 -0.21721741 -0.21721741 -0.21721742 -0.21721742 -0.21721742 -0.2172174
2
 -0.21721742]
[-0.21721339 -0.21725095 -0.2172539  -0.21725452 -0.21725466 -0.2172547
 -0.21725471 -0.21725471 -0.21725471 -0.21725471 -0.21725471 -0.2172547
1
 -0.21725471 -0.21725471]
[-0.21725327 -0.21726656 -0.2172676  -0.21726782 -0.21726787 -0.2172678
8
 -0.21726788 -0.21726789 -0.21726789 -0.21726789 -0.21726789 -0.2172678
9
 -0.21726789 -0.21726789 -0.21726789]
[-0.21726737 -0.21727207 -0.21727244 -0.21727252 -0.21727254 -0.2172725
4
 -0.21727254 -0.21727254 -0.21727254 -0.21727254 -0.21727254 -0.2172725
4
 -0.21727254 -0.21727254 -0.21727254 -0.21727254]
[-0.21727236 -0.21727402 -0.21727415 -0.21727418 -0.21727419 -0.2172741
9
 -0.21727419 -0.21727419 -0.21727419 -0.21727419 -0.21727419 -0.2172741
9
 -0.21727419 -0.21727419 -0.21727419 -0.21727419 -0.21727419]
[-0.21727412 -0.21727471 -0.21727476 -0.21727477 -0.21727477 -0.2172747
7
 -0.21727477 -0.21727477 -0.21727477 -0.21727477 -0.21727477 -0.2172747
7
 -0.21727477 -0.21727477 -0.21727477 -0.21727477 -0.21727477 -0.2172747
7]
```

Ostatecznie otrzymujemy:

In [3]:
```
print(I1)
print(I2)
print(I1 + I2)
```

```
-0.21727476915713467
5.118524277996492e-09
-0.2172747640386104
```