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

$$I = \int_0^\infty \sin\left(\pi \frac{1+\sqrt{x}}{1+x^2}\right) e^{-x}\, dx \tag{9}$$

z dokładnością do 10^{-7}.
Wskazówka:

$$I = \underbrace{\int_0^A \sin\left(\pi \frac{1+\sqrt{x}}{1+x^2}\right) e^{-x}\, dx}_{I_1} + \underbrace{\int_A^\infty \sin\left(\pi \frac{1+\sqrt{x}}{1+x^2}\right) e^{-x}\, dx}_{I_{\text{ogon}}} \tag{10}$$

prz czym

$$|I_{\text{ogon}}| \leqslant \int_A^\infty \left|\sin\left(\pi \frac{1+\sqrt{x}}{1+x^2}\right)\right| e^{-x}\, dx \leqslant \int_A^\infty e^{-x}\, dx = e^{-A}\,. \tag{11}$$

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

Całka obliczona w przedziale [0,17], ponieważ funkcja podcałkowa jest nie większa od e(-x).

Rozwiązanie

A=17, I=-0.2172725417

```
import numpy
import math as m

def f(x):
    return (m.sin(m.pi*(1.0+m.sqrt(x))/(1.0+x*x)))*m.pow(m.e,-x)

def romberg(f, a, b, n):
    r = numpy.zeros((n+1,n+1))
    h = b - a
    r[0, 0] = 0.5 * h * (f(a) + f(b))

    powerOf2 = 1
    for i in range(1, n + 1):
        h = 0.5 * h
        sum = 0.0
        powerOf2 = 2 * powerOf2
        for k in range(1, powerOf2, 2):
            sum +=  f(a + k * h)

        r[i, 0] = 0.5 * r[i - 1, 0] + sum * h

        powerOf4 = 1
        for j in range(1, i + 1):
            powerOf4 = 4 * powerOf4
```

```
          r[i, j] = r[i, j - 1] + (r[i, j - 1] - r[i - 1, j - 1]) / (powerOf4 - 1)
       if(numpy.linalg.norm(r)<tol):
          break

    return r

tol=1.0e-7
val=0
precision=15
a=0
b=0
while(True):
    val=m.pow(m.e,-b)
    if(val<tol):
       break
    b+=1

print(val)
print(b)
sol=romberg(f,a,b,precision)
print (sol[precision,precision])
```