Jun 25, 2019

CK0030: Partial evaluation III

Exercise 01.

Let $\mathcal{X} = \{-5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5\}$ and let $\mathcal{Y} = \{-5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5\}$.

Write Python code that computes for each pair (x, y), with $x \in \mathcal{X}$ and $y \in \mathcal{Y}$, and for $K = 32$, the following mathematical function

$$f(x,y) = -\Big| \underbrace{\Big[\sum_{k=0}^{K} \frac{(-1)^k}{(2k+1)!} x^{2k+1}\Big]}_{a(x;K)} \underbrace{\Big\{\prod_{k=1}^{K} \Big[1 - \frac{y^2}{\pi^2(k-1/2)^2}\Big]\Big\}}_{b(y;K)} \underbrace{\Big(\sum_{k=0}^{K} \frac{z^k}{k!}\Big)}_{c(z;K)} \Big|,$$

$$\text{with } z = |1 - (x^2 + y^2)^{1/2}/\pi|.$$

In your code, use Python functions that evaluate $a(x; K)$, $b(y; K)$ and $c(z; K)$.

Compare the resulting $f(x, y)$ with $g(x, y) = -\big|\sin(x)\cos(y)e^{|1-\sqrt{x^2+y^2}/\pi|}\big|$.

Sejam $\mathcal{X} = \{-5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5\}$ e $\mathcal{Y} = \{-5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5\}$.

Escreva um código Python que compute para cada par (x, y), com $x \in \mathcal{X}$ e $y \in \mathcal{Y}$, e para $K = 32$ a seguinte função matemática

$$f(x,y) = -\Big| \underbrace{\Big[\sum_{k=0}^{K} \frac{(-1)^k}{(2k+1)!} x^{2k+1}\Big]}_{a(x;K)} \underbrace{\Big\{\prod_{k=1}^{K} \Big[1 - \frac{y^2}{\pi^2(k-1/2)^2}\Big]\Big\}}_{b(y;K)} \underbrace{\Big(\sum_{k=0}^{K} \frac{z^k}{k!}\Big)}_{c(z;K)} \Big|,$$

$$\text{com } z = |1 - (x^2 + y^2)^{1/2}/\pi|.$$

No seu código, use as funções Python que calculam $a(x; K)$, $b(x; K)$, $c(x; K)$.

Compare o resultado de $f(x, y)$ com $g(x, y) = -\big|\sin(x)\cos(y)e^{|1-\sqrt{x^2+y^2}/\pi|}\big|$.

Solution:

```
def mysin(x,K):
    s = 0
    k = 0
    while k <= K:
        t = (-1)**k * x**(2*k+1) / factorial(2*k+1)
        s = s + t
        k = k + 1
    return s

def mycos_sum(x,K):
    s = 0
    k = 0
    while k <= K:
        t = (-1)**k * x**(2*k) / factorial(2*k)
        s = s + t
        k = k + 1
    return s

def mycos_pro(x,K):
    p = 1
    k = 1
    while k <= K:
        t = 1 - x**2 / pi**2 / (k-1/2)**2
        p = p * t
        k = k + 1
    return p

def myexp(x,K):
    s = 0
    k = 0
    while k <= K:
        t = x**k / factorial(k)
        s = s + t
        k = k + 1
    return s

from math import e, pi, sin, cos, exp, fabs, sqrt, factorial

X = [-5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5]
Y = [-5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5]

K = 32
for x in X:
    for y in Y:
        z = fabs(1 - sqrt(x**2 + y**2)/pi)
        Fs = -fabs(mysin(x,K) * mycos_sum(y,K) * myexp(z,K))
        Fp = -fabs(mysin(x,K) * mycos_pro(y,K) * myexp(z,K))
        G = -fabs(sin(x) * cos(y) * exp(z))
        print(x,y,Fs,Fp,G)
```