```python
import numpy as np

def error(point): #error function
    u = point[0]
    v = point[1]
    return 1.0*(u*np.exp(v)-2*v*np.exp(-u))**2

def gradu(point): #partial derivative wrt u
    u = point[0]
    v = point[1]
    return 2.0*((np.exp(v)+2*v*np.exp(-u))*(u*np.exp(v)-2*v*np.exp(-u)))

def gradv(point): #partial derivative wrt v
    u = point[0]
    v = point[1]
    return 2.0*((u*np.exp(v)-2*np.exp(-u))*(u*np.exp(v)-2*v*np.exp(-u)))

point = np.array([1.0,1.0])
counter = 0

while error(point) > 10**-14: #implement gradient descent
    point += np.array([-0.1*gradu(point),-0.1*gradv(point)])
    counter += 1

print(point)
print(counter)

i = 0
p2 = np.array([1.0,1.0])

while i < 16: #implement coordinate descent
    p2 += np.array([-0.1*gradu(p2),0])
    p2 += np.array([0,-0.1*gradv(p2)])
    i += 1

print(error(p2))
```