import numpy as np

import matplotlib.pyplot as plt

import matplotlib.style as style

style.use('ggplot')

def finite\_difference\_solver(M, N):

sigma = 0.40

T = 1.00

r = 0.05

a = r - (sigma\*\*2)/2

b = (sigma\*\*2)/2

c = -r

x\_min = np.log(50)

x\_max = np.log(100)

s\_min = 0

s\_max = T

x\_range, dx = np.linspace(x\_min, x\_max, N, retstep=True)

s\_range, ds = np.linspace(s\_min, s\_max, M, retstep=True)

V = np.zeros((M,N))

V[0,:] = (np.exp(x\_range) - 50) \* (100 - np.exp(x\_range))

for m\_ix, m in enumerate(s\_range):

if m\_ix == len(s\_range) - 1:

continue

for n\_ix, n in enumerate(x\_range):

if n\_ix == 0:

V[m\_ix, n\_ix] = 0 # lower boundary condition

continue

if n\_ix == len(x\_range) - 1:

V[m\_ix, n\_ix] = 0 # upper boundary condition

continue

prev\_left = V[m\_ix, n\_ix - 1]

prev\_right = V[m\_ix , n\_ix + 1]

prev\_center = V[m\_ix, n\_ix]

V[m\_ix+1, n\_ix] = ds\*a\*(prev\_right - prev\_left)/(2\*dx) + ds\*b\*(prev\_right - 2\*prev\_center + prev\_left)/(dx\*\*2) + prev\_center\*(ds\*c + 1)

return V, x\_range, s\_range

if \_\_name\_\_ == '\_\_main\_\_':

N = 100

k = 2

x\_range, dx = np.linspace(np.log(50), np.log(100), N, retstep=True)

ds = k \* dx\*\*2

M = int(round(1/ds))

V, x\_range, s\_range = finite\_difference\_solver(M, N)