

#Exemplul 2

import numpy as np

def f(x):

miu = np.array([1, -1, 0, 2 ])

cov = np.array ([[1.0 , 0.5, 0.8, 0.3],[0.5, 1.0, 0.6, 0.1], [0.8, 0.6, 1.0, 0.2], [0.3, 0.1, 0.2, 1.0]])

det = np.linalg.det(cov)

inv = np.linalg.inv(cov)

d = x - miu

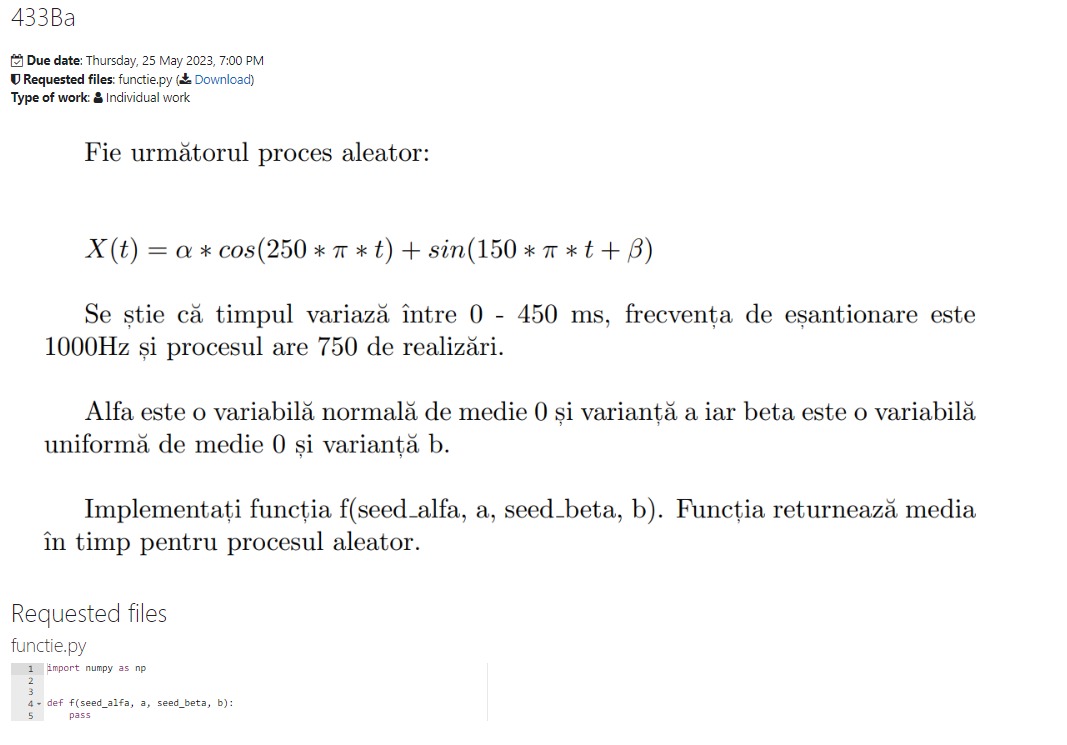
exp = - (1/2) \* np.dot(np.dot(d,inv),d) # produsul scalar intre 3 vectori

coef = 1 / (((2\*np.pi)(4/2)) \* (det \*\* (1/2)))

dens = coef \* np.exp(exp)

return dens

print (f([1, 2, 3, 4]))



#Exemplul 3

def f(seed\_alfa, a, seed\_beta, b): #a, b variantele

fs = 1000

t = np.arange(0,450/1000, 1/fs)

x = np.empty((750, len(t)))

np.random.seed(seed\_alfa)

np.random.seed(seed\_beta)

alfa = np.random.normal(0, a, size = 750)

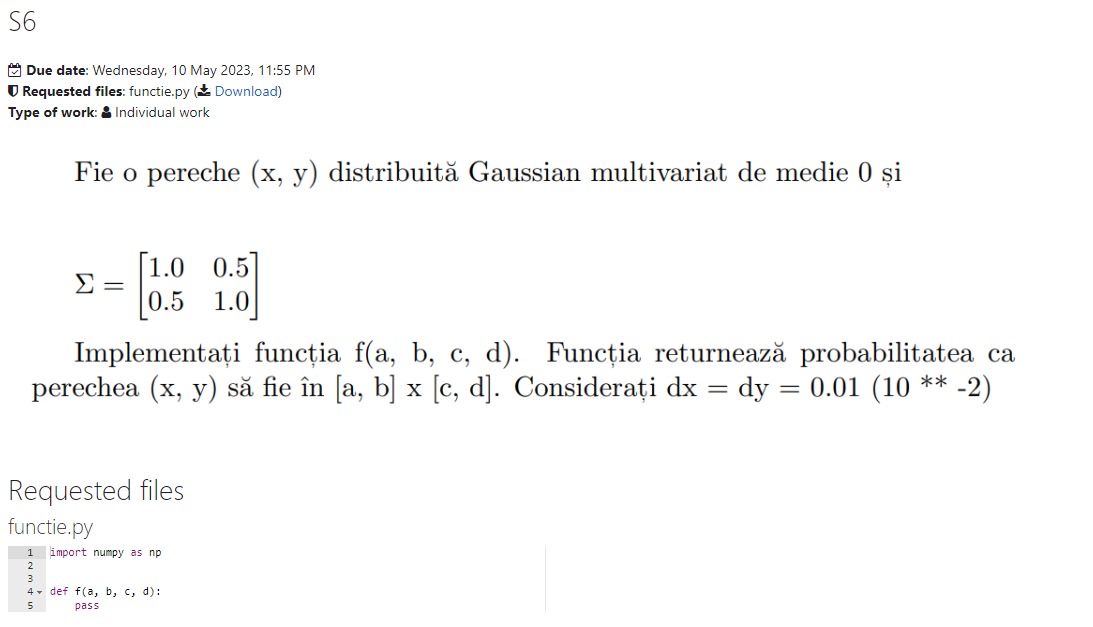
beta = np.random.normal(0, b, size = 750)

for i in range(750):

x[i] = alfa[i] \* np.cos(250 \*np.pi\*t) + np.sin(150 \* np.pi + beta[i])

mean = np.mean(x)

return mean



def f(a,b,c,d):

dx=0.01\*(10\*\*(-2))

dy=0.01\*(10\*\*(-2))

miu=np.zeros(2)

sgm=np.array([[1,0.5],[0.5,1]])

X = np.arange(1, 7, 10\*\*(-2))

Y = np.arange(1, 7, 10\*\*(-2))

X, Y = np.meshgrid(X, Y)

det = np.linalg.det(sgm)

inv = np.linalg.inv(sgm)

f = np.empty((len(X), len(Y)))

for i in range(len(X)):

for j in range(len(Y)):

if X[i,j]<a or X[i,j]>b:

X[i,j]=0

if Y[i,j]<c or Y[i,j]>d:

Y[i,j]=0

C = np.array([X[i,j], Y[i,j]])

f[i, j] = np.exp((-1/2) \* (C-miu) @ inv @ (C-miu))/((2\*np.pi)\*\*(2/2)\*det\*\*0.5)

P=np.sum(f)\*dx\*dy

return P

print(f(5,6,3,4))