Session 7 – Assignment 1:

Problem Statement 1:

Given a sequence of n values x1, x2, ..., xn and a window size k>0, the k-th moving

average of the given sequence is defined as follows:

The moving average sequence has n-k+1 elements as shown below.

The moving averages with k=4 of a ten-value sequence (n=10) is shown below

i 1 2 3 4 5 6 7 8 9 10

===== == == == == == == == == == ==

Input 10 20 30 40 50 60 70 80 90 100

y1 25 = (10+20+30+40)/4

y2 35 = (20+30+40+50)/4

y3 45 = (30+40+50+60)/4

y4 55 = (40+50+60+70)/4

y5 65 = (50+60+70+80)/4

y6 75 = (60+70+80+90)/4

y7 85 = (70+80+90+100)/4

Thus, the moving average sequence has n-k+1=10-4+1=7 values.

Code:

import numpy as np

def running\_average(x, N):

out = np.zeros\_like(x, dtype=np.float64)

dim\_len = x.shape[0]

for i in range(dim\_len):

if N%2 == 0:

a, b = i - (N-1)//2, i + (N-1)//2 + 2

else:

a, b = i - (N-1)//2, i + (N-1)//2 + 1

#cap indices to min and max indices

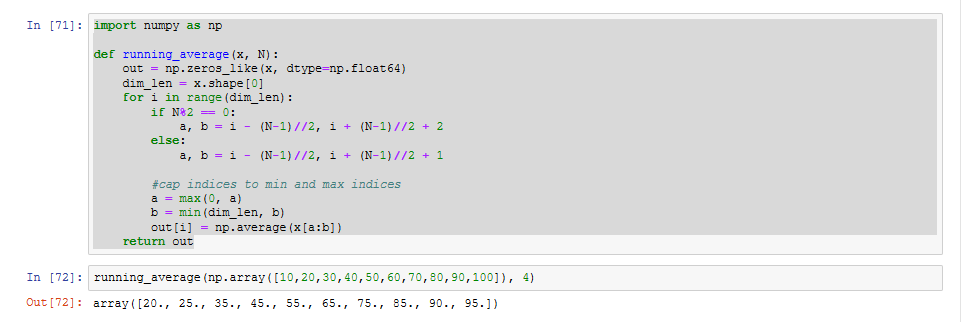
a = max(0, a)

b = min(dim\_len, b)

out[i] = np.average(x[a:b])

return out

Output:



Problem Statement 2:

Write a function to find moving average in an array over a window:

Test it over [3, 5, 7, 2, 8, 10, 11, 65, 72, 81, 99, 100, 150] and window of 3.

Code:

import numpy as np

def moving\_average(a, n) :

ret = np.cumsum(a, dtype=float)

ret[n:] = ret[n:] - ret[:-n]

return ret[n - 1:] / n

Output:

