test_addition

April 18, 2025

[1]:
```
import numpy as np
import matplotlib.pyplot as plt
```

[2]:
```
import random
```

1 1. Preparing a dataset

[3]:
```
rangeData=20
lenData =1000
testProportion = 0.3
testEnd = round(lenData * testProportion)
```

Now generate a random number

[4]:
```
dataIn = np.random.randint(-rangeData,rangeData+1,size=(lenData,2))

print(dataIn)
```

```
[[ -5  -6]
 [ -8  -7]
 [  6  15]
 …
 [ 10   9]
 [  5  20]
 [ -6 -11]]
```

[5]:
```
dataOut=dataIn[:,0] + dataIn[:,1]

print(dataOut)
```

```
[-11 -15  21  -2 -16   5  18  -4  12 -11 -23   6 -13 -18 -16  -5  11  13
  22 -26 -27  13  10  10 -11 -26  -1   8 -18 -19 -19  -1  24  -2  38  -7
 -10  13   3 -32  10   3   5  -4 -19  23  -1 -10  -5  16 -30  14  12  11
  22 -12  19  28 -12 -10  -2 -19 -25  29  16   5  29 -10   5   7   7   7
   8   1   8  30  11  -1  28   6 -12   3   3  -9   1 -26 -28 -33  20  -7
 -16   2 -30 -13  14   7  29  -1 -14   7   1 -11 -36  24  23   7   8 -19
   1  -3 -31  -5  33 -28  19   8  28  -5 -23   7 -31   0 -28   6   3  17
```

35	17	13	-1	-5	-3	-17	-22	-29	31	-6	16	-6	-13	-20	-21	-17	-24
-2	-24	9	6	-6	-17	-3	31	-35	-5	-16	-11	29	-23	31	9	15	0
25	10	-20	-3	9	29	-1	12	-2	8	-11	-18	-6	-8	-9	-11	14	4
-3	11	-9	6	5	-4	-3	-1	39	1	29	15	-25	28	-1	-18	13	23
-10	-10	12	-1	23	-17	1	11	-23	5	19	-40	-30	30	-2	15	10	2
-12	9	19	13	29	16	4	-21	32	-18	-15	-4	-14	3	-17	-6	7	5
13	15	-20	-19	7	14	1	-18	-11	2	16	14	-31	29	-9	-3	-15	2
7	-27	35	-19	-26	-20	17	1	22	-25	-24	7	-16	15	-13	-7	-2	14
21	-8	7	-11	35	-21	0	-9	5	5	20	-17	8	-8	-8	-27	6	30
2	-4	2	5	-3	-17	0	-3	11	17	6	-13	-14	-1	-19	-10	7	-7
8	24	-10	31	6	29	-32	7	16	11	-9	-14	-23	15	11	-18	-19	25
8	18	1	-29	-5	-14	29	-25	12	1	15	7	4	-3	3	8	-3	10
10	-4	13	-4	16	-28	-3	-12	-12	-12	-6	-20	-9	5	5	-26	-10	-18
19	-8	-17	-6	6	28	-39	3	-3	15	-6	8	11	-25	-20	18	-6	-30
37	-9	-1	5	4	-7	-1	0	25	21	27	-13	26	30	15	29	-2	2
-11	-21	-14	-18	5	22	-20	16	14	23	-12	-2	22	-2	-7	19	-25	6
-16	4	1	32	-2	-10	-17	19	-4	-26	-1	-18	36	3	13	-4	-18	1
4	-19	-37	33	-2	17	-23	-10	-1	11	-32	12	-10	-3	-16	11	-15	4
18	38	-9	-16	-6	3	36	11	0	-13	-17	-28	-18	-23	18	-25	10	25
-27	15	-1	-12	12	1	7	16	15	20	3	10	-17	-30	7	10	3	-8
-8	23	-3	-12	7	18	-1	-2	-2	-15	-28	-5	-12	-1	-3	-31	6	-7
-15	-12	33	-5	-26	-31	7	-19	4	9	8	-14	2	-15	-19	-6	10	17
-16	32	29	28	-37	-15	-11	-25	-13	-7	16	-33	-14	5	17	-14	1	-5
8	3	8	-21	-8	-7	-13	-35	-3	-1	6	-5	9	10	-8	-4	1	38
-25	-28	-12	-8	21	12	4	-18	-9	36	6	9	20	13	7	2	-16	-35
15	5	2	24	-27	-28	17	-6	4	-13	-2	-24	-10	-8	-26	33	-10	-12
-22	-18	-9	12	12	13	-18	-11	-14	19	-32	5	-8	16	1	6	-5	-20
20	24	6	20	5	8	-25	-2	10	8	3	-13	-4	-20	-20	30	25	4
3	-15	2	2	10	11	15	-6	-8	-25	21	-6	12	19	-15	-10	21	30
-2	-15	35	-5	11	24	15	-22	33	11	-24	28	-28	-5	14	-8	-4	-14
-9	-5	-27	1	20	13	16	24	16	1	-2	-26	-8	22	17	-6	-13	13
19	25	9	-14	-20	-28	-13	23	-22	-11	-6	27	-9	-26	-10	-11	-3	-1
6	-13	1	-5	24	24	-3	0	9	-9	4	7	7	9	2	15	-8	0
-15	9	-30	-8	-16	-1	-10	2	5	-17	3	2	7	10	12	4	-10	4
9	12	3	-15	25	-5	12	-26	-26	30	-17	0	-12	4	22	32	19	11
-7	-18	36	-26	8	2	-4	1	-24	0	25	2	20	-21	25	13	2	29
33	-3	8	-6	-21	25	-20	-18	1	8	10	-6	-14	-28	-13	-8	-5	5
-27	10	17	-20	-1	1	-5	-4	-18	6	-17	-18	18	-29	-16	-10	28	0
23	-37	7	13	11	-2	-21	5	-17	-17	-10	17	-7	-18	0	-14	-4	15
0	40	-32	4	-23	31	21	5	22	26	3	18	-4	1	-22	-4	2	-32
-23	38	19	20	-4	-18	1	5	35	-12	17	-11	-2	23	-5	1	21	1
0	-5	-32	19	-7	-3	11	-12	-10	4	2	-11	-12	4	10	11	34	1
4	2	21	-26	8	-2	-19	1	-6	4	-3	10	3	-1	30	-1	6	-20
-10	16	-16	3	12	20	21	11	20	27	-24	11	-23	4	17	26	2	-28
-3	10	5	-28	-1	6	4	-19	19	7	10	11	5	-2	10	18	-12	-5
15	13	7	24	-21	-5	4	27	-31	32	-10	5	-27	-3	20	18	-33	14
-12	-15	8	-26	20	4	26	-19	-10	8	9	-4	26	-15	-8	-5	35	-23
-28	-12	-23	19	12	-19	31	10	23	6	30	5	24	3	-7	13	34	23

```
 -20  36 -15  -3  -9  -1  -7  19  25 -17]
```

[6]:
```
# It creates a column vector of ones: np.ones([lenData, 1])

# Then it horizontally concatenates this with dataIn:


dataIn = np.concatenate( [     np.ones([lenData,1])    ,
                               dataIn
                        ] ,
                         axis=1
                       )

# dataIn = np.concatenate([np.ones([lenData,1]), dataIn], axis=1)
```

In NumPy:

axis=0 refers to rows (vertical stacking)

axis=1 refers to columns (horizontal stacking)

[7]:
```
a = np.array([[1, 2],
              [3, 4]])

b = np.array([[5],
              [6]])

np.concatenate([a, b], axis=1)
```

[7]:
```
array([[1, 2, 5],
       [3, 4, 6]])
```

Now, making the final dataset.

[8]:
```
print(testEnd)
print(f"Testing = 0 : {testEnd} and training is {testEnd+1} : {lenData}")
print(f"Also we have Input and Output : in Testing and Training ")
print("------------------")
print(f"Normally 70% in training and 30% in testing")

testingIn = dataIn[0:testEnd]
testingOut = dataOut[0:testEnd]

trainingIn = dataIn[testEnd:]
trainingOut=dataOut[testEnd:]
```

```
300
Testing = 0 : 300 and training is 301 : 1000
```

```
Also we have Input and Output : in Testing and Training
-----------------
Normally 70% in training and 30% in testing
```

[9]:
```python
print(testingIn[0])
print(testingOut[0])

print(trainingIn[0])
print(trainingOut[0])
```

```
[ 1. -5. -6.]
-11
[  1.   0. -14.]
-14
```

2 2. Setting up Neural Network

[10]:
```python
# Only one input layer ----> Length = 3 (1 bias + 2 numbers )
# Output layer lenth = 1 (result)

# setting a random weight in the range of -2 to 2
weights = 4*np.random.rand(3)-2

print(weights)
```

```
[-0.2769033  -1.75657219 -0.80155973]
```

[11]:
```python
# No need of activation function

# Output of neural network is DOT PRODUCT of two vectors.

def calculateOut(x,w):
    # x = input
    # w = weights
    return np.dot(x,w)
```

[12]:
```python
testIndex=10

calculateOut(trainingIn[testIndex] , weights)
```

[12]: -11.771348918645202

[13]:
```python
print(trainingIn[testIndex])
one = trainingIn[testIndex]

print(weights)
two = weights
```

```python
print(np.dot(one,two))
```

```
[ 1.  7. -1.]
[-0.2769033  -1.75657219 -0.80155973]
-11.771348918645202
```

[14]:
```python
# Since we are giving only index, so we need to check everything as our output␣
↪matches or not
#
calculateOut(testingIn,weights)
```

[14]:
```
array([ 13.31531606,  19.38659237, -22.83973247,  16.6064155 ,
        23.05318949,   9.08547245, -31.89520277,   9.61442284,
       -18.49073224,  -0.05485836,  29.61912008, -10.81633646,
         8.23334832,  25.61132141,  29.7382767 ,  -5.81922922,
       -11.95909775,  -8.78715493, -27.46134204,  32.97881174,
        29.96032164,  -9.74216739, -17.84262523, -16.88761277,
        17.1353659 ,  39.66389895,  -2.34038095, -23.87960543,
        18.9262342 ,  17.81776902,  15.9077441 ,  16.75986823,
       -28.10944905,  -9.17892088, -47.92639745,  13.92912696,
        20.153856  ,  -5.92211755,  -7.4566448 ,  41.60821998,
         0.30261148,   4.95851716, -12.8798141 ,   8.65941038,
        32.1429559 , -23.48783948,   9.11976856,   2.96363174,
        -1.04416693, -22.65198363,  39.05008805, -26.77893892,
        -9.89562011, -18.64418496, -21.73126729,   1.70171383,
       -31.74175005, -34.18072536,   5.52176367,  20.153856  ,
        18.51644042,  20.68280639,  27.40218971, -32.11724772,
       -29.33707085,  -0.46465214, -35.93729755,  23.97390583,
       -16.69986394,  -6.8428339 , -22.12303324,   6.52734052,
       -21.01456806,  15.15674876,   1.90573095, -35.78384483,
       -14.82413513,  10.07478102, -32.27070044, -20.36646105,
        25.5770253 ,   3.04849225,   5.91352962,   8.84715922,
         3.69659926,  32.02379928,  33.62691875,  38.58972988,
       -31.58829732,  23.47925155,  12.54805244,  15.31020148,
        36.18505068,  23.51354766, -12.45375204, -21.16802078,
       -37.84732247,   6.25473118,  11.89994543, -17.34797095,
        -4.89851287,  26.68549048,  43.85944646, -29.06446151,
       -26.35287685,  -4.93280898, -10.50943101,  22.59283131,
       -14.44863746,  17.40797524,  43.67169762,  -7.72925414,
       -44.87361124,  35.53694367, -30.78673759,  -2.86933134,
       -38.95578765,  20.92111962,  28.66410762,   6.52734052,
        36.98661041,  14.04828358,  31.71689383,  -6.99628663,
       -12.23170709, -33.00366796, -45.52171825, -24.40855583,
       -14.51722968, -17.62058029,  -2.95419185,  -5.51232377,
        14.30462463,  20.22244822,  35.38349094, -39.45044194,
        -8.83771932, -32.20210822,  -6.9276944 ,   7.27833586,
```

```
 31.03449071,  28.97101307,  20.98971184,  33.28571719,
  5.146266  ,  25.64561752, -10.35597829,  -1.26621187,
 -4.06265703,  27.67479905,  15.49795032, -39.45044194,
 44.01289918,   0.86585799,  18.27812719,   5.67521639,
-39.75734739,  21.97902041, -40.4054544 ,  -1.76086616,
-17.07536161,   7.36319637, -29.86602124, -24.52771244,
 32.94451563,  12.63291294, -17.0410655 , -32.11724772,
 16.75986823, -13.71566995,  -7.26889596,  -5.73436872,
  8.54025377,  17.01620928,  -6.9276944 ,   4.22554965,
  7.89214676,  15.22534098, -10.54372712,  10.84204464,
 10.72288803, -13.86912267,   5.98212184, -13.68137384,
  9.08547245,  17.25452251,   9.76787557, -11.89050553,
-49.68296964,  -0.12345058, -36.89231001, -18.03037407,
 24.53715234, -30.36067552,  10.07478102,  17.01620928,
 -4.96710509, -33.99297652,  21.10886846,   1.05360682,
-10.85063257,  -4.25040586, -31.12793915,  20.03469938,
-12.53861254,  -7.18403546,  27.70909516,  -5.23971443,
-31.74175005,  50.88573523,  37.14006314, -37.69386974,
  2.28122862, -30.44553603, -18.79763769, -14.29518474,
  9.3418135 , -18.95109041, -17.41656317, -19.29229197,
-41.6673723 , -16.92190888,   8.93201972,  24.19595078,
-38.34197675,  17.01620928,  30.84674188, -10.44083879,
 20.49505756,  -7.4566448 ,  26.7197866 ,  16.94761706,
-24.03305816,  -6.19472689,  -5.92211755, -31.40054849,
 24.3494035 ,  18.77278148,  -9.70787128,  -6.72367729,
  8.47166155,  22.74628404,  20.00040327,   8.62511427,
-11.19183413, -23.91390154,  38.89663533, -38.80233493,
 18.39728381,  -1.69227394,   7.92644287,  -9.52012244,
-11.61789619,  29.96032164, -42.65668087,  21.63781885,
 26.29372453,  27.21444088, -13.90341878,  -2.0334755 ,
-24.59630466,  35.99730184,  23.7355926 ,   6.52734052,
 12.54805244, -23.76044882,  10.14337324,   4.37900237,
 -3.44884613, -26.77893892, -26.6597823 ,   5.18056211,
 -9.70787128,   9.49526623, -46.47673071,  28.01600061,
 15.95830849,  25.08237102, -14.78983902, -10.96978918,
-22.03817274,  30.53983643, -16.23950576,  13.77567424,
  3.27053719,  29.00530918,   8.28391271, -43.4239445 ,
  1.94002706,  -5.6657765 , -13.34017228, -20.51991377,
 -7.42234869,  20.98971184,   6.40818391,  20.27301261,
-18.64418496, -11.03838141, -23.23149842,  17.78347291])
```

[15]:
```
# Testing the accuracy

# For that , we only aim to calculated how many is correct ;

# If TestingOutput - CalculateOutput = 0 ; Then it is correct
# Else the calculation is incorrect
```

```
# So,  (TestingOutput - CalculateOutput) > o ----Then represent is as +1
# And
# So,  (TestingOutput - CalculateOutput) < o ----Then represent is as -1

# Then we will get the output as -1 , -1 , 1 , 0 , 1, 1 like this

# Since all +1 and -1 are the incorrect values; else we would have got (0)

# Therefore, makking all error as (1) and correct as (0). And on calculating␣
↪total, we would get the total incorrect values
```

[16]:
```
testingOut-calculateOut(testingIn,weights)
np.round(testingOut-calculateOut(testingIn,weights))
np.sign(np.round(testingOut-calculateOut(testingIn,weights)))
np.abs(np.sign(np.round(testingOut-calculateOut(testingIn,weights))))
np.sum(np.abs(np.sign(np.round(testingOut-calculateOut(testingIn,weights)))))

np.sum(np.abs(np.sign(np.round(testingOut-calculateOut(testingIn,weights))))) /␣
↪testEnd

1- np.sum(np.abs(np.sign(np.
↪round(testingOut-calculateOut(testingIn,weights))))) / testEnd
```

[16]: 0.01666666666666672

[17]:
```
def manual_accuracy(testingIn, testingOut, weights):
    predictions = calculateOut(testingIn, weights)
    error = testingOut - predictions

    # Convert error to 1 or -1 based on sign
    for i in range(len(error)):
        if error[i] > 0:
            error[i] = 1
        elif error[i] < 0:
            error[i] = -1
        else:
            error[i] = 0  # optional: handle case where prediction == actual

    # Count correct predictions (i.e., where error == 0)
    correct = 0
    for i in range(len(error)):
        if error[i] == 0:
            correct += 1

    # Calculate accuracy as a percentage
    return correct / len(error)
```

[18]:
```python
def accuracy(testingIn,testingOut,weights):
    return 1-np.sum(np.abs(np.sign(np.
↪round(testingOut-calculateOut(testingIn,weights))))) / testEnd
```

[19]:
```python
accuracy(testingIn,testingOut,weights)
```

[19]: 0.01666666666666672

2.0.1 Now, for the neural network, the accuracy means how much it is write or wrong;

But this means nothing for the learining. Only the error helps in learning

So with the help of error function, we can improve the network.

2.0.2 Error

[20]:
```python
def error(predictedValues,corectValues):
    return np.sum((predictedValues-corectValues)**2)
```

[21]:
```python
# error(calculateOut,testingOut)

# error(calculateOut(trainingIn,weights),trainingOut)

error(calculateOut(trainingIn[testIndex],weights),trainingOut[testIndex])
```

[21]: 315.820842388232

Accuracy is used with the testing Dataset while

Error is used with the training dataset

---

3 Now with the error minimization

3.1 Gradient of Error

[22]:
```python
def gradientSlow(x,w,correctValues):
    return np.array([  2 * (calculateOut(x,w) - correctValues) * x[indexWeight]␣
↪for indexWeight in range(3) ])
```

[23]:
```python
gradientSlow(trainingIn[testIndex],weights,testingOut[testIndex])

# it is giving three output of weights : w1, w2 and w3
```

[23]: array([22.45730216, 157.20111514, -22.45730216])

[24]:
```
def gradient(x,w,correctValues):
    return 2 * (calculateOut(x,w) - correctValues) * x
```

[25]:
```
gradient(trainingIn[testIndex],weights,testingOut[testIndex])
```

[25]:
```
array([ 22.45730216, 157.20111514, -22.45730216])
```

3.2 Learning Rate

[26]:
```
iterations = 100000

learningRate = 0.0001

# weights = weights - learningRate * gradient()


# For documentation ; making a list
errorlist = [error(calculateOut(trainingIn[testIndex],weights) ,␣
↪trainingOut[testIndex] )]
weight_list = [weights]
```

[27]:
```
for i in range(iterations):
    # pick random input
    index = np.random.randint(lenData-testEnd)
    # update weight
    weights = weights - learningRate *␣
↪gradient(trainingIn[index],weights,trainingOut[index])
    weight_list.append(weights)
    # Calculate error
    er = (error(calculateOut(trainingIn[index],weights),trainingOut[index]))
    errorlist.append(er)
```

[28]:
```
errorlist
```

[28]:
```
[315.820842388232,
 518.1792001346853,
 4750.934483074319,
 896.6774794212665,
 408.1249681440157,
 2237.57616025153,
 53.49986955008886,
 648.2500240443367,
 3180.104285453929,
 278.2280542671019,
 77.35380806959002,
 118.74050431240813,
```

209.78873866595575,
136.45534070745117,
833.7499367590063,
3.3402347960360705,
237.73788471825384,
1864.9275994191519,
995.6070716746941,
556.596516085255,
2.687690147294702,
994.9129282477743,
270.74256282299393,
9.975395088293801,
5.882393809266465,
101.12844128521463,
191.5436318444525,
84.12325658872653,
41.781627129805486,
40.995575991693926,
51.795001884338205,
227.59709808343936,
700.0548308401964,
152.16615420259467,
297.8817749067755,
156.63296637993423,
84.28725511849161,
3.3560036255355654,
45.623476240708456,
12.995061579678142,
1.2818258527253135,
38.079272518928136,
23.736956802853587,
3.200336098018333,
0.6945041101482481,
39.40767185025295,
4.880661236200896,
0.2951393917012479,
6.91105815519295,
84.20547639388896,
202.64884255061642,
3.618592597491607,
58.32122695698092,
33.04567728026098,
127.5785155520089,
60.17928621809471,
9.049800641780193,
132.7456481035311,
126.78931925977001,

```
150.57407929366332,
14.905708652464343,
0.0010053633116649198,
0.8684003506793343,
111.64750252438382,
47.56858101898743,
0.34108479523634716,
54.953117236400026,
38.185792190405486,
61.76653237757244,
31.58822110893368,
70.45757921126958,
0.017410964880927644,
1.3001660717890118,
32.10228248290299,
11.32272067000505,
1.9018524329074802,
51.68471998531909,
1.190837358939049,
2.5685167464370267,
6.798628031546653,
4.687400051794021,
20.73046540651832,
20.251608080247475,
1.4924602866756609,
6.492358203227307,
31.384406492511978,
0.32075099248675254,
8.614801688929289,
8.171683670535744,
0.44479102429051715,
18.838445373307703,
0.06980138418575824,
0.036603569655675586,
15.304074616514313,
2.985599294439382,
0.058879856690653695,
7.045855941741748,
1.6840197518317084,
2.703970901432967,
0.02940951744708954,
0.007453003522121001,
0.02882697966113189,
5.379183002248453,
0.22817917568288762,
0.1524111193735972,
0.08723476471670097,
```

```
0.08780050432071695,
1.78063372075458,
0.16500951138198622,
1.3921264871298042,
1.3037897393890552,
0.03283141799284693,
0.04638038090034248,
0.9798072877837077,
1.4730019092715114,
1.2177235809604792,
0.23144519997242363,
0.6100704135896325,
2.3370570478212445,
1.117815355780521,
1.3420463822129367,
0.47630145150595354,
0.4539617276268679,
0.12622649153044962,
0.25944030858273115,
0.11592125628212062,
0.33195618262808924,
0.004552669050092646,
1.3170915917685877,
0.446404569862404,
0.19184825761059743,
0.06684440714439933,
0.31385702145022204,
1.06551816828357,
0.2875793952656449,
0.002660907598116851,
0.28650268548744273,
0.0002902959474312641,
0.75231288044169,
0.162412475820004,
0.4018067666304306,
0.028121572339665814,
0.1468446218807878,
0.0049106229218236095,
0.0028398068673124487,
0.06408153803129912,
0.21112337981928764,
0.000565108776348156,
0.011886088358555653,
0.23997467076810922,
0.6666096779912377,
0.6276494380806439,
0.1572235425058775,
```

0.4646495874783191,
0.007908064568978851,
1.1042541423999264,
0.00654779533037365,
0.052202185111795876,
0.13177010063050065,
0.5329975680506176,
0.0947191720850914,
0.016744213059104684,
0.0379037348182499,
0.03452537935714728,
0.04962878002146915,
0.0005110119075261489,
0.0018677346960761926,
0.01740347028567333,
0.0646497458063355,
0.28176052607251145,
0.4157545385138413,
0.2659522798473051,
0.000882699943019732,
0.0037774241340791845,
0.35119592070804273,
0.007445439133179267,
0.07460315781571175,
0.36638686554826827,
0.06771633971178947,
0.00039804251055040326,
0.0034356767608754203,
0.09081938555495422,
0.0020020314607737955,
0.027682823160540843,
0.015594458533136743,
0.029028106688398204,
0.11088541834274662,
1.2827190991298356e-05,
0.20784629624509837,
0.06680361485624355,
0.19490576471466112,
0.1836645805558408,
0.09008846137579529,
0.03710241676701951,
0.06239413505757202,
0.11553179818017059,
0.06654392329848977,
0.09846052985636156,
0.04429895686427743,
0.02848885635456339,

0.03481507053809858,
0.04232415226365321,
0.02829440000614942,
0.11987501122269546,
0.08717154158893743,
0.09929471961798032,
0.05673422547332943,
0.02346856184169672,
0.08522979205015349,
0.030498395676261194,
0.02066409454607421,
0.0191797756648077,
0.034852750170884975,
0.049572611641716925,
0.04796556734481623,
0.06244433127782948,
0.11230335817330118,
0.02622612289250654,
0.008397984773195423,
0.08824933124332317,
0.07675200385362682,
0.04605114095423844,
0.11776155961692342,
0.03954542350516655,
0.02687689541753247,
0.05684978146375615,
0.026446775530538827,
0.08902003496020125,
0.09049216281362253,
0.06093346008039113,
0.08684253704027099,
0.0621340517192423,
0.03897313814608583,
0.07219688513778268,
0.06200784445507611,
0.05896785565185441,
0.061411560529778655,
0.051024110026049965,
0.058110450349660635,
0.05694852440977055,
0.04554549309503228,
0.05083191344093667,
0.04893716451389651,
0.04396042424222716,
0.05804465015458033,
0.06180313858978864,
0.0741932686858005,

0.06163112530370409,
0.05785880654033426,
0.05628198196625327,
0.056400951901544975,
0.0653454558812039З,
0.05843284171689026,
0.0536851880742148,
0.05748589556144305,
0.05039902132876749,
0.05992422327636923,
0.05781425570205855,
0.055771377076300414,
0.05141723547973227,
0.050102305276676654,
0.05771079733362915,
0.06223345841292853,
0.05699957668260167,
0.0582073411183706,
0.05154173173204965,
0.05889245388427987,
0.05500695740776451,
0.04416122231668834,
0.03736093739600644,
0.03453918674109234,
0.0689670201488077,
0.07001771688285834,
0.06367476382322416,
0.06658769397614706,
0.05765704887522834,
0.04867160110499761,
0.058865566606448594,
0.03974169344765215,
0.06444675253867888,
0.05159889689027116,
0.04753928542241531,
0.06478592731282735,
0.047806975751988945,
0.04787326768159882,
0.044477543074189385,
0.04794828498236466,
0.0719479117501925,
0.06560315766514624,
0.06591968702951669,
0.04868551825811187,
0.046339117530406905,
0.045517271312879595,
0.0452302226979499З,

0.03263629771932131,
0.04471476912469271,
0.035838217076455924,
0.044271929502869574,
0.027478178032648095,
0.023101289204360845,
0.023189961919997358,
0.02595420294954103,
0.011474965046791331,
0.05374483554944442,
0.01995127175580268,
0.015465175845089888,
0.12993999286235405,
0.10879910177256154,
0.01723018975977872,
0.07835011807970173,
0.03973209590944559,
0.031822995106183256,
0.09469126549654538,
0.06070949732218866,
0.049612191110665116,
0.0709134491503001,
0.04839691504153137,
0.04930997166811189,
0.021032555209857303,
0.03850806007924536,
0.03685289422066637,
0.042134194844949166,
0.02643659857728762,
0.06954498910207194,
0.04317081871021212,
0.015331743261549994,
0.02296155484054693,
0.01379940190391015,
0.11706716504309979,
0.022145161790991498,
0.09793723848669318,
0.0753487819769045,
0.0424453800749195,
0.06654653697728606,
0.03219851102552884,
0.05228023939564017,
0.10947649833262227,
0.061314007894495884,
0.030151076993402994,
0.021609509185973264,
0.0314194258012131,

```
0.05885488809911969,
0.05692931200016743,
0.046320608688116154,
0.08690546619295811,
0.023974166933338346,
0.033272679212665954,
0.04070887737892609,
0.01657539208782861,
0.07507788186148244,
0.07185340686677787,
0.0548134293091378,
0.015254944039648655,
0.10212981821043253,
0.026171526160537008,
0.08059413642808842,
0.09491519545451498,
0.034369128355206384,
0.045576496435626696,
0.04082447748371999,
0.01770463093430351,
0.028189416594502963,
0.03575321402247655,
0.12027910216107368,
0.029117676256470815,
0.09893901267427417,
0.05073282976780131,
0.023905489057927047,
0.07785038236815568,
0.05591865449308726,
0.0328716292173884,
0.02683976852814465,
0.08686823858109023,
0.06023579612374704,
0.04276883671658483,
0.04192595138128325,
0.03259838822233371,
0.030839412896259855,
0.09652435007841452,
0.043335948714330254,
0.05054120463735055,
0.08470136297269185,
0.06905742078571665,
0.039636053061650155,
0.07272713422705909,
0.04488306434636181,
0.055232051094047815,
0.028131787480378874,
```

```
0.02918612487492136,
0.02144217220279538,
0.02967753676042759,
0.07288371586833814,
0.07639452468401602,
0.052813948918641926,
0.08115818357025939,
0.06405774662734125,
0.05649293794429453,
0.05302743112455546,
0.06800793798707411,
0.048041834506417795,
0.06636837518173119,
0.04616078629130069,
0.03197359895876663,
0.06216597025291133,
0.028893127056576328,
0.07173812790703268,
0.07305428558640456,
0.06185552648340804,
0.05251794473174834,
0.05211189504522865,
0.04820728748406466,
0.04044998800689028,
0.06201112149330699,
0.05040975334313366,
0.05720587907474544,
0.045547909884167674,
0.06093153236304972,
0.04537853739491926,
0.06044661452998078,
0.05596778478555221,
0.04894429624525371,
0.054548982382534805,
0.046312031608566104,
0.05694481400280491,
0.05178038023220814,
0.05194941957281972,
0.048978785330180415,
0.0548594549541196,
0.043005095513480855,
0.058941821672008565,
0.0592090686647399,
0.049848019805325564,
0.042659943039997913,
0.05629632396738562,
0.052282213336372424,
```

0.041973045404094696,
0.0659810190282014 2,
0.0566884919057796 3,
0.046929535278727466,
0.05095352670212056,
0.0397111957277964,
0.0652243713291650 5,
0.0547785912313724 1,
0.0456986619070731 4,
0.039950011620918054,
0.0603095279459029 3,
0.044370088887575265,
0.042630602354414064,
0.0643209553356935 4,
0.059492071933323724,
0.0575894369495212 4,
0.0521522855269028,
0.0588600415708876 1,
0.0366246064827626 7,
0.0649441795610263 3,
0.045547502483155936,
0.0536541542219474 8,
0.053307993727081324,
0.043103875750962216,
0.029106257833156467,
0.059436502637484005,
0.0637463226167382 5,
0.051895420727592644,
0.052269013578576044,
0.0487221294676080 8,
0.049772387478746315,
0.0536788161506867,
0.0400866328473187 1,
0.0564746797356135 6,
0.0523517085600247 5,
0.0521620285251262 1,
0.048270934189673285,
0.0519372861266791 2,
0.055114982621914645,
0.049689873959392235,
0.034718236302676654,
0.0415517855006332 7,
0.0666498835642733,
0.0413182701993626 7,
0.0492940345537012 6,
0.0611142875259347 2,
0.0544733343130251 2,

0.049540640513760184,
0.05783023433651857,
0.03460374836960141,
0.03135710326515357,
0.060182600544549845,
0.046846356642419686,
0.056896745181785446,
0.060796663274707854,
0.03843345544770264,
0.04239262180705866,
0.029831713497000086,
0.06093671052270212,
0.05658290017993341,
0.0510832965451028,
0.02372096290288429,
0.06831972630440497,
0.04945319034063188,
0.07389936118506583,
0.04603984231133089,
0.0625769641436537,
0.041583494531531415,
0.05658755701031442,
0.039013464844467875,
0.025654259049492957,
0.04902816861050271,
0.022903436939662753,
0.030267374348006745,
0.0661910166302303,
0.04695546154465218,
0.035242305447853554,
0.02720118470318746,
0.06341154427827196,
0.05451124011258805,
0.04682863359628285,
0.07643804669643697,
0.047736693566755584,
0.024746492917773318,
0.021921864865116533,
0.08323007245140451,
0.08363833227767972,
0.03659399970990711,
0.03776089302639819,
0.07714599755659145,
0.07706765812747632,
0.06244980422849473,
0.06028381440422284,
0.03856054199073515,

0.05365167748821171,
0.035780541765479415,
0.06966730132732944,
0.05113806108495346,
0.04480277106399553,
0.05803724957724536,
0.04993714845616683,
0.059326061350319786,
0.050416572754728196,
0.04867003647709783,
0.03983859622250591,
0.053234331803177486,
0.043575807520667385,
0.03993426146095911,
0.056372981310802935,
0.04386668672385958,
0.0559069282276504,
0.047775993484502365,
0.05051848749474681,
0.059328168961197005,
0.044332306426310614,
0.04568322795970419,
0.0362142253338649,
0.053324281161494194,
0.04066480445331941,
0.05219757139262867,
0.02868458601674829,
0.0433866500877288,
0.03637523802737711,
0.032662796586855385,
0.02622853081099289,
0.055759457801772834,
0.02895321727954514,
0.09085330335008215,
0.06180781285089592,
0.053472947635359654,
0.05149410634674477,
0.03178553635174305,
0.03963682478393313,
0.07180871961337017,
0.038773010548928885,
0.05260186475453543,
0.048183174813226225,
0.041842609677701224,
0.045584458392389476,
0.034831348296902404,
0.0649416959422647,

0.035922568373999264,
0.05251708244287872,
0.0749258702494688,
0.03120685155884471,
0.063520726494144,
0.030822321558532133,
0.0671959971336341,
0.06359030611394902,
0.06181690123868258,
0.05105347918006516,
0.038118450049358016,
0.04789594754893757,
0.040160907645659806,
0.05915235228944168,
0.05567549066703265,
0.03522522057938059,
0.034258856873615184,
0.036395905524303274,
0.03555468523382378,
0.06809663089021309,
0.035122534835472084,
0.048764494424660784,
0.06689421003442443,
0.05603641841537711,
0.05454019101156657,
0.04340571053358812,
0.048641396329184586,
0.0503652991471469,
0.05415532222949331,
0.05822287982141887,
0.03682948625648684,
0.045813238224128534,
0.04840760139545758,
0.05036522252954987,
0.05556221881155062,
0.038505848027699796,
0.03068041999984569,
0.05941589005736271,
0.03904465336668339,
0.025063400968187134,
0.04134926616331436,
0.015648876645671272,
0.052896290693038736,
0.061812420144405425,
0.017731359590055604,
0.040705056454941387,
0.0299374010713704,

0.04774813352491333,
0.03876531389777605,
0.07807223608068566,
0.05179118337313296,
0.020506847350737552,
0.02003009651356191,
0.052773480300070084,
0.025547975248687682,
0.06140933703010286,
0.08252237030157535,
0.047347133878798485,
0.0176060758347394,
0.017512059993786216,
0.032241097978918616,
0.04265100746647054,
0.08446621641296741,
0.067583139464953,
0.08912593087097963,
0.024328223845607285,
0.06855668019107466,
0.021110052425760667,
0.03656571955681882,
0.0771450507901548,
0.024985519274633546,
0.06193515192219198,
0.023988738110862787,
0.024806987474301295,
0.027545469592064324,
0.05055501986146608,
0.0352491484939852,
0.02062749145876945,
0.03698430584595793,
0.10326388330041308,
0.06386929646260828,
0.07963482091119535,
0.027976966410009987,
0.0263328753317364,
0.06585749020591793,
0.026603681984143864,
0.0599600039738967,
0.029072020340771613,
0.046194730722259984,
0.05590669782603366,
0.03359966199719377,
0.05508586414148689,
0.04488724993295662,
0.05042542415102555,

```
0.07731441763442139,
0.024932549929127922,
0.045807621902977054,
0.0383450417745698З,
0.04910725536460062,
0.011099778758966531,
0.08049404029287417,
0.042524783665496196,
0.033935230058694844,
0.038206160368777324,
0.019946845192156067,
0.03692161899044542,
0.0514220822917304,
0.05961172175184299,
0.025450270039793395,
0.03131581959331221,
0.0639471559614003,
0.05258903023907933,
0.05919170321680743,
0.05957142784410667,
0.05287812787967047,
0.07155627555196081,
0.026257055807313412,
0.057778857319040505,
0.029075122327037194,
0.07788008603966098,
0.06644628749211623,
0.02997010728504811,
0.043915393547935055,
0.048452256381275864,
0.043269186326026554,
0.060153004356942534,
0.03300152080663862,
0.025676190550215403,
0.025897575579442705,
0.08022333617189473,
0.06355015177214446,
0.03384490919978681,
0.03312218503424548,
0.04760056159781677,
0.05288398503013503,
0.04803943381454364,
0.04131829592471289,
0.030532953334281843,
0.029702377806046343,
0.026421251788241406,
0.025116828349053526,
```

0.05881797568700132,
0.07223040800585448,
0.07349896597898815,
0.03399139992845073,
0.047203504329697846,
0.036679537473327035,
0.05425350703836872,
0.05271445484802729,
0.05193599185148247,
0.04423101446020893,
0.05478841482166328,
0.03871607846982811,
0.06199611184166472,
0.05554080630118083,
0.048967314134626266,
0.05352325131838021,
0.05423756839878628,
0.03759097934410432,
0.05446725567738501,
0.05122280927104632,
0.033349805944816124,
0.031802640158060516,
0.057376916916918014,
0.05179072004891553,
0.03528741701243586,
0.035231431392851266,
0.03131396577459591,
0.04075865766914382,
0.036087660719628656,
0.06012779236250966,
0.05374802648075755,
0.04584109612576218,
0.06097060565774951,
0.03443035345613376,
0.03538747740483966,
0.045018107964523005,
0.04242068170818142,
0.02041715324966792,
0.051184645963489074,
0.03965706076023569,
0.04583418787802663,
0.020198165732717696,
0.03606710744100507,
0.01743320556515403,
0.0644417835059591,
0.05676739806777717,
0.023891177939052873,

0.05872002034761229,
0.05487145194185045,
0.01147364346752973,
0.033202594338279455,
0.020446673032242284,
0.034759178598652934,
0.05275354113568451,
0.03499464870996751,
0.06766974672730518,
0.029839752592159343,
0.01747855294988939,
0.059770549250832715,
0.013211756702255982,
0.022925474787377706,
0.03954406021481912,
0.08968839212883818,
0.05264823768112538,
0.054854268345837086,
0.051544037926352586,
0.09298845875419189,
0.014555644614700287,
0.062065061951720714,
0.06677800728420724,
0.015528945550497539,
0.06304547990681196,
0.044308517681589,
0.06423843884013306,
0.03761147296496521,
0.018659806063171042,
0.03163908621454976,
0.0769967178023615,
0.02619756204349468,
0.07039162983141749,
0.05556047377619208,
0.024856864802808475,
0.04227068543574682,
0.03616594088781245,
0.06917927274890666,
0.06227790833661459,
0.05070522064567128,
0.0463130297234466,
0.04807989765032024,
0.03755065958343125,
0.03300646356933805,
0.036528739351173733,
0.020791025875504545,
0.026928824714423732,

0.04617515807133055,
0.01733915183125761,
0.0764510792227456,
0.0336667875170935 9,
0.021144252634711216,
0.030077910135479453,
0.05065101539534269,
0.036468766525300705,
0.07862759387798973,
0.033580717633218946,
0.05049811228939923,
0.04156786753713562,
0.021789104920768368,
0.06037621383220202,
0.0605221218715984,
0.046301331278399874,
0.026894871409914242,
0.06027113913809759,
0.0388423504050545,
0.034316762357898836,
0.03736776420340355,
0.055113814053014624,
0.01835831658372283,
0.02021085054531486,
0.05801582355074469,
0.04419971220390764,
0.03880597712395214,
0.03088204959089607,
0.04192543204340269,
0.08236475195664825,
0.056140542413750424,
0.04748298036956498,
0.03147454383581224,
0.06272616434069281,
0.05194666280553275,
0.062273007300026335,
0.022378291673375572,
0.03748337814972889,
0.027882712641932386,
0.022560464586928012,
0.048832419814917864,
0.07330908561685688,
0.02883466243340474,
0.02196677064324882,
0.0617003408664221,
0.04574487416573983,
0.04463810453956348,

0.04591670278094825,
0.05951939011786163,
0.05357846441269909,
0.04403143542317379,
0.03239762567127244,
0.04250417834753101,
0.04052386825281388,
0.03297046876803467,
0.05423631309737586,
0.05010617941355606,
0.029836911697856138,
0.017031399726514518,
0.05586284979905366,
0.06517232404725892,
0.05615706692493446,
0.04402634597577104,
0.041228124887300314,
0.03889128678100701,
0.02649394833571351,
0.05802787661083934,
0.028301313383542286,
0.045648609792123526,
0.031089180449663224,
0.050582261678010575,
0.019893522320527546,
0.01933243651028235,
0.05096578089353249,
0.04474822773147835,
0.052166409619514104,
0.05250743555325631,
0.050167264679589214,
0.041901011312064584,
0.043690541100094855,
0.045855417690835155,
0.03140697698379711,
0.06217189860737108,
0.0402189978059486,
0.02931368834100918,
0.040550053539418686,
0.0646709645172487,
0.044240884430318546,
0.04861660759238078,
0.05245203952508936,
0.03236207387709548,
0.05426318889765059,
0.03769298656434223,
0.05061072182870532,

0.04774708817427302,
0.04273103046223492,
0.04604213584664455,
0.040710855913825755,
0.02993924368168649,
0.03583963116172624,
0.048228229721371675,
0.05696833023447227,
0.0347939451586385,
0.040791718136312445,
0.033988297030290625,
0.040929017626272934,
0.055222304759231365,
0.04929666594019473,
0.04883753687644782,
0.040499399001204715,
0.04572380833965358,
0.04630291252062832,
0.045849805151670274,
0.03969922152751419,
0.04376749657182871,
0.03562874986913105,
0.02792912238264526,
0.02843862535566821,
0.029143030175108137,
0.05318965142111027,
0.02389416673171131,
0.061888171119270446,
0.04619865775398284,
0.02342210197925539,
0.05138393271411683,
0.021577329768808773,
0.036400507783610425,
0.023067949725817473,
0.06588183463227132,
0.05249230205999239,
0.03901107676808709,
0.025795807550492208,
0.0536904097910329,
0.05109359140001471,
0.05854084626236377,
0.0570144473841936,
0.04733559701839208,
0.034026918435031306,
0.03952107825067338,
0.025825363249839098,
0.047278556542713845,

0.04879561604041643,
0.03434332675548066,
0.03771955558145218,
0.018119692163387106,
0.05012229190021011,
0.02848671386784454,
0.03653436735414545,
0.02152435620700116,
0.03710782526700771,
0.018285340739810918,
0.042833964754342375,
0.009782027630648026,
0.006425030155330298,
0.026708647458876432,
0.021325803927619232,
0.016629488665659914,
0.09821994032519103,
0.05416412806493261,
0.05997099293485347,
0.04106632497139927,
0.047226296076355354,
0.012940823346857287,
0.08060277860960853,
0.04339276846068912,
0.060851635211368896,
0.039630798052273875,
0.06713535282299934,
0.021099772057470677,
0.04827617060069565,
0.04709274763813246,
0.04954335082290543,
0.023610723214567135,
0.03649377485875362,
0.02840245886016207,
0.036562676597217174,
0.0368203523368286,
0.05596396176483355,
0.04931362115509663,
0.049349999400125105,
0.04437899520335902,
0.043235756886262244,
0.032109672620103634,
0.03139418656526165,
0.028952926777839158,
0.018555548240109687,
0.039009043523354,
0.012069319114477587,

```
 0.06078259018742006,
 …]
```

[29]:
```
weight_list
```

[29]:
```
[array([-0.2769033 , -1.75657219, -0.80155973]),
 array([-0.28155367, -1.74727146, -0.75505606]),
 array([-0.26583844, -1.48011245, -0.47218182]),
 array([-0.25941394, -1.37089603, -0.51715329]),
 array([-0.25515099, -1.31121478, -0.55125686]),
 array([-0.26545891, -1.19782771, -0.37602229]),
 array([-0.26391514, -1.2070903 , -0.35286581]),
 array([-0.26915938, -1.1441595 , -0.35286581]),
 array([-0.25608742, -0.89579235, -0.11757062]),
 array([-0.25271155, -0.87216127, -0.10744302]),
 array([-0.2546227 , -0.88362816, -0.07113121]),
 array([-0.25237779, -0.88587307, -0.04419224]),
 array([-0.25539154, -0.84970806, -0.0652885 ]),
 array([-0.25299683, -0.84970806, -0.03894666]),
 array([-0.24686501, -0.74546717, -0.04507847]),
 array([-0.24723807, -0.74322879, -0.04806297]),
 array([-0.24398448, -0.69767857, -0.07409167]),
 array([-0.25400416, -0.52734406,  0.12630187]),
 array([-0.24718773, -0.4114648 ,  0.18764971]),
 array([-0.24216061, -0.36622067,  0.26305659]),
 array([-0.24248974, -0.36523327,  0.26206919]),
 array([-0.24954775, -0.23113111,  0.3538233 ]),
 array([-0.25295725, -0.18680758,  0.3606423 ]),
 array([-0.25225052, -0.17762006,  0.3472144 ]),
 array([-0.25277087, -0.18126253,  0.35606039]),
 array([-0.25482527, -0.16071858,  0.3519516 ]),
 array([-0.25198281, -0.14082138,  0.37753373]),
 array([-0.25394513, -0.14278369,  0.41285545]),
 array([-0.25259002, -0.14549391,  0.43318209]),
 array([-0.25405117, -0.12211544,  0.40542015]),
 array([-0.2555311 , -0.10731619,  0.3965406 ]),
 array([-0.25242052, -0.07621038,  0.41831467]),
 array([-0.2585537 ,  0.04032009,  0.52871196]),
 array([-0.25601708,  0.06568625,  0.54393166]),
 array([-0.25233314,  0.12831321,  0.56235135]),
 array([-0.25499032,  0.14957063,  0.60220902]),
 array([-0.25705714,  0.18884019,  0.57327356]),
 array([-0.25668972,  0.18994244,  0.57253873]),
 array([-0.25532435,  0.1981347 ,  0.57800023]),
 array([-0.25458123,  0.19664846,  0.58691765]),
 array([-0.25435139,  0.19595895,  0.58875634]),
 array([-0.25565079,  0.21285112,  0.57706176]),
```

```
array([-0.25453052,  0.23189573,  0.55577661]),
array([-0.25417237,  0.23261202,  0.55577661]),
array([-0.25434785,  0.23419137,  0.55349533]),
array([-0.25562404,  0.23929611,  0.56370481]),
array([-0.2551635 ,  0.24344099,  0.55863885]),
array([-0.25504951,  0.2443529 ,  0.557157  ]),
array([-0.25558755,  0.24865726,  0.55339068]),
array([-0.25370484,  0.26936713,  0.55715611]),
array([-0.25678478,  0.32788597,  0.56947587]),
array([-0.25718067,  0.32590651,  0.57462246]),
array([-0.25562531,  0.33834942,  0.58239928]),
array([-0.25446022,  0.34300976,  0.59055489]),
array([-0.25204416,  0.38649879,  0.59055489]),
array([-0.25367048,  0.41089354,  0.58730225]),
array([-0.25428006,  0.41089354,  0.59217891]),
array([-0.25675781,  0.45549301,  0.60456765]),
array([-0.25921903,  0.4801052 ,  0.64886959]),
array([-0.25652036,  0.52328396,  0.686651  ]),
array([-0.25574166,  0.52717745,  0.6897658 ]),
array([-0.25573498,  0.52711736,  0.6898526 ]),
array([-0.25592582,  0.52635399,  0.69176101]),
array([-0.25361017,  0.57266711,  0.70565495]),
array([-0.25510529,  0.58014272,  0.73406227]),
array([-0.25522221,  0.58037657,  0.73406227]),
array([-0.25680823,  0.59623672,  0.7578525 ]),
array([-0.25809723,  0.61299372,  0.7655865 ]),
array([-0.25641576,  0.6382157 ,  0.78240116]),
array([-0.25524923,  0.649881  ,  0.79289992]),
array([-0.25338351,  0.68719552,  0.81155718]),
array([-0.25335618,  0.68700425,  0.81185775]),
array([-0.25359284,  0.68937089,  0.80972778]),
array([-0.25238811,  0.70141828,  0.82659412]),
array([-0.25311004,  0.71369112,  0.8215406 ]),
array([-0.25338886,  0.71564291,  0.82098294]),
array([-0.25176271,  0.74328746,  0.84862749]),
array([-0.25152682,  0.74163618,  0.85287362]),
array([-0.25185329,  0.74457445,  0.8518942 ]),
array([-0.25240049,  0.74621605,  0.86010223]),
array([-0.25284001,  0.74973216,  0.86142077]),
array([-0.2538236 ,  0.76842051,  0.86437156]),
array([-0.25479641,  0.78593093,  0.87118117]),
array([-0.25454881,  0.78766414,  0.87019077]),
array([-0.25508298,  0.79567675,  0.86912242]),
array([-0.25378437,  0.82035023,  0.89249729]),
array([-0.2536694 ,  0.82000531,  0.89341707]),
array([-0.25305356,  0.82493204,  0.901423  ]),
array([-0.25366412,  0.83164814,  0.90997076]),
```

```
array([-0.2538037 ,  0.83318355,  0.90857493]),
array([-0.2528114 ,  0.85302959,  0.92345946]),
array([-0.2527553 ,  0.85347833,  0.92261806]),
array([-0.25271693,  0.85340159,  0.92273317]),
array([-0.25180184,  0.87170354,  0.93920493]),
array([-0.25144088,  0.8767569 ,  0.93776111]),
array([-0.25149042,  0.87670736,  0.93825652]),
array([-0.25210629,  0.88717717,  0.95057394]),
array([-0.2518414 ,  0.88876651,  0.95269306]),
array([-0.25149339,  0.89120262,  0.9579133 ]),
array([-0.2515295 ,  0.89152762,  0.95744386]),
array([-0.25151074,  0.89164018,  0.95708743]),
array([-0.25154885,  0.89209751,  0.9563252 ]),
array([-0.25209611,  0.9030428 ,  0.96672323]),
array([-0.25219762,  0.90324581,  0.96844881]),
array([-0.25211227,  0.90392862,  0.96682713]),
array([-0.25217894,  0.90479535,  0.96549369]),
array([-0.25223942,  0.90527923,  0.96513079]),
array([-0.25195873,  0.90724406,  0.96906046]),
array([-0.25204445,  0.9074155 ,  0.97043193]),
array([-0.2522939 ,  0.91065831,  0.9729264 ]),
array([-0.25206096,  0.9127547 ,  0.97385813]),
array([-0.25209943,  0.9127547 ,  0.97451213]),
array([-0.25214587,  0.9127547 ,  0.97539437]),
array([-0.25235404,  0.91525275,  0.97747608]),
array([-0.25209437,  0.91914772,  0.97487943]),
array([-0.2518565 ,  0.92057497,  0.9791612 ]),
array([-0.25195586,  0.9216679 ,  0.97856505]),
array([-0.2521177 ,  0.92377191,  0.97888874]),
array([-0.25178348,  0.9304564 ,  0.97721762]),
array([-0.25201921,  0.93399241,  0.9812251 ]),
array([-0.25177067,  0.93821767,  0.97948529]),
array([-0.25162962,  0.9394871 ,  0.97878005]),
array([-0.2514886 ,  0.94103826,  0.97736991]),
array([-0.25141746,  0.9411094 ,  0.97751219]),
array([-0.2513145 ,  0.94172717,  0.97710034]),
array([-0.25124424,  0.9415164 ,  0.97794344]),
array([-0.2511198 ,  0.94300968,  0.97607685]),
array([-0.25110603,  0.94302345,  0.97593909]),
array([-0.25086386,  0.94689818,  0.97642343]),
array([-0.2507062 ,  0.9498936 ,  0.97327036]),
array([-0.25079969,  0.95148294,  0.97280291]),
array([-0.25074762,  0.95163916,  0.97254254]),
array([-0.2508682 ,  0.95260383,  0.97459246]),
array([-0.2506474 ,  0.95657822,  0.97459246]),
array([-0.25052947,  0.95846523,  0.97294132]),
array([-0.25054065,  0.95840932,  0.97315379]),
```

```
array([-0.25043284,  0.95894835,  0.97347721]),
array([-0.25043644,  0.95899518,  0.97344119]),
array([-0.25025225,  0.96212644,  0.97325699]),
array([-0.25017057,  0.9626982 ,  0.97293028]),
array([-0.25003596,  0.96256359,  0.97521866]),
array([-0.24999903,  0.96208351,  0.97584646]),
array([-0.24991533,  0.96325538,  0.97459088]),
array([-0.24990029,  0.96336062,  0.97433529]),
array([-0.24991113,  0.96345816,  0.97432445]),
array([-0.24986036,  0.96335662,  0.97447676]),
array([-0.24976798,  0.96363375,  0.97484627]),
array([-0.24977288,  0.96368764,  0.97482178]),
array([-0.24974907,  0.96335423,  0.975179  ]),
array([-0.24964379,  0.96493344,  0.97402091]),
array([-0.24947358,  0.96646527,  0.97589314]),
array([-0.24930736,  0.96779509,  0.9780541 ]),
array([-0.24922386,  0.96737762,  0.97930652]),
array([-0.24907727,  0.97001627,  0.97857356]),
array([-0.24909602,  0.97001627,  0.97887354]),
array([-0.24885021,  0.97493246,  0.98329811]),
array([-0.24883385,  0.97489974,  0.98318359]),
array([-0.24878272,  0.97556437,  0.98221221]),
array([-0.24870986,  0.97585582,  0.98228507]),
array([-0.24855323,  0.97851858,  0.98338151]),
array([-0.2484916 ,  0.97864183,  0.98344313]),
array([-0.24846561,  0.97853788,  0.98339115]),
array([-0.24842476,  0.97886467,  0.98286011]),
array([-0.24838364,  0.97824777,  0.98351814]),
array([-0.24833804,  0.97792861,  0.98388289]),
array([-0.24834268,  0.9779518 ,  0.98392927]),
array([-0.2483519 ,  0.97810862,  0.98388315]),
array([-0.2483794 ,  0.97841113,  0.98413066]),
array([-0.24843513,  0.97952559,  0.984465  ]),
array([-0.24831908,  0.98184658,  0.98388475]),
array([-0.24817872,  0.98310977,  0.98641114]),
array([-0.24806732,  0.983444  ,  0.98852788]),
array([-0.2480735 ,  0.98351818,  0.98857115]),
array([-0.24806094,  0.98345538,  0.9884581 ]),
array([-0.24793137,  0.98513969,  0.9905311 ]),
array([-0.24791311,  0.98486576,  0.99065893]),
array([-0.24785766,  0.98469942,  0.99110252]),
array([-0.24771614,  0.98738837,  0.99379148]),
array([-0.24766407,  0.98744044,  0.99379148]),
array([-0.24766839,  0.9875095 ,  0.99383896]),
array([-0.24765623,  0.98736355,  0.99376598]),
array([-0.24759305,  0.98723719,  0.99471366]),
array([-0.24758368,  0.98712472,  0.99462931]),
```

```
array([-0.24755004,  0.98688924,  0.99469659]),
array([-0.24752469,  0.98668644,  0.99462053]),
array([-0.24748946,  0.98636942,  0.99493755]),
array([-0.24742197,  0.98663938,  0.99540997]),
array([-0.24742275,  0.98665027,  0.99542086]),
array([-0.24732437,  0.98851952,  0.99522409]),
array([-0.24727128,  0.98836027,  0.99580801]),
array([-0.2471762 ,  0.98988164,  0.99675887]),
array([-0.247076  ,  0.99178546,  0.99866269]),
array([-0.24701535,  0.99221   ,  0.99860204]),
array([-0.24697641,  0.99197638,  0.9984463 ]),
array([-0.24692635,  0.99202645,  0.99829609]),
array([-0.24685292,  0.99327473,  0.99763524]),
array([-0.24679678,  0.99316245,  0.99875803]),
array([-0.24673225,  0.9938077 ,  0.99914518]),
array([-0.24668722,  0.99376267,  0.99833463]),
array([-0.24665035,  0.9932465 ,  0.99888766]),
array([-0.24661229,  0.99290393,  0.99903992]),
array([-0.24656888,  0.9928171 ,  0.99834533]),
array([-0.2465327 ,  0.99263623,  0.9976942 ]),
array([-0.24645918,  0.99359206,  0.99850297]),
array([-0.24639289,  0.99471896,  0.99744236]),
array([-0.24632589,  0.99538898,  0.99838039]),
array([-0.24627819,  0.99534129,  0.998285  ]),
array([-0.24624414,  0.99466027,  0.9986255 ]),
array([-0.24618473,  0.99519496,  0.99874432]),
array([-0.24614833,  0.9947945 ,  0.99841667]),
array([-0.24611747,  0.99436254,  0.99804642]),
array([-0.24608729,  0.99403057,  0.99753338]),
array([-0.24604919,  0.99380193,  0.99722852]),
array([-0.24599991,  0.99424545,  0.99624291]),
array([-0.24595117,  0.99483026,  0.99531697]),
array([-0.24589901,  0.994413  ,  0.99594286]),
array([-0.24582761,  0.99505567,  0.99701398]),
array([-0.24579236,  0.99498518,  0.9963091 ]),
array([-0.24577186,  0.99459557,  0.99604253]),
array([-0.24571125,  0.99483797,  0.99658794]),
array([-0.24565417,  0.99546593,  0.99630251]),
array([-0.24560476,  0.99467534,  0.99729074]),
array([-0.24552729,  0.99622461,  0.99829777]),
array([-0.24548556,  0.99609941,  0.99767177]),
array([-0.24545096,  0.99558039,  0.99746416]),
array([-0.24540287,  0.99538803,  0.99770461]),
array([-0.24536718,  0.99470998,  0.99802579]),
array([-0.2453049 ,  0.99520818,  0.9987731 ]),
array([-0.24523935,  0.99645368,  0.99831423]),
array([-0.24518543,  0.99720856,  0.99750543]),
```

```
array([-0.24511525,  0.99861218,  0.99890905]),
array([-0.24506514,  0.99861218,  0.99915962]),
array([-0.24502182,  0.99822229,  0.99833651]),
array([-0.24496308,  0.99927945,  0.99892382]),
array([-0.24491136,  0.9999519 ,  0.99913073]),
array([-0.24486194,  1.00039665,  0.99922956]),
array([-0.24481088,  1.00014133,  0.99979127]),
array([-0.24476213,  0.99950764,  0.99910884]),
array([-0.24471092,  0.99920036,  0.99992825]),
array([-0.24466147,  0.99929926,  1.00057108]),
array([-0.24461372,  0.99839193,  0.99995028]),
array([-0.24456764,  0.99797723,  1.00018067]),
array([-0.24452251,  0.99757107,  1.00036118]),
array([-0.244479  ,  0.99704897,  1.00010013]),
array([-0.24442677,  0.99731011,  0.99910781]),
array([-0.24437699,  0.99735989,  0.99920737]),
array([-0.24431845,  0.99823798,  0.9998513 ]),
array([-0.24426351,  0.99911715,  1.00067553]),
array([-0.24421522,  0.99892403,  1.00062724]),
array([-0.24416753,  0.99878094,  1.00081803]),
array([-0.24411968,  0.99878094,  1.00110514]),
array([-0.24406499,  0.99960131,  1.00055822]),
array([-0.24401635,  0.99935812,  1.00046095]),
array([-0.24396905,  0.99907435,  1.00083931]),
array([-0.24391392,  1.00001157,  0.99979184]),
array([-0.24386589,  1.0000596 ,  0.99892728]),
array([-0.24381307,  1.00016525,  0.99993096]),
array([-0.24376394,  0.99967404,  1.00002921]),
array([-0.24371414,  1.00047087,  0.9999794 ]),
array([-0.24366457,  1.00007427,  1.00092132]),
array([-0.24361822,  1.00002792,  1.0015239 ]),
array([-0.24356954,  1.00041735,  1.0015239 ]),
array([-0.24351539,  0.99965926,  1.00076581]),
array([-0.24346668,  1.00009759,  1.00096062]),
array([-0.24341472,  0.99957797,  1.00012922]),
array([-0.24336469,  1.00052851,  1.0006295 ]),
array([-0.24331339,  0.99986153,  1.00011644]),
array([-0.24326471,  1.00029964,  1.00060323]),
array([-0.24321876,  1.0008969 ,  1.00133833]),
array([-0.24317664,  1.00136023,  1.00205438]),
array([-0.24313592,  1.00217481,  1.00229875]),
array([-0.24308276,  1.00201536,  1.0019267 ]),
array([-0.24302693,  1.00112198,  1.00181503]),
array([-0.24297558,  1.00065982,  1.00171232]),
array([-0.24292037,  1.00032857,  1.00077377]),
array([-0.24286957,  1.00053175,  0.99996102]),
array([-0.24281977,  0.99988435,  0.99896501]),
```

```
array([-0.24277036,  0.99948904,  0.99921208]),
array([-0.24272585,  0.99868789,  0.99858896]),
array([-0.24267035,  0.99896539,  0.99969899]),
array([-0.24262269,  0.9985841 ,  1.00031858]),
array([-0.24257604,  0.99811762,  0.99961886]),
array([-0.24252399,  0.99863813,  0.9994627 ]),
array([-0.24247838,  0.99804522,  0.99973635]),
array([-0.24243363,  0.99759777,  0.99987059]),
array([-0.24238804,  0.99727865,  1.0006912 ]),
array([-0.24234286,  0.99714309,  1.00123343]),
array([-0.2422876 ,  0.99758517,  1.00073609]),
array([-0.24223595,  0.99789507,  1.00063279]),
array([-0.24218388,  0.99831161,  1.00052866]),
array([-0.2421358 ,  0.99797504,  0.9996151 ]),
array([-0.24209144,  0.99748711,  0.99983688]),
array([-0.24204545,  0.99721117,  1.00066471]),
array([-0.24200172,  0.99694877,  1.00110204]),
array([-0.24196254,  0.99624355,  1.0007886 ]),
array([-0.24191932,  0.99602746,  1.00117755]),
array([-0.24187891,  0.99582542,  1.00186448]),
array([-0.24183582,  0.99543753,  1.00160589]),
array([-0.24180124,  0.99498802,  1.00181335]),
array([-0.24176838,  0.99439662,  1.00158336]),
array([-0.24173465,  0.99382311,  1.00111107]),
array([-0.24170053,  0.99351606,  1.00158871]),
array([-0.24167744,  0.99310041,  1.00172726]),
array([-0.24163083,  0.99310041,  1.0019603 ]),
array([-0.24160027,  0.99261145,  1.00162414]),
array([-0.24157408,  0.99227103,  1.00185982]),
array([-0.24149683,  0.99366162,  1.00162805]),
array([-0.24142626,  0.99472017,  1.00092235]),
array([-0.24139644,  0.9943028 ,  1.00151859]),
array([-0.24133989,  0.99469872,  1.00146203]),
array([-0.24129882,  0.99432913,  1.00113351]),
array([-0.24126213,  0.99392553,  1.00098675]),
array([-0.24119713,  0.99470556,  1.00027173]),
array([-0.24114783,  0.99475486,  1.00027173]),
array([-0.24110083,  0.99466085,  0.99951971]),
array([-0.2410464 ,  0.99493297,  1.00000953]),
array([-0.24100225,  0.99475634,  0.99996537]),
array([-0.24095623,  0.99466432,  0.9993672 ]),
array([-0.24092475,  0.9942236 ,  0.99892649]),
array([-0.24088449,  0.9940223 ,  0.99852387]),
array([-0.24084559,  0.99371103,  0.99856278]),
array([-0.24080047,  0.99339524,  0.99946505]),
array([-0.24076659,  0.99305643,  0.99912624]),
array([-0.24071357,  0.99321549,  0.99933832]),
```

```
array([-0.24067076,  0.99300142,  0.99980927]),
array([-0.24064293,  0.99258404,  1.00031012]),
array([-0.24061155,  0.99223887,  1.00052978]),
array([-0.24058693,  0.99186947,  1.00057903]),
array([-0.24051303,  0.99297795,  0.99969225]),
array([-0.24048215,  0.99260738,  0.99950696]),
array([-0.24041335,  0.99363931,  0.99847504]),
array([-0.24035071,  0.99464158,  0.99728484]),
array([-0.24030906,  0.99435   ,  0.99736815]),
array([-0.24025678,  0.99435   ,  0.99778633]),
array([-0.24022023,  0.99402102,  0.99767667]),
array([-0.24017298,  0.99421003,  0.99710965]),
array([-0.24010064,  0.99515042,  0.99826706]),
array([-0.24004992,  0.99545474,  0.99781057]),
array([-0.24001336,  0.99490628,  0.99799339]),
array([-0.2399811 ,  0.99429337,  0.99828372]),
array([-0.23994176,  0.99378198,  0.99899179]),
array([-0.23989313,  0.99383061,  0.99884591]),
array([-0.23984247,  0.99377994,  0.99970728]),
array([-0.23979567,  0.99377994,  0.99877133]),
array([-0.23973553,  0.99432119,  0.99901188]),
array([-0.23970329,  0.99386977,  0.99904412]),
array([-0.23966619,  0.99353589,  0.99908122]),
array([-0.23962546,  0.99329153,  0.99920341]),
array([-0.23959582,  0.99281722,  0.99979629]),
array([-0.23954055,  0.99309354,  1.00001735]),
array([-0.23948577,  0.99336746,  1.0005104 ]),
array([-0.23943859,  0.99332028,  1.0002273 ]),
array([-0.23941235,  0.99290044,  1.00038474]),
array([-0.23933886,  0.99414965,  0.99898857]),
array([-0.23930519,  0.99371196,  0.99915691]),
array([-0.23923933,  0.99483172,  0.99783955]),
array([-0.23917485,  0.99573443,  0.99816194]),
array([-0.23913626,  0.99523286,  0.99835486]),
array([-0.23909317,  0.99497429,  0.99848414]),
array([-0.2390513 ,  0.99489056,  0.99793987]),
array([-0.23902274,  0.99449073,  0.99759715]),
array([-0.23898816,  0.99407574,  0.99759715]),
array([-0.23894898,  0.99399738,  0.9970878 ]),
array([-0.23887256,  0.99544944,  0.99785205]),
array([-0.238833  ,  0.99473745,  0.99860359]),
array([-0.23876182,  0.9960187 ,  0.99974247]),
array([-0.23871399,  0.99597087,  1.00055561]),
array([-0.23868   ,  0.99546104,  1.00106544]),
array([-0.23861823,  0.99614051,  0.9998918 ]),
array([-0.23856995,  0.99618879,  0.99940901]),
array([-0.23853238,  0.9957004 ,  0.99933387]),
```

```
array([-0.23849737,  0.99514018,  0.99961398]),
array([-0.23843295,  0.99629979,  0.99896975]),
array([-0.23838268,  0.99660143,  0.9985173 ]),
array([-0.23833906,  0.99664504,  0.99781952]),
array([-0.23829723,  0.99622674,  0.99790318]),
array([-0.2382583 ,  0.9961878 ,  0.99716338]),
array([-0.23822144,  0.99607724,  0.99661056]),
array([-0.23815479,  0.99694377,  0.99747709]),
array([-0.23811226,  0.99651849,  0.99756214]),
array([-0.23806178,  0.99732608,  0.99670407]),
array([-0.23800025,  0.99794138,  0.99750396]),
array([-0.23794678,  0.99831572,  0.99782482]),
array([-0.23790504,  0.9983992 ,  0.99719876]),
array([-0.23784609,  0.99946023,  0.99778822]),
array([-0.23780284,  0.99950348,  0.99735569]),
array([-0.23775568,  0.99931483,  0.99735569]),
array([-0.23771901,  0.99894822,  0.99669579]),
array([-0.23768235,  0.99843486,  0.99625576]),
array([-0.2376506 ,  0.99827614,  0.99565264]),
array([-0.23761469,  0.9978811 ,  0.99532943]),
array([-0.23755962,  0.99837677,  0.99554972]),
array([-0.23750263,  0.99894665,  0.99594864]),
array([-0.23745424,  0.99822078,  0.9961906 ]),
array([-0.23739374,  0.99900724,  0.99685608]),
array([-0.23734116,  0.99969072,  0.99706638]),
array([-0.23729243,  1.00022681,  0.99711511]),
array([-0.23724542,  1.00069686,  0.99711511]),
array([-0.23719222,  1.00048406,  0.9975939 ]),
array([-0.23714786,  1.00035098,  0.99728338]),
array([-0.23709123,  0.99950149,  0.99813287]),
array([-0.23704705,  0.99905969,  0.99786778]),
array([-0.23700817,  0.99905969,  0.99709017]),
array([-0.23695784,  0.9989087 ,  0.99739215]),
array([-0.23692088,  0.9989087 ,  0.99665295]),
array([-0.23686529,  0.99946461,  0.99715327]),
array([-0.23680703,  0.9998142 ,  0.99820202]),
array([-0.23675485,  0.99950116,  0.99893244]),
array([-0.23670413,  0.99904465,  0.9999469 ]),
array([-0.23665725,  0.998904  ,  1.00046263]),
array([-0.23660978,  0.99857176,  0.99960831]),
array([-0.23656557,  0.99790857,  1.00027149]),
array([-0.2365108 ,  0.99878483,  0.99950477]),
array([-0.23646329,  0.99911745,  0.998792  ]),
array([-0.23641529,  0.99930945,  0.998792  ]),
array([-0.23636845,  0.99841961,  0.9992135 ]),
array([-0.2363185 ,  0.99876932,  0.99936338]),
array([-0.23627115,  0.99829583,  1.00031035]),
```

```
array([-0.23621838,  0.99903462,  0.9996771 ]),
array([-0.23616727,  0.99990344,  1.00013707]),
array([-0.23612038,  1.0001379 ,  1.00088733]),
array([-0.23606955,  1.00003624,  0.99987076]),
array([-0.23602265,  1.00092725,  0.99954249]),
array([-0.2359723 ,  1.00012158,  0.9996432 ]),
array([-0.23592438,  0.99940284,  0.99988278]),
array([-0.23587789,  0.99921685,  0.99946432]),
array([-0.23583266,  0.99880978,  0.99923817]),
array([-0.23578571,  0.99885673,  0.99909732]),
array([-0.23574256,  0.99833893,  0.99879528]),
array([-0.23569017,  0.99844372,  0.9997907 ]),
array([-0.23564044,  0.99889129,  0.99954205]),
array([-0.23559136,  0.99962749,  0.99880585]),
array([-0.23554803,  0.99936753,  0.99819926]),
array([-0.23550042,  0.99955799,  0.99819926]),
array([-0.2354487 ,  0.99867875,  0.9990785 ]),
array([-0.23540607,  0.99816721,  0.9987801 ]),
array([-0.23535117,  0.99904558,  0.99921928]),
array([-0.23530288,  0.99943186,  0.99912271]),
array([-0.23525553,  1.00028418,  0.9986492 ]),
array([-0.2352099 ,  1.00055796,  0.99846668]),
array([-0.23516788,  1.00059998,  0.9977943 ]),
array([-0.23511235,  0.9999336 ,  0.99868281]),
array([-0.23506538,  0.99974572,  0.99868281]),
array([-0.23501886,  0.99881527,  0.99877585]),
array([-0.23497462,  0.99921342,  0.99789106]),
array([-0.23492452,  0.9996643 ,  0.99809145]),
array([-0.23488137,  0.99940542,  0.99770313]),
array([-0.2348388 ,  0.99897971,  0.99740513]),
array([-0.23478527,  0.99865854,  0.99820805]),
array([-0.23473518,  0.99850826,  0.99875909]),
array([-0.23468505,  0.99921009,  0.99855856]),
array([-0.23463674,  0.99998308,  0.99836532]),
array([-0.23458721,  0.99968594,  0.9987615 ]),
array([-0.23454538,  0.99926764,  0.99800855]),
array([-0.23449053,  0.99970645,  0.99894103]),
array([-0.23444487,  1.00039136,  0.99848442]),
array([-0.23439338,  0.99936165,  0.99899927]),
array([-0.2343442 ,  0.99901734,  0.99978627]),
array([-0.23430039,  0.99831638,  0.99987389]),
array([-0.23425976,  0.99750378,  1.00068649]),
array([-0.2342052 ,  0.99854028,  1.00139568]),
array([-0.23415067,  0.99919466,  1.00057771]),
array([-0.23410465,  0.99919466,  1.00089986]),
array([-0.23405574,  0.99890118,  1.00006832]),
array([-0.23400498,  0.99986551,  1.00093114]),
```

```
array([-0.23395637,  1.00078921,  1.00059083]),
array([-0.23390544,  1.00002524,  0.99982686]),
array([-0.2338594 ,  1.00080787,  0.99895215]),
array([-0.23380841,  0.99994112,  0.99930905]),
array([-0.23375933,  1.00038282,  1.00009431]),
array([-0.23371149,  0.99980874,  0.99966375]),
array([-0.23366311,  1.00043772,  1.00048625]),
array([-0.23361472,  1.00043772,  0.99966369]),
array([-0.23356734,  1.00029555,  0.99994803]),
array([-0.23352089,  1.00038845,  0.99929773]),
array([-0.23347798,  1.00124651,  0.99861128]),
array([-0.23343564,  1.00179698,  0.99844191]),
array([-0.23337889,  1.00094569,  0.99929319]),
array([-0.23333532,  1.00107639,  0.99850904]),
array([-0.23328817,  1.00168932,  0.99902768]),
array([-0.2332374 ,  1.00133391,  0.99948463]),
array([-0.23318875,  1.00079871,  0.99904674]),
array([-0.23314047,  1.00128152,  0.99986753]),
array([-0.23309084,  1.00068603,  0.99971866]),
array([-0.23304768,  1.00141976,  0.99885545]),
array([-0.2330086 ,  1.00188874,  0.99815198]),
array([-0.23295647,  1.001993  ,  0.99903818]),
array([-0.23291135,  1.00239907,  0.9995345 ]),
array([-0.23286325,  1.00230287,  0.9998231 ]),
array([-0.23281335,  1.00195355,  0.99997281]),
array([-0.23277222,  1.00252931,  1.00021956]),
array([-0.23272927,  1.002787  ,  1.00077789]),
array([-0.23269258,  1.0033741 ,  1.00099806]),
array([-0.23264262,  1.00322422,  1.00064833]),
array([-0.23259441,  1.00317601,  1.00026265]),
array([-0.23254916,  1.00326651,  1.00026265]),
array([-0.23251566,  1.00393644,  1.00022915]),
array([-0.23246096,  1.00333468,  1.0007762 ]),
array([-0.23241398,  1.00347561,  1.0000246 ]),
array([-0.23235503,  1.0024144 ,  1.00049624]),
array([-0.23231154,  1.00271882,  1.00032229]),
array([-0.23225865,  1.00203129,  1.00085116]),
array([-0.23221684,  1.00249123,  1.00089297]),
array([-0.23216901,  1.0024434 ,  1.0006538 ]),
array([-0.23212769,  1.00273259,  1.00119087]),
array([-0.23209242,  1.00319112,  1.00179049]),
array([-0.23204776,  1.00341442,  1.00161185]),
array([-0.2320151 ,  1.00387164,  1.00203641]),
array([-0.23197925,  1.00426598,  1.00221566]),
array([-0.23192572,  1.00362359,  1.00259038]),
array([-0.23188223,  1.00379754,  1.00259038]),
array([-0.23184207,  1.00440001,  1.00218874]),
```

```
array([-0.23180801,  1.00477469,  1.00239311]),
array([-0.23175593,  1.00425387,  1.00280976]),
array([-0.23170783,  1.00386907,  1.00324266]),
array([-0.23166333,  1.00360207,  1.00368765]),
array([-0.23160682,  1.00309354,  1.00340513]),
array([-0.23156298,  1.00326892,  1.00340513]),
array([-0.23152955,  1.00333578,  1.0039734 ]),
array([-0.23149809,  1.0033987 ,  1.00450826]),
array([-0.23143822,  1.00268031,  1.00414906]),
array([-0.23137534,  1.00268031,  1.00289138]),
array([-0.23133506,  1.0024789 ,  1.0034956 ]),
array([-0.23129555,  1.00279493,  1.00365362]),
array([-0.23123614,  1.00279493,  1.0025842 ]),
array([-0.23117659,  1.00225901,  1.00163144]),
array([-0.23112561,  1.00205509,  1.00117262]),
array([-0.23107569,  1.00160577,  1.00117262]),
array([-0.23103482,  1.00213705,  1.00137696]),
array([-0.23098309,  1.00110253,  1.00194595]),
array([-0.2309434 ,  1.00122162,  1.00254141]),
array([-0.23088588,  1.00058894,  1.00156362]),
array([-0.23084002,  1.00095582,  1.00147189]),
array([-0.23079297,  1.00001487,  1.00194237]),
array([-0.23074334,  1.00041192,  1.00149569]),
array([-0.23069772,  1.00086816,  1.00145007]),
array([-0.23064561,  1.00055549,  1.00056417]),
array([-0.23059633,  1.00011203,  0.99962799]),
array([-0.23055107,  1.00033835,  0.99917535]),
array([-0.23050803,  1.0002953 ,  0.99835751]),
array([-0.23045901,  0.99951102,  0.99865162]),
array([-0.23041405,  1.00023038,  0.99820202]),
array([-0.23037236,  0.99998027,  0.99766012]),
array([-0.2303178 ,  1.00096252,  0.99864237]),
array([-0.23027411,  1.00070038,  0.99807441]),
array([-0.2302263 ,  1.00036574,  0.99817002]),
array([-0.23017696,  1.00135255,  0.99881145]),
array([-0.23013191,  1.0013976 ,  0.99867629]),
array([-0.23007953,  1.00045473,  0.9989382 ]),
array([-0.23003615,  1.00080175,  0.9985478 ]),
array([-0.22999096,  1.00021431,  0.99809593]),
array([-0.22995089,  1.00041467,  0.99749486]),
array([-0.22990331,  1.00093809,  0.99773277]),
array([-0.22986078,  1.00161849,  0.99769025]),
array([-0.22981297,  1.00209666,  0.99821623]),
array([-0.22977663,  1.00271438,  0.99796187]),
array([-0.22973177,  1.00226576,  0.99724408]),
array([-0.22969021,  1.00180859,  0.99653754]),
array([-0.2296518 ,  1.00246146,  0.99646074]),
```

```
array([-0.22961609,  1.00217576,  0.9957465 ]),
array([-0.22956225,  1.00309101,  0.99671559]),
array([-0.2295272 ,  1.00337145,  0.9964001 ]),
array([-0.22945989,  1.00202523,  0.99714052]),
array([-0.22940975,  1.00192496,  0.99744132]),
array([-0.22936349,  1.0018787 ,  0.99744132]),
array([-0.22931443,  1.00099573,  0.99709794]),
array([-0.22927545,  1.00177529,  0.99690305]),
array([-0.22923091,  1.00092885,  0.99632391]),
array([-0.22917069,  0.99984492,  0.99722719]),
array([-0.22913046,  0.99976447,  0.99682492]),
array([-0.22908217,  0.99904005,  0.9970664 ]),
array([-0.22903492,  0.99823686,  0.99744437]),
array([-0.22899185,  0.99759077,  0.99765973]),
array([-0.22894489,  0.99834205,  0.99700236]),
array([-0.22890635,  0.99791806,  0.99677109]),
array([-0.2288522 ,  0.99883863,  0.9968794 ]),
array([-0.22881312,  0.99844785,  0.99660585]),
array([-0.22876644,  0.99807446,  0.99683921]),
array([-0.22870851,  0.99847998,  0.99770818]),
array([-0.22867072,  0.99851777,  0.99702787]),
array([-0.22861964,  0.9987221 ,  0.99738544]),
array([-0.22858179,  0.99894917,  0.99670423]),
array([-0.22852633,  0.99978107,  0.99725883]),
array([-0.2284737 ,  1.00020215,  0.99789044]),
array([-0.22842063,  0.9999368 ,  0.99879262]),
array([-0.22837484,  1.00030308,  0.99879262]),
array([-0.22833115,  0.99947302,  0.99822469]),
array([-0.2282843 ,  1.00031638,  0.99813098]),
array([-0.22824215,  1.00090654,  0.9978359 ]),
array([-0.22818725,  0.99980852,  0.99854961]),
array([-0.22813958,  0.99980852,  0.99888329]),
array([-0.22809758,  0.99901058,  0.99833733]),
array([-0.22805821,  0.99889246,  0.99766799]),
array([-0.22801627,  0.99952153,  0.99703892]),
array([-0.22797575,  1.0001294 ,  0.99659315]),
array([-0.2279219 ,  0.99996785,  0.99723934]),
array([-0.22788231,  1.0004429 ,  0.99680387]),
array([-0.22783559,  1.00119036,  0.99699074]),
array([-0.22778091,  1.00047951,  0.99753754]),
array([-0.22773165,  0.99983906,  0.99778387]),
array([-0.22768461,  1.00007425,  0.99792498]),
array([-0.22764239,  1.00007425,  0.99758725]),
array([-0.22759792,  1.00034109,  0.9974983 ]),
array([-0.22754726,  1.00135429,  0.99815687]),
array([-0.22749815,  1.00056859,  0.99805866]),
array([-0.22744701,  1.00082431,  0.99887696]),
```

```
array([-0.22740562,  1.00078292,  0.99809061]),
array([-0.22736245,  1.00099878,  0.99791793]),
array([-0.22731485,  1.00009433,  0.99772752]),
array([-0.22726922,  0.99968364,  0.99772752]),
array([-0.22721738,  0.99895788,  0.99855696]),
array([-0.22717593,  0.99908223,  0.99789374]),
array([-0.22713813,  0.99915783,  0.99717557]),
array([-0.22708897,  0.99925614,  0.99747049]),
array([-0.22704778,  0.99962686,  0.99701738]),
array([-0.22701214,  0.9989497 ,  0.99651842]),
array([-0.22697072,  0.99857689,  0.99639415]),
array([-0.22694273,  0.998213  ,  0.9958623 ]),
array([-0.22689446,  0.998937  ,  0.9957175 ]),
array([-0.22684374,  0.99939347,  0.99592038]),
array([-0.22681472,  0.99919035,  0.99536905]),
array([-0.22677415,  0.99914978,  0.9951662 ]),
array([-0.22673881,  0.99907908,  0.99481273]),
array([-0.22669498,  0.99921056,  0.99472508]),
array([-0.22665454,  0.99876575,  0.99460377]),
array([-0.22659108,  0.99787731,  0.99587297]),
array([-0.22654529,  0.99810626,  0.99578139]),
array([-0.226515  ,  0.99798508,  0.99529667]),
array([-0.22648394,  0.9982646 ,  0.99470659]),
array([-0.22643795,  0.99835658,  0.99470659]),
array([-0.22640475,  0.99799135,  0.99444097]),
array([-0.22635394,  0.99855028,  0.99449178]),
array([-0.2262922 ,  0.99947635,  0.9951709 ]),
array([-0.22624856,  0.99960727,  0.99508361]),
array([-0.22621877,  1.00008393,  0.99457717]),
array([-0.22619084,  1.00002808,  0.99413038]),
array([-0.22615159,  1.0008132 ,  0.9939341 ]),
array([-0.22610702,  1.00165997,  0.99402324]),
array([-0.22604619,  1.00099087,  0.9946315 ]),
array([-0.22599197,  1.00028591,  0.99495687]),
array([-0.22592737,  1.00002755,  0.9961841 ]),
array([-0.22589438,  0.99989556,  0.99565615]),
array([-0.2258385 ,  1.00062194,  0.99632666]),
array([-0.22580717,  1.00059061,  0.99573133]),
array([-0.22576777,  1.00023606,  0.99541617]),
array([-0.22570915,  1.00058779,  0.9962955 ]),
array([-0.22567478,  1.00113771,  0.99588306]),
array([-0.22562166,  1.00023467,  0.99614866]),
array([-0.22558844,  0.99993567,  0.9956171 ]),
array([-0.22555562,  0.99987003,  0.99515762]),
array([-0.22552053,  0.99941388,  0.99480673]),
array([-0.22547554,  0.99936889,  0.99480673]),
array([-0.22543744,  0.99914025,  0.9945781 ]),
```

```
array([-0.22540748,  0.9989605 ,  0.99418863]),
array([-0.2253685 ,  0.99868766,  0.99403273]),
array([-0.22529825,  0.99903894,  0.99543783]),
array([-0.22524579,  0.99856685,  0.99596237]),
array([-0.2251841,  0.9993689,  0.9969495]),
array([-0.22514708,  0.99862847,  0.99661631]),
array([-0.22511302,  0.9985263 ,  0.99610548]),
array([-0.22506031,  0.99836818,  0.99668525]),
array([-0.22502515,  0.9977353 ,  0.99647429]),
array([-0.22497316,  0.9986193 ,  0.99642229]),
array([-0.22493763,  0.99869036,  0.99592488]),
array([-0.22489406,  0.99899529,  0.99575063]),
array([-0.22484417,  0.99979358,  0.99585042]),
array([-0.22480423,  1.00055252,  0.99557081]),
array([-0.22475251,  1.00153519,  0.99608801]),
array([-0.22470971,  1.0018348 ,  0.99608801]),
array([-0.22465765,  1.0028238 ,  0.99702496]),
array([-0.22459664,  1.00166448,  0.99757412]),
array([-0.22456253,  1.00217604,  0.99716487]),
array([-0.22451898,  1.00191471,  0.99685999]),
array([-0.22447545,  1.00278521,  0.99729524]),
array([-0.22443112,  1.00282955,  0.99729524]),
array([-0.22440669,  1.00326914,  0.99683123]),
array([-0.22434563,  1.00223101,  0.99731976]),
array([-0.22430264,  1.00184411,  0.99684688]),
array([-0.22426504,  1.00206973,  0.99654606]),
array([-0.2242213 ,  1.00123871,  0.99597747]),
array([-0.22419063,  1.00169875,  0.99557877]),
array([-0.2241498 ,  1.00239288,  0.99566043]),
array([-0.22410164,  1.0030189 ,  0.99619014]),
array([-0.22405244,  1.00272367,  0.99623935]),
array([-0.22401756,  1.00244465,  0.99557667]),
array([-0.22397976,  1.00202889,  0.99504752]),
array([-0.22392808,  1.0022873 ,  0.99551265]),
array([-0.22387538,  1.003236  ,  0.99646136]),
array([-0.22382635,  1.003236  ,  0.99675548]),
array([-0.22377705,  1.003236  ,  0.99710064]),
array([-0.22373086,  1.00305126,  0.99700827]),
array([-0.22367519,  1.00232754,  0.99728662]),
array([-0.22364046,  1.00295264,  0.99718244]),
array([-0.22359155,  1.0025124 ,  0.99708461]),
array([-0.22355439,  1.0022523 ,  0.99637862]),
array([-0.22349494,  1.00183613,  0.99732986]),
array([-0.22344066,  1.0011305 ,  0.99781837]),
array([-0.22340313,  1.00094286,  0.99710533]),
array([-0.22336004,  1.00051193,  0.99684678]),
array([-0.223316  ,  1.00051193,  0.99680274]),
```

```
array([-0.22327179,  1.00126336,  0.99689114]),
array([-0.2232202 ,  1.00162449,  0.99761341]),
array([-0.22318063,  1.00241605,  0.99773214]),
array([-0.22314678,  1.00282223,  0.99735981]),
array([-0.22311289,  1.00299166,  0.99685151]),
array([-0.22305117,  1.00194241,  0.99753044]),
array([-0.2229994 ,  1.00157998,  0.99799641]),
array([-0.22296057,  1.00150234,  0.99737529]),
array([-0.22292138,  1.00111037,  0.99674812]),
array([-0.22287256,  1.00203794,  0.99738278]),
array([-0.22282595,  1.00222437,  0.99770904]),
array([-0.22278146,  1.00195746,  0.99744213]),
array([-0.22273475,  1.00279838,  0.99828305]),
array([-0.22269778,  1.00338983,  0.99843091]),
array([-0.22266227,  1.00378047,  0.99825335]),
array([-0.22262838,  1.00408544,  0.99788061]),
array([-0.22259133,  1.00478948,  0.99858465]),
array([-0.22253725,  1.00397836,  0.99766538]),
array([-0.22248199,  1.0034257 ,  0.99799698]),
array([-0.22242299,  1.00224569,  0.99811498]),
array([-0.22238025,  1.0030577 ,  0.99888425]),
array([-0.22232984,  1.00220074,  0.99787606]),
array([-0.2222865 ,  1.00298086,  0.9985695 ]),
array([-0.22223964,  1.00274653,  0.99847577]),
array([-0.22219109,  1.0021155 ,  0.99799036]),
array([-0.22214101,  1.00111377,  0.99763976]),
array([-0.22209872,  1.00107148,  0.9974283 ]),
array([-0.22205147,  1.00078799,  0.99757005]),
array([-0.22200987,  1.0002472 ,  0.99715406]),
array([-0.22195849,  1.00009306,  0.99777062]),
array([-0.22190936,  1.00053522,  0.99831104]),
array([-0.22186509,  1.00057949,  0.99831104]),
array([-0.22181835,  1.00057949,  0.9986382 ]),
array([-0.22177002,  1.00028953,  0.99921813]),
array([-0.22172854,  1.00033101,  0.99847147]),
array([-0.22168046,  1.00028293,  0.99904843]),
array([-0.22163426,  1.00000574,  0.99941802]),
array([-0.22159189,  0.99915832,  0.99869771]),
array([-0.22155341,  0.99865803,  0.99815894]),
array([-0.22150377,  0.9987573 ,  0.99880418]),
array([-0.2214578 ,  0.9987573 ,  0.99912601]),
array([-0.22141791,  0.99807914,  0.99908612]),
array([-0.22137882,  0.99753191,  0.9991252 ]),
array([-0.22134034,  0.99691627,  0.99958693]),
array([-0.22129748,  0.99691627,  0.99885825]),
array([-0.22125772,  0.99647893,  0.99925584]),
array([-0.22120809,  0.99682632,  0.99940472]),
```

```
array([-0.22115802,  0.99742719,  0.99865363]),
array([-0.2211131 ,  0.99715771,  0.99928243]),
array([-0.22106219,  0.99766682,  0.99963881]),
array([-0.2210228 ,  0.99735166,  0.99904787]),
array([-0.22098255,  0.99731141,  0.99832342]),
array([-0.22093902,  0.99752907,  0.9978881 ]),
array([-0.22089765,  0.99736361,  0.99780537]),
array([-0.22086674,  0.99686897,  0.99746531]),
array([-0.22082012,  0.99663587,  0.99797811]),
array([-0.22077985,  0.99635403,  0.99805864]),
array([-0.22073667,  0.99618131,  0.99827454]),
array([-0.22070627,  0.99563411,  0.99827454]),
array([-0.2206679 ,  0.99536549,  0.99831292]),
array([-0.22063986,  0.99488883,  0.99828488]),
array([-0.22058858,  0.99524782,  0.99828488]),
array([-0.22053508,  0.99610371,  0.99737549]),
array([-0.22050147,  0.99610371,  0.99670331]),
array([-0.22045264,  0.99615254,  0.99699633]),
array([-0.22040554,  0.99615254,  0.9972318 ]),
array([-0.22038184,  0.99567837,  0.99701842]),
array([-0.22034178,  0.99507753,  0.99761926]),
array([-0.22031178,  0.99483751,  0.99722923]),
array([-0.22027124,  0.99435104,  0.99787785]),
array([-0.22022506,  0.99435104,  0.99810873]),
array([-0.22018742,  0.99416285,  0.99803346]),
array([-0.22013021,  0.99507821,  0.99723251]),
array([-0.22009492,  0.99472531,  0.99730309]),
array([-0.22006736,  0.99442215,  0.99705505]),
array([-0.22001427,  0.9952185 ,  0.99636488]),
array([-0.21998997,  0.99490259,  0.99612187]),
array([-0.21995903,  0.99462409,  0.99596715]),
array([-0.21991289,  0.99540845,  0.99504438]),
array([-0.21985073,  0.99596787,  0.99566597]),
array([-0.21980384,  0.99638992,  0.9954315 ]),
array([-0.21975656,  0.99624809,  0.99571516]),
array([-0.21971111,  0.99633899,  0.99571516]),
array([-0.21964492,  0.99726566,  0.99664183]),
array([-0.21961855,  0.99700195,  0.99616715]),
array([-0.21956813,  0.9973549 ,  0.99631841]),
array([-0.21951354,  0.99822829,  0.99648217]),
array([-0.21948619,  0.99798218,  0.9959626 ]),
array([-0.21943462,  0.99782747,  0.99652986]),
array([-0.21939021,  0.99844932,  0.99617452]),
array([-0.21933671,  0.99812836,  0.99697692]),
array([-0.21929751,  0.99789314,  0.99682011]),
array([-0.21926803,  0.99750992,  0.99640742]),
array([-0.21922998,  0.99789048,  0.99583658]),
```

```
array([-0.21917107,  0.9985973 ,  0.99654339]),
array([-0.21913563,  0.99902268,  0.99594077]),
array([-0.2190787 ,  0.99976266,  0.99668076]),
array([-0.2190245 ,  0.99873288,  0.99760215]),
array([-0.2189906 ,  0.99856335,  0.99699185]),
array([-0.21894879,  0.99818703,  0.99703366]),
array([-0.21890819,  0.99749696,  0.99723662]),
array([-0.21885198,  0.99839637,  0.99768632]),
array([-0.21879949,  0.99876382,  0.99842124]),
array([-0.218754  ,  0.99908226,  0.99842124]),
array([-0.21871059,  0.99929928,  0.99824761]),
array([-0.2186665 ,  0.99951975,  0.99820352]),
array([-0.21862556,  1.00001095,  0.99775325]),
array([-0.21858767,  0.99978357,  0.99726059]),
array([-0.2185478 ,  0.99926529,  0.99702139]),
array([-0.21851628,  0.99895012,  0.99645408]),
array([-0.21848209,  0.99901851,  0.99597536]),
array([-0.21843549,  0.99817967,  0.99634818]),
array([-0.21840713,  0.99789606,  0.99589441]),
array([-0.21834812,  0.99866313,  0.99660247]),
array([-0.21831067,  0.9983635 ,  0.99637775]),
array([-0.2182799 ,  0.99824045,  0.99588556]),
array([-0.21824308,  0.99864557,  0.99540678]),
array([-0.21819625,  0.99808363,  0.99573457]),
array([-0.21815539,  0.99869649,  0.995326  ]),
array([-0.21809091,  0.99992154,  0.99642209]),
array([-0.21805285,  1.00034027,  0.99611756]),
array([-0.21800497,  0.99952642,  0.99630906]),
array([-0.21796385,  0.99973203,  0.99614457]),
array([-0.2179327 ,  0.99966973,  0.9956462 ]),
array([-0.21788255,  0.99936879,  0.99604745]),
array([-0.21783236,  0.99916805,  0.99649912]),
array([-0.21778667,  0.99994486,  0.99645343]),
array([-0.21775147,  1.00043766,  0.99603103]),
array([-0.21770069,  0.9999299 ,  0.99643724]),
array([-0.21766015,  0.99948401,  0.99627509]),
array([-0.21762246,  0.99925782,  0.99601121]),
array([-0.21758196,  0.99865032,  0.99597071]),
array([-0.21753467,  0.99888674,  0.99611256]),
array([-0.21750561,  0.99862517,  0.99564755]),
array([-0.21747558,  0.99829483,  0.99528718]),
array([-0.21742508,  0.99905227,  0.99538817]),
array([-0.21738021,  0.9982894 ,  0.99561254]),
array([-0.21733816,  0.99757459,  0.99582278]),
array([-0.21729898,  0.99824069,  0.99523505]),
array([-0.21725788,  0.9980763 ,  0.99519395]),
array([-0.21719558,  0.99894842,  0.99606608]),
```

```
array([-0.2171435 ,  0.99816713,  0.99684737]),
array([-0.21709332,  0.99736435,  0.99785085]),
array([-0.217055  ,  0.99671281,  0.99819578]),
array([-0.21700024,  0.99780792,  0.997922  ]),
array([-0.21695419,  0.99813029,  0.99787595]),
array([-0.21690089,  0.99887648,  0.99846224]),
array([-0.21686678,  0.99826255,  0.99788242]),
array([-0.21682738,  0.99790796,  0.99780362]),
array([-0.21679258,  0.99752505,  0.99749033]),
array([-0.21676013,  0.99690854,  0.99739299]),
array([-0.21671182,  0.99652204,  0.99831093]),
array([-0.21664848,  0.99778874,  0.99945095]),
array([-0.21661122,  0.99745337,  0.99874295]),
array([-0.21657907,  0.9968748 ,  0.99848581]),
array([-0.21652561,  0.99730251,  0.9993947 ]),
array([-0.21648034,  0.99761937,  0.99871572]),
array([-0.21643804,  0.99757707,  0.99863111]),
array([-0.21639509,  0.99762002,  0.99850225]),
array([-0.21634423,  0.99807774,  0.99906169]),
array([-0.21629712,  0.99850168,  0.9991559 ]),
array([-0.21625398,  0.99824278,  0.9995874 ]),
array([-0.21621475,  0.99796822,  0.99884216]),
array([-0.21617019,  0.99765625,  0.99964436]),
array([-0.21612955,  0.99741244,  0.99976627]),
array([-0.2160922 ,  0.99700163,  0.99961688]),
array([-0.21604169,  0.99750681,  1.00047569]),
array([-0.21599679,  0.9976415 ,  1.0003859 ]),
array([-0.21596042,  0.99716865,  1.00071325]),
array([-0.21593041,  0.9965985 ,  1.00122338]),
array([-0.21588157,  0.99679388,  1.00063726]),
array([-0.21582595,  0.99790624,  1.00080412]),
array([-0.21577383,  0.99874011,  1.00007448]),
array([-0.21572586,  0.99969953,  1.00079405]),
array([-0.21568193,  0.99899664,  1.00031081]),
array([-0.21564116,  0.9986297 ,  1.00067775]),
array([-0.21560458,  0.99808104,  1.00133614]),
array([-0.21555196,  0.99887033,  1.00059948]),
array([-0.21551457,  0.99812247,  1.0009734 ]),
array([-0.21546985,  0.99856964,  1.00146529]),
array([-0.21543286,  0.99827367,  1.00194623]),
array([-0.21538449,  0.99803184,  1.00107563]),
array([-0.21535202,  0.99751222,  1.00172515]),
array([-0.2153215 ,  0.9969325 ,  1.00199976]),
array([-0.21527348,  0.99678843,  1.00118337]),
array([-0.21522739,  0.99729549,  1.00196701]),
array([-0.21518055,  0.99724866,  1.00145186]),
array([-0.21513448,  0.99747905,  1.00140578]),
```

```
array([-0.21508446,  0.99847941,  1.00195598]),
array([-0.21504318,  0.99856196,  1.00220365]),
array([-0.21499894,  0.99922553,  1.00251331]),
array([-0.2149561 ,  0.99918268,  1.00251331]),
array([-0.21491806,  0.998498  ,  1.00266546]),
array([-0.21486574,  0.99891653,  1.00198534]),
array([-0.21482526,  0.99867364,  1.00210679]),
array([-0.21478943,  0.99827947,  1.00246512]),
array([-0.21474764,  0.99777803,  1.0022144 ]),
array([-0.21469271,  0.99843714,  1.00139052]),
array([-0.21465047,  0.99826816,  1.00130603]),
array([-0.21460504,  0.99876794,  1.0015332 ]),
array([-0.21455702,  0.99886396,  1.000813  ]),
array([-0.21451735,  0.9980706 ,  1.00049566]),
array([-0.21446835,  0.9987566 ,  1.00015266]),
array([-0.21442835,  0.99831654,  1.00035269]),
array([-0.2143795 ,  0.99904933,  0.9997176 ]),
array([-0.21433445,  0.99958987,  0.99962751]),
array([-0.21429225,  0.99963207,  0.99920547]),
array([-0.21424932,  0.999675  ,  0.99920547]),
array([-0.21420782,  0.99921851,  0.99903947]),
array([-0.21417  ,  0.9988403,  0.9983587]),
array([-0.21413093,  0.99837151,  0.9982415 ]),
array([-0.21408664,  0.99828294,  0.99850721]),
array([-0.21403514,  0.99900403,  0.99917679]),
array([-0.21399493,  0.99828026,  0.99941805]),
array([-0.21394884,  0.99901771,  0.99854233]),
array([-0.21391054,  0.99859645,  0.99823595]),
array([-0.2138682 ,  0.99906221,  0.99781254]),
array([-0.21381749,  0.99896079,  0.99877604]),
array([-0.21377271,  0.9991847 ,  0.99895517]),
array([-0.21372477,  0.99966408,  0.99977011]),
array([-0.21368258,  0.99974846,  0.99913726]),
array([-0.2136386 ,  1.00023224,  0.99931318]),
array([-0.21359468,  0.99988085,  0.99957673]),
array([-0.2135515 ,  0.99979449,  0.9998358 ]),
array([-0.21350887,  1.00022078,  0.99919637]),
array([-0.21346625,  0.99983722,  0.99906852]),
array([-0.21342618,  0.99935642,  0.99858771]),
array([-0.21338977,  0.9991015 ,  0.99789578]),
array([-0.21335418,  0.99903033,  0.99732642]),
array([-0.21331861,  0.99910147,  0.9968284 ]),
array([-0.21327214,  0.99910147,  0.99710722]),
array([-0.21323932,  0.99910147,  0.9965493 ]),
array([-0.21318764,  0.99951489,  0.99711775]),
array([-0.21314281,  0.99893203,  0.99738675]),
array([-0.21310636,  0.99966098,  0.99665781]),
```

```
array([-0.21305965,  1.00017479,  0.99689135]),
array([-0.21302823,  1.00017479,  0.99632578]),
array([-0.21298769,  1.00086399,  0.99624469]),
array([-0.21295519,  1.00135142,  0.99591975]),
array([-0.21290093,  1.00064597,  0.9964624 ]),
array([-0.21284736,  1.00166377,  0.9974802 ]),
array([-0.21280764,  1.00170349,  0.99728161]),
array([-0.21277378,  1.00170349,  0.99673981]),
array([-0.21272689,  1.00137522,  0.9968805 ]),
array([-0.21268107,  1.00155847,  0.99720118]),
array([-0.2126312 ,  1.00145871,  0.99779971]),
array([-0.21258092,  1.00080508,  0.99825222]),
array([-0.21253356,  1.00123131,  0.99910468]),
array([-0.21249463,  1.00185411,  0.99918253]),
array([-0.21245314,  1.00218606,  0.99968046]),
array([-0.21241706,  1.00279943,  1.00025775]),
array([-0.21236968,  1.00246776,  0.99935749]),
array([-0.21232431,  1.00233165,  0.99985654]),
array([-0.21228645,  1.00271024,  0.99993226]),
array([-0.21224719,  1.0029458 ,  1.0000893 ]),
array([-0.21221624,  1.00353388,  1.00061548]),
array([-0.21217054,  1.00325968,  1.00098107]),
array([-0.21213548,  1.00354017,  1.00136673]),
array([-0.2120959 ,  1.00350059,  1.00188129]),
array([-0.21206296,  1.00415922,  1.00148611]),
array([-0.21202424,  1.00427541,  1.00164102]),
array([-0.21199531,  1.00456472,  1.00207499]),
array([-0.21195357,  1.00448125,  1.0023254 ]),
array([-0.2119319 ,  1.00491475,  1.00245545]),
array([-0.21191408,  1.00527108,  1.00263361]),
array([-0.21188066,  1.00560529,  1.00253335]),
array([-0.21184811,  1.00612603,  1.00201262]),
array([-0.21181992,  1.00626699,  1.00257648]),
array([-0.21175102,  1.00523358,  1.00154306]),
array([-0.21170308,  1.00485   ,  1.00197458]),
array([-0.21165319,  1.00440103,  1.00212424]),
array([-0.21161262,  1.00448217,  1.00212424]),
array([-0.21156087,  1.00344709,  1.00315932]),
array([-0.21153619,  1.00369385,  1.00357882]),
array([-0.21147504,  1.00308238,  1.00260047]),
array([-0.21143337,  1.00312406,  1.00260047]),
array([-0.21138275,  1.002871  ,  1.00209435]),
array([-0.21134164,  1.00262431,  1.00254661]),
array([-0.21128429,  1.00147732,  1.00203046]),
array([-0.211252  ,  1.00212306,  1.00235333]),
array([-0.21120574,  1.00142906,  1.00258466]),
array([-0.21116024,  1.00074665,  1.00267565]),
```

```
array([-0.21111548,  1.0007914 ,  1.00245186]),
array([-0.21108133,  1.00147448,  1.00279339]),
array([-0.21104273,  1.00143588,  1.0030636 ]),
array([-0.21100738,  1.00122376,  1.00355855]),
array([-0.21096878,  1.00145534,  1.00367434]),
array([-0.21092742,  1.00219989,  1.00342616]),
array([-0.21087666,  1.00255517,  1.00256333]),
array([-0.21083142,  1.00214804,  1.00269904]),
array([-0.21078604,  1.00169422,  1.0027898 ]),
array([-0.21074307,  1.00130744,  1.0029617 ]),
array([-0.21069937,  1.00065192,  1.00313651]),
array([-0.21066065,  0.99999384,  1.00348491]),
array([-0.21062338,  1.00051562,  1.0037458 ]),
array([-0.21058741,  1.00004797,  1.00410554]),
array([-0.21055747,  0.99959879,  1.00455471]),
array([-0.21051769,  0.99979769,  1.00467405]),
array([-0.21049367,  0.99960555,  1.0051304 ]),
array([-0.21043862,  0.99861476,  1.00435979]),
…]
```

[30]:
```
print(weights)
print(er)
```

```
[-5.38857708e-10  1.00000000e+00  1.00000000e+00]
1.8418196179133608e-19
```

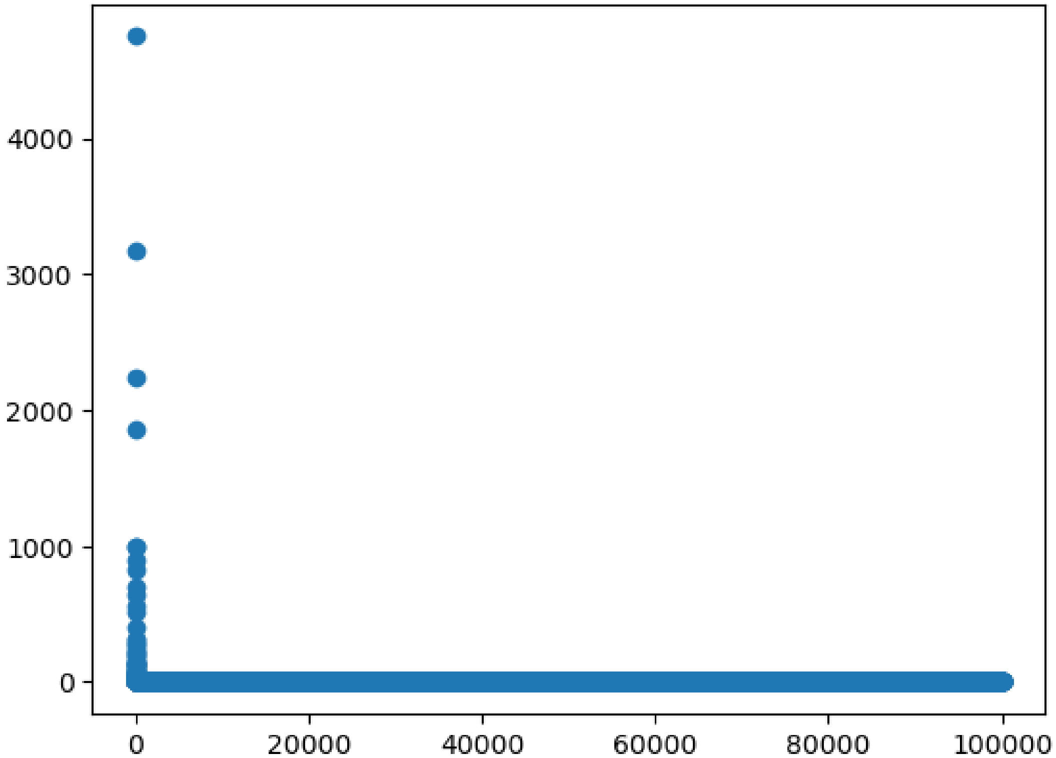
[31]:
```
print(len(errorlist))
```

```
100001
```

4 Plotting the plot of error (convergence rate)

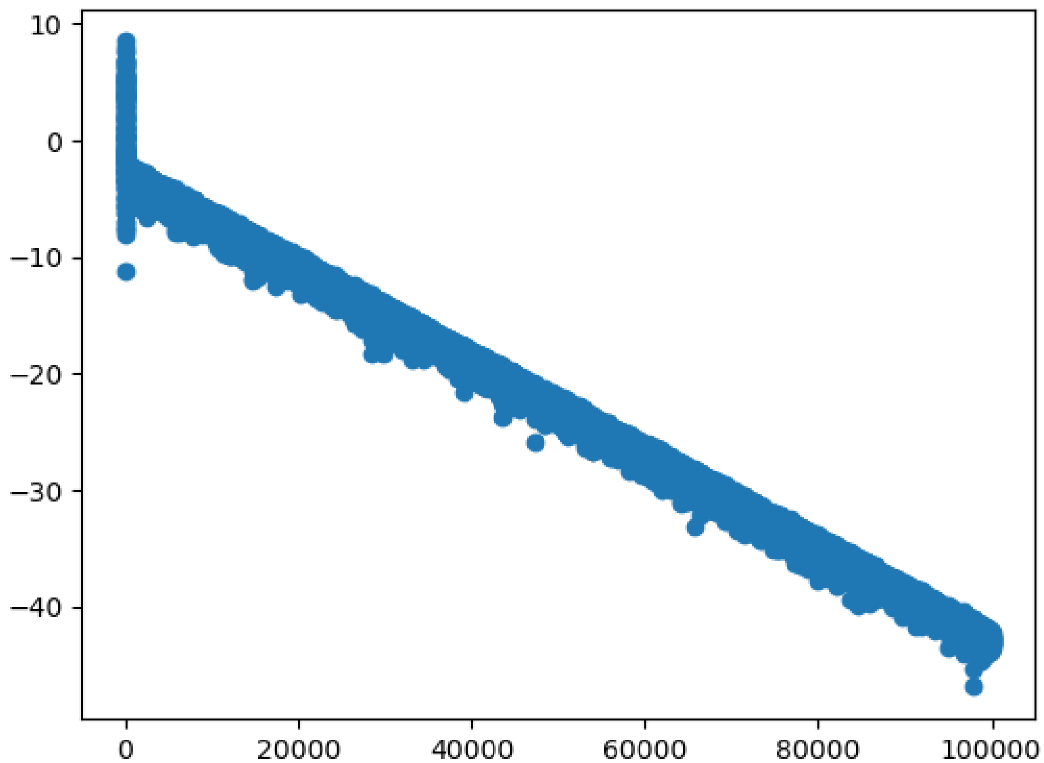
[32]:
```
plt.scatter(range(iterations+1),errorlist)
```

[32]:
```
<matplotlib.collections.PathCollection at 0x7baa8d534a30>
```

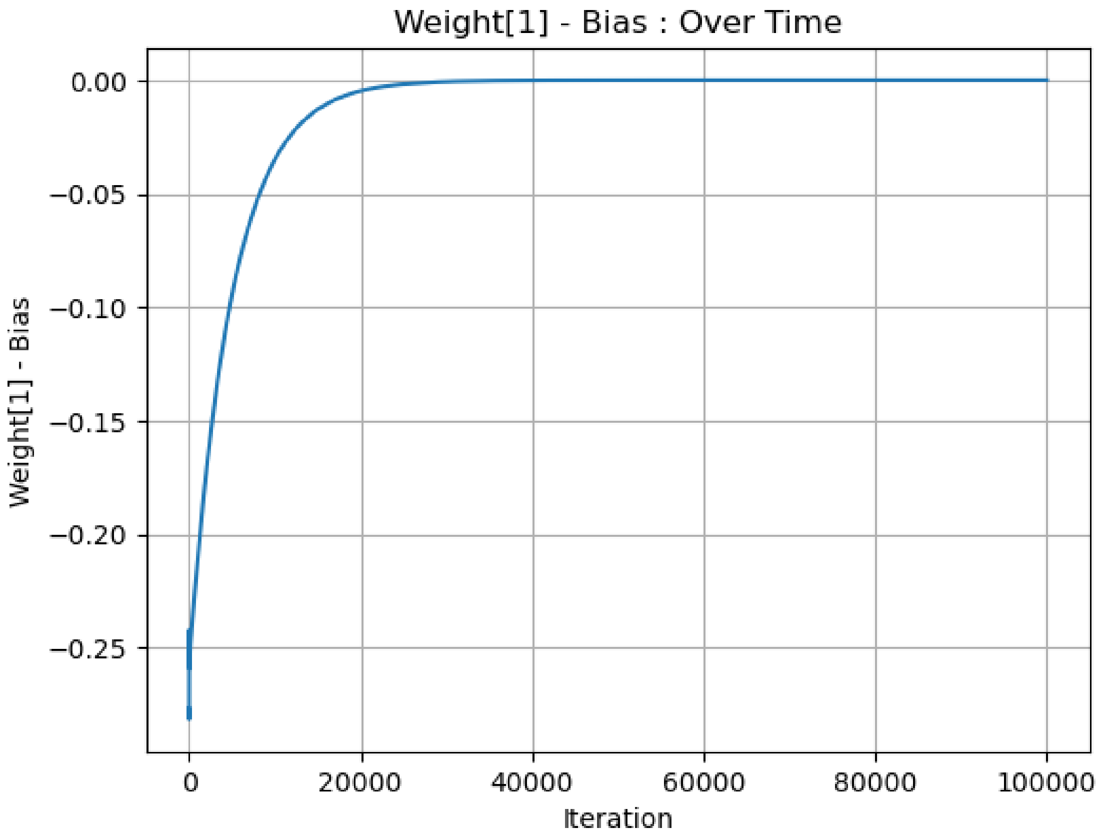


[33]:
```
plt.scatter(range(iterations+1),np.log(errorlist))
```

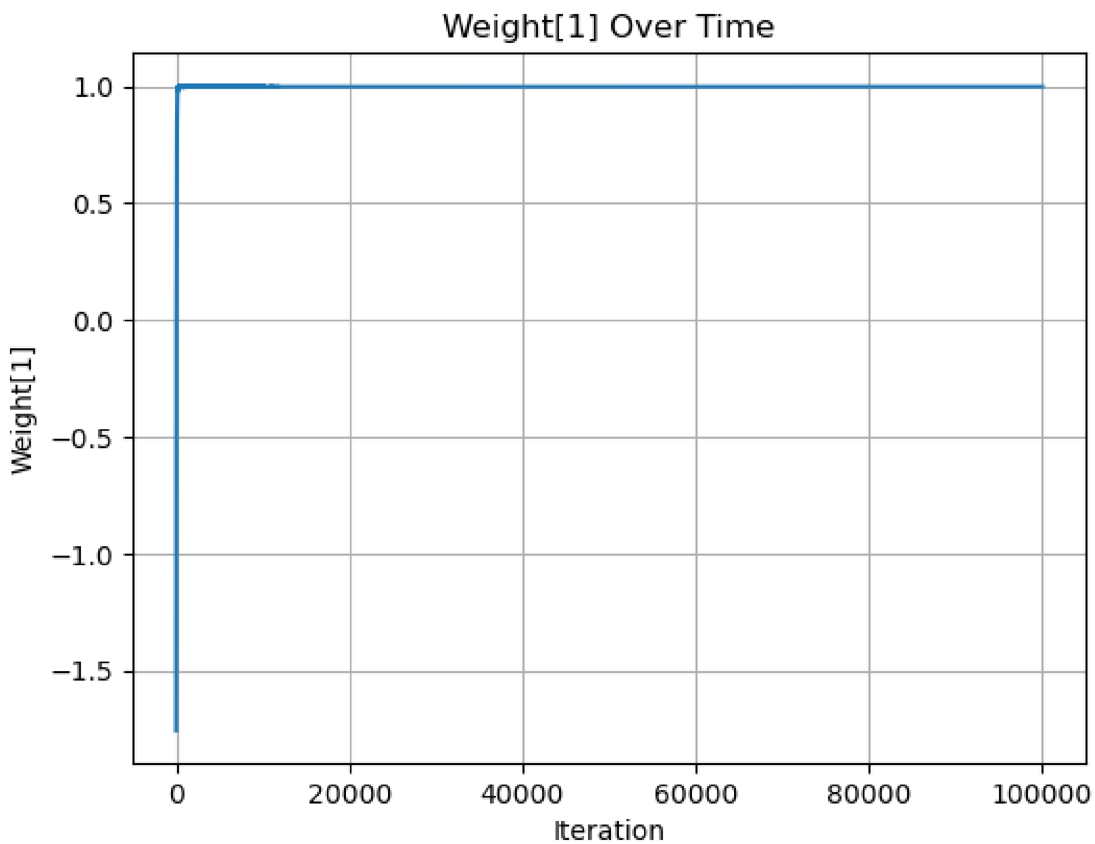
[33]: <matplotlib.collections.PathCollection at 0x7baa6f5fefb0>



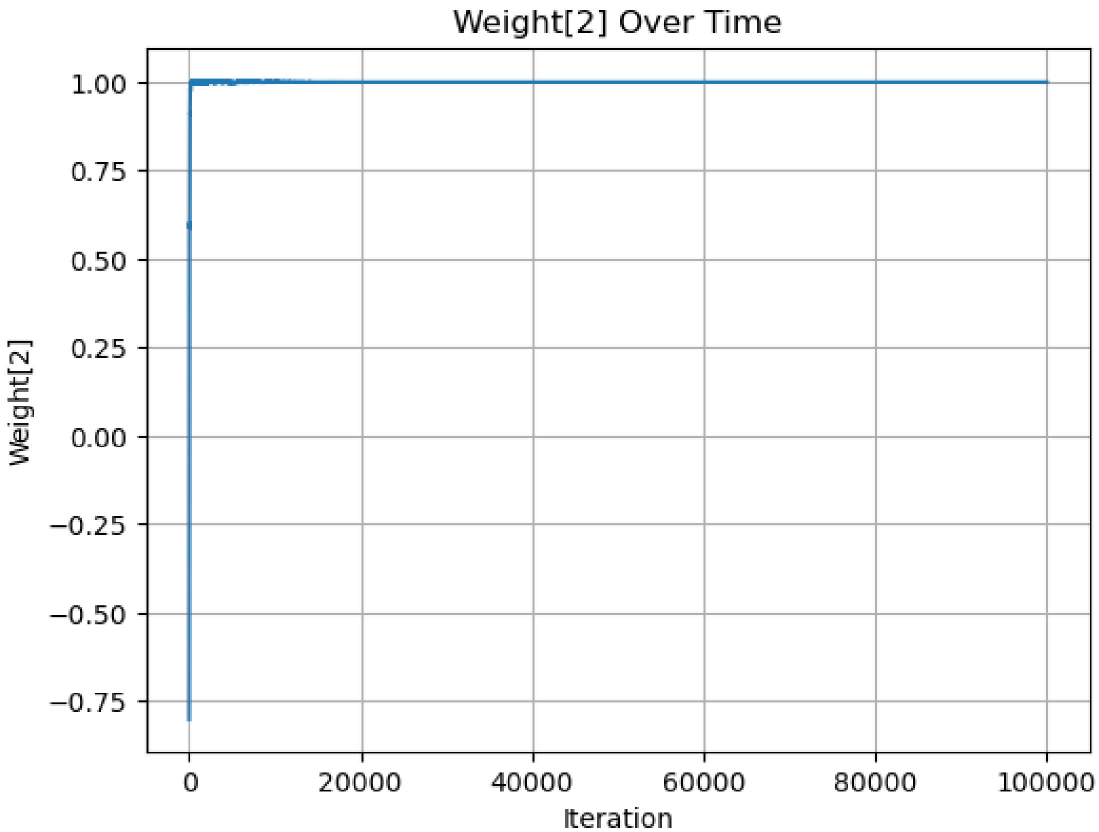
[42]:
```
weights_array = np.array(weight_list)  # shape: (iterations+1, num_weights)
plt.plot(range(len(weights_array)), weights_array[:, 0])  # first weight
plt.xlabel("Iteration")
plt.ylabel("Weight[1] - Bias")
plt.title("Weight[1] - Bias : Over Time")
plt.grid(True)
plt.show()
```



[38]:
```
weights_array = np.array(weight_list)  # shape: (iterations+1, num_weights)
plt.plot(range(len(weights_array)), weights_array[:, 1])  # second weight
plt.xlabel("Iteration")
plt.ylabel("Weight[1]")
plt.title("Weight[1] Over Time")
plt.grid(True)
plt.show()
```

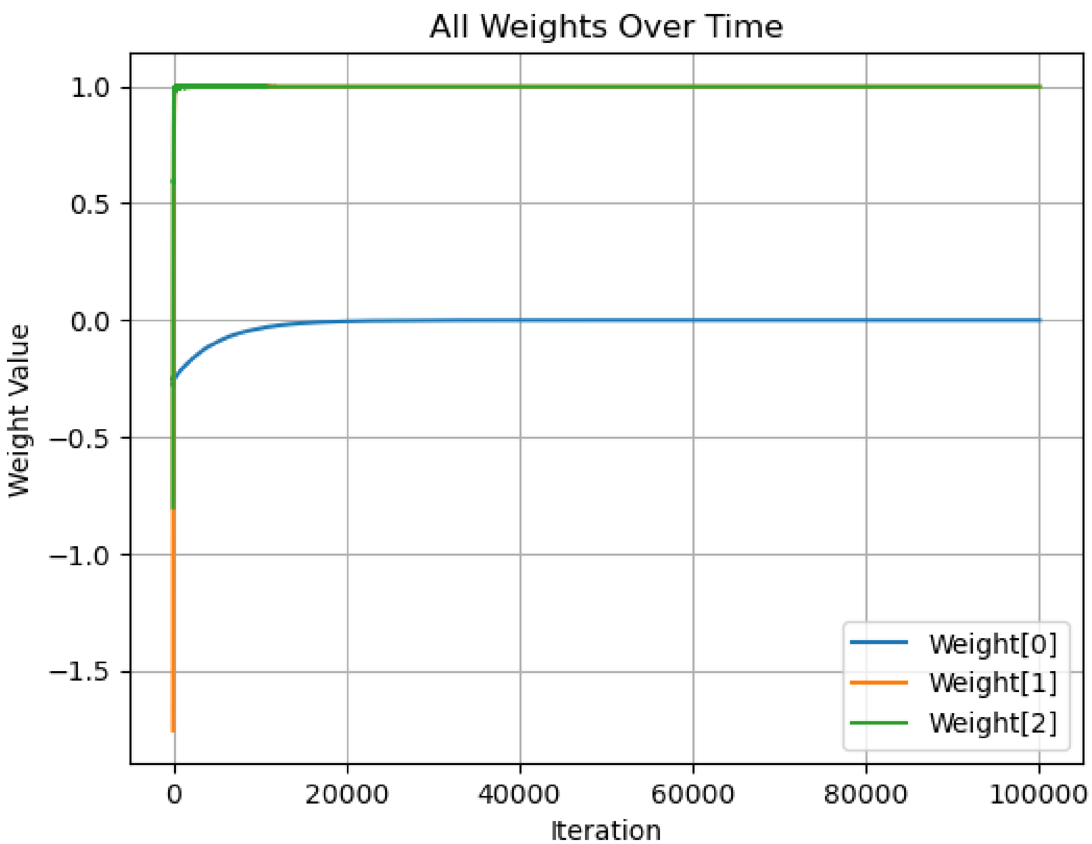


[39]:
```
weights_array = np.array(weight_list)  # shape: (iterations+1, num_weights)
plt.plot(range(len(weights_array)), weights_array[:, 2])  # third weight
plt.xlabel("Iteration")
plt.ylabel("Weight[2]")
plt.title("Weight[2] Over Time")
plt.grid(True)
plt.show()
```



[40]:
```
for i in range(weights_array.shape[1]):
    plt.plot(weights_array[:, i], label=f'Weight[{i}]')

plt.xlabel("Iteration")
plt.ylabel("Weight Value")
plt.title("All Weights Over Time")
plt.legend()
plt.grid(True)
plt.show()
```



5 Now what to do with those 300 testing dataset?

Those 300 datasets are quite untouched so far

So, the network does not know this dataset, and this network has not learned from this (testing) dataset.

Now, it is a good dataset (testing) to verify that the network has actually learned and that it has not just memorized the input data from the trainingg.

But that it can also apply its capabilities to a new testing dataset.

[43]:
```
testingOut
```

[43]:
```
array([-11, -15,  21,  -2, -16,   5,  18,  -4,  12, -11, -23,   6, -13,
       -18, -16,  -5,  11,  13,  22, -26, -27,  13,  10,  10, -11, -26,
        -1,   8, -18, -19, -19,  -1,  24,  -2,  38,  -7, -10,  13,   3,
       -32,  10,   3,   5,  -4, -19,  23,  -1, -10,  -5,  16, -30,  14,
        12,  11,  22, -12,  19,  28, -12, -10,  -2, -19, -25,  29,  16,
         5,  29, -10,   5,   7,   7,   7,   8,   1,   8,  30,  11,  -1,
        28,   6, -12,   3,   3,  -9,   1, -26, -28, -33,  20,  -7, -16,
```

```
 2, -30, -13,  14,   7,  29,  -1, -14,   7,   1, -11, -36,  24,
23,   7,   8, -19,   1,  -3, -31,  -5,  33, -28,  19,   8,  28,
-5, -23,   7, -31,   0, -28,   6,   3,  17,  35,  17,  13,  -1,
-5,  -3, -17, -22, -29,  31,  -6,  16,  -6, -13, -20, -21, -17,
-24,  -2, -24,   9,   6,  -6, -17,  -3,  31, -35,  -5, -16, -11,
29, -23,  31,   9,  15,   0,  25,  10, -20,  -3,   9,  29,  -1,
12,  -2,   8, -11, -18,  -6,  -8,  -9, -11,  14,   4,  -3,  11,
-9,   6,   5,  -4,  -3,  -1,  39,   1,  29,  15, -25,  28,  -1,
-18,  13,  23, -10, -10,  12,  -1,  23, -17,   1,  11, -23,   5,
19, -40, -30,  30,  -2,  15,  10,   2, -12,   9,  19,  13,  29,
16,   4, -21,  32, -18, -15,  -4, -14,   3, -17,  -6,   7,   5,
13,  15, -20, -19,   7,  14,   1, -18, -11,   2,  16,  14, -31,
29,  -9,  -3, -15,   2,   7, -27,  35, -19, -26, -20,  17,   1,
22, -25, -24,   7, -16,  15, -13,  -7,  -2,  14,  21,  -8,   7,
-11,  35, -21,   0,  -9,   5,   5,  20, -17,   8,  -8,  -8, -27,
 6,  30,   2,  -4,   2,   5,  -3, -17,   0,  -3,  11,  17,   6,
-13])
```

6 This is what we want to receive from our network

[45]:
```
calculateOut(testingIn,weights)

np.round(calculateOut(testingIn,weights))
```

[45]:
```
array([-11., -15.,  21.,  -2., -16.,   5.,  18.,  -4.,  12., -11., -23.,
         6., -13., -18., -16.,  -5.,  11.,  13.,  22., -26., -27.,  13.,
        10.,  10., -11., -26.,  -1.,   8., -18., -19., -19.,  -1.,  24.,
        -2.,  38.,  -7., -10.,  13.,   3., -32.,  10.,   3.,   5.,  -4.,
       -19.,  23.,  -1., -10.,  -5.,  16., -30.,  14.,  12.,  11.,  22.,
       -12.,  19.,  28., -12., -10.,  -2., -19., -25.,  29.,  16.,   5.,
        29., -10.,   5.,   7.,   7.,   7.,   8.,   1.,   8.,  30.,  11.,
        -1.,  28.,   6., -12.,   3.,   3.,  -9.,   1., -26., -28., -33.,
        20.,  -7., -16.,   2., -30., -13.,  14.,   7.,  29.,  -1., -14.,
         7.,   1., -11., -36.,  24.,  23.,   7.,   8., -19.,   1.,  -3.,
       -31.,  -5.,  33., -28.,  19.,   8.,  28.,  -5., -23.,   7., -31.,
        -0., -28.,   6.,   3.,  17.,  35.,  17.,  13.,  -1.,  -5.,  -3.,
       -17., -22., -29.,  31.,  -6.,  16.,  -6., -13., -20., -21., -17.,
       -24.,  -2., -24.,   9.,   6.,  -6., -17.,  -3.,  31., -35.,  -5.,
       -16., -11.,  29., -23.,  31.,   9.,  15.,  -0.,  25.,  10., -20.,
        -3.,   9.,  29.,  -1.,  12.,  -2.,   8., -11., -18.,  -6.,  -8.,
        -9., -11.,  14.,   4.,  -3.,  11.,  -9.,   6.,   5.,  -4.,  -3.,
        -1.,  39.,   1.,  29.,  15., -25.,  28.,  -1., -18.,  13.,  23.,
       -10., -10.,  12.,  -1.,  23., -17.,   1.,  11., -23.,   5.,  19.,
       -40., -30.,  30.,  -2.,  15.,  10.,   2., -12.,   9.,  19.,  13.,
        29.,  16.,   4., -21.,  32., -18., -15.,  -4., -14.,   3., -17.,
        -6.,   7.,   5.,  13.,  15., -20., -19.,   7.,  14.,   1., -18.,
```

```
 -11.,   2.,  16.,  14., -31.,  29.,  -9.,  -3., -15.,   2.,   7.,
 -27.,  35., -19., -26., -20.,  17.,   1.,  22., -25., -24.,   7.,
 -16.,  15., -13.,  -7.,  -2.,  14.,  21.,  -8.,   7., -11.,  35.,
 -21.,  -0.,  -9.,   5.,   5.,  20., -17.,   8.,  -8.,  -8., -27.,
   6.,  30.,   2.,  -4.,   2.,   5.,  -3., -17.,  -0.,  -3.,  11.,
  17.,   6., -13.])
```

7 Now check the accuracy

[47]:
```
accuracy(testingIn,testingOut,weights)

# previously on the top, the accuracy was 0.016666666666666672
```

[47]:
```
1.0
```

Now its 100% perfect of time .

It is able to predict all 300 out of 300 perfectly.

These dataset(testing) was not seen by the network but now did task ; very correctly.

8 Now you can check (or perform addition with the neural network)

[48]:
```
calculateOut([50, 90,90 ],weights)
```

[48]:
```
179.99999997372225
```

[52]:
```
np.round(calculateOut([50, 170,90 ],weights))
```

[52]:
```
260.0
```