import time  
import math  
import matplotlib.pyplot as plt  
import numpy as np  
import numpy.polynomial  
  
  
#def f(x):  
# return 1.76\*x + 2.5  
def f(x):  
 return math.sin(math.pi\*x - 4)  
  
# polynomial order  
K = 300  
  
# N = number of points  
N = 100  
  
a = -3  
b = 3  
step = (b-a)/N  
X = np.linspace(a, b, N)  
Y = np.zeros(N)  
print(X)  
  
#A will be the container i keep the values for the coefficient matrix  
A = np.zeros(2\*K+1)  
  
the\_random\_noise = np.random.random(N)  
  
for n in range(0,N):  
 X[n] = X[n] + ((-1)\*\*n)\*0.1\*step\*the\_random\_noise[n]  
  
#fills in the Y values with the function evaluated @ each x value  
for n in range(0, N):  
 Y[n] = f(X[n]) + ((-1)\*\*n)\*0.5\*the\_random\_noise[n]  
  
print("xi's that we need to sum")  
print(X)  
  
#this loop fills A with the appropriate sums  
for k in range(0,2\*K+1):  
 A[k] = A[k] + sum(i\*\*k for i in X)  
  
#this double loop fills the list 'a' with the values from A in the appropriate order  
a = []  
for i in range(0, K+1):  
 a.append(A[i])  
 for i in range(i+1, i+K+1):  
 a.append(A[i])  
  
#transform the list 'a' into an array  
a = np.asarray(a)  
  
#resize the array to the appropriate size for matrix multiplication  
a = np.resize(a, (K+1, K+1))  
  
#this loop fills the array B with the appropriate sums  
B = np.zeros(K+1)  
for i in range(0,K+1):  
 B[i] = B[i] + sum(X\*\*i \* Y)  
print("Array for B")  
print(B)  
  
  
  
tic = time.time()  
transformation\_coefs = np.linalg.solve(a, B)  
toc = time.time()  
print("Coefficient matrix")  
print(transformation\_coefs)  
print("Time to process")  
print(abs(tic - toc))  
  
  
the\_g\_values = np.zeros(N)  
error\_squared = 0  
  
#polynomial defined the by the coefficients a\_i  
poly = numpy.polynomial.Polynomial(transformation\_coefs)  
  
for n in range(0,N):  
 error\_squared = error\_squared + (poly(X[n]) - Y[n])\*\*2  
 the\_g\_values[n] = poly(X[n])  
  
print("Error")  
print(math.sqrt(error\_squared)/N)  
plt.plot(X, Y, 'o')  
plt.plot(X, the\_g\_values)  
plt.show()