Assignment 2

Mathematical Foundations For Data Science

Akhil Sudhakaran

2021MT12054

Q1

Write a code to minimize f (x, y)= (x – y)^2 +1– x using the Gradient descent method. Take a suitable initial value and do the iterations till the tolerance is 0.01. Please attach the code along with the final output.

NOTE : THE CODE AND METRICS USED FOR THIS QUESTION CAN BE FOUND IN THE GIT REPOSITORY HERE : https://git.io/JPypY

THE FOLLOWING IS THE GENERAL ITERATIVE FORMULAE FOR MINIMIZING VARIABLES

$$X_{k+1} = X_k - \gamma \nabla f(X_k)$$

- $X_k \rightarrow$ VARIABLE VALUES FOR PREVIOUS ITERATION
- $X_{k+1} \rightarrow$ VARIABLE VALUES FOR CURRENT ITERATION
- $\gamma \rightarrow$ LEARNING RATE / STEP SIZE
- $\nabla f(X_k) \rightarrow$ GRADIENT OF $f(x,y)$ W.R.T. X_k

THE ABOVE ITERATIVE FORMULAE IS IMPLIMENTED IN MATLAB R2021b. THE SCRIPT SNIPPET IS SHOWN BELOW:

```
format long
syms X1 X2;

% Function given in question:
f = (X1 - X2)^2 + 1 - X1

% Tolerance Criterias
max_iterations = 100
tol = 0.01;

% Preallocate for speed
x1 = zeros(1,max_iterations);
x2 = zeros(1,max_iterations);
obj = zeros(1,max_iterations);

% Initial Values for x1, x2:
x1(1) = 2;
x2(1) = -2;

i = 1;

% Gradient Computation:
df_dx1 = diff(f, X1);
df_dx2 = diff(f, X2);
G = [subs(df_dx1,[X1,X2], [x1(1),x2(1)]) subs(df_dx2, [X1,X2], [x1(1),x2(1
)])]; % Gradient

```

```
% Iterative process
while (true)
    X = [x1(i),x2(i)]';
    syms r;

    % X_k+1 = X_k - r * Gradient
    f_new = subs(f, [X1,X2], [x1(i) - r * G(1) ,x2(i) - r * G(2)]);
    dg_dh = diff(f_new,r);
    r = solve(dg_dh, r);
    x1(i+1) = X(1) - r * G(1);
    x2(i+1) = X(2) - r * G(2);
    obj(i+1) = eval(f_new);
    i = i+1;

    % Checking for tolerance criteria
    % If max iterations are reached or
    % if the difference between the previous values for x_1 and x_2
    % and the current values are less than the tolerance criteria
    if(i > max_iterations - 1 || ((abs(r * G(1)) <= tol) && (abs(r * G(2
))<=tol)))
        fprintf('Tolerance criteria reached...\n');
        break
    end

    % Update Gradient
    G = [subs(df_dx1,[X1,X2], [x1(i),x2(i)]) subs(df_dx2, [X1,X2], [x1(i),x2(
i)])];
end
% Result Table:
header = {'Iteration Number', 'X', 'Y', 'Function' };
T = table((1:i)', x1',x2',obj');
T.Properties.VariableNames = header;

% Contour Plot x1 and x2:
fcontour(f, 'Fill', 'On');
hold on;
plot(x1,x2,'*-r');

% Final Output:
disp(T);
fprintf('Initial Objective Value: %d\n',subs(f,[X1,X2], [x1(1),x2(1)]));
fprintf('Number of Iterations for Convergence: %d\n', i);
fprintf('Point of Minima (x, y): (%d,%d)\n', x1(i), x2(i));

```

FOR THE ABOVE CODE TO WORK PROPERLY MAKE SURE THAT THE "SYMBOLIC MATH TOOLBOX" ADD-ON IS ENABLED

THE CRITERIAS / TOLERANCES USED TO TERMINATE THE GRADIENT DESCENT IN THE CODE IS:

(1) THE ITERATIONS ARE RUN TILL THE DIFFERENCE BETWEEN THE PREVIOUS VALUES FOR X AND CURRENT VALUE OF X IS LESS THAN 0.01 $(X_{k+1} - X_k < 0.01)$

(2) IF THE ABOVE CRITERIA IS NOT MEANT, THEN THE LOOP TERMINATES AFTER A MAXIMUM ITERATION (SET IN max_iterations VARIABLE IN CODE)

LET US CONSIDER TWO INITIAL VALUES TO VALIDATE THE GRADIENT DESCENT ALGORITHM:

CASE 1:

INITIAL VALUES: [X, Y] = [2, -2]

REPLACING THE ABOVE INITIAL VALUES IN THE HIGHLIGHTED PART OF CODE THE VALUES OF X & Y AND THE RESPECTIVE OBJECTIVE FUNCTION VALUE FOR EVERY ITERATION IS SHOWN BELOW:

Iteration Number	X	Y	Function
1	2	-2	0
2	0.242222222222222	0.00888888888888889	0.812222222222222
3	30.3755555555556	26.3755555555556	-13.3755555555556
4	28.6177777777778	28.3844444444444	-27.5633333333333
5	58.7511111111111	54.7511111111111	-41.7511111111111
6	56.9933333333333	56.76	-55.9388888888889
7	87.1266666666667	83.1266666666667	-70.1266666666667
8	85.3688888888889	85.1355555555556	-84.3144444444444
9	115.502222222222	111.502222222222	-98.5022222222222
10	113.744444444444	113.511111111111	-112.69
11	143.877777777778	139.877777777778	-126.877777777778
12	142.12	141.886666666667	-141.065555555556
13	172.253333333333	168.253333333333	-155.253333333333
14	170.495555555556	170.262222222222	-169.441111111111
15	200.628888888889	196.628888888889	-183.628888888889
16	198.871111111111	198.637777777778	-197.816666666667
17	229.004444444444	225.004444444444	-212.004444444444
18	227.246666666667	227.013333333333	-226.192222222222
19	257.38	253.38	-240.38
20	255.622222222222	255.388888888889	-254.567777777778
21	285.755555555556	281.755555555556	-268.755555555556
22	283.997777777778	283.764444444444	-282.943333333333
23	314.131111111184	310.13111111118	-297.131111111146
24	312.373333333404	312.140000000071	-311.31888888896
25	342.50666666679	338.506666666787	-325.506666666763
26	340.748888889011	340.515555555677	-339.694444444566
27	370.882222222397	366.882222222394	-353.88222222237
28	369.124444444617	368.891111111284	-368.070000000173
29	399.257777778004	395.257777778	-382.257777777976
30	397.500000000224	397.266666666891	-396.44555555578

31	427.63333333361	423.633333333607	-410.633333333583
32	425.875555555831	425.642222222498	-424.821111111386
33	456.008888889217	452.008888889214	-439.00888888919
34	454.251111111438	454.017777778104	-453.196666666993
35	484.384444444824	480.38444444482	-467.384444444796
36	482.626666667044	482.393333333711	-481.5722222226
37	512.760000000431	508.760000000427	-495.760000000403
38	511.002222222651	510.768888889318	-509.947777778207
39	541.135555556037	537.135555556034	-524.13555555601
40	539.377777778258	539.144444444924	-538.323333333813
41	569.511111111644	565.511111111641	-552.511111111617
42	567.753333333864	567.520000000531	-566.69888888942
43	597.886666667251	593.886666667247	-580.886666667223
44	596.128888889471	595.895555556138	-595.074444445027
45	626.262222222857	622.262222222854	-609.26222222283
46	624.504444445078	624.271111111745	-623.450000000634
47	654.637777778464	650.637777778461	-637.637777778437
48	652.880000000685	652.646666667351	-651.82555555624
49	683.013333334071	679.013333334068	-666.013333334044
50	681.255555556291	681.022222222958	-680.201111111847
51	711.388888889678	707.388888889674	-694.38888888965
52	709.631111111898	709.397777778565	-708.576666667454
53	739.764444445284	735.764444445281	-722.764444445257
54	738.006666667505	737.773333334172	-736.952222223061
55	768.140000000891	764.140000000888	-751.140000000864
56	766.382222223112	766.148888889778	-765.327777778667
57	796.515555556498	792.515555556495	-779.515555556471
58	794.757777778718	794.524444445385	-793.703333334274
59	824.891111112105	820.891111112101	-807.891111112077
60	823.133333334325	822.900000000992	-822.078888889881
61	853.266666667711	849.266666667708	-836.266666667684
62	851.508888889932	851.275555556599	-850.454444445488
63	881.642222223318	877.642222223315	-864.642222223291
64	879.884444445539	879.651111112205	-878.830000001094
65	910.017777778925	906.017777778922	-893.017777778898
66	908.260000001145	908.026666667812	-907.205555556701
67	938.393333334532	934.393333334528	-921.393333334504
68	936.635555556752	936.402222223419	-935.581111112308
69	966.768888890138	962.768888890135	-949.768888890111
70	965.011111112359	964.777777779026	-963.956666667915
71	995.144444445745	991.144444445742	-978.144444445718
72	993.386666667966	993.153333334632	-992.332222223521
73	1023.52000000135	1019.52000000135	-1006.52000000132
74	1021.76222222357	1021.52888889024	-1020.70777777913
75	1051.89555555696	1047.89555555696	-1034.89555555693
76	1050.13777777918	1049.90444444585	-1049.08333333473
77	1080.27111111257	1076.27111111256	-1063.27111111254
78	1078.51333333479	1078.28000000145	-1077.45888889034
79	1108.64666666817	1104.64666666817	-1091.64666666814
80	1106.88888889039	1106.65555555706	-1105.83444444595
81	1137.02222222378	1133.02222222378	-1120.02222222375

82	1135.264444446	1135.03111111267	-1134.21000000156
83	1165.39777777939	1161.39777777938	-1148.39777777936
84	1163.64000000161	1163.40666666827	-1162.58555555716
85	1193.77333333499	1189.77333333499	-1176.77333333497
86	1192.01555555721	1191.78222222388	-1190.96111111277
87	1222.1488888906	1218.1488888906	-1205.14888889057
88	1220.39111111282	1220.15777777949	-1219.33666666838
89	1250.52444444621	1246.5244444462	-1233.52444444618
90	1248.76666666843	1248.53333333509	-1247.71222222398
91	1278.90000000181	1274.90000000181	-1261.90000000179
92	1277.14222222403	1276.9088888907	-1276.08777777959
93	1307.27555555742	1303.27555555742	-1290.27555555739
94	1305.51777777964	1305.28444444631	-1304.4633333352
95	1335.65111111303	1331.65111111302	-1318.651111113
96	1333.89333333525	1333.66000000191	-1332.8388888908
97	1364.02666666863	1360.02666666863	-1347.02666666861
98	1362.26888889085	1362.03555555752	-1361.21444444641
99	1392.40222222424	1388.40222222424	-1375.40222222421
100	1390.64444444646	1390.41111111313	-1389.59000000202

INFERENCES:

→ FROM THE ABOVE TABLE WE SEE THAT THE VALUES OF (X,Y) KEEPS UPDATING CONVERGING TOWARDS $X = Y$ AND ITERATION GOES ON NONSTOP.

→ WE SEE THAT THE CODE STOPPED EXECUTING BECAUSE OF THE MAX_ITERATION CONDITION

→ THE CODE DID NOT ENTER THE 0.01 TOLERANCE CRITERIA

→ WE CAN ALSO SEE THAT THE OBJECTIVE FUNCTION VALUE IS EVER DECREASING

→ THE MINIMUM VALUE REACHED AT ITERATION 100 = -1389.59

TO MAKE A MORE CONCRETE CONCLUSION LETS CONSIDER ANOTHER CASE

CASE 2:

INITIAL VALUES: $[x, y] = [-10, -15]$

max_iteration SET TO = 50

REPLACING THE ABOVE INITIAL VALUES IN THE HIGHLIGHTED PART OF CODE THE VALUES OF X & Y AND THE RESPECTIVE OBJECTIVE FUNCTION VALUE FOR EVERY ITERATION IS SHOWN BELOW:

Iteration Number	X	Y	Function
1	-10	-15	0
2	-12.2562326869806	-12.4930747922438	13.3123268698061
3	35.3753462603878	30.3753462603878	-9.37534626038781
4	33.1191135734072	32.882271468144	-32.0630193905817
5	80.7506925207756	75.7506925207756	-54.7506925207756
6	78.494459833795	78.2576177285319	-77.4383656509695
7	126.126038781163	121.126038781163	-100.126038781163
8	123.869806094183	123.63296398892	-122.813711911357
9	171.501385041551	166.501385041551	-145.501385041551
10	169.245152354571	169.008310249307	-168.189058171745
11	216.876731302025	211.876731302021	-190.876731301981
12	214.620498615042	214.383656509779	-213.564404432217
13	262.252077562558	257.25207756255	-236.252077562482
14	259.995844875574	259.75900277031	-258.939750692748
15	307.627423823089	302.627423823082	-281.627423823014
16	305.371191136105	305.134349030842	-304.31509695328
17	353.002770083621	348.002770083613	-327.002770083545
18	350.746537396637	350.509695291373	-349.690443213811
19	398.378116344152	393.378116344145	-372.378116344077
20	396.121883657168	395.885041551905	-395.065789474343
21	443.753462604684	438.753462604676	-417.753462604608
22	441.497229917699	441.260387812436	-440.441135734874
23	489.128808865215	484.128808865208	-463.12880886514
24	486.872576178231	486.635734072968	-485.816481995405
25	534.504155125747	529.504155125739	-508.504155125671
26	532.247922438762	532.011080333499	-531.191828255937
27	579.879501386678	574.87950138665	-553.879501386397
28	577.623268699684	577.38642659442	-576.567174516858
29	625.2548476476	620.254847647572	-599.254847647319
30	622.998614960605	622.761772855342	-621.942520777779
31	670.630193908521	665.630193908493	-644.63019390824
32	668.373961221526	668.137119116263	-667.317867038701
33	716.005540169442	711.005540169414	-690.005540169161
34	713.749307482448	713.512465377184	-712.693213299622
35	761.380886430364	756.380886430336	-735.380886430083
36	759.124653743369	758.887811638106	-758.068559560543
37	806.756232691285	801.756232691257	-780.756232691004
38	804.50000000429	804.263157899027	-803.443905821465
39	852.131578952206	847.131578952178	-826.131578951925
40	849.875346265212	849.638504159948	-848.819252082386
41	897.506925213128	892.5069252131	-871.506925212847
42	895.250692526133	895.01385042087	-894.194598343307
43	942.882271474049	937.882271474021	-916.882271473768
44	940.626038787054	940.389196681791	-939.569944604229
45	988.25761773497	983.257617734942	-962.257617734689
46	986.001385047976	985.764542942712	-984.94529086515
47	1033.63296399589	1028.63296399586	-1007.63296399561
48	1031.3767313089	1031.13988920363	-1030.32063712607
49	1079.00831025601	1074.00831025603	-1053.00831025614
50	1076.75207756904	1076.51523546378	-1075.69598338621

INFERENCES:

→ SIMILAR TO CASE 1, WE SEE THAT THE (X, Y) VALUES CHANGES TILL ALMOST $X = Y$, THEN X & Y KEEPS INCREASING INDEFINITELY

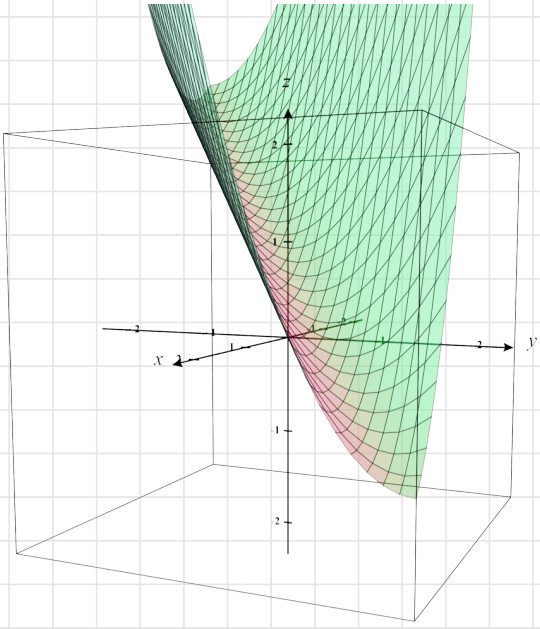
→ THE CODE EXITED BECAUSE OF MAX ITERATIONS BEING REACHED

→ SIMILAR TO CASE 1 THE OBJECTIVE FUNCTION VALUE IS EVER DECREASING

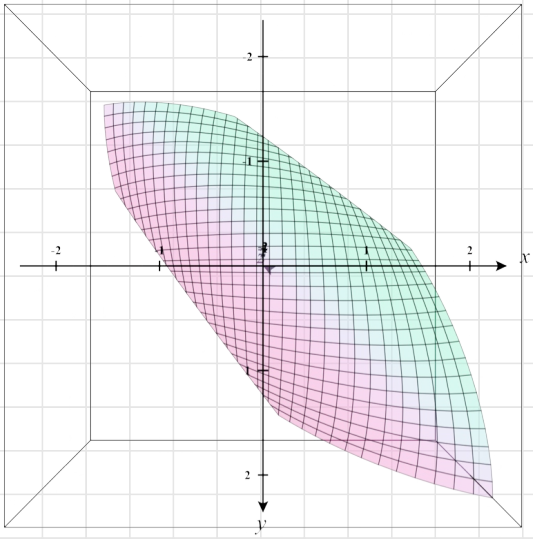
→ THE MINIMUM VALUE OBTAINED AT ITERATION 50 = -1075.695

FROM THE DATA OBTAINED IN THE 2 CASES IT SEEMS TO SHOW THAT THE GIVEN FUNCTION $f(x,y) = (x-y)^2 + 1 - x$ DOES NOT HAVE A GLOBAL MINIMUM.

THE ABOVE CONCLUSION CAN FURTHER BE SEEN IF WE SURFACE PLOT THE GIVEN FUNCTION $f(x,y) = Z = (x-y)^2 + 1 - x$:



SIDE VIEW



BOTTOM VIEW

FROM THE GRAPH WE CAN SEE THAT THE FUNCTION IS MINIMUM IN THE X-Y PLANE. MEANING, WHEN $X = Y$ THE FUNCTION VALUE IS MINIMUM AS SEEN BELOW:

$f(x,y) = Z = (x-y)^2 + 1 - x$

WHEN $x = y$

$\boxed{Z = 1 - x} \Rightarrow$ WHEN x & y KEEPS INCREASING, EQUALLY Z KEEPS DECREASING INDEFINITELY WHICH WE ALSO NOTICED IN CASE 1 & CASE 2 AS WELL.

CONCLUSIONS:

→ FROM THESE DATA POINTS WE CAN CONCLUDE THAT THE FUNCTION DOES NOT OBTAIN A MINIMUM AND THE SURFACE PLOT CONFIRMS THIS NATURE AS WELL.

→ WE CAN NOW CONCLUDE THAT THERE IS NO GLOBAL MINIMA FOR THIS FUNCTION AS THERE IS NO CONVERGENCE SEEN IN THE ALGORITHM EXECUTION.

Q2

Find the least natural number that divides 11^(n+1) +12^(2n−1) by examining the values of this expression for small values of n ∈ N , and prove the formula you conjectured.

GIVEN EXPRESSION: $f(n) = 11^{n+1} + 12^{2n-1}$

FOR $n = 1$

$\rightarrow f(1) = 11^2 + 12^1 = 121 + 12 = \underline{133}$

FOR $n = 2$

$\rightarrow f(2) = 11^3 + 12^3 = 1331 + 1728 = \underline{3059}$

FOR $n = 3$

$\rightarrow f(3) = 11^4 + 12^5 = \underline{263473}$

FINDING GCD FOR THE 3 NUMBERS

7 | 133
19 | 19
1

7 | 3059
19 | 437
23 | 23
1

7 | 263473
7 | 37639
19 | 5377
283 | 283
1

GREATEST COMMON DIVISOR FOR THE 3 NUMBERS ⇒ $7 \times 19 = 133$ (1)

FROM (1) WE SEE THAT 133 DIVIDES THE FUNCTION $f(n)$ FOR SMALL VALUES OF n, AND SINCE 133 CAN BE DIVIDED BY 7 AND 19, AND SINCE THE QUESTION NEEDS THE SMALLEST NATURAL NUMBER THAT DIVIDES $f(n)$ WE CAN TAKE 7 AS THE SMALLEST FACTOR FOR $f(n)$ FOR SMALL VALUES OF n

NOW: TO PROVE THAT ALL VALUES OF $f(n)$ IS DIVISIBLE BY 7, WE NEED TO PROVE THE FOLLOWING FOR ALL VALUES OF $n \in N$:

$$\text{IF } f(n) \,\%\, 7 = 0 \text{ THEN } f(n+1) \,\%\, 7 = 0$$

WE SHALL PROVE THIS BY USING PRINCIPLE OF MATHEMATICAL INDUCTION

BASIS :–

P(1): TO PROVE THAT:

$\Rightarrow f(1) \,\%\, 7 = 0$

$\Rightarrow (11^2 + 12^1) \,\%\, 7 = 0$

$\Rightarrow 133 \,\%\, 7 = 0$

$0 = 0 \rightarrow$ LHS = RHS

HENCE P(1) IS TRUE

NOW WE SHALL ASSUME $P(k)$ IS TRUE WHERE $k \in N$

$\Rightarrow f(k) \% 7 = 0$

$\Rightarrow$ $P(k) \Rightarrow (11^{k+1} + 12^{2k-1}) \% 7 = 0$ $\longrightarrow (2)$

TO PROVE THAT $P(k+1)$ IS TRUE WE NEED TO PROVE:

$f(k+1) \% 7 = 0$ $\longrightarrow (3)$

$$
\begin{aligned}
f(k+1) &= 11^{k+1+1} + 12^{2k+2-1} \\
&= 11\,(11^{k+1}) + 12^2\,(12^{2k-1}) \\
&= 11\,(11^{k+1}) + 144\,(12^{2k-1}) \\
&= 11\,(11^{k+1}) + 11\,(12^{2k-1}) + 133\,(12^{2k-1}) \\
&= \underbrace{11\,(11^{k+1} + 12^{2k-1})}_{\text{DIVISIBLE BY 7 FROM (2)}} + \underbrace{(7 \times 19)(12^{2k-1})}_{\text{DIVISIBLE BY 7}} \qquad \longrightarrow (4)
\end{aligned}
$$

SINCE ALL TERMS OF (4) ARE DIVISIBLE BY 7 WE CAN SAY THAT $f(k+1)$ IS ALSO DIVISIBLE BY 7 PROVING (3)

SINCE IT WAS HIGHLIGHTED THAT 7 IS THE SMALLEST FACTOR > 1 FOR $f(n)$ FOR SMALL VALUES OF n AND SINCE 7 IS A FACTOR FOR ALL $f(n)$ $n \in N$ FROM (4) WE CAN SAY THAT THE SMALLEST NATURAL NUMBER > 1 THAT DIVIDES $f(n)$ IS 7