Απειροστικός Λογισμός ΙΙ-Άσκηση bonus

Τζανάκης Νικόλαος,AM:4349

Ιούλιος 2020

1 Part 1: Finding minimun

1.1 [f_1 AND g_1]

$$f_1(x) = x^2 + 0.5x + 4 \tag{1}$$

$$g_1(x) = x + 5 \tag{2}$$

$$L_1(x) = (f_1(x) - g_1(x))^2 = (x^2 - 0.5x - 1)^2 \tag{3}$$

Βρισκουμε τα ελάχιστα με τον μηδενισμο της παραγωγου $L_1'(x) = 0$:
$L_1'(x) = 0 => 2(x^2 - 0.5x - 1) * (2x - 0.5) = 0 => x^2 - 0.5x - 1 = 0, 2x - 0.5 = 0$
$x^2 - 0.5x - 1 = 0 => \Delta = 1/4 + 4 = 17/4.x_1 = \frac{1+\sqrt{17}}{4}, x_2 = \frac{1-\sqrt{17}}{4}$
$2x - 0.5 = 0 => x_3 = \frac{1}{4}$

Εχουμε ελάχιστα για $x_1 = \frac{1+\sqrt{17}}{4}, x_2 = \frac{1-\sqrt{17}}{4}, x_3 = \frac{1}{4}$
$L(x_1) = 0, L(x_2) = 0, L(x_3) = \frac{1}{4} = \frac{289}{256}$

1.2 [f_2 AND g_2]

$$f_2(x) = x^3 - x^2 + 10 \tag{4}$$

$$g_2(x) = -x + 8 \tag{5}$$

$$L_2(x) = (f_2(x) - g_2(x))^2 = (x^3 - x^2 + x + 2)^2 \tag{6}$$

Βρισκουμε τα ελάχιστα με τον μηδενισμο της παραγωγου $L_2'(x) = 0$:
$L_2'(x) = 0 => 2(x^3 - x^2 + x + 2)(3x^2 - 2x + 1) = 0 => 3x^2 - 2x + 1 = 0 => \Delta = 4 - 4 * 1 * 3 = -8 < 0$
$x^3 - x^2 + x + 2 = 0 => x \approx -0.81053$
Έχουμε ελάχιστο για : $L_2(\approx -0.81053) = 0$

1.3 [f_3 AND g_3]

$$f_3(x) = e^x - 1 \tag{7}$$

$$g_3(x) = -10 * e^x + 2 \tag{8}$$

$$L_3(x) = (f_3(x) - g_3(x))^2 = (11e^x - 3)^2 \tag{9}$$

Βρισκουμε τα ελάχιστα με τον μηδενισμο της παραγωγου $L_3'(x) = 0$:
$L_3'(x) = 0 => 2(11e^x - 3)(11e^x) = 0 => 11e^x = 0$ Δεν υπάρχει λύση
$11e^x - 3 = 0 => 11e^x = 3 => e^x = \frac{3}{11} => x = ln(\frac{3}{11})$
Έχουμε ελάχιστο για $x = ln(\frac{3}{11}), L_3(ln(\frac{3}{11})) = 0$

2 Part3 Review of gradient descent for f_1, f_2, f_3

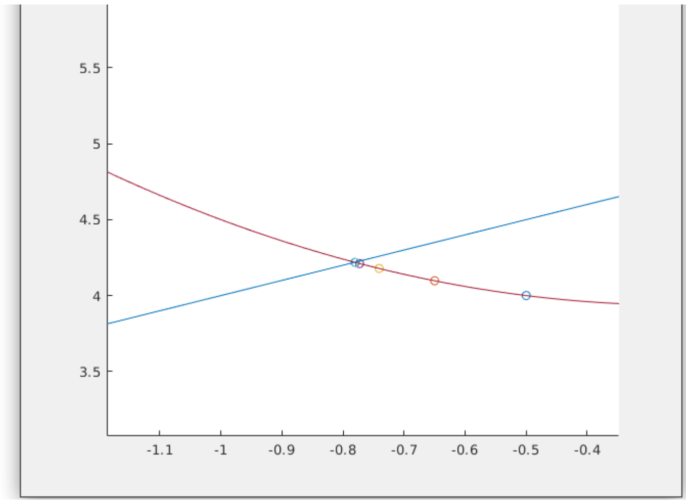
2.1 f_1 KAI g_1

$a = 0.1$

Iterations	X_cord	Y_cord
1	-0.5	4
2	-0.65	4.0975
3	-0.7409	4.1785
4	-0.77285	4.2109
5	-0.77951	4.2179
6	-0.78058	4.219
7	-0.78075	4.2192
8	-0.78077	4.2192
9	-0.78078	4.2192
10	-0.78078	4.2192
11	-0.78078	4.2192
12	-0.78078	4.2192
13	-0.78078	4.2192
14	-0.78078	4.2192
15	-0.78078	4.2192
16	-0.78078	4.2192
17	-0.78078	4.2192
18	-0.78078	4.2192
19	-0.78078	4.2192
20	-0.78078	4.2192

error =

8.8818e-16



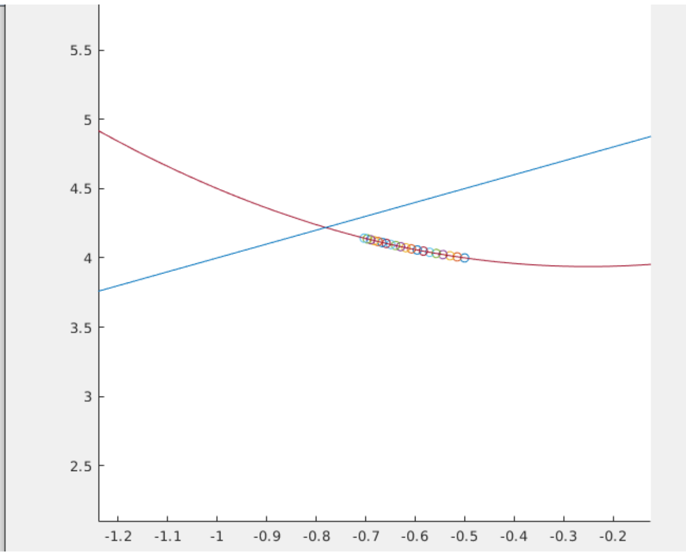
$a = 0.01$

Iterations	X_cord	Y_cord
1	-0.5	4
2	-0.515	4.0077
3	-0.5296	4.0157
4	-0.54378	4.0238
5	-0.55751	4.0321
6	-0.57077	4.0404
7	-0.58354	4.0487
8	-0.5958	4.0571
9	-0.60754	4.0653
10	-0.61876	4.0735
11	-0.62946	4.0815
12	-0.63962	4.0893
13	-0.64927	4.0969
14	-0.6584	4.1043
15	-0.66702	4.1114
16	-0.67515	4.1183
17	-0.6828	4.1248
18	-0.68997	4.1311
19	-0.6967	4.137
20	-0.703	4.1427

```
error =

    0.1543

fx >>
```



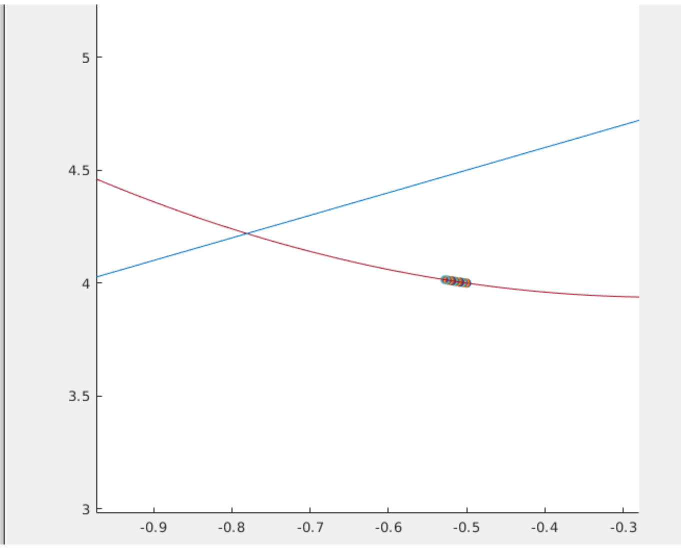
$a = 0.001$

Iterations	X_cord	Y_cord
1	-0.5	4
2	-0.5015	4.0008
3	-0.503	4.0015
4	-0.50449	4.0023
5	-0.50598	4.003
6	-0.50746	4.0038
7	-0.50894	4.0046
8	-0.51042	4.0053
9	-0.51189	4.0061
10	-0.51336	4.0069
11	-0.51483	4.0076
12	-0.51629	4.0084
13	-0.51774	4.0092
14	-0.5192	4.01
15	-0.52065	4.0107
16	-0.52209	4.0115
17	-0.52353	4.0123
18	-0.52497	4.0131
19	-0.5264	4.0139
20	-0.52783	4.0147

```
error =

    0.4575

fx >>
```



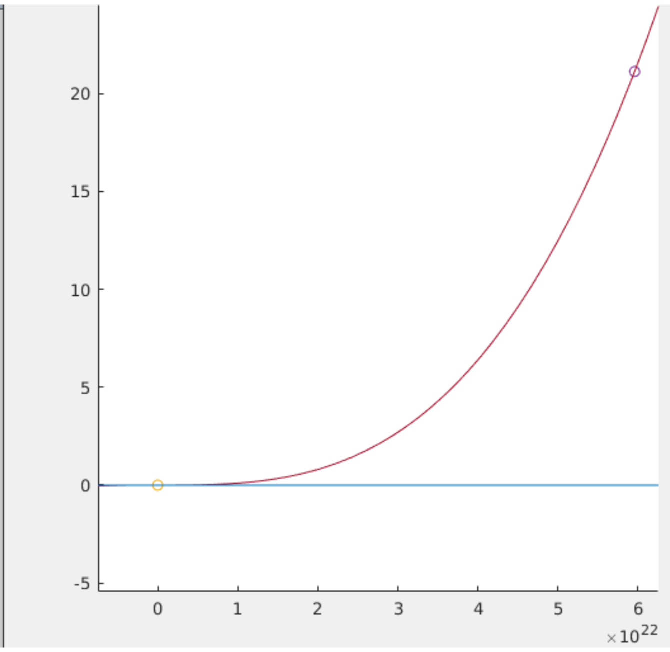
2.2 f_2 KAI g_2

$a = 0.1$

Iterations	X_cord	Y_cord
1	-1.5	4.375
2	9.5188	781.85
3	-39752	-6.2817e+13
4	5.956e+22	2.1128e+68
5	-4.4968e+113	-Inf
6	Inf	NaN
7	NaN	NaN
8	NaN	NaN
9	NaN	NaN
10	NaN	NaN
11	NaN	NaN
12	NaN	NaN
13	NaN	NaN
14	NaN	NaN
15	NaN	NaN
16	NaN	NaN
17	NaN	NaN
18	NaN	NaN
19	NaN	NaN
20	NaN	NaN

error =

NaN

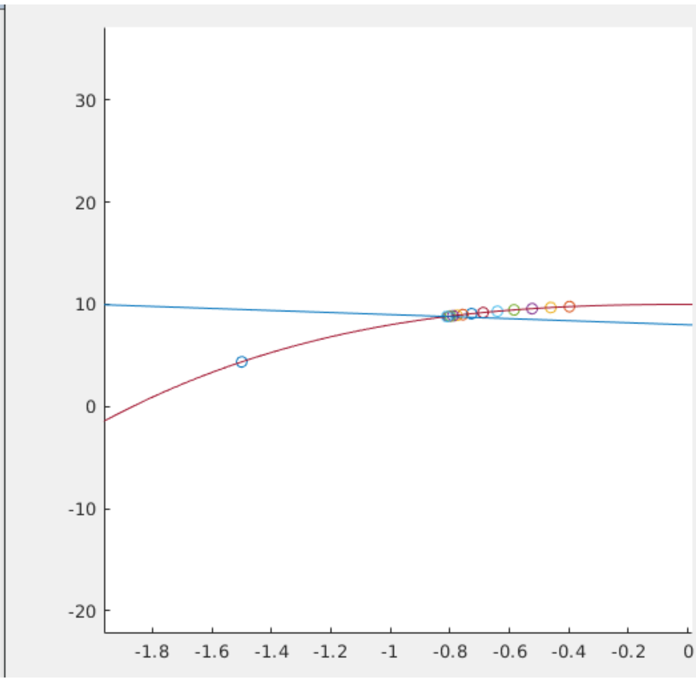


$a = 0.01$

Iterations	X_cord	Y_cord
1	-1.5	4.375
2	-0.39813	9.7784
3	-0.46084	9.6898
4	-0.52373	9.5821
5	-0.58448	9.4587
6	-0.64033	9.3274
7	-0.68857	9.1994
8	-0.72739	9.086
9	-0.75638	8.9951
10	-0.77658	8.9286
11	-0.78984	8.8834
12	-0.79817	8.8544
13	-0.80324	8.8366
14	-0.80627	8.8258
15	-0.80805	8.8194
16	-0.80909	8.8157
17	-0.8097	8.8135
18	-0.81005	8.8123
19	-0.81026	8.8115
20	-0.81037	8.8111

error =

7.4446e-04

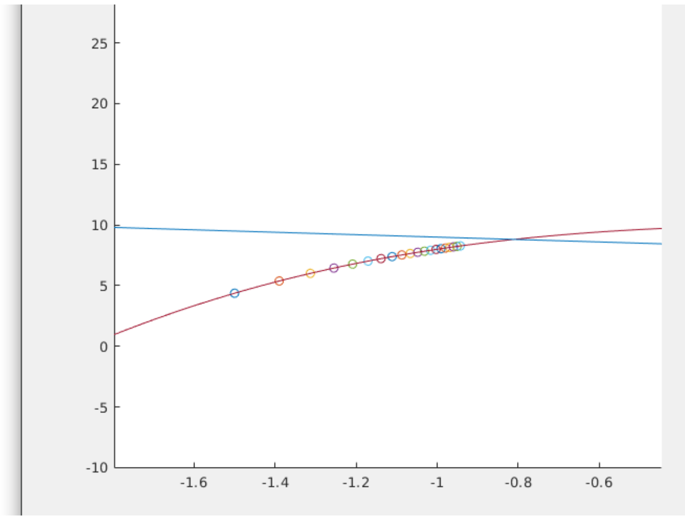


$a = 0.001$

Iterations	X_cord	Y_cord
1	-1.5	4.375
2	-1.3898	5.3839
3	-1.3131	6.0116
4	-1.255	6.4483
5	-1.2088	6.7727
6	-1.1708	7.0245
7	-1.1388	7.2264
8	-1.1114	7.3922
9	-1.0875	7.531
10	-1.0666	7.6489
11	-1.048	7.7504
12	-1.0315	7.8387
13	-1.0165	7.9162
14	-1.003	7.9847
15	-0.99077	8.0458
16	-0.97957	8.1005
17	-0.96931	8.1497
18	-0.95987	8.1943
19	-0.95117	8.2347
20	-0.94312	8.2716

error =

0.6715



2.3 f_3 KAI g_3

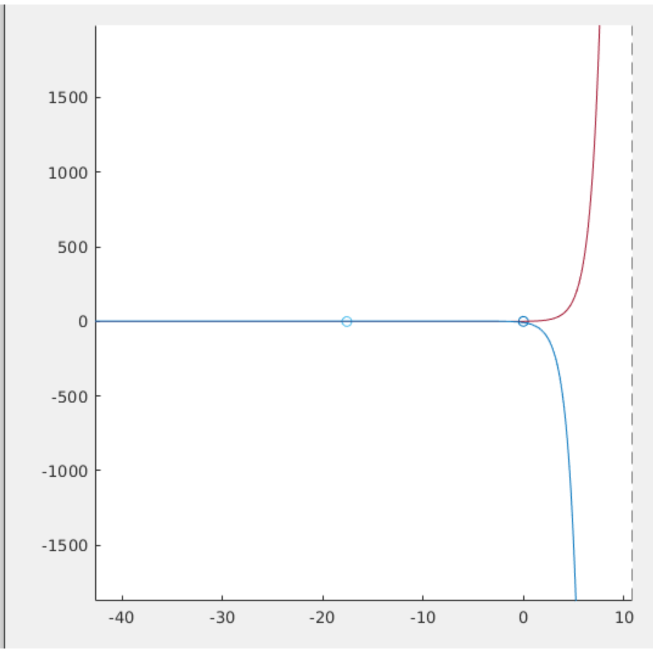
$a = 0.1$

Iterations	X_cord	Y_cord
1	0	0
2	-17.6	-1
3	-17.6	-1
4	-17.6	-1
5	-17.6	-1
6	-17.6	-1
7	-17.6	-1
8	-17.6	-1
9	-17.6	-1
10	-17.6	-1
11	-17.6	-1
12	-17.6	-1
13	-17.6	-1
14	-17.6	-1
15	-17.6	-1
16	-17.6	-1
17	-17.6	-1
18	-17.6	-1
19	-17.6	-1
20	-17.6	-1

```
error =

    3.0000

>>
```



$a = 0.01$

Iterations	X_cord	Y_cord
1	0	0
2	-1.76	-0.82796
3	-1.7181	-0.82059
4	-1.6776	-0.81317
5	-1.6387	-0.80577
6	-1.6018	-0.79847
7	-1.5671	-0.79135
8	-1.5348	-0.78449
9	-1.5049	-0.77796
10	-1.4777	-0.77183
11	-1.4531	-0.76615
12	-1.4311	-0.76095
13	-1.4116	-0.75624
14	-1.3945	-0.75204
15	-1.3796	-0.74833
16	-1.3668	-0.74508
17	-1.3558	-0.74227
18	-1.3465	-0.73984
19	-1.3386	-0.73778
20	-1.3319	-0.73602

error =

0.0963

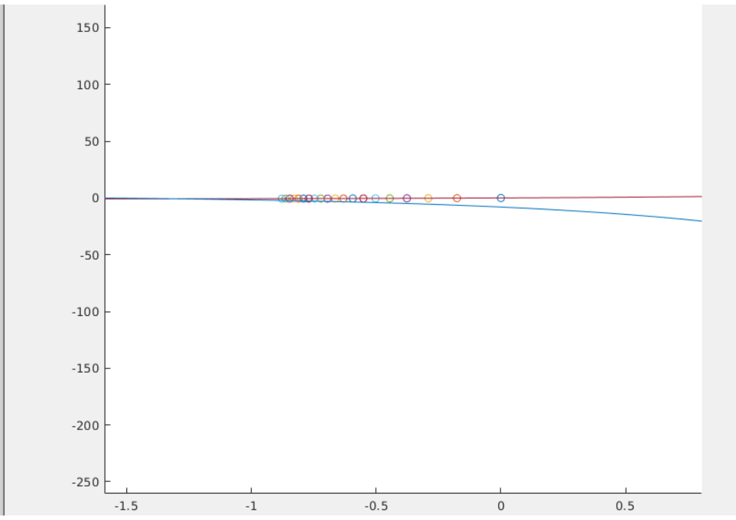
$a = 0.001$

Iterations	X_cord	Y_cord
1	0	0
2	-0.176	-0.16138
3	-0.29085	-0.25237
4	-0.37677	-0.31392
5	-0.4454	-0.35943
6	-0.50242	-0.39493
7	-0.55108	-0.42367
8	-0.59342	-0.44757
9	-0.63082	-0.46784
10	-0.66423	-0.48533
11	-0.69436	-0.50061
12	-0.72176	-0.5141
13	-0.74682	-0.52613
14	-0.76989	-0.53694
15	-0.79122	-0.54671
16	-0.81102	-0.5556
17	-0.82949	-0.56373
18	-0.84675	-0.5712
19	-0.86295	-0.57808
20	-0.87818	-0.58446

error =

1.5709

>>



Βλέπουμε οτι για α=0.1 , για τις συναρτήσεις $f_1 kai g_1$ το λάθος ειναι πάρα πολυ μικρό(8.8818ε-16),ενω σε αντίθεση με τις συναρτήσεις $f_2 g_2 kai f_3 g_3$ όπου για α=0.1 δεν έχουμε καν προσεγγιστικό αποτέλεσμα διότι το αποτέλεσμα στην αρχή ¨φεύγει' σε μεγαλύτερες τιμές και η συμπεριφορα του αλλοιώνεται .Για α=0.01 , και τα τρια παραδείγματα εμφανίζουν μικρότερο σφάλμα απο τις άλλες εκτιμήσεις(α=0.1,α=0.001) οπότε ειναι η καλύτερη επιλογή στα συγκεκριμμένα παραδείγματα.Για α = 0.001 , έχουμε καλή προσέγγιση αλλά σε αυτές τις περιπτώσεις με (μαξ ιτερατιονς = 20) το α=0.01 ειναι προτιμότερο,γιατι συγκλίνει πιο γρήγορα και με 20 επαναληψεις θα εχει καλυτερο αποτέλεσμα.

3 Part4 Quality of approach for different max iteration

Χρησιμοποίησα α=0.01 για κάθε παράδειγμα με διαφορετικό max iteration

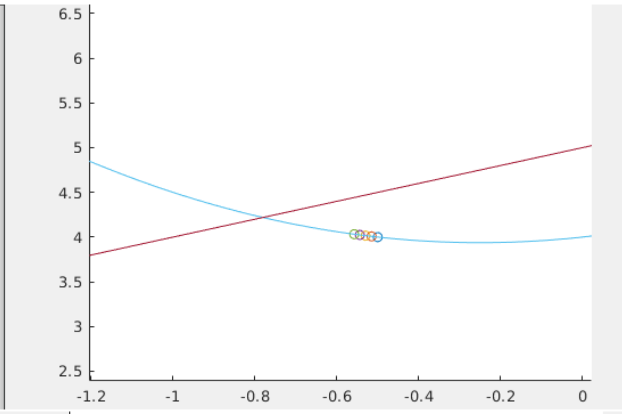
3.1 f_1 AND g_1

Iterations	X_cord	Y_cord
1	-0.5	4
2	-0.515	4.0077
3	-0.5296	4.0157
4	-0.54378	4.0238
5	-0.55751	4.0321

```
error =

    0.4104

>>
```

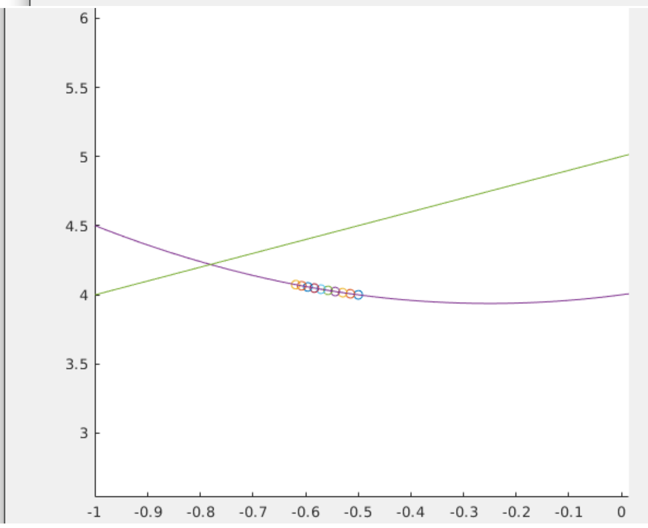


Iterations	X_cord	Y_cord
1	-0.5	4
2	-0.515	4.0077
3	-0.5296	4.0157
4	-0.54378	4.0238
5	-0.55751	4.0321
6	-0.57077	4.0404
7	-0.58354	4.0487
8	-0.5958	4.0571
9	-0.60754	4.0653
10	-0.61876	4.0735

```
error =

    0.3078

>>
```

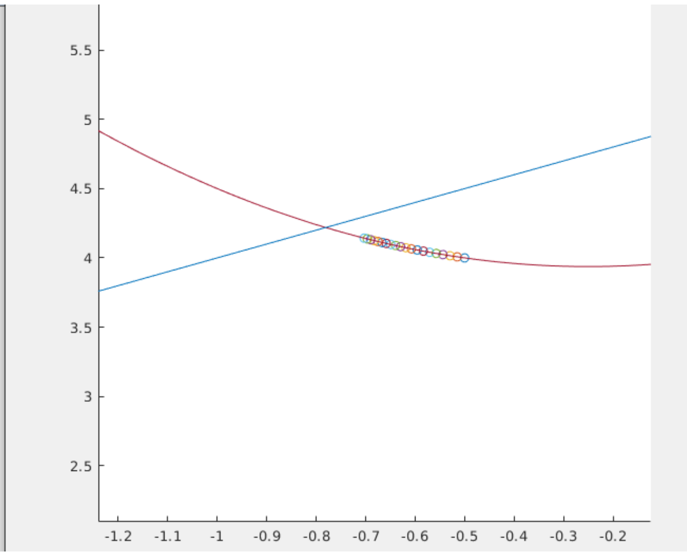


Iterations	X_cord	Y_cord
1	-0.5	4
2	-0.515	4.0077
3	-0.5296	4.0157
4	-0.54378	4.0238
5	-0.55751	4.0321
6	-0.57077	4.0404
7	-0.58354	4.0487
8	-0.5958	4.0571
9	-0.60754	4.0653
10	-0.61876	4.0735
11	-0.62946	4.0815
12	-0.63962	4.0893
13	-0.64927	4.0969
14	-0.6584	4.1043
15	-0.66702	4.1114
16	-0.67515	4.1183
17	-0.6828	4.1248
18	-0.68997	4.1311
19	-0.6967	4.137
20	-0.703	4.1427

```
error =

    0.1543

>>
```



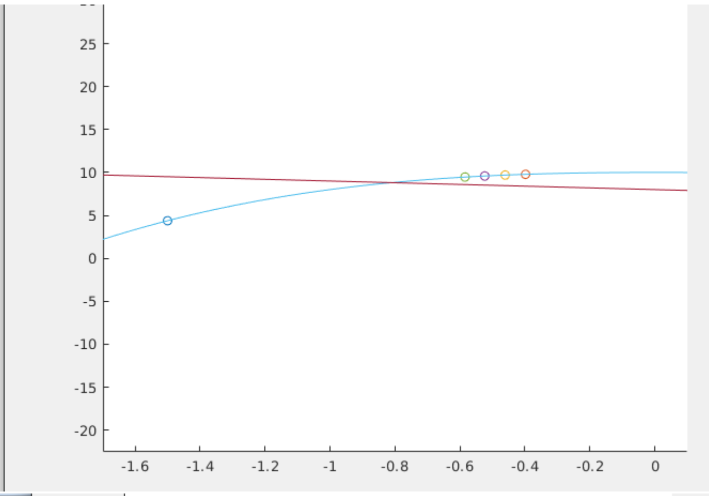
3.2 f_2 AND g_2

Iterations	X_cord	Y_cord
1	-1.5	4.375
2	-0.39813	9.7784
3	-0.46084	9.6898
4	-0.52373	9.5821
5	-0.58448	9.4587

error =

0.8742

fx >>

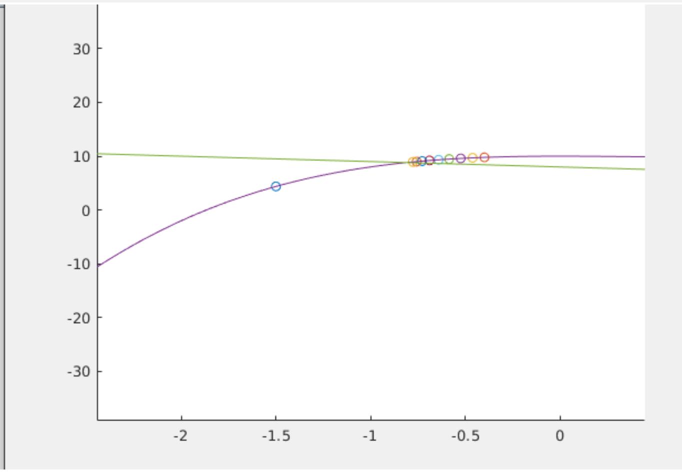


Iterations	X_cord	Y_cord
1	-1.5	4.375
2	-0.39813	9.7784
3	-0.46084	9.6898
4	-0.52373	9.5821
5	-0.58448	9.4587
6	-0.64033	9.3274
7	-0.68857	9.1994
8	-0.72739	9.086
9	-0.75638	8.9951
10	-0.77658	8.9286

error =

0.1520

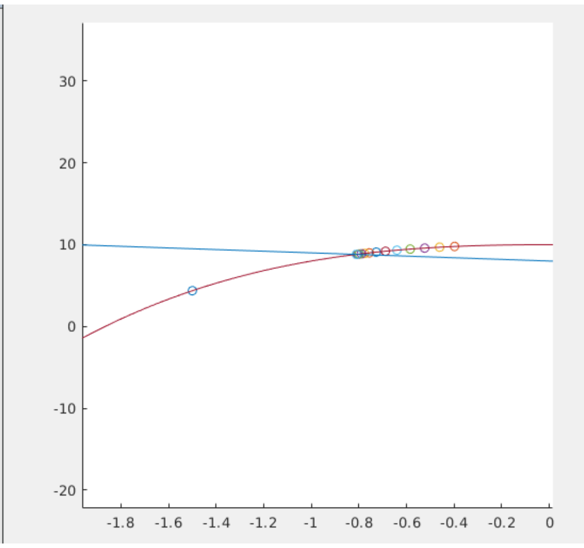
fx >>



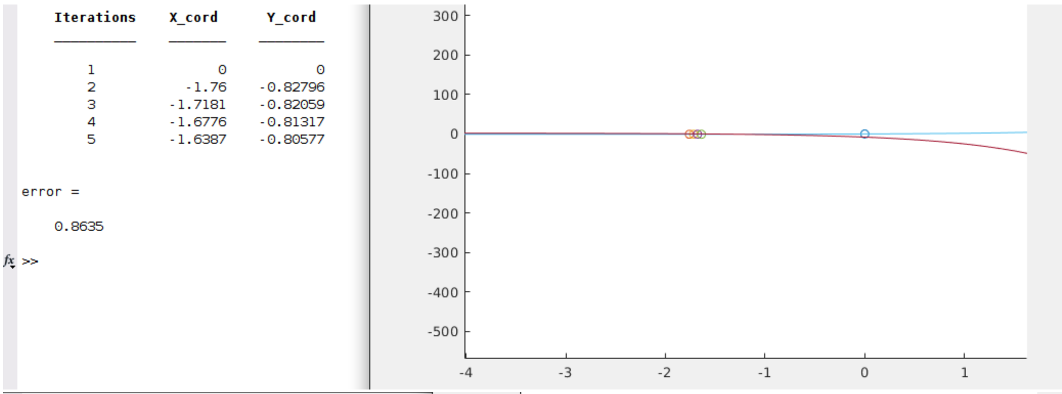
Iterations	X_cord	Y_cord
1	-1.5	4.375
2	-0.39813	9.7784
3	-0.46084	9.6898
4	-0.52373	9.5821
5	-0.58448	9.4587
6	-0.64033	9.3274
7	-0.68857	9.1994
8	-0.72739	9.086
9	-0.75638	8.9951
10	-0.77658	8.9286
11	-0.78984	8.8834
12	-0.79817	8.8544
13	-0.80324	8.8366
14	-0.80627	8.8258
15	-0.80805	8.8194
16	-0.80909	8.8157
17	-0.8097	8.8135
18	-0.81005	8.8123
19	-0.81026	8.8115
20	-0.81037	8.8111

```
error =

   7.4446e-04
```



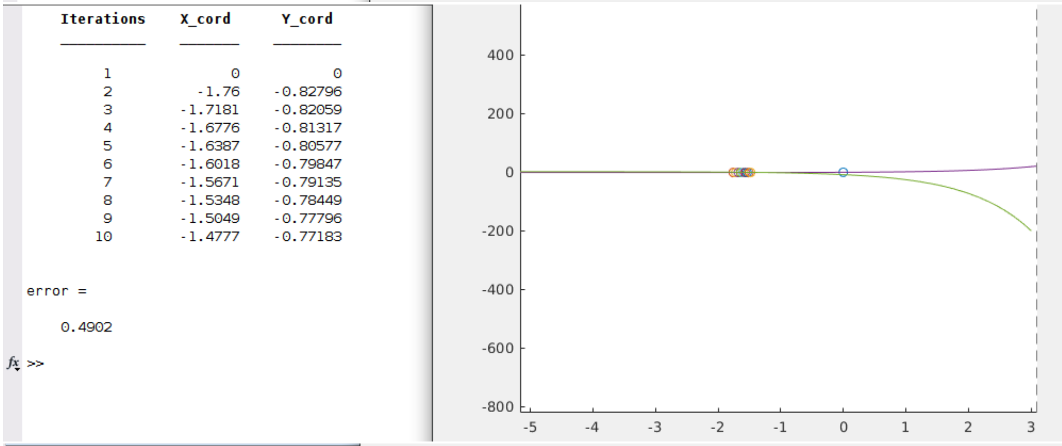
3.3 f_3 AND g_3



Iterations	X_cord	Y_cord
1	0	0
2	-1.76	-0.82796
3	-1.7181	-0.82059
4	-1.6776	-0.81317
5	-1.6387	-0.80577

error =

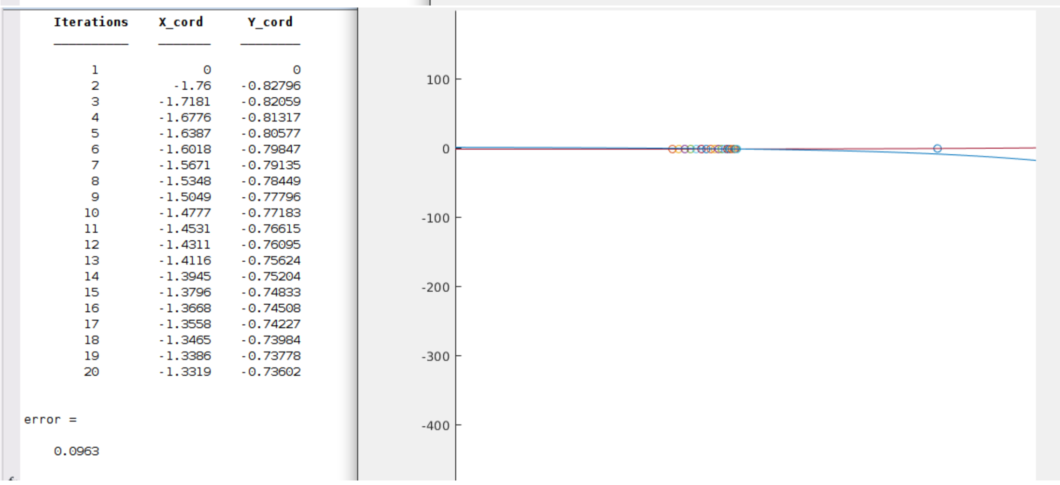
0.8635



Iterations	X_cord	Y_cord
1	0	0
2	-1.76	-0.82796
3	-1.7181	-0.82059
4	-1.6776	-0.81317
5	-1.6387	-0.80577
6	-1.6018	-0.79847
7	-1.5671	-0.79135
8	-1.5348	-0.78449
9	-1.5049	-0.77796
10	-1.4777	-0.77183

error =

0.4902



Iterations	X_cord	Y_cord
1	0	0
2	-1.76	-0.82796
3	-1.7181	-0.82059
4	-1.6776	-0.81317
5	-1.6387	-0.80577
6	-1.6018	-0.79847
7	-1.5671	-0.79135
8	-1.5348	-0.78449
9	-1.5049	-0.77796
10	-1.4777	-0.77183
11	-1.4531	-0.76615
12	-1.4311	-0.76095
13	-1.4116	-0.75624
14	-1.3945	-0.75204
15	-1.3796	-0.74833
16	-1.3668	-0.74508
17	-1.3558	-0.74227
18	-1.3465	-0.73984
19	-1.3386	-0.73778
20	-1.3319	-0.73602

error =

0.0963

Βλέπουμε οτι σε κάθε περίπτωση,περισσότερες επαναλήψεις μας φέρνουν πιο κοντά στο σωστό αποτέλεσμα.Η προσέγγιση με 20 επαναλήψεις ειναι αρκετά καλή,και αν θέλουμε ακόμα μεγαλύτερη ακρίβεια θα μπορουσαμε να κάνουμε και περισσότερες επαναλήψεις μέχρι το χ που βρίσκει η μέθοδος να είναι σχεδόν ίδιο κάθε φορά μετά απο ενα συγκεκριμμένο αριθμό επαναλήψεων.

4 Part5.Tests gradient descent method for two functions without common points

4.1 $f_1(x)$ AND $g_1\beta(x)$

$$f_1(x) = x^2 + 0.5 * x + 4 \tag{10}$$

$$g_1\beta(x) = x + 3 \tag{11}$$

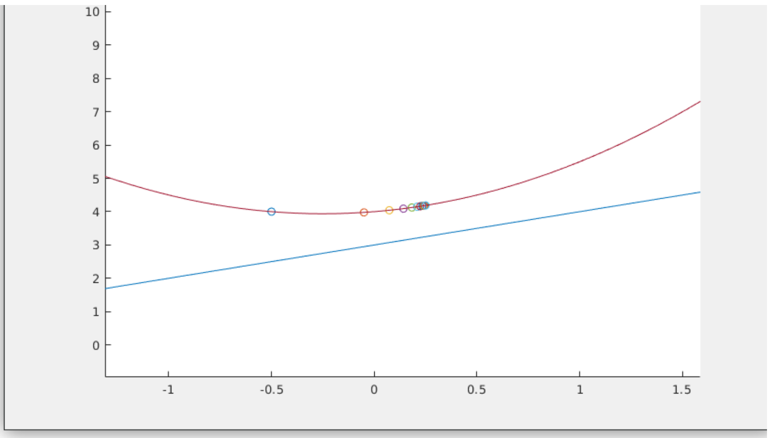
$L(x) = (f_1(x) - g_1\beta(x))^2 = (x^2 - 0.5 * x + 1)^2$
$L'(x) = 0 => 2(x^2 - 0.5 * x + 1)(2x - 0.5) = 0$
$x^2 - 0.5 * x + 1 = 0, \Delta = \frac{1}{4} - 4 < 0$
$2x - 0.5 = 0 => 2x = \frac{1}{2} => x = \frac{1}{4}$
Άρα έχουμε ελάχιστο για $x = \frac{1}{4}$
$L(\frac{1}{4}) \approx 0.8789$
Βρήκαμε για την καινούργια συνάρτηση $L(x) = (f_1(x) - g_1\beta(x))^2$ ελάχιστο για $x = \frac{1}{4}$ θα δούμε τώρα και πως η μέθοδος gradient descent συμπεριφέρεται στην συνάρτηση $L(x)$
Για αρχικό σημείο $x_0 = -0.5$ έχουμε:
$a = 0.1$

Iterations	X_cord	Y_cord
1	-0.5	4
2	-0.05	3.9775
3	0.0733	4.042
4	0.14177	4.091
5	0.18286	4.1249
6	0.20816	4.1474
7	0.22388	4.1621
8	0.23368	4.1714
9	0.2398	4.1774
10	0.24363	4.1812
11	0.24602	4.1835
12	0.24751	4.185
13	0.24844	4.1859
14	0.24903	4.1865
15	0.24939	4.1869
16	0.24962	4.1871
17	0.24976	4.1873
18	0.24985	4.1874
19	0.24991	4.1874
20	0.24994	4.1874

error =

0.9375

fx >>

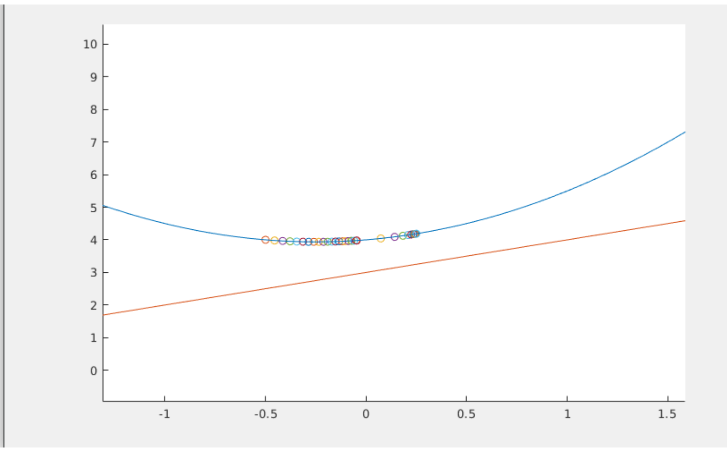


$a = 0.01$

Iterations	X_cord	Y_cord
1	-0.5	4
2	-0.455	3.9795
3	-0.41455	3.9646
4	-0.37789	3.9539
5	-0.34444	3.9464
6	-0.31375	3.9416
7	-0.28544	3.9388
8	-0.25922	3.9376
9	-0.23484	3.9377
10	-0.2121	3.9389
11	-0.19083	3.941
12	-0.17087	3.9438
13	-0.1521	3.9471
14	-0.13442	3.9509
15	-0.11774	3.955
16	-0.10196	3.9594
17	-0.087014	3.9641
18	-0.072845	3.9689
19	-0.059393	3.9738
20	-0.046606	3.9789

error =

1.0255

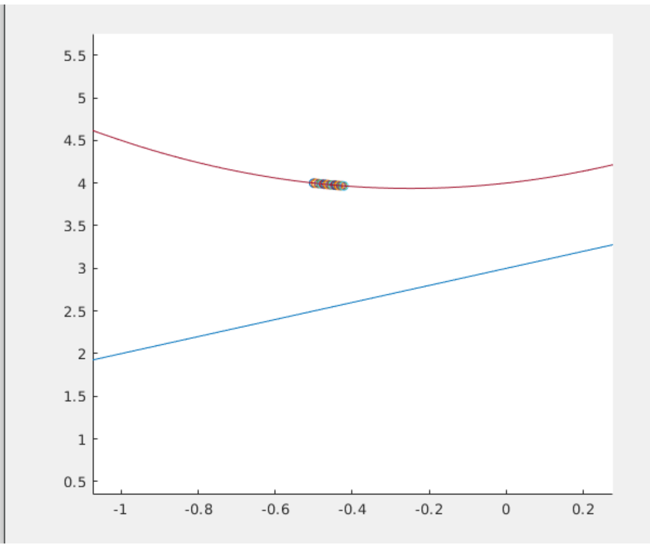


$a = 0.001$

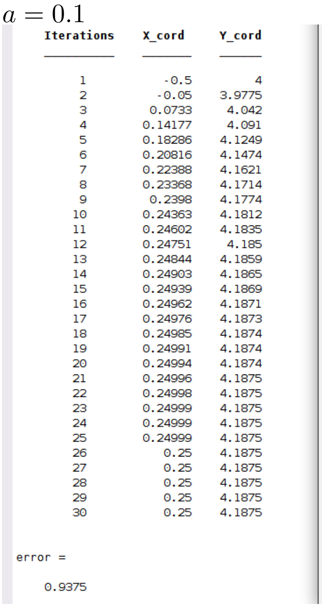
Iterations	X_cord	Y_cord
1	-0.5	4
2	-0.4955	3.9978
3	-0.49105	3.9956
4	-0.48664	3.9935
5	-0.48228	3.9915
6	-0.47796	3.9895
7	-0.47369	3.9875
8	-0.46946	3.9857
9	-0.46527	3.9838
10	-0.46113	3.9821
11	-0.45702	3.9804
12	-0.45296	3.9787
13	-0.44893	3.9771
14	-0.44494	3.9755
15	-0.44099	3.974
16	-0.43708	3.9725
17	-0.43321	3.9711
18	-0.42937	3.9697
19	-0.42557	3.9683
20	-0.4218	3.967

error =

1.3888



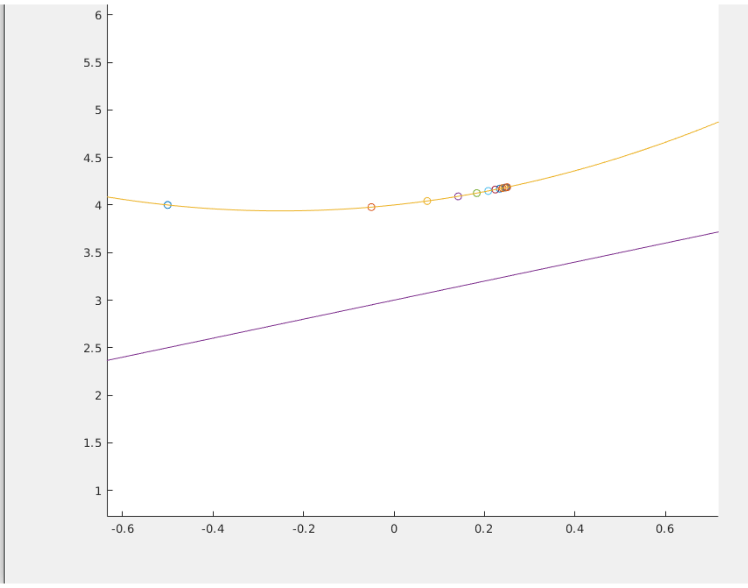
$a = 0.1$



Iterations	X_cord	Y_cord
1	-0.5	4
2	-0.05	3.9775
3	0.0733	4.042
4	0.14177	4.091
5	0.18286	4.1249
6	0.20816	4.1474
7	0.22388	4.1621
8	0.23368	4.1714
9	0.2398	4.1774
10	0.24363	4.1812
11	0.24602	4.1835
12	0.24751	4.185
13	0.24844	4.1859
14	0.24903	4.1865
15	0.24939	4.1869
16	0.24962	4.1871
17	0.24976	4.1873
18	0.24985	4.1874
19	0.24991	4.1874
20	0.24994	4.1874
21	0.24996	4.1875
22	0.24998	4.1875
23	0.24999	4.1875
24	0.24999	4.1875
25	0.24999	4.1875
26	0.25	4.1875
27	0.25	4.1875
28	0.25	4.1875
29	0.25	4.1875
30	0.25	4.1875

error =

0.9375



Βλέπουμε οτι η μέθοδος δουλέυει σε κάθε περίπτωση και προσπαθεί να συγκλίνει στην τιμή που ελαχιστοποιεί την συνάρτηση $L(x)$.Όμως οταν το πετυχένει,στο τελευταίο παράδειγμα,το σφάλμα αντι να ειναι 0,είναι 0.9375 και αυτό συμβαίνει διότι η συνάρτηση $f_1(x)$ και $g_1\beta(x)$ δεν έχουν κοινό σημείο

5 Part 2: Gradient descent algorithm in Matlab

```
clc
clear

syms x

%https://www.mathworks.com/matlabcentral/answers/297575-get-
    equation-as-input-from-user-str2func

 messageF = inputdlg(' Function f:' );
```

```
sF = messageF{:};
f = str2func(['@(x)' vectorize(sF)]); %vectorized f for plot

messageG = inputdlg('Function g:' );
sG = messageG{:};
g = str2func(['@(x)' vectorize(sG)]); % vectorized g for plot


messagex0 = inputdlg('x0:');
x0 = str2double(messagex0);

messageA = inputdlg('a:');
a = str2double(messageA);

messageMax = inputdlg('Max_iteration:');
max_iteration = str2double(messageMax);

L = (f(x) - g(x))^2;

A = [x0]; % an array to save x values
i = 1;

dL = diff(L,x);


while i < max_iteration

     result = double(subs(dL,x0));
     x0 = x0 - a*result;
     A = [A,x0];


i = i + 1;
end

%prints results in command window
iterations = 1:i;
Iterations = iterations';

X_cord = A';

Y_cord = double(subs(f,A))';

T = table(Iterations,X_cord,Y_cord);

```

```
disp(T);

%plot results, f and g


hold on;

for i = 1:length(A)
 plot(A(i),subs(f,A(i)),'o');
end

fplot(f);
fplot(g);


error = abs(f(x0) - g(x0))
```