Реализация градиентного спуска (steepest descent)

Для реализации градиентного спуска в общем виде нужно зафиксировать его сигнатуру. Скажем, что градиентный спуск принимает: оптимизируемую функцию, функцию для подсчёта градиента, исходную точку, функцию для линейного поиска, функцию реализующую терминирующие условия и возвращает массив полученных точек. Такой общности будет достаточно для нужд данной работы. Непосредственная реализация такой функции довольно тривиальна.

Градиентный спуск с фиксированным шагом (learning rate)

Для реализации данного метода достаточно реализовать подходящий метод линейного поиска. Он тривиален: нужно просто вернуть заданную константу. Рассмотрим поведение данного метода на различных функциях и с различной длиной шага. Проведём тесты на следующих функциях: $f(x, y) = \alpha(x - 5)^2 + (y - 7)^2$
$f(x, y) = \sin(0.5x^2 - 0.25y^2 + 3)\cos(2x + 1 - \exp(y))$
$f(x, y) = 100(y - x^2)^2 + (1 - x)^2$ (функция Розенброка)

In [29]:
```
%load_ext autoreload
%autoreload 2

from core.gradient_descent import *
from core.visualizer import *
```

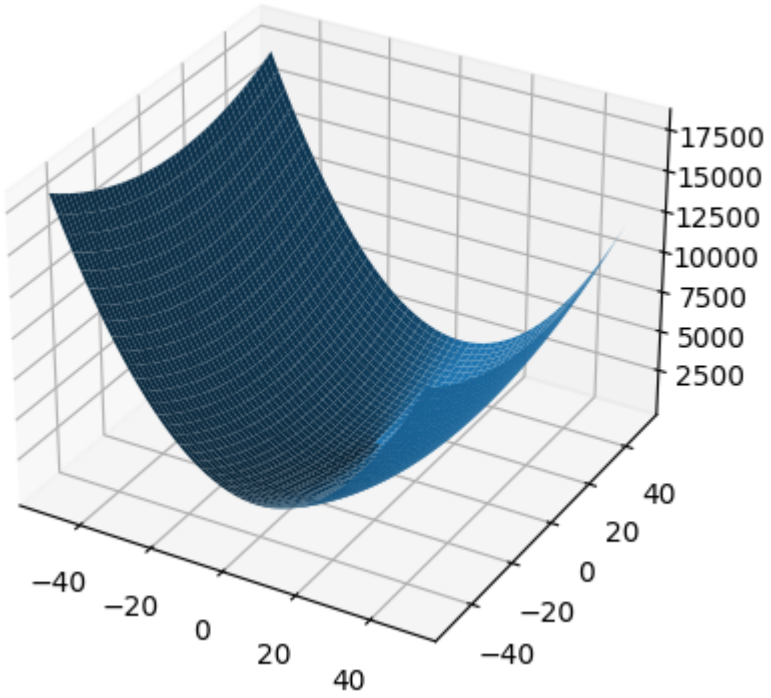
```
The autoreload extension is already loaded. To reload it, use:
  %reload_ext autoreload
```

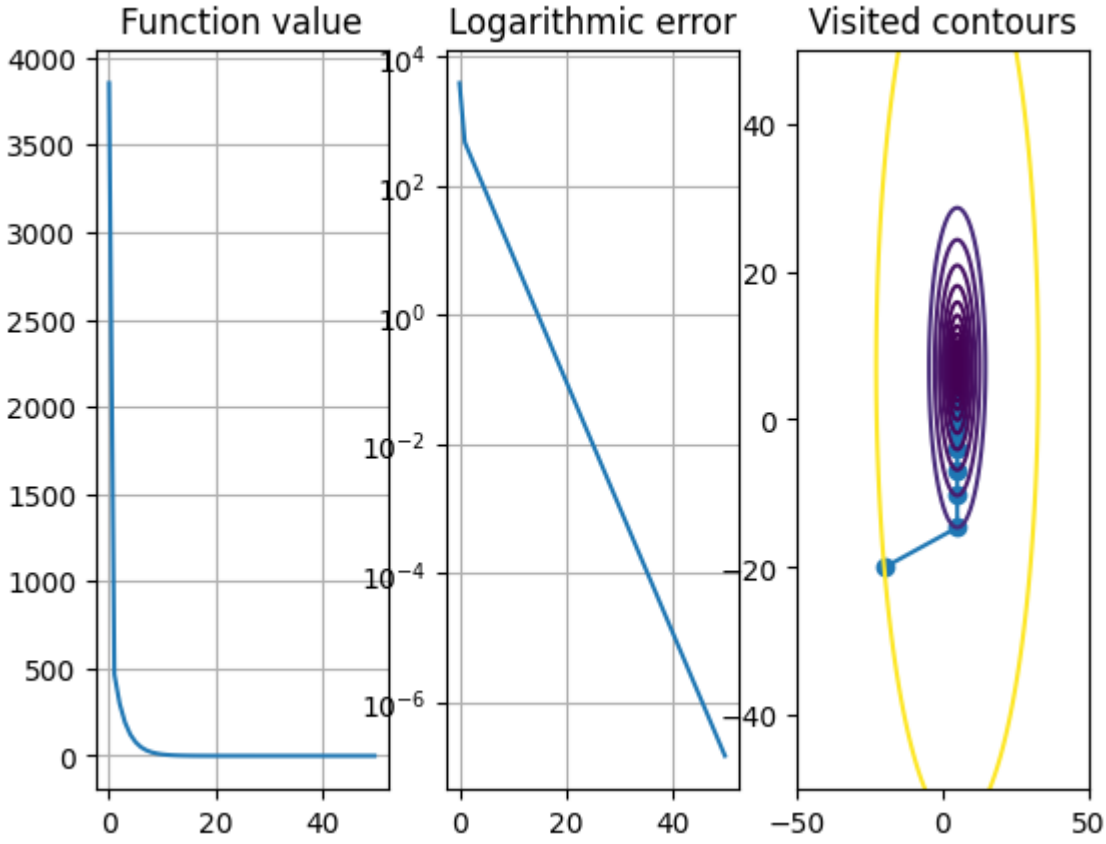
In [30]:
```
test_linear_search(fixed_step_search(0.1))
```

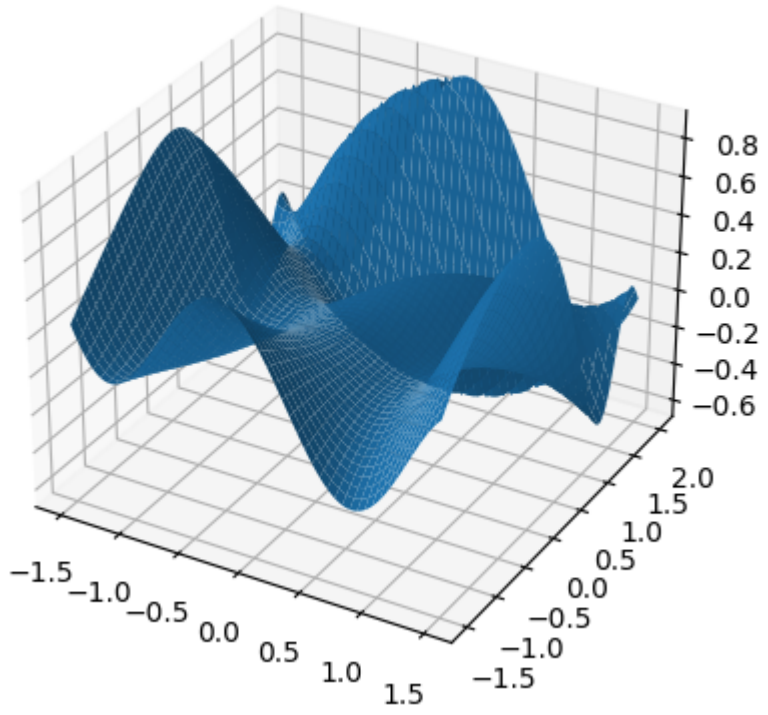
```
Optimizer trajectory:
[[-20.         -20.        ]
 [  5.         -14.6       ]
 [  5.         -10.28      ]
 [  5.          -6.824     ]
 [  5.          -4.0592    ]
 [  5.          -1.84736   ]
 [  5.          -0.077888  ]
 [  5.           1.3376896 ]
 [  5.           2.47015168]
 [  5.           3.37612134]
 [  5.           4.10089708]
 [  5.           4.68071766]
 [  5.           5.14457413]
 [  5.           5.5156593 ]
 [  5.           5.81252744]
 [  5.           6.05002195]
 [  5.           6.24001756]
 [  5.           6.39201405]
 [  5.           6.51361124]
 [  5.           6.61088899]
 [  5.           6.68871119]
 [  5.           6.75096896]
 [  5.           6.80077516]
 [  5.           6.84062013]
 [  5.           6.8724961 ]
 [  5.           6.89799688]
 [  5.           6.91839751]
 [  5.           6.93471801]
 [  5.           6.9477744 ]
 [  5.           6.95821952]
 [  5.           6.96657562]
 [  5.           6.9732605 ]
 [  5.           6.9786084 ]
 [  5.           6.98288672]
 [  5.           6.98630937]
 [  5.           6.9890475 ]
 [  5.           6.991238  ]
 [  5.           6.9929904 ]
 [  5.           6.99439232]
 [  5.           6.99551386]
 [  5.           6.99641108]
 [  5.           6.99712887]
 [  5.           6.99770309]
 [  5.           6.99816248]
 [  5.           6.99852998]
 [  5.           6.99882398]
 [  5.           6.99905919]
 [  5.           6.99924735]
 [  5.           6.99939788]
 [  5.           6.9995183 ]
 [  5.           6.99961464]]
Best value found: x* = [5.         6.99961464] with f(x*) = 1.4849992267483374e-07
Optimizer trajectory:
[[-0.1        -0.4       ]
 [-0.10521824 -0.38199919]
```

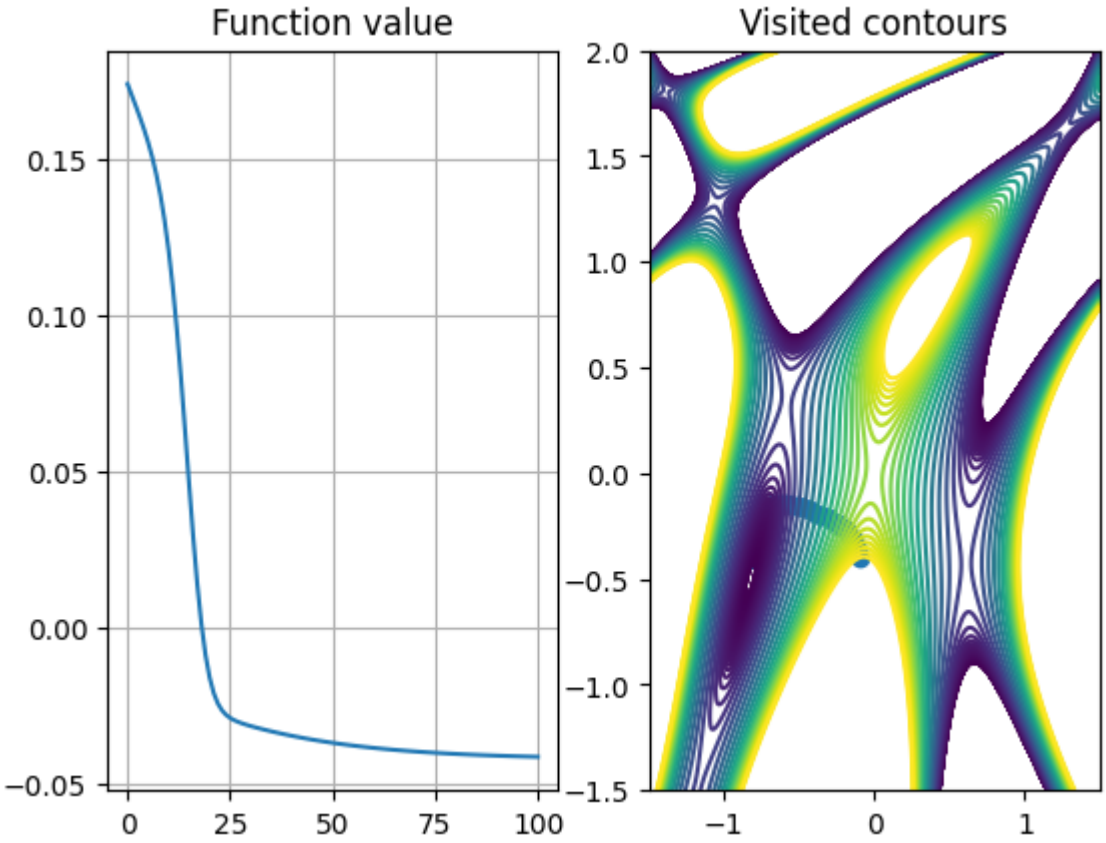
```
[-0.11185515 -0.36454272]
[-0.12010457 -0.34758569]
[-0.13018685 -0.33108343]
[-0.14234858 -0.3149924 ]
[-0.15685956 -0.29927173]
[-0.1740055  -0.28388572]
[-0.19407368 -0.2688077 ]
[-0.21732877 -0.25402572]
[-0.243975   -0.23955054]
[-0.27410232 -0.22542585]
[-0.30761707 -0.21174018]
[-0.34416528 -0.1986377 ]
[-0.38306793 -0.1863236 ]
[-0.42330022 -0.17505703]
[-0.46355026 -0.16512577]
[-0.50237431 -0.15680219]
[-0.53842277 -0.15029041]
[-0.57066446 -0.1456833 ]
[-0.59852584 -0.1429465 ]
[-0.62190274 -0.14193421]
[-0.64106521 -0.1424263 ]
[-0.65651444 -0.14417041]
[-0.66884749 -0.14691624]
[-0.67865904 -0.15043715]
[-0.68648477 -0.15454017]
[-0.69277728 -0.15906789]
[-0.69790282 -0.16389577]
[-0.70214899 -0.16892736]
[-0.70573683 -0.17408919]
[-0.70883359 -0.17932598]
[-0.71156439 -0.18459646]
[-0.71402202 -0.18987013]
[-0.71627483 -0.19512454]
[-0.71837291 -0.20034329]
[-0.72035285 -0.20551448]
[-0.72224124 -0.21062954]
[-0.72405738 -0.21568237]
[-0.72581523 -0.22066868]
[-0.72752489 -0.22558551]
[-0.72919364 -0.23043091]
[-0.73082673 -0.23520363]
[-0.73242794 -0.23990298]
[-0.734      -0.24452866]
[-0.73554489 -0.24908067]
[-0.73706406 -0.25355922]
[-0.73855855 -0.25796469]
[-0.74002917 -0.2622976 ]
[-0.7414765  -0.26655855]
[-0.74290103 -0.27074822]
[-0.74430311 -0.27486733]
[-0.74568308 -0.27891666]
[-0.74704121 -0.28289702]
[-0.74837774 -0.28680925]
[-0.74969292 -0.2906542 ]
[-0.75098696 -0.29443275]
[-0.75226008 -0.29814577]
```

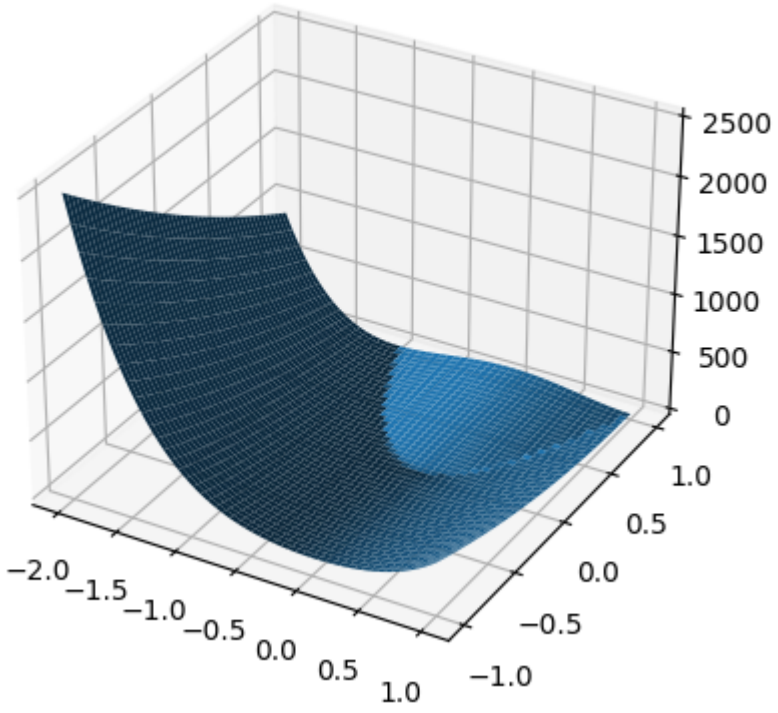
```
 [-0.75351251 -0.30179416]
 [-0.75474446 -0.30537883]
 [-0.75595616 -0.30890067]
 [-0.75714783 -0.3123606 ]
 [-0.75831969 -0.31575952]
 [-0.75947199 -0.31909834]
 [-0.76060496 -0.32237798]
 [-0.76171883 -0.32559934]
 [-0.76281384 -0.32876332]
 [-0.76389025 -0.33187082]
 [-0.76494828 -0.33492274]
 [-0.76598819 -0.33791996]
 [-0.76701022 -0.34086337]
 [-0.76801463 -0.34375384]
 [-0.76900166 -0.34659224]
 [-0.76997157 -0.34937943]
 [-0.77092459 -0.35211627]
 [-0.77186099 -0.35480359]
 [-0.77278101 -0.35744224]
 [-0.77368489 -0.36003304]
 [-0.77457289 -0.3625768 ]
 [-0.77544524 -0.36507433]
 [-0.7763022  -0.36752644]
 [-0.77714401 -0.36993389]
 [-0.7779709  -0.37229748]
 [-0.77878312 -0.37461797]
 [-0.7795809  -0.37689611]
 [-0.78036448 -0.37913265]
 [-0.78113409 -0.38132833]
 [-0.78188995 -0.38348386]
 [-0.78263231 -0.38559996]
 [-0.78336138 -0.38767734]
 [-0.78407738 -0.38971668]
 [-0.78478055 -0.39171868]
 [-0.78547108 -0.39368399]
 [-0.78614921 -0.39561328]
 [-0.78681513 -0.3975072 ]
 [-0.78746907 -0.39936638]
 [-0.78811122 -0.40119147]
 [-0.78874179 -0.40298307]
 [-0.78936098 -0.40474179]
 [-0.789969   -0.40646824]
 [-0.79056602 -0.408163  ]]
Best value found: x* = [-0.79056602 -0.408163  ] with f(x*) = -0.04113237042735722
Optimizer trajectory:
[[-1.50000000e+00  2.50000000e-01]
 [ 1.19000000e+02  4.02500000e+01]
 [-6.72146746e+07  2.82455250e+05]]
Best value found: x* = [-67214674.60000001    282455.25      ] with f(x*) = 2.0410
629616202993e+33
```

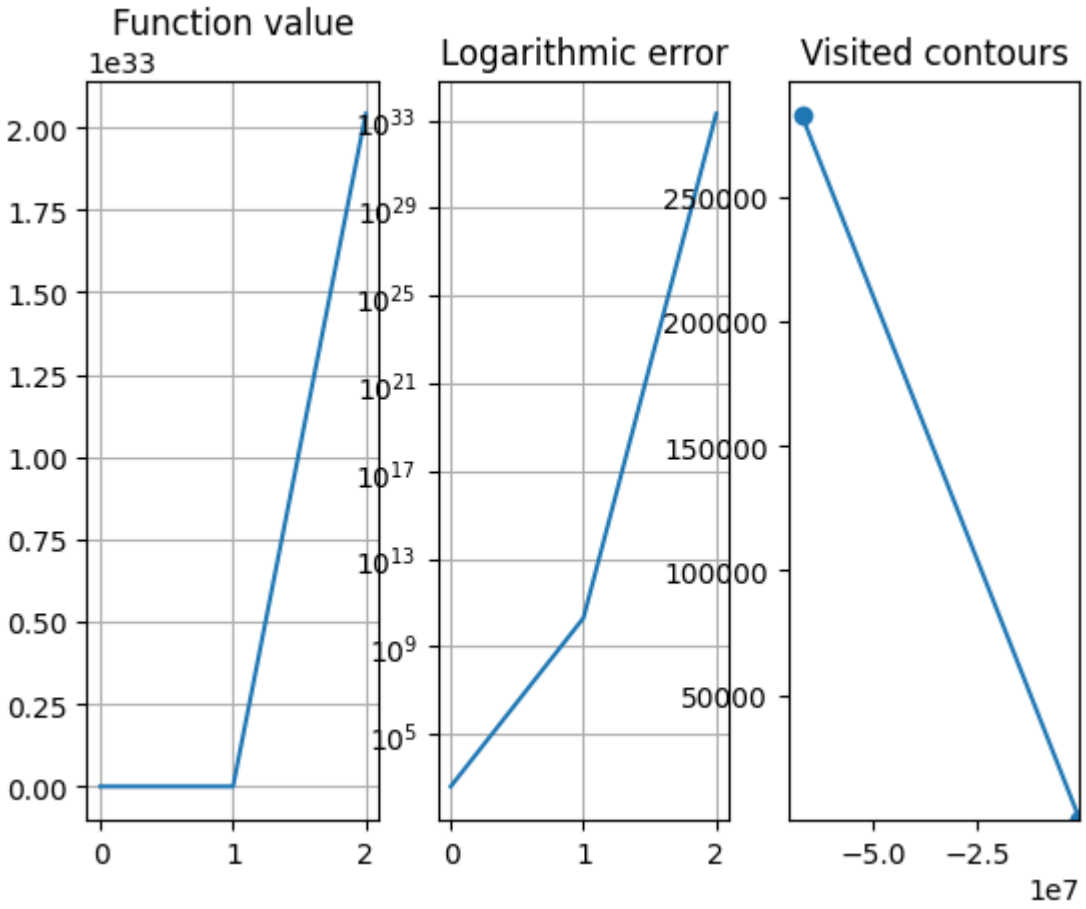

17500
15000
12500
10000
7500
5000
2500
−40
−20
0
20
40
40
20
0
−20
−40


Function value
Logarithmic error
Visited contours
4000
3500
3000
2500
2000
1500
1000
500
0
0
20
40
10^{4}
10^{2}
10^{0}
10^{-2}
10^{-4}
10^{-6}
0
20
40
40
20
0
−20
−40
−50
0
50


0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5


Function value
0.15
0.10
0.05
0.00
−0.05
0
25
50
75
100
Visited contours
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1
0
1



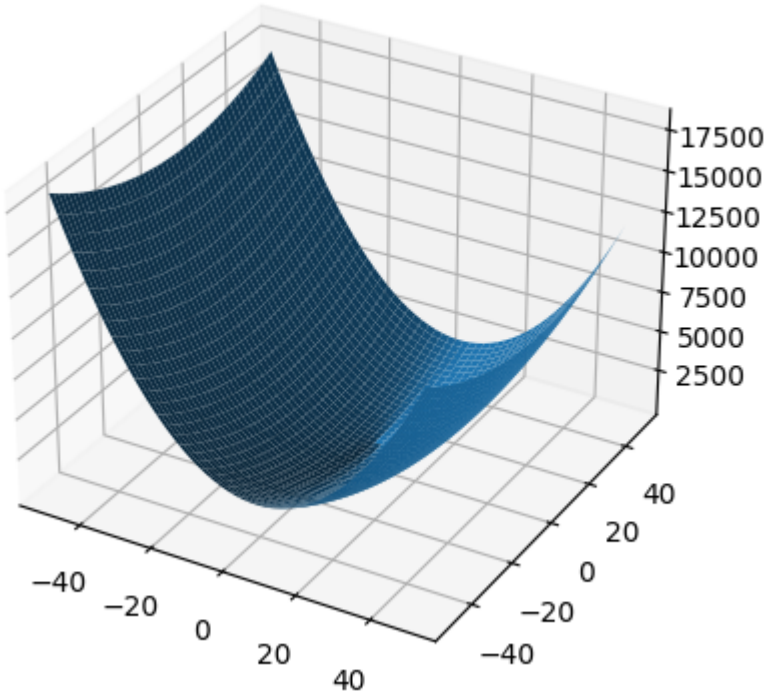


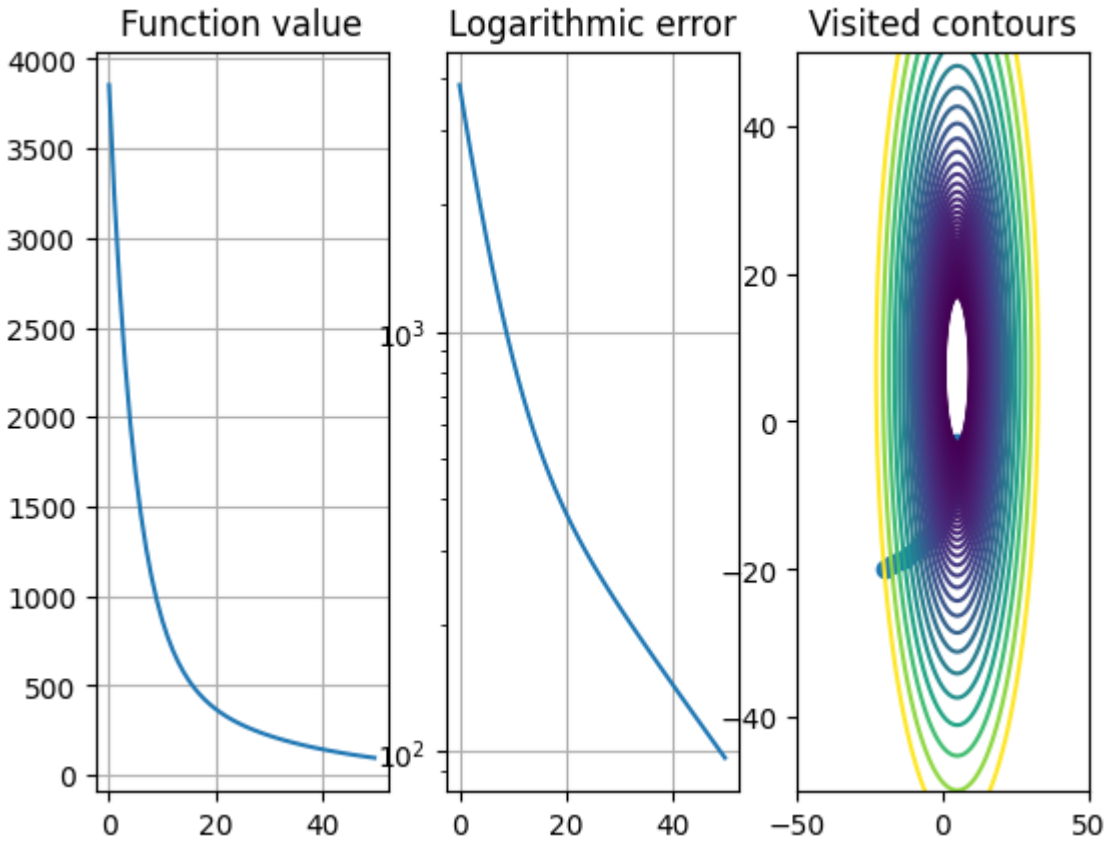
In [31]:
```
test_linear_search(fixed_step_search(0.01))
```

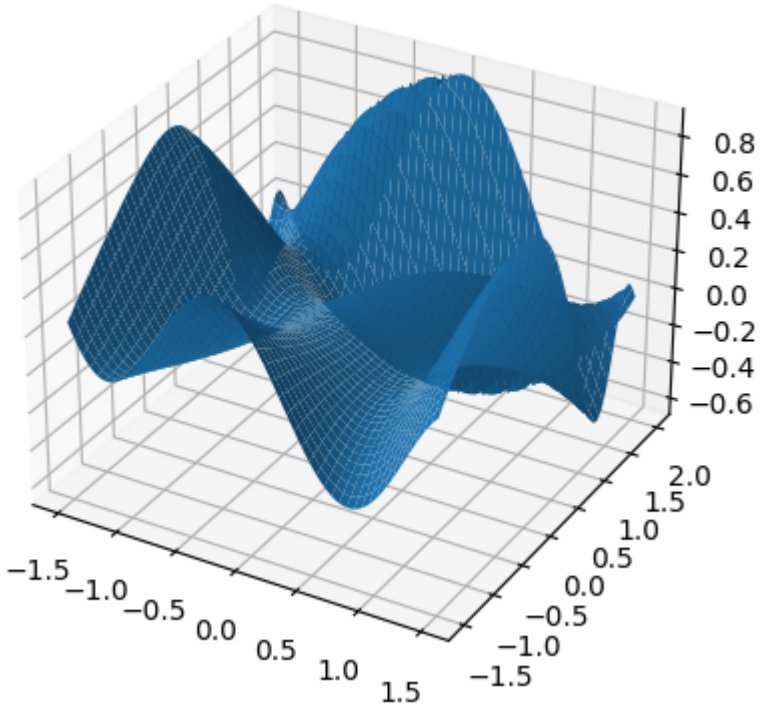
```
Optimizer trajectory:
[[-20.         -20.        ]
 [-17.5        -19.46      ]
 [-15.25       -18.9308    ]
 [-13.225      -18.412184  ]
 [-11.4025     -17.90394032]
 [ -9.76225    -17.40586151]
 [ -8.286025   -16.91774428]
 [ -6.9574225  -16.4393894 ]
 [ -5.76168025 -15.97060161]
 [ -4.68551223 -15.51118958]
 [ -3.716961   -15.06096579]
 [ -2.8452649  -14.61974647]
 [ -2.06073841 -14.18735154]
 [ -1.35466457 -13.76360451]
 [ -0.71919811 -13.34833242]
 [ -0.1472783  -12.94136577]
 [  0.36744953 -12.54253846]
 [  0.83070458 -12.15168769]
 [  1.24763412 -11.76865393]
 [  1.62287071 -11.39328085]
 [  1.96058364 -11.02541524]
 [  2.26452527 -10.66490693]
 [  2.53807274 -10.31160879]
 [  2.78426547  -9.96537662]
 [  3.00583892  -9.62606909]
 [  3.20525503  -9.2935477 ]
 [  3.38472953  -8.96767675]
 [  3.54625657  -8.64832321]
 [  3.69163092  -8.33535675]
 [  3.82246783  -8.02864962]
 [  3.94022104  -7.72807662]
 [  4.04619894  -7.43351509]
 [  4.14157904  -7.14484479]
 [  4.22742114  -6.86194789]
 [  4.30467903  -6.58470894]
 [  4.37421112  -6.31301476]
 [  4.43679001  -6.04675446]
 [  4.49311101  -5.78581937]
 [  4.54379991  -5.53010298]
 [  4.58941992  -5.27950093]
 [  4.63047793  -5.03391091]
 [  4.66743013  -4.79323269]
 [  4.70068712  -4.55736803]
 [  4.73061841  -4.32622067]
 [  4.75755657  -4.09969626]
 [  4.78180091  -3.87770234]
 [  4.80362082  -3.66014829]
 [  4.82325874  -3.44694532]
 [  4.84093286  -3.23800642]
 [  4.85683958  -3.03324629]
 [  4.87115562  -2.83258136]]
Best value found: x* = [ 4.87115562 -2.83258136] with f(x*) = 96.7626606187988
Optimizer trajectory:
[[-0.1        -0.4       ]
 [-0.10052182 -0.39819992]
```

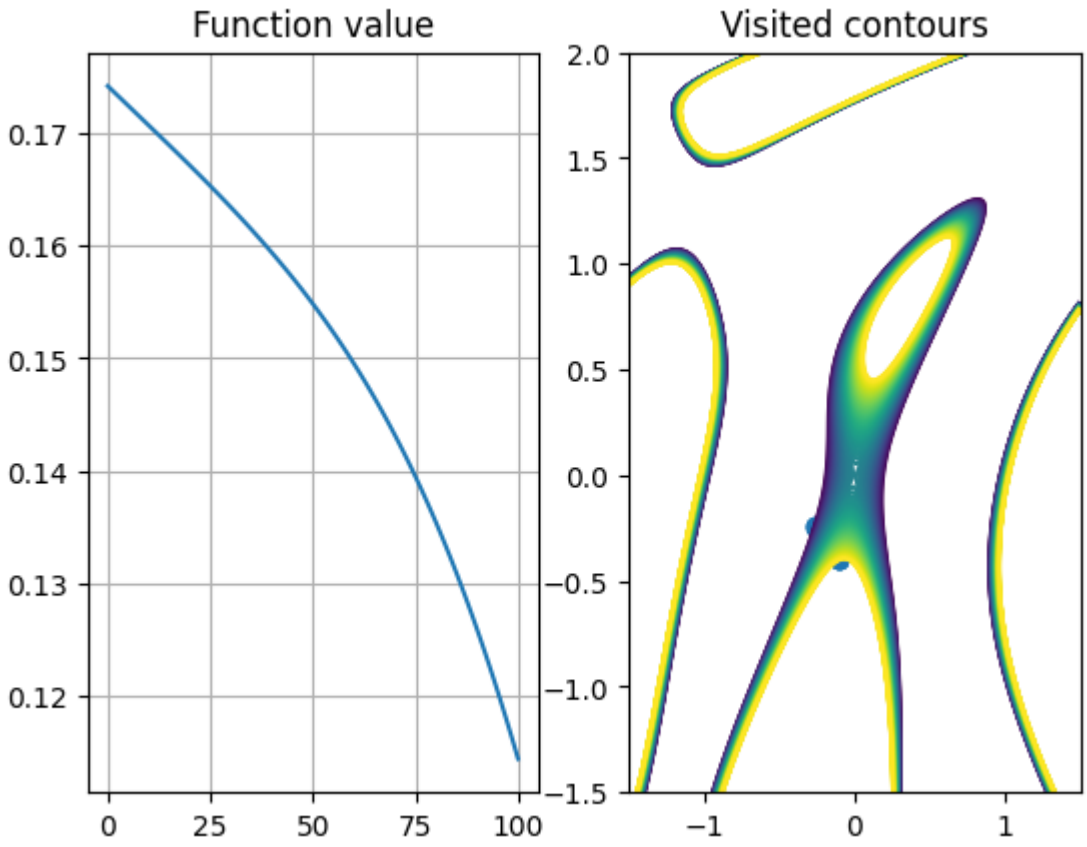
```
[-0.10105797 -0.39640521]
[-0.10160865 -0.39461583]
[-0.10217406 -0.39283172]
[-0.1027544  -0.39105285]
[-0.10334988 -0.38927916]
[-0.10396073 -0.38751061]
[-0.10458714 -0.38574715]
[-0.10522935 -0.38398874]
[-0.10588756 -0.38223532]
[-0.10656202 -0.38048686]
[-0.10725294 -0.37874329]
[-0.10796056 -0.37700459]
[-0.10868511 -0.37527069]
[-0.10942683 -0.37354155]
[-0.11018596 -0.37181713]
[-0.11096275 -0.37009737]
[-0.11175744 -0.36838224]
[-0.11257028 -0.36667168]
[-0.11340153 -0.36496564]
[-0.11425145 -0.36326408]
[-0.11512029 -0.36156696]
[-0.11600833 -0.35987422]
[-0.11691582 -0.35818582]
[-0.11784304 -0.35650171]
[-0.11879027 -0.35482185]
[-0.11975777 -0.35314619]
[-0.12074583 -0.35147468]
[-0.12175474 -0.34980728]
[-0.12278477 -0.34814394]
[-0.12383623 -0.34648462]
[-0.12490939 -0.34482927]
[-0.12600456 -0.34317784]
[-0.12712203 -0.34153029]
[-0.12826211 -0.33988659]
[-0.1294251  -0.33824667]
[-0.13061131 -0.3366105 ]
[-0.13182104 -0.33497804]
[-0.1330546  -0.33334924]
[-0.13431232 -0.33172407]
[-0.1355945  -0.33010247]
[-0.13690147 -0.3284844 ]
[-0.13823355 -0.32686984]
[-0.13959105 -0.32525873]
[-0.14097431 -0.32365104]
[-0.14238364 -0.32204672]
[-0.14381938 -0.32044575]
[-0.14528186 -0.31884808]
[-0.1467714  -0.31725368]
[-0.14828833 -0.3156625 ]
[-0.149833   -0.31407453]
[-0.15140572 -0.31248971]
[-0.15300684 -0.31090803]
[-0.15463667 -0.30932944]
[-0.15629557 -0.30775393]
[-0.15798385 -0.30618145]
[-0.15970185 -0.30461199]
```

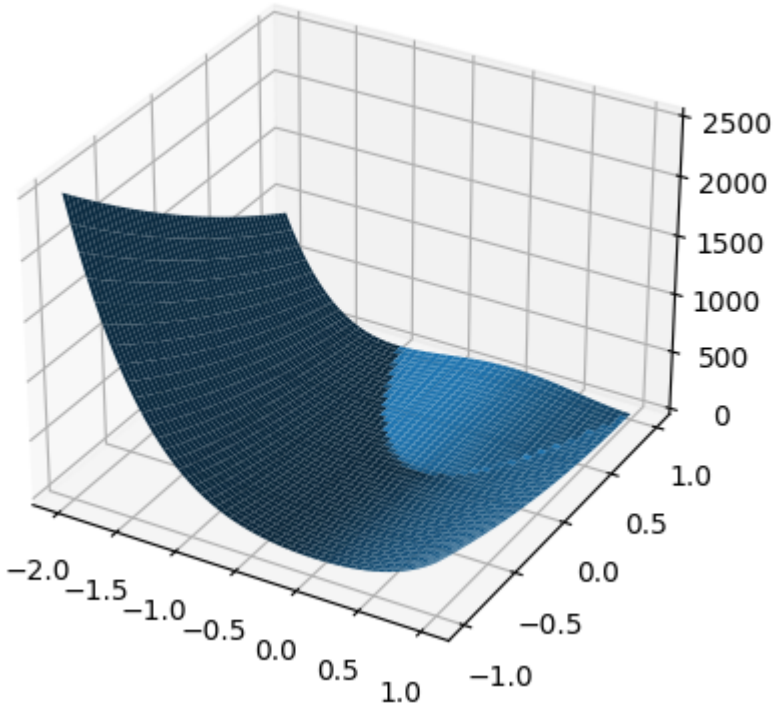
```
 [-0.16144989 -0.30304551]
 [-0.16322832 -0.30148199]
 [-0.16503744 -0.29992141]
 [-0.16687759 -0.29836375]
 [-0.16874909 -0.29680899]
 [-0.17065225 -0.2952571 ]
 [-0.17258739 -0.29370807]
 [-0.17455483 -0.2921619 ]
 [-0.17655487 -0.29061856]
 [-0.17858781 -0.28907804]
 [-0.18065396 -0.28754035]
 [-0.18275359 -0.28600546]
 [-0.184887   -0.28447339]
 [-0.18705448 -0.28294412]
 [-0.18925627 -0.28141767]
 [-0.19149267 -0.27989403]
 [-0.1937639  -0.27837322]
 [-0.19607023 -0.27685524]
 [-0.19841189 -0.27534012]
 [-0.20078911 -0.27382786]
 [-0.20320209 -0.27231849]
 [-0.20565105 -0.27081203]
 [-0.20813616 -0.26930851]
 [-0.21065761 -0.26780797]
 [-0.21321556 -0.26631043]
 [-0.21581016 -0.26481595]
 [-0.21844153 -0.26332455]
 [-0.2211098  -0.2618363 ]
 [-0.22381505 -0.26035124]
 [-0.22655737 -0.25886943]
 [-0.22933681 -0.25739093]
 [-0.23215342 -0.25591582]
 [-0.23500722 -0.25444415]
 [-0.23789819 -0.25297602]
 [-0.24082631 -0.2515115 ]
 [-0.24379154 -0.25005068]
 [-0.2467938  -0.24859365]
 [-0.24983298 -0.24714052]
 [-0.25290896 -0.24569138]
 [-0.25602159 -0.24424636]
 [-0.25917068 -0.24280555]
 [-0.26235602 -0.2413691 ]
 [-0.26557737 -0.23993712]]
Best value found: x* = [-0.26557737 -0.23993712] with f(x*) = 0.11439490253855422
Optimizer trajectory:
[[-1.50000000e+00  2.50000000e-01]
 [ 1.05500000e+01  4.25000000e+00]
 [-4.50725650e+03  2.18355000e+02]
 [ 3.66262233e+11  4.06305040e+07]]
Best value found: x* = [3.66262233e+11 4.06305040e+07] with f(x*) = 1.799569223005
2147e+48
```

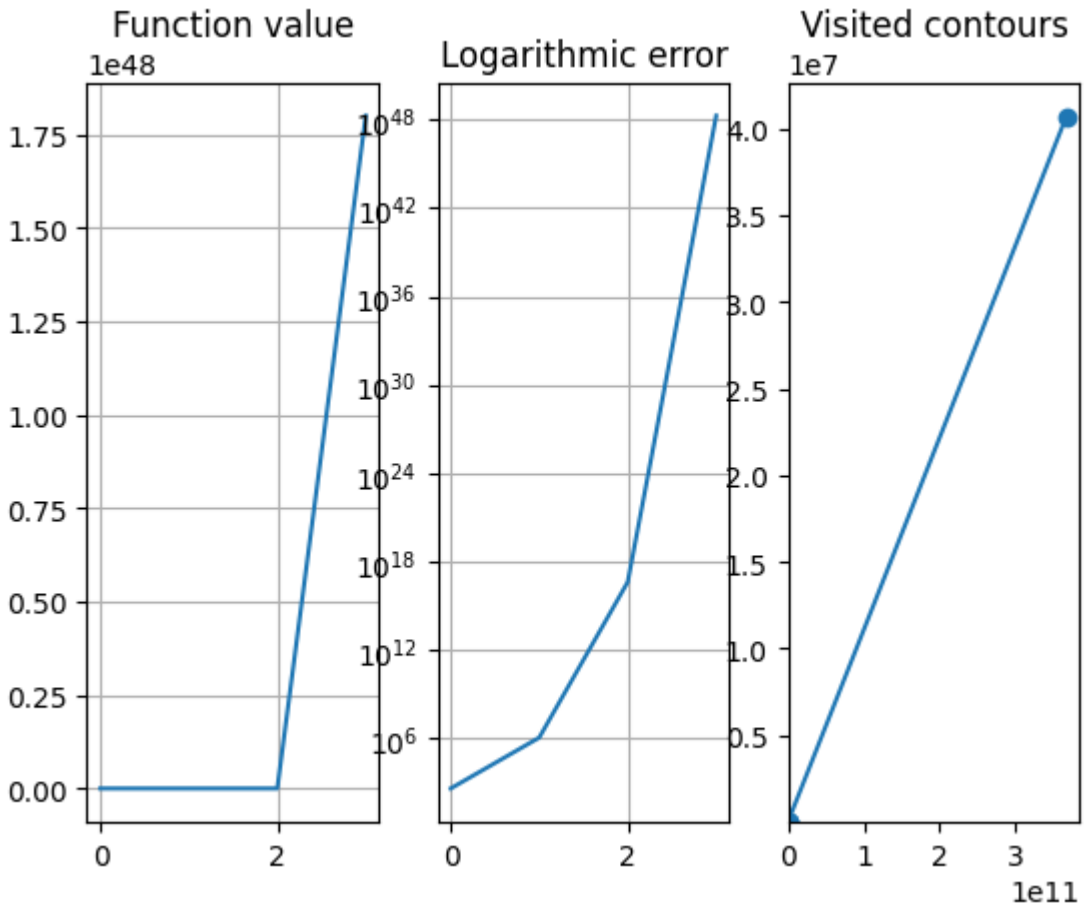

17500
15000
12500
10000
7500
5000
2500
−40
−20
0
20
40
40
20
0
−20
−40


Function value
Logarithmic error
Visited contours
4000
3500
3000
2500
2000
1500
1000
500
0
10^3
10^2
0
20
40
40
20
0
−20
−40
−50
0
50


0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5


Function value
0.17
0.16
0.15
0.14
0.13
0.12
0
25
50
75
100
Visited contours
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1
0
1





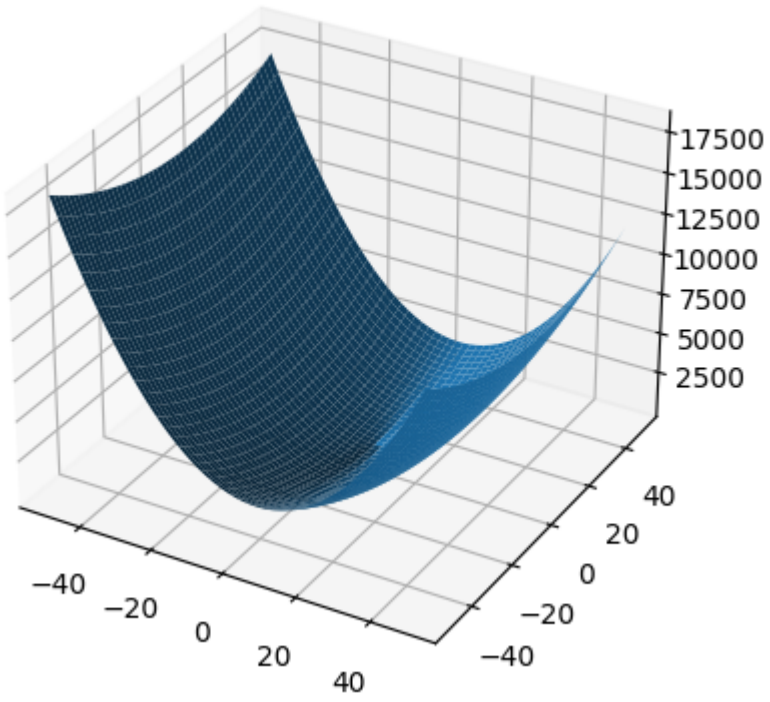
In [32]:
```
test_linear_search(fixed_step_search(0.001))
```

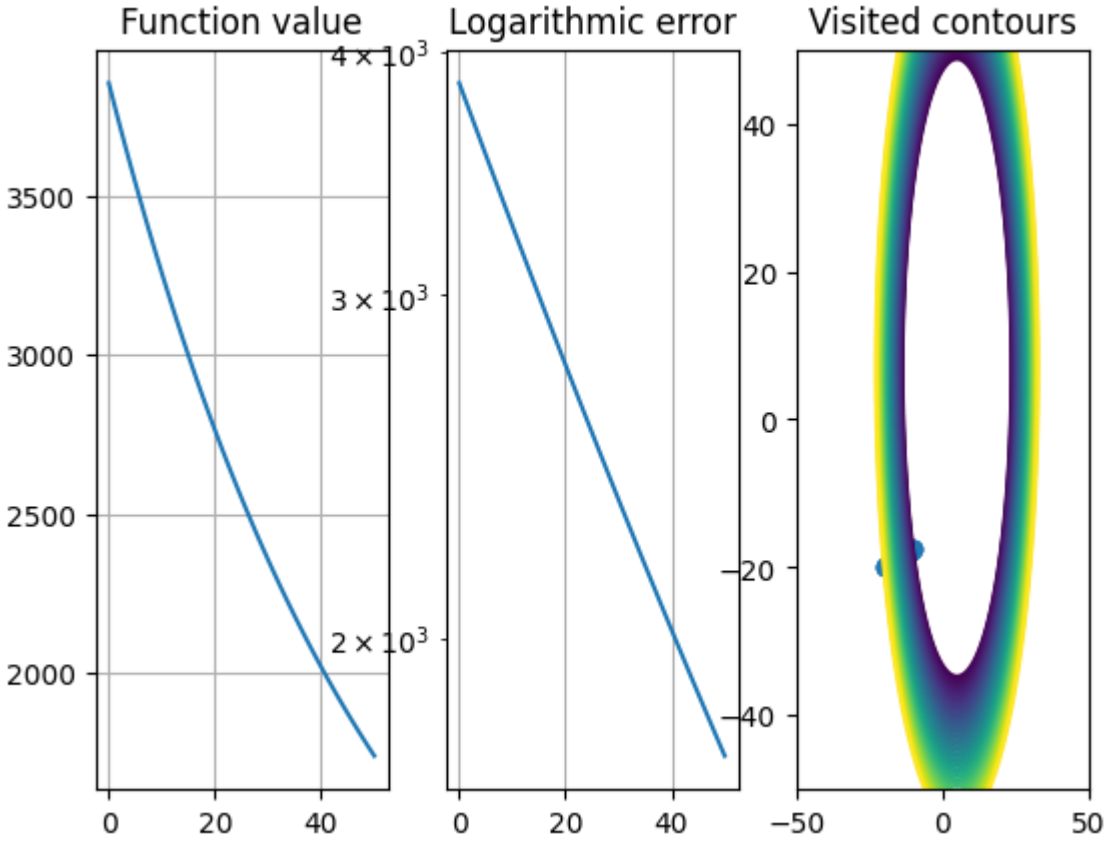
```
Optimizer trajectory:
[[-20.         -20.        ]
 [-19.75       -19.946     ]
 [-19.5025     -19.892108  ]
 [-19.257475   -19.83832378]
 [-19.01490025 -19.78464714]
 [-18.77475125 -19.73107784]
 [-18.53700374 -19.67761569]
 [-18.3016337  -19.62426046]
 [-18.06861736 -19.57101193]
 [-17.83793119 -19.51786991]
 [-17.60955188 -19.46483417]
 [-17.38345636 -19.4119045 ]
 [-17.15962179 -19.35908069]
 [-16.93802557 -19.30636253]
 [-16.71864532 -19.25374981]
 [-16.50145887 -19.20124231]
 [-16.28644428 -19.14883982]
 [-16.07357983 -19.09654214]
 [-15.86284404 -19.04434906]
 [-15.6542156  -18.99226036]
 [-15.44767344 -18.94027584]
 [-15.24319671 -18.88839529]
 [-15.04076474 -18.8366185 ]
 [-14.84035709 -18.78494526]
 [-14.64195352 -18.73337537]
 [-14.44553398 -18.68190862]
 [-14.25107865 -18.6305448 ]
 [-14.05856786 -18.57928371]
 [-13.86798218 -18.52812514]
 [-13.67930236 -18.47706889]
 [-13.49250933 -18.42611476]
 [-13.30758424 -18.37526253]
 [-13.1245084  -18.324512  ]
 [-12.94326331 -18.27386298]
 [-12.76383068 -18.22331525]
 [-12.58619237 -18.17286862]
 [-12.41033045 -18.12252288]
 [-12.23622715 -18.07227784]
 [-12.06386488 -18.02213328]
 [-11.89322623 -17.97208902]
 [-11.72429396 -17.92214484]
 [-11.55705102 -17.87230055]
 [-11.39148051 -17.82255595]
 [-11.22756571 -17.77291084]
 [-11.06529005 -17.72336501]
 [-10.90463715 -17.67391828]
 [-10.74559078 -17.62457045]
 [-10.58813487 -17.57532131]
 [-10.43225352 -17.52617066]
 [-10.27793099 -17.47711832]
 [-10.12515168 -17.42816409]]
Best value found: x* = [-10.12515168 -17.42816409] with f(x*) = 1740.5862670967067
Optimizer trajectory:
[[-0.1        -0.4       ]
 [-0.10005218 -0.39981999]
```

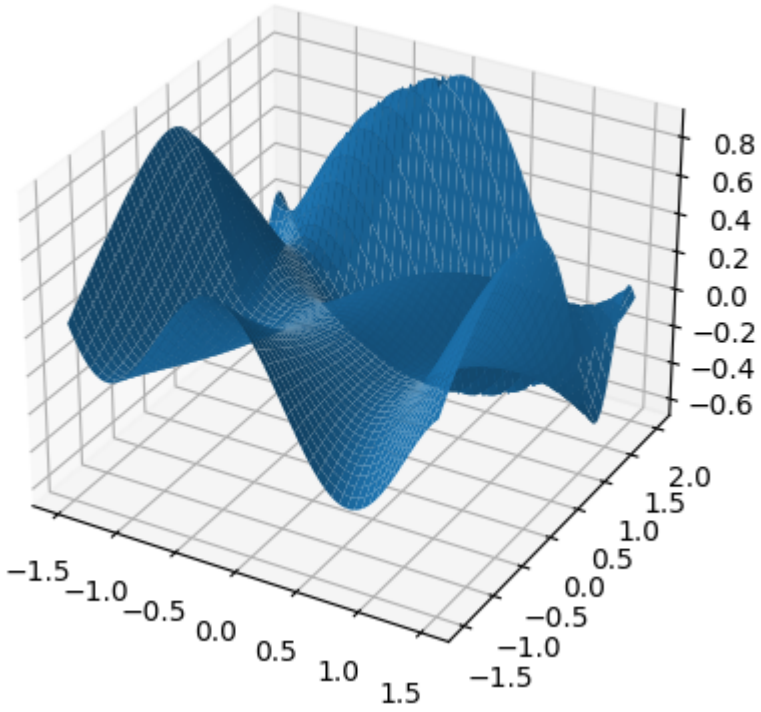
```
[-0.10010451 -0.39964004]
[-0.10015698 -0.39946014]
[-0.10020959 -0.39928029]
[-0.10026235 -0.3991005 ]
[-0.10031525 -0.39892076]
[-0.1003683  -0.39874107]
[-0.10042149 -0.39856143]
[-0.10047482 -0.39838185]
[-0.1005283  -0.39820233]
[-0.10058193 -0.39802285]
[-0.1006357  -0.39784343]
[-0.10068961 -0.39766407]
[-0.10074368 -0.39748475]
[-0.10079789 -0.39730549]
[-0.10085224 -0.39712628]
[-0.10090674 -0.39694713]
[-0.10096139 -0.39676802]
[-0.10101618 -0.39658897]
[-0.10107112 -0.39640998]
[-0.10112621 -0.39623103]
[-0.10118145 -0.39605214]
[-0.10123683 -0.3958733 ]
[-0.10129237 -0.39569452]
[-0.10134805 -0.39551578]
[-0.10140387 -0.3953371 ]
[-0.10145985 -0.39515847]
[-0.10151598 -0.3949799 ]
[-0.10157225 -0.39480137]
[-0.10162868 -0.3946229 ]
[-0.10168525 -0.39444448]
[-0.10174197 -0.39426612]
[-0.10179884 -0.3940878 ]
[-0.10185587 -0.39390954]
[-0.10191304 -0.39373133]
[-0.10197036 -0.39355317]
[-0.10202783 -0.39337507]
[-0.10208546 -0.39319701]
[-0.10214323 -0.39301901]
[-0.10220116 -0.39284106]
[-0.10225924 -0.39266316]
[-0.10231747 -0.39248531]
[-0.10237585 -0.39230752]
[-0.10243438 -0.39212977]
[-0.10249307 -0.39195208]
[-0.10255191 -0.39177444]
[-0.1026109  -0.39159685]
[-0.10267004 -0.39141931]
[-0.10272934 -0.39124183]
[-0.10278879 -0.39106439]
[-0.10284839 -0.39088701]
[-0.10290815 -0.39070968]
[-0.10296806 -0.3905324 ]
[-0.10302813 -0.39035517]
[-0.10308835 -0.39017799]
[-0.10314873 -0.39000086]
[-0.10320926 -0.38982379]
```

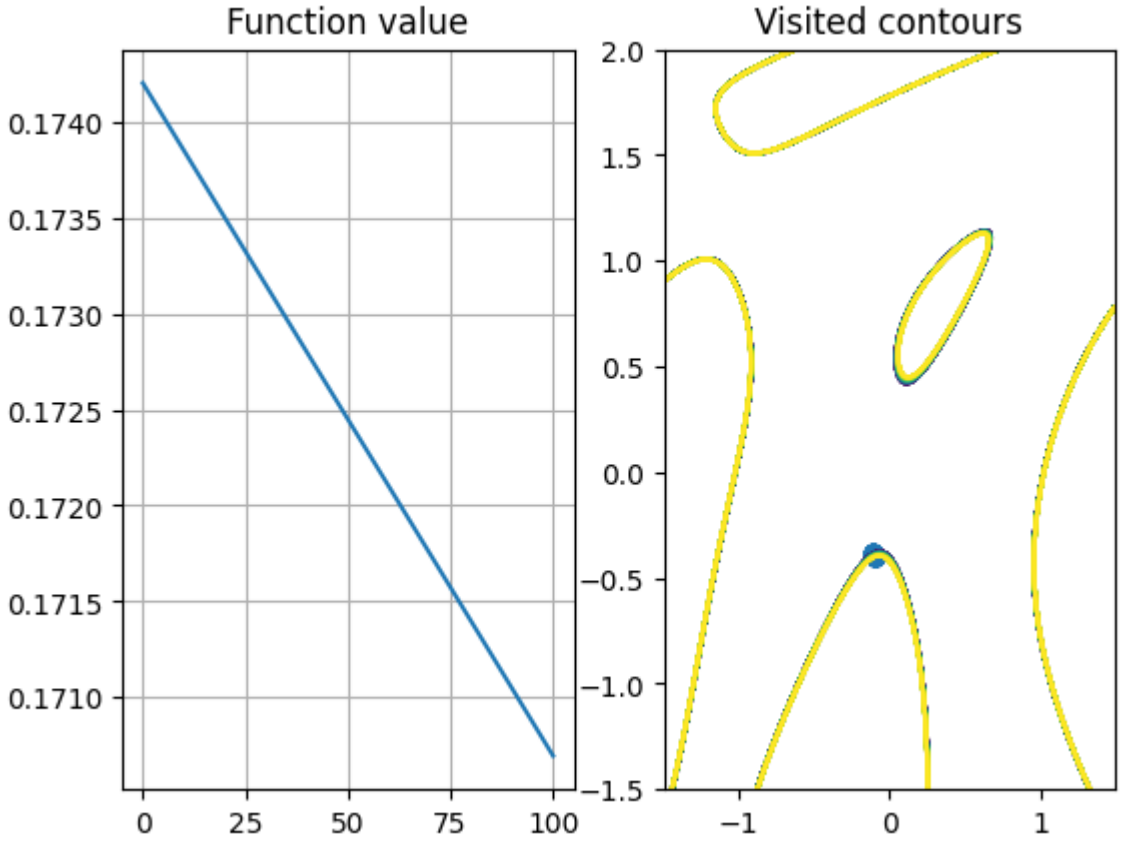
```
 [-0.10326994 -0.38964676]
 [-0.10333078 -0.38946979]
 [-0.10339178 -0.38929287]
 [-0.10345293 -0.38911599]
 [-0.10351424 -0.38893917]
 [-0.10357571 -0.3887624 ]
 [-0.10363733 -0.38858568]
 [-0.1036991  -0.38840901]
 [-0.10376104 -0.38823239]
 [-0.10382313 -0.38805582]
 [-0.10388538 -0.38787931]
 [-0.10394778 -0.38770284]
 [-0.10401035 -0.38752642]
 [-0.10407307 -0.38735006]
 [-0.10413595 -0.38717374]
 [-0.10419899 -0.38699748]
 [-0.10426219 -0.38682126]
 [-0.10432554 -0.3866451 ]
 [-0.10438906 -0.38646898]
 [-0.10445273 -0.38629292]
 [-0.10451657 -0.3861169 ]
 [-0.10458056 -0.38594094]
 [-0.10464471 -0.38576502]
 [-0.10470903 -0.38558916]
 [-0.1047735  -0.38541334]
 [-0.10483814 -0.38523758]
 [-0.10490293 -0.38506186]
 [-0.10496789 -0.3848862 ]
 [-0.10503301 -0.38471058]
 [-0.10509829 -0.38453501]
 [-0.10516373 -0.3843595 ]
 [-0.10522933 -0.38418403]
 [-0.1052951  -0.38400861]
 [-0.10536103 -0.38383324]
 [-0.10542712 -0.38365793]
 [-0.10549337 -0.38348266]
 [-0.10555979 -0.38330744]
 [-0.10562637 -0.38313227]
 [-0.10569312 -0.38295714]
 [-0.10576002 -0.38278207]
 [-0.1058271  -0.38260705]
 [-0.10589433 -0.38243207]
 [-0.10596174 -0.38225715]]
Best value found: x* = [-0.10596174 -0.38225715] with f(x*) = 0.17069544613148885
Optimizer trajectory:
[[-1.5         0.25      ]
 [-0.295       0.65      ]
 [-0.35884105  0.537405  ]
 [-0.41477782  0.45567738]
 [-0.45900675  0.39895003]
 [-0.4906543   0.36129747]
 [-0.51133349  0.3371863 ]
 [-0.52379898  0.32204143]
 [-0.53074045  0.31250622]
 [-0.53422211  0.30634206]
 [-0.53563019  0.3021523 ]
```

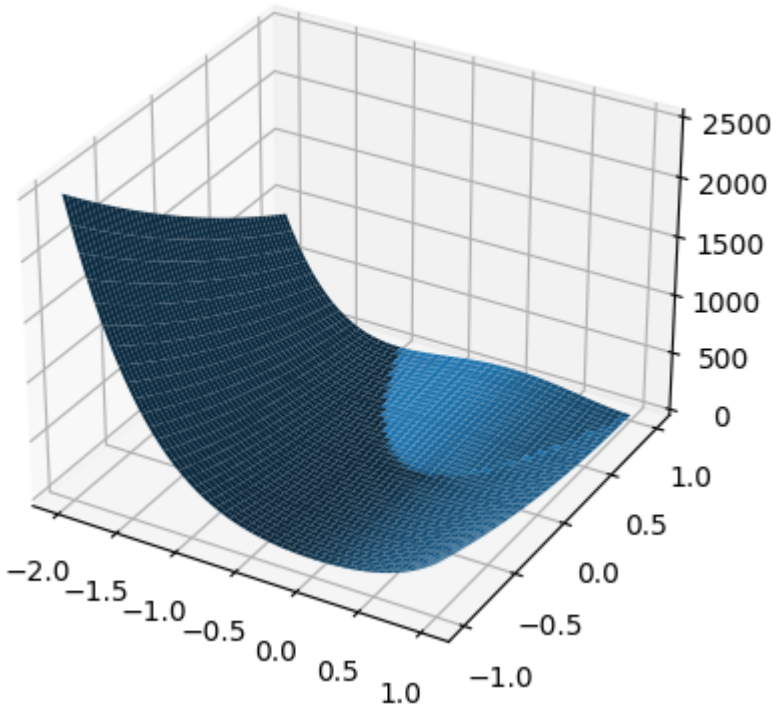
```
 [-0.53582683  0.29910178]
 [-0.5353253   0.2967035 ]
 [-0.53442386  0.29467744]
 [-0.53329359  0.29286372]
 [-0.53203203  0.29117139]
 [-0.53069458  0.28954873]
 [-0.52931273  0.28796633]
 [-0.52790437  0.28640746]
 [-0.52647966  0.28486257]
 [-0.52504441  0.28332622]]
Best value found: x* = [-0.52504441  0.28332622] with f(x*) = 2.3316197339595957
```

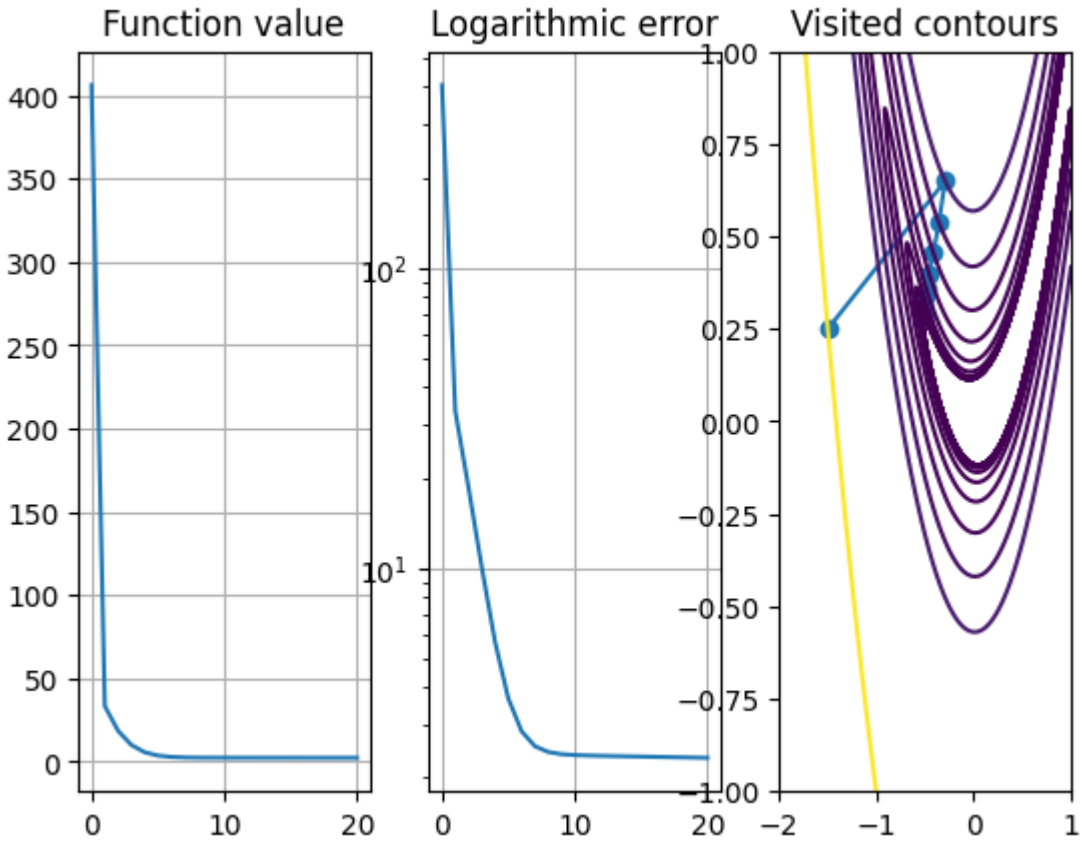



Function value
Logarithmic error
Visited contours
4 × 10^3
3 × 10^3
2 × 10^3
3500
3000
2500
2000
0
20
40
40
20
0
−20
−40
−50
0
50


0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5


Function value
0.1740
0.1735
0.1730
0.1725
0.1720
0.1715
0.1710
0
25
50
75
100
Visited contours
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1
0
1


2500
2000
1500
1000
500
0
1.0
0.5
0.0
−0.5
−1.0
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0



Другие методы одномерного поиска

Метод дихотомии (двоичного поиска)

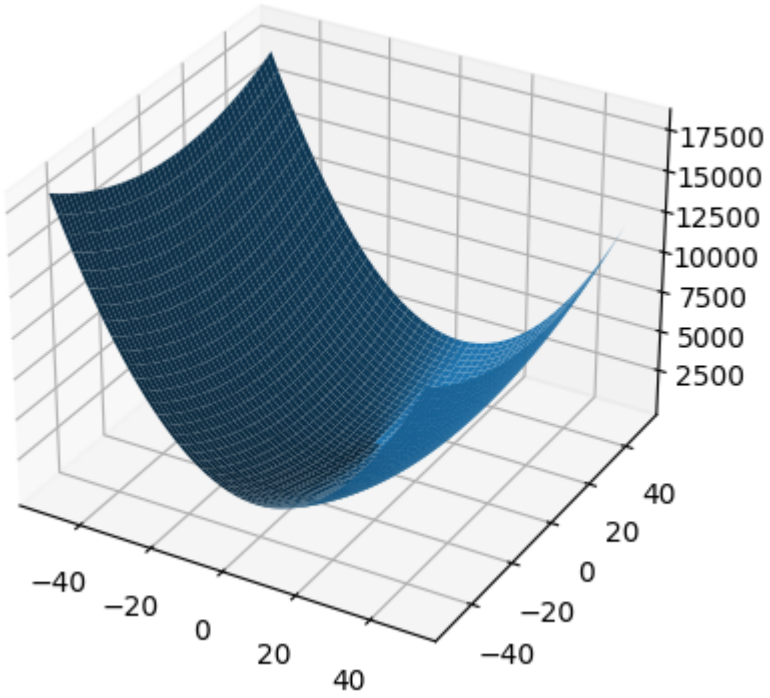
Метод дихотомии подходит для нахождения точного минимума унимодальной функции. В его реализации требуется зафиксировать отрезок, внутри которого обязан лежать минимум, а затем делить его пополам, выбирая для следующей итерации правую либо левую часть, в зависимости от значения производной в середине. Сужение заканчивается, когда длина полученного отрезка становится меньше заранее заданного ε. Для выбора размера изначального отрезка будем постепенно увеличивать длину отрезка, ожидая, что после какого-то из увеличений значение функции на конце увеличится, а значит минимум оказался внутри.

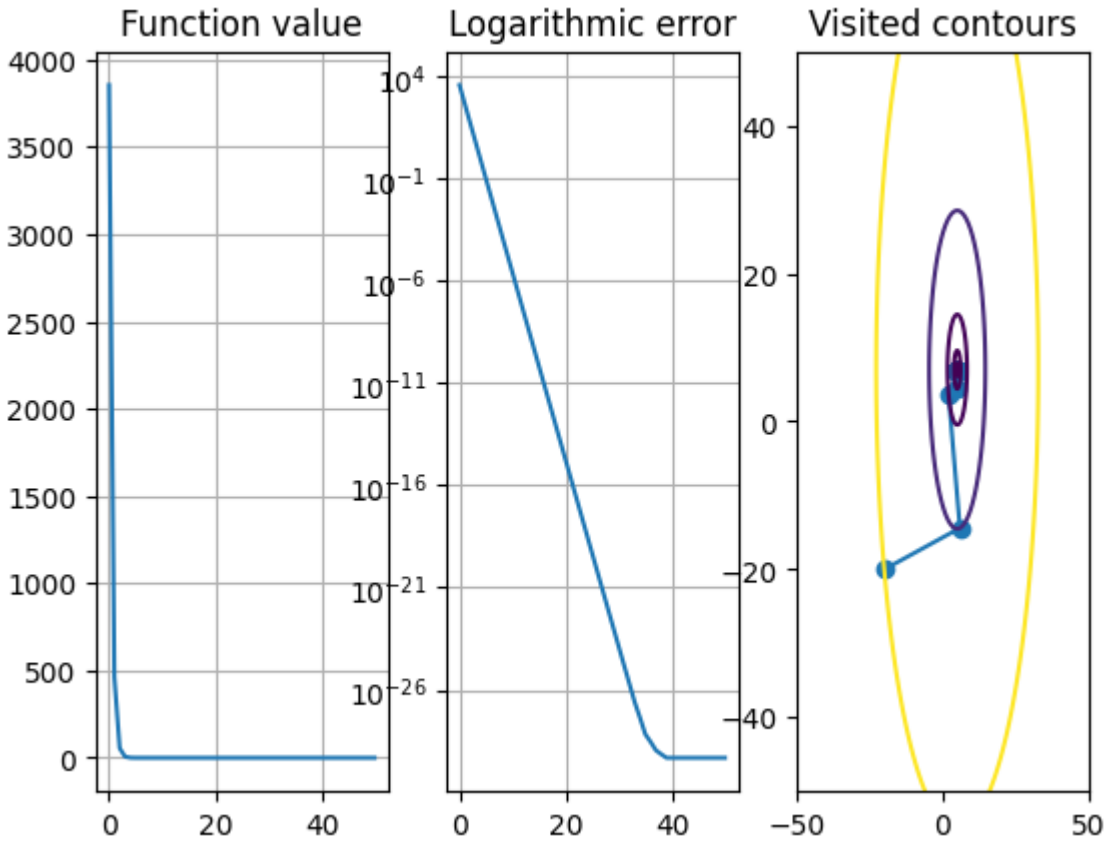
In [33]:
```
test_linear_search(bin_search)
```

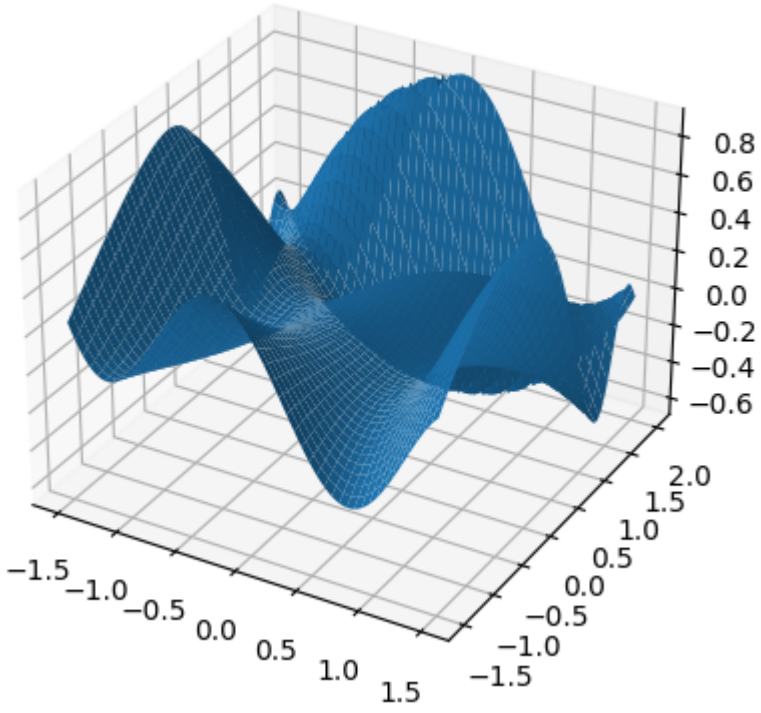
```
Optimizer trajectory:
[[-20.         -20.        ]
 [  5.92468262 -14.40026855]
 [  2.00110257   3.76070154]
 [  5.11114993   4.43257324]
 [  4.63978542   6.61016138]
 [  5.01337833   6.69102482]
 [  4.95664658   6.953071  ]
 [  5.00161345   6.9628061 ]
 [  4.99477508   6.99433431]
 [  5.00019485   6.99550971]
 [  4.99936909   6.99931559]
 [  5.00002358   6.99945759]
 [  4.99992371   6.99991711]
 [  5.00000286   6.99993431]
 [  4.99999076   6.99998995]
 [  5.00000035   6.99999204]
 [  4.99999888   6.99999878]
 [  5.00000004   6.99999903]
 [  4.99999986   6.99999985]
 [  5.00000001   6.99999988]
 [  4.99999998   6.99999998]
 [  5.           6.99999999]
 [  5.           7.        ]
 [  5.           7.        ]
 [  5.           7.        ]
 [  5.           7.        ]
 [  5.           7.        ]
 [  5.           7.        ]
 [  5.           7.        ]
 [  5.           7.        ]
 [  5.           7.        ]
 [  5.           7.        ]
 [  5.           7.        ]
 [  5.           7.        ]
 [  5.           7.        ]
 [  5.           7.        ]
 [  5.           7.        ]
 [  5.           7.        ]
 [  5.           7.        ]
 [  5.           7.        ]
 [  5.           7.        ]
 [  5.           7.        ]
 [  5.           7.        ]
 [  5.           7.        ]
 [  5.           7.        ]
 [  5.           7.        ]
 [  5.           7.        ]
 [  5.           7.        ]
 [  5.           7.        ]
 [  5.           7.        ]
 [  5.           7.        ]
 [  5.           7.        ]
 [  5.           7.        ]]
Best value found: x* = [5. 7.] with f(x*) = 4.733165431326071e-30
Optimizer trajectory:
[[-0.1        -0.4       ]
 [-0.5636805   1.19950929]
```

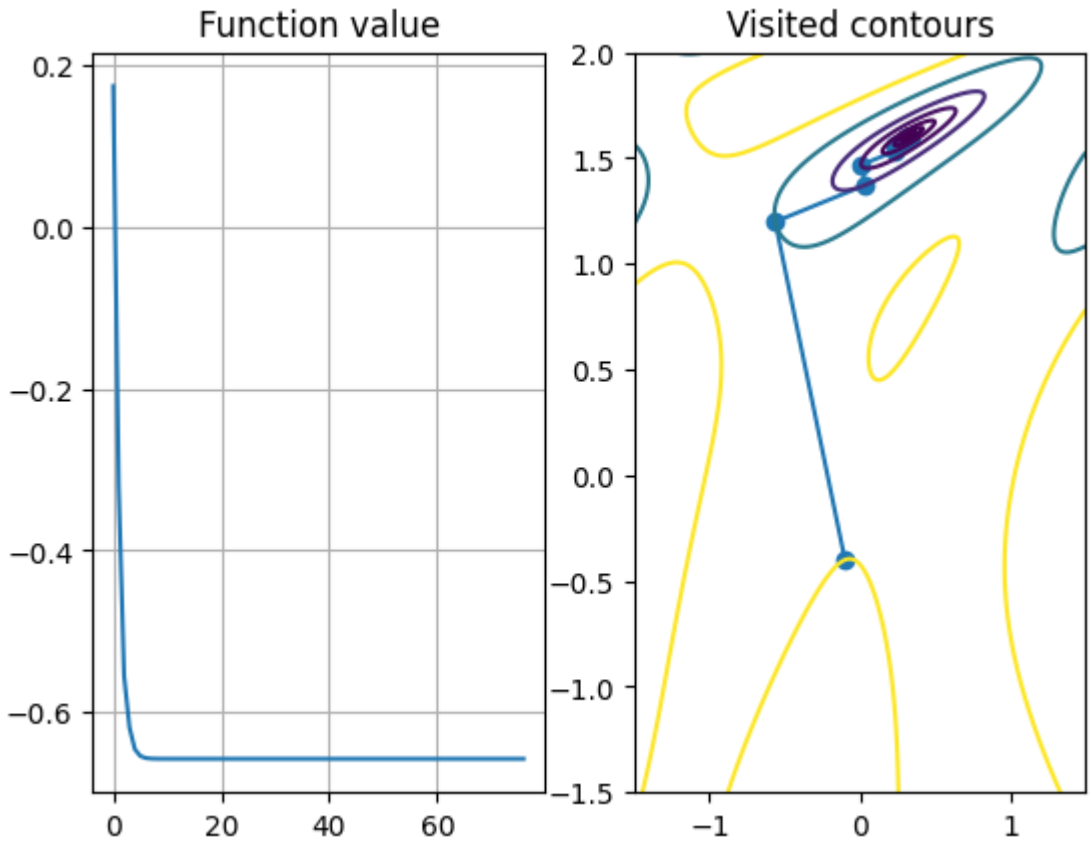
```
[ 0.03595589  1.37333543]
[ 0.00845876  1.4681904 ]
[ 0.23547275  1.5339512 ]
[ 0.22711564  1.56280076]
[ 0.29885225  1.5835532 ]
[ 0.29651168  1.59164401]
[ 0.31638642  1.59738468]
[ 0.31576078  1.59955067]
[ 0.32104151  1.60106972]
[ 0.32087604  1.60164495]
[ 0.32227988  1.60204663]
[ 0.32223569  1.60220107]
[ 0.32261557  1.60230944]
[ 0.32260358  1.60235149]
[ 0.32270715  1.60238091]
[ 0.32270386  1.60239249]
[ 0.32273263  1.60240066]
[ 0.32273171  1.60240388]
[ 0.32273972  1.60240615]
[ 0.32273946  1.60240705]
[ 0.32274168  1.60240767]
[ 0.32274161  1.60240792]
[ 0.32274224  1.6024081 ]
[ 0.32274222  1.60240817]
[ 0.3227424   1.60240822]
[ 0.32274239  1.60240824]
[ 0.32274244  1.60240826]
[ 0.32274244  1.60240826]
[ 0.32274246  1.60240827]
[ 0.32274246  1.60240827]
[ 0.32274246  1.60240827]
[ 0.32274246  1.60240827]
[ 0.32274246  1.60240827]
[ 0.32274246  1.60240827]
[ 0.32274246  1.60240827]
[ 0.32274246  1.60240827]
[ 0.32274246  1.60240827]
[ 0.32274246  1.60240827]
[ 0.32274246  1.60240827]
[ 0.32274246  1.60240827]
[ 0.32274246  1.60240827]
[ 0.32274246  1.60240827]
[ 0.32274246  1.60240827]
[ 0.32274246  1.60240827]
[ 0.32274246  1.60240827]
[ 0.32274246  1.60240827]
[ 0.32274246  1.60240827]
[ 0.32274246  1.60240827]
[ 0.32274246  1.60240827]
[ 0.32274246  1.60240827]
[ 0.32274246  1.60240827]
[ 0.32274246  1.60240827]
[ 0.32274246  1.60240827]
[ 0.32274246  1.60240827]
[ 0.32274246  1.60240827]
[ 0.32274246  1.60240827]
```

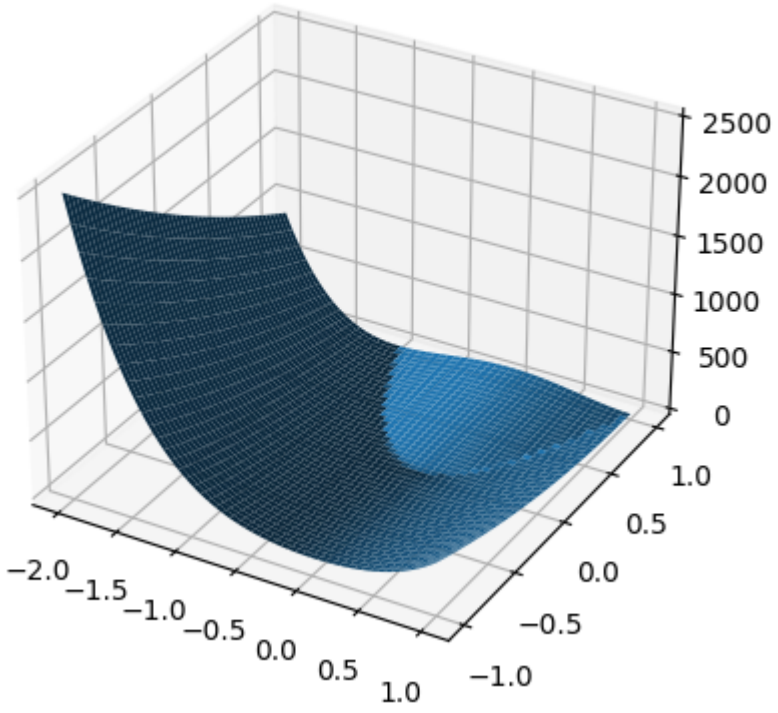
```
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240827]]
Best value found: x* = [0.32274246 1.60240827] with f(x*) = -0.6574000294758535
Optimizer trajectory:
[[-1.5         0.25      ]
 [ 1.04657745  1.09533691]
 [ 1.04647418  1.09533397]
 [ 1.04648095  1.09513726]
 [ 1.04637838  1.09513374]
 [ 1.04638623  1.0949413 ]
 [ 1.04628436  1.09493722]
 [ 1.04629318  1.09474906]
 [ 1.04619213  1.09474436]
 [ 1.04620174  1.09455954]
 [ 1.04610118  1.09455437]
 [ 1.04611175  1.09437381]
 [ 1.04601207  1.09436798]
 [ 1.04602325  1.0941898 ]
 [ 1.04592374  1.09418366]
 [ 1.04593597  1.09400998]
 [ 1.04583747  1.09400312]
 [ 1.04585007  1.09383124]
 [ 1.04575166  1.09382411]
 [ 1.04576526  1.09365639]
 [ 1.04566777  1.09364858]]
Best value found: x* = [1.04566777 1.09364858] with f(x*) = 0.0020907203681311123
```

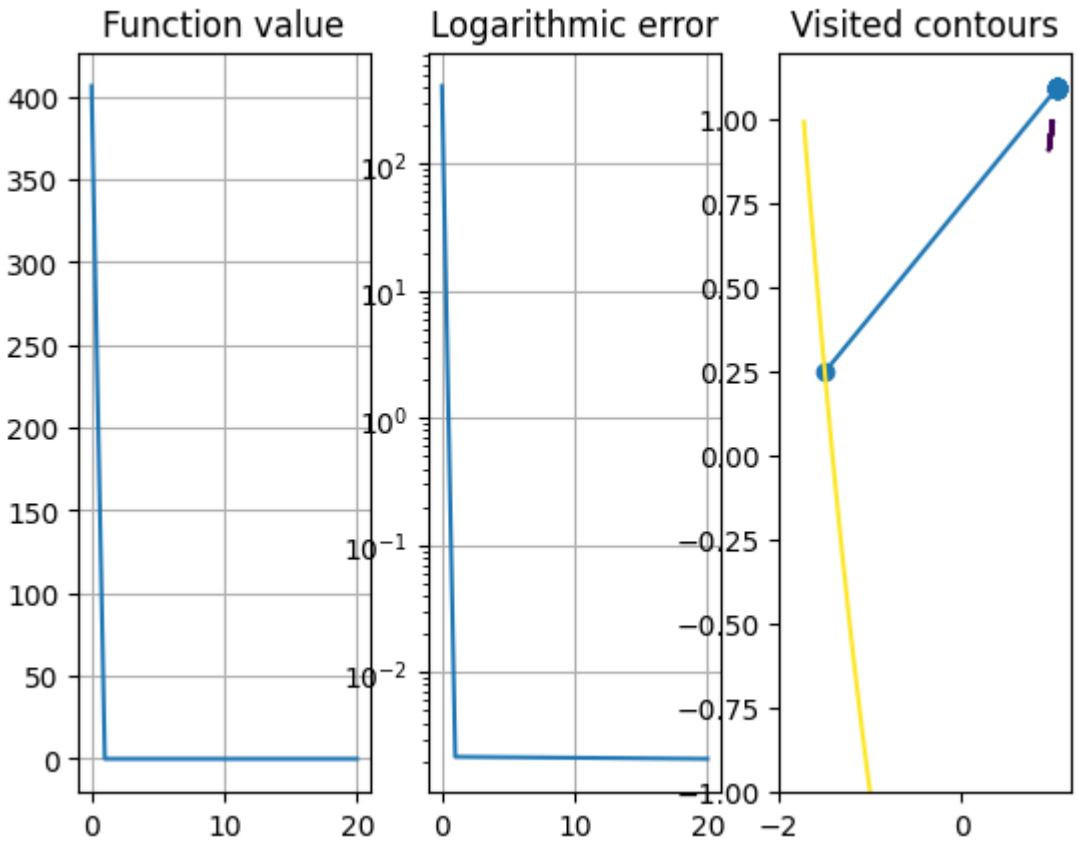

17500
15000
12500
10000
7500
5000
2500
−40
−20
0
20
40
40
20
0
−20
−40


Function value
Logarithmic error
Visited contours
4000
3500
3000
2500
2000
1500
1000
500
0
0
20
40
10^{4}
10^{-1}
10^{-6}
10^{-11}
10^{-16}
10^{-21}
10^{-26}
0
20
40
40
20
0
−20
−40
−50
0
50


0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5


Function value
0.2
0.0
−0.2
−0.4
−0.6
0
20
40
60
Visited contours
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1
0
1





Метод золотого сечения

Метод золотого сечения также используется для поиска минимума унимодальной функции, однако в отличии от метода дихотомии он не требует вычисления

производной. Зафиксируем отрезок, в котором обязан лежать минимум. Затем будет сужать его с помощью следующей процедуры: фиксируются две точки $x_1, x_2 : \frac{b-a}{x_2-a} = \frac{b-a}{b-x_1} = \varphi$, затем из отрезка удаляется $[a, x_1]$ либо $[x_2, b]$ в зависимости от того, какое из значений $\{f(x_1), f(x_2)\}$ больше.

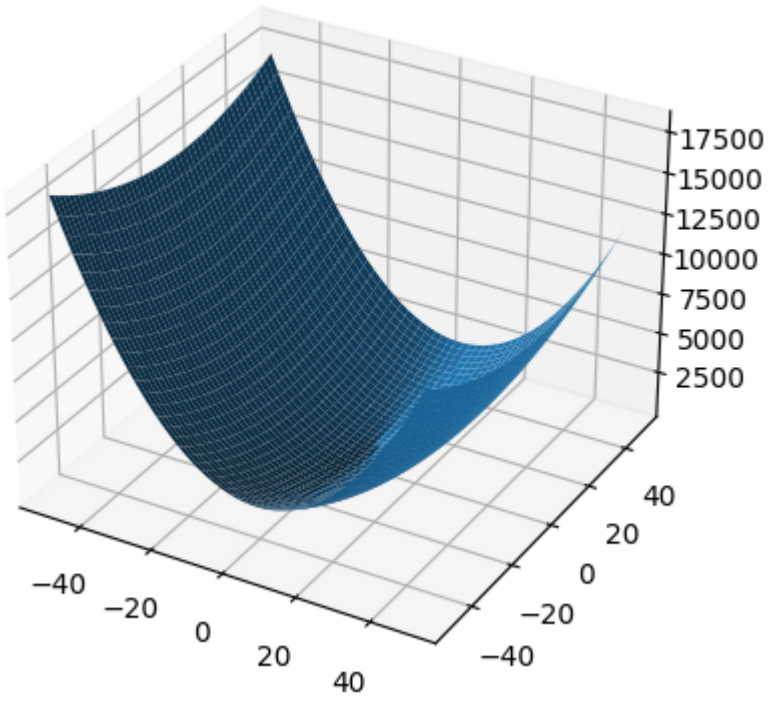
In [34]:
```
test_linear_search(golden_ratio_search)
```

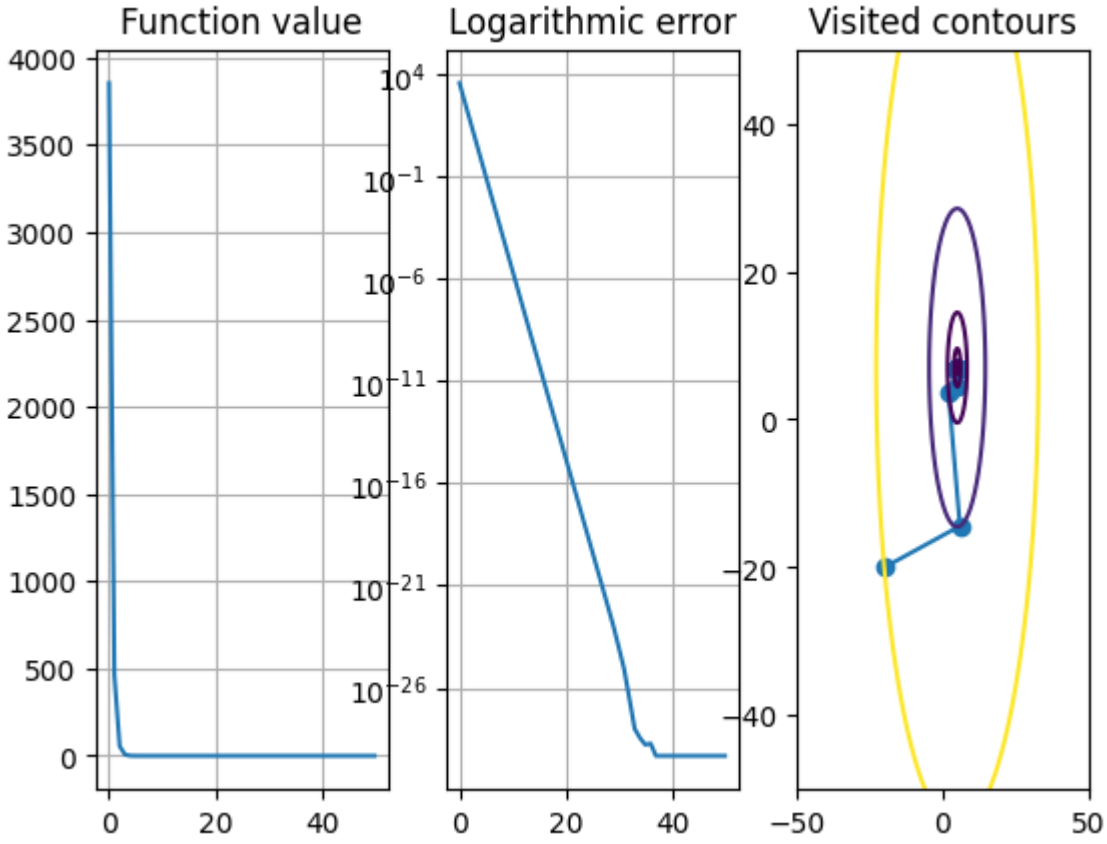
```
Optimizer trajectory:
[[-20.         -20.        ]
 [  5.92572832 -14.40004268]
 [  1.99900415   3.75475789]
 [  5.11141372   4.42790261]
 [  4.63900473   6.60910968]
 [  5.01343698   6.69019769]
 [  4.95648437   6.95281704]
 [  5.00162394   6.96260579]
 [  4.99474334   6.99429352]
 [  5.00019648   6.99547748]
 [  4.99936408   6.99930939]
 [  5.00002381   6.99945268]
 [  4.99992299   6.99991625]
 [  5.00000289   6.99993363]
 [  4.99999065   6.99998982]
 [  5.00000035   6.99999194]
 [  4.99999886   6.99999876]
 [  5.00000004   6.99999902]
 [  4.99999986   6.99999985]
 [  5.00000001   6.99999988]
 [  4.99999998   6.99999998]
 [  5.           6.99999999]
 [  5.           7.        ]
 [  5.           7.        ]
 [  5.           7.        ]
 [  5.           7.        ]
 [  5.           7.        ]
 [  5.           7.        ]
 [  5.           7.        ]
 [  5.           7.        ]
 [  5.           7.        ]
 [  5.           7.        ]
 [  5.           7.        ]
 [  5.           7.        ]
 [  5.           7.        ]
 [  5.           7.        ]
 [  5.           7.        ]
 [  5.           7.        ]
 [  5.           7.        ]
 [  5.           7.        ]
 [  5.           7.        ]
 [  5.           7.        ]
 [  5.           7.        ]
 [  5.           7.        ]
 [  5.           7.        ]
 [  5.           7.        ]
 [  5.           7.        ]
 [  5.           7.        ]
 [  5.           7.        ]
 [  5.           7.        ]
 [  5.           7.        ]]
Best value found: x* = [5. 7.] with f(x*) = 4.733165431326071e-30
Optimizer trajectory:
[[-0.1        -0.4       ]
 [-0.56368051  1.19950934]
```

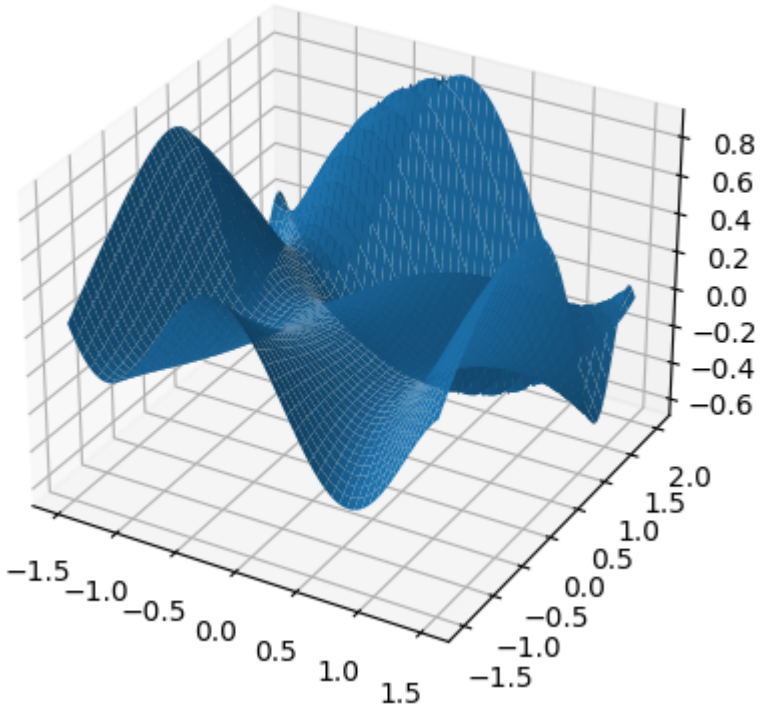
```
[ 0.03595618  1.37333534]
[ 0.00845879  1.46819102]
[ 0.23543539  1.53393197]
[ 0.22707823  1.56278568]
[ 0.29879142  1.58351911]
[ 0.29644971  1.59161863]
[ 0.31635853  1.59736611]
[ 0.31573135  1.59953863]
[ 0.32103331  1.60106355]
[ 0.32086715  1.60164128]
[ 0.32228117  1.60204669]
[ 0.32223678  1.60220151]
[ 0.32261605  1.60230975]
[ 0.32260408  1.6023517 ]
[ 0.32270755  1.60238115]
[ 0.32270427  1.60239266]
[ 0.32273239  1.60240056]
[ 0.32273148  1.60240379]
[ 0.322739    1.60240578]
[ 0.32273874  1.60240675]
[ 0.32274122  1.6024074 ]
[ 0.32274114  1.60240773]
[ 0.32274208  1.60240799]
[ 0.32274205  1.6024081 ]
[ 0.32274248  1.60240827]
[ 0.32274248  1.60240828]
[ 0.32274247  1.60240828]
[ 0.32274247  1.60240828]
[ 0.32274247  1.60240827]
[ 0.32274247  1.60240828]
[ 0.32274247  1.60240827]
[ 0.32274247  1.60240828]
[ 0.32274247  1.60240827]
[ 0.32274247  1.60240828]
[ 0.32274247  1.60240827]
[ 0.32274246  1.60240828]
[ 0.32274247  1.60240827]
[ 0.32274246  1.60240828]
[ 0.32274246  1.60240827]
[ 0.32274247  1.60240827]
[ 0.32274246  1.60240827]
[ 0.32274247  1.60240827]
[ 0.32274246  1.60240827]
[ 0.32274246  1.60240827]
[ 0.32274246  1.60240827]
[ 0.32274246  1.60240828]
[ 0.32274246  1.60240827]
[ 0.32274246  1.60240828]
[ 0.32274246  1.60240827]
[ 0.32274246  1.60240828]
[ 0.32274246  1.60240827]
[ 0.32274246  1.60240827]
[ 0.32274246  1.60240828]
[ 0.32274246  1.60240827]
[ 0.32274246  1.60240827]
[ 0.32274246  1.60240827]
```

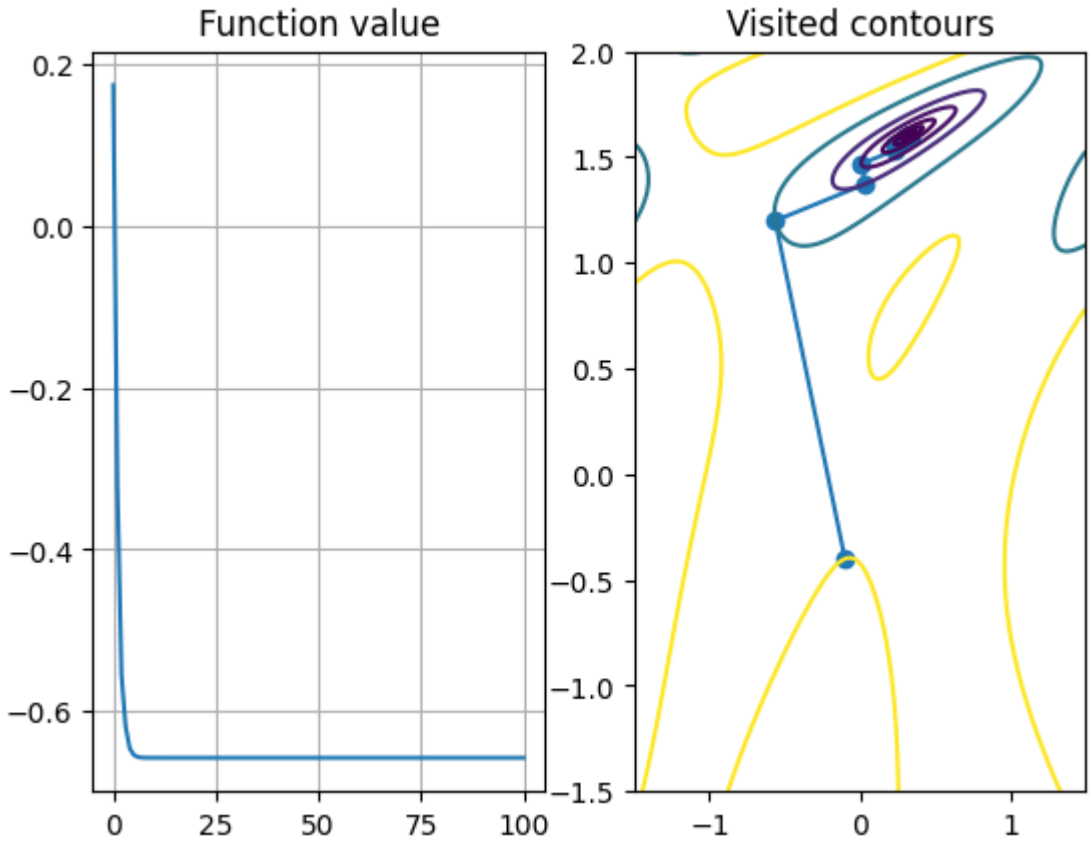
```
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240828]
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240828]
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240828]
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240828]
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240827]]
Best value found: x* = [0.32274246 1.60240827] with f(x*) = -0.6574000294758535
Optimizer trajectory:
[[-1.5         0.25      ]
 [ 1.05182803  1.09707984]
 [-1.50743603  2.28213741]
 [-1.50839278  2.28000877]
 [-1.50643635  2.27914039]
 [-1.50739817  2.27701551]
 [-1.50543739  2.27614247]
 [-1.5063989   2.27402693]
 [-1.50443823  2.27314396]
 [-1.50540341  2.27102652]
 [-1.5034416   2.27014395]
```

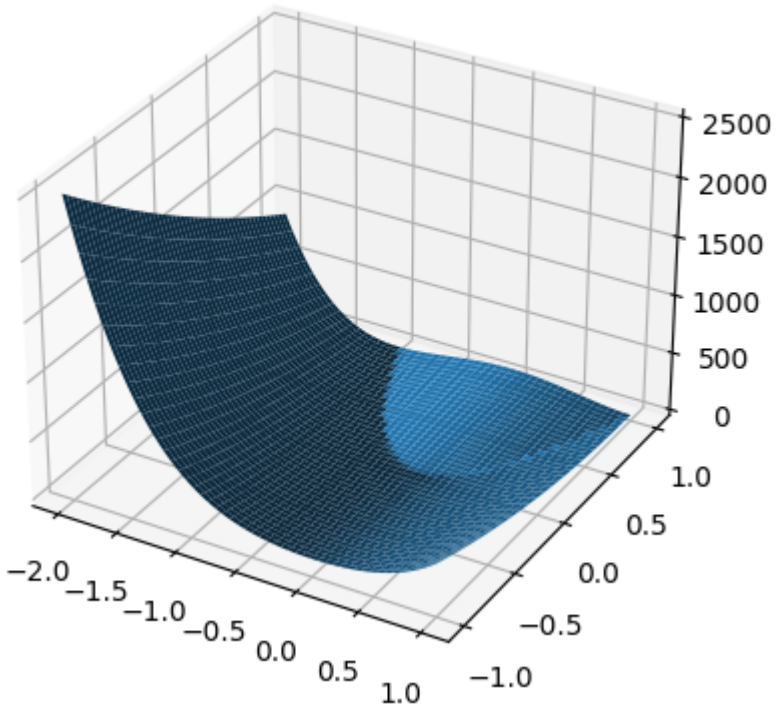
```
 [-1.50440324  2.26802699]
 [-1.50243968  2.26713891]
 [-1.50340352  2.26502052]
 [-1.50142766  2.26412662]
 [-1.50239859  2.26202124]
 [-1.50042219  2.26111763]
 [-1.501397    2.25901026]
 [-1.49941931  2.25810709]
 [-1.50039406  2.25599291]
 [-1.49841265  2.255091  ]]
Best value found: x* = [-1.49841265  2.255091  ] with f(x*) = 6.251769073080365
```

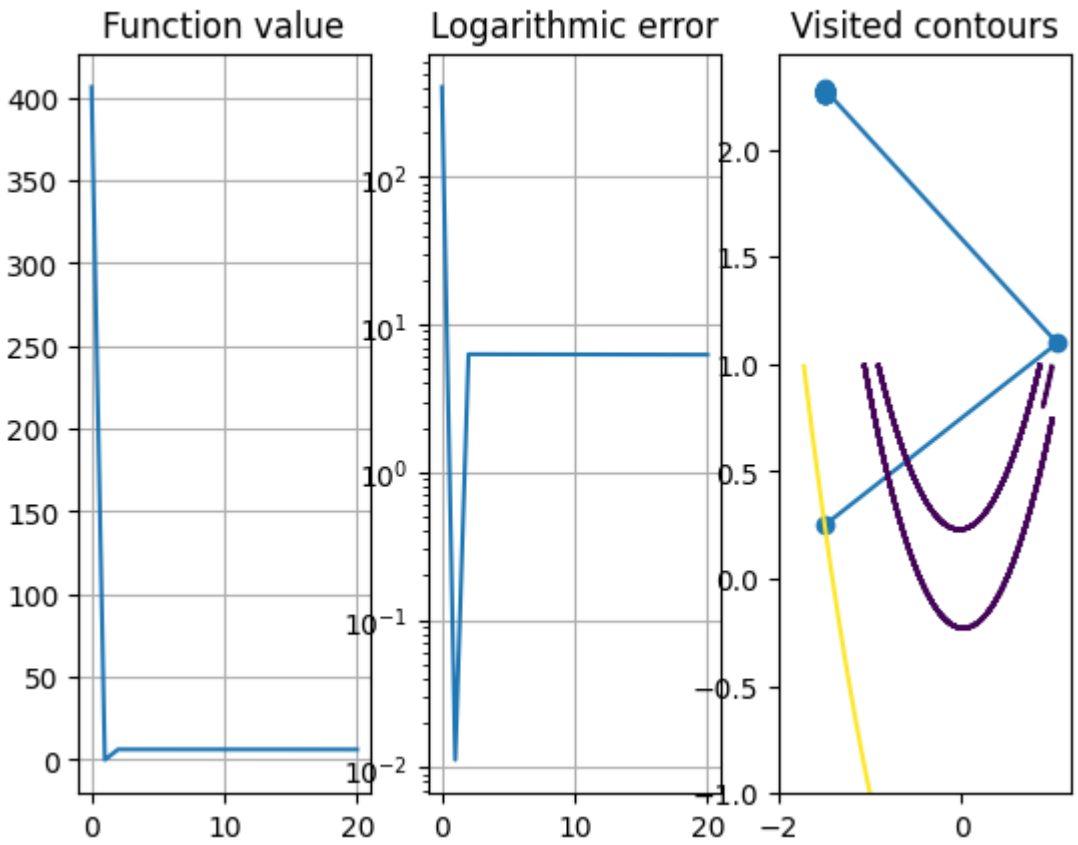



Function value
Logarithmic error
Visited contours
4000
3500
3000
2500
2000
1500
1000
500
0
0
20
40
10^{4}
10^{-1}
10^{-6}
10^{-11}
10^{-16}
10^{-21}
10^{-26}
0
20
40
40
20
0
−20
−40
−50
0
50


0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5


Function value
Visited contours
0.2
0.0
−0.2
−0.4
−0.6
0
25
50
75
100
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1
0
1


2500
2000
1500
1000
500
0
1.0
0.5
0.0
−0.5
−1.0
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0



Метод Фибоначчи

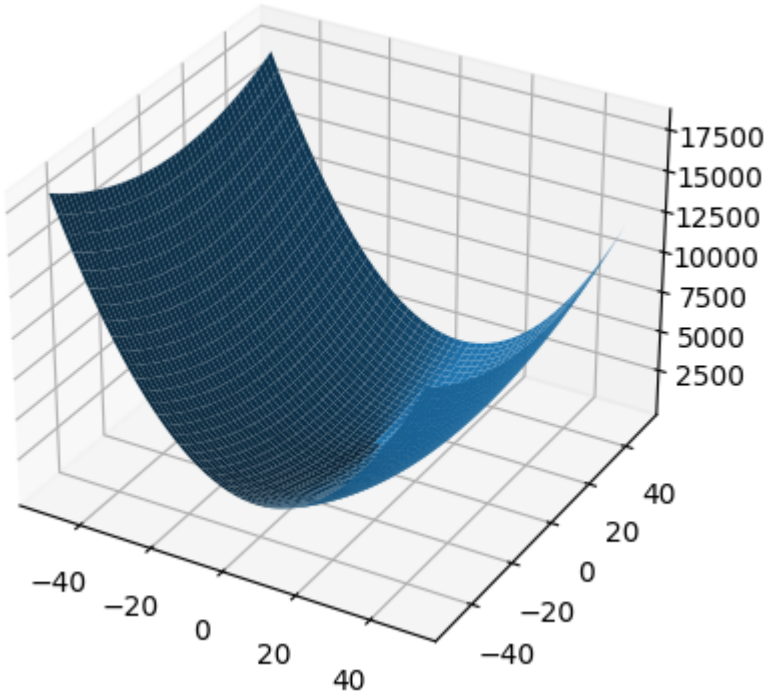
Метод Фибоначчи является модификацией метода золотого сечения, в котором x_1 и x_2 мы выбираем используя не точное значение φ, а отношение соседних чисел из последовательности Фибоначчи, а так же сужение проводится не до достижения требуемой точности, а количество итераций зафиксировано заранее. Строго говоря, на итерации алгоритма с номером k $\frac{b-a}{x_2-a} = \frac{F_{n-k+1}}{F_{n-k}}$, $\frac{b-a}{b-x_1} = \frac{F_{n-k+1}}{F_{n-k-1}}$, где n - количество итераций

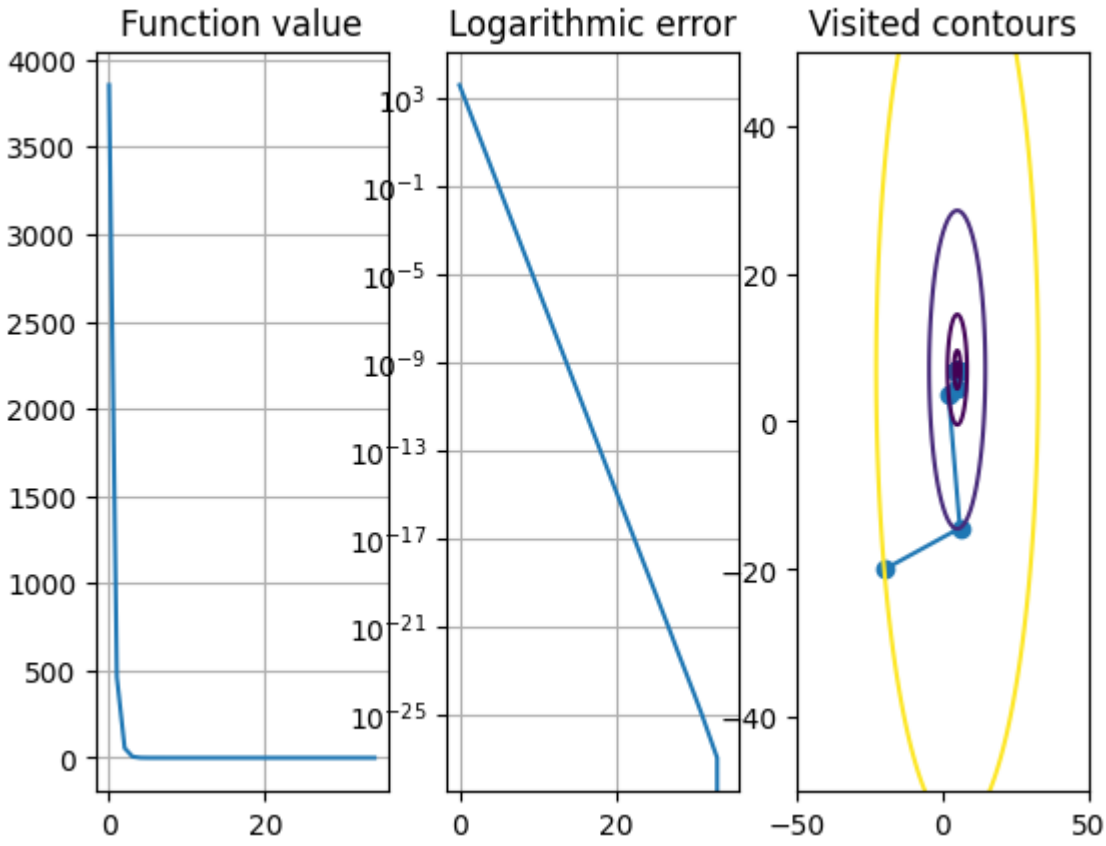
In [35]:
```
test_linear_search(fibonacci_search(50))
```

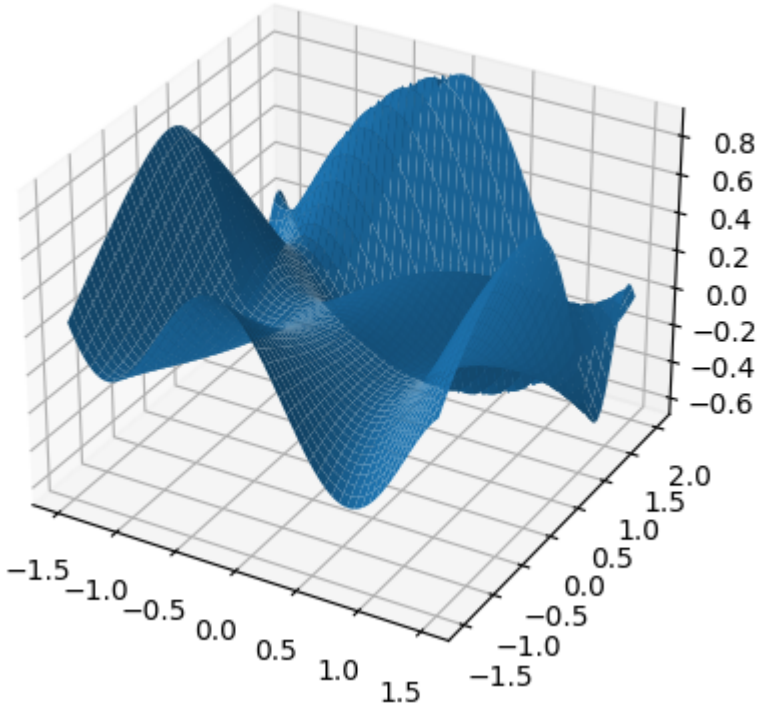
```
Optimizer trajectory:
[[-20.         -20.        ]
 [  5.92449333 -14.40030944]
 [  2.00151355   3.76163472]
 [  5.11088322   4.43325855]
 [  4.64036317   6.61159231]
 [  5.01329927   6.69214649]
 [  4.95686537   6.95341463]
 [  5.0015951    6.96307624]
 [  4.99482646   6.99441259]
 [  5.00019132   6.99557139]
 [  4.99937949   6.99932985]
 [  5.00002295   6.99946884]
 [  4.99992558   6.99991962]
 [  5.00000275   6.99993629]
 [  4.99999107   6.99999036]
 [  5.00000033   6.99999236]
 [  4.99999893   6.99999884]
 [  5.00000004   6.99999908]
 [  4.99999987   6.99999986]
 [  5.           6.99999989]
 [  4.99999998   6.99999998]
 [  5.           6.99999999]
 [  5.           7.        ]
 [  5.           7.        ]
 [  5.           7.        ]
 [  5.           7.        ]
 [  5.           7.        ]
 [  5.           7.        ]
 [  5.           7.        ]
 [  5.           7.        ]
 [  5.           7.        ]
 [  5.           7.        ]
 [  5.           7.        ]
 [  5.           7.        ]]
Best value found: x* = [5. 7.] with f(x*) = 0.0
Optimizer trajectory:
[[-0.1        -0.4       ]
 [-0.56368038  1.19950889]
 [ 0.0359617   1.37333866]
 [ 0.00846529  1.4681901 ]
 [ 0.23567268  1.53405506]
 [ 0.22731588  1.56288262]
 [ 0.29908501  1.58368777]
 [ 0.29675036  1.59174133]
 [ 0.31653107  1.5974756 ]
 [ 0.31591175  1.59961199]
 [ 0.32112694  1.60112391]
 [ 0.32096536  1.60168126]
 [ 0.32232326  1.60207491]
 [ 0.3222813   1.60221965]
 [ 0.3226336   1.60232174]
 [ 0.32262272  1.60235929]
 [ 0.3227146   1.60238602]
 [ 0.32271178  1.60239572]
```

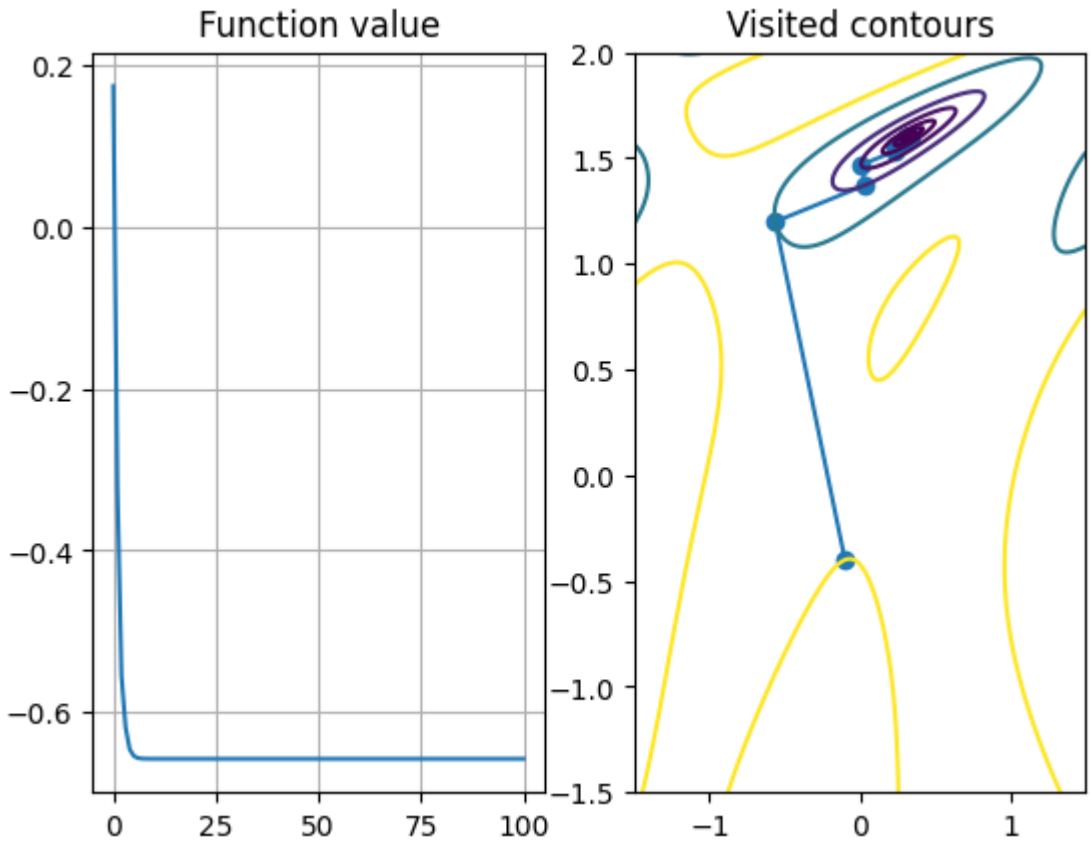
```
[ 0.32273533  1.60240258]
[ 0.32273461  1.60240506]
[ 0.32274065  1.60240682]
[ 0.32274046  1.60240746]
[ 0.32274191  1.60240785]
[ 0.32274186  1.60240802]
[ 0.32274232  1.60240826]
[ 0.32274234  1.60240822]
[ 0.32274238  1.60240823]
[ 0.32274238  1.60240823]
[ 0.32274238  1.60240824]
[ 0.32274242  1.60240825]
[ 0.32274242  1.60240825]
[ 0.32274242  1.60240825]
[ 0.32274242  1.60240825]
[ 0.32274243  1.60240826]
[ 0.32274243  1.60240826]
[ 0.32274243  1.60240826]
[ 0.32274244  1.60240826]
[ 0.32274244  1.60240827]
[ 0.32274244  1.60240826]
[ 0.32274244  1.60240826]
[ 0.32274244  1.60240826]
[ 0.32274244  1.60240827]
[ 0.32274245  1.60240827]
[ 0.32274245  1.60240827]
[ 0.32274245  1.60240827]
[ 0.32274246  1.60240827]
[ 0.32274246  1.60240827]
[ 0.32274246  1.60240827]
[ 0.32274246  1.60240827]
[ 0.32274246  1.60240827]
[ 0.32274246  1.60240827]
[ 0.32274246  1.60240827]
[ 0.32274246  1.60240827]
[ 0.32274246  1.60240827]
[ 0.32274246  1.60240827]
[ 0.32274246  1.60240827]
[ 0.32274246  1.60240827]
[ 0.32274246  1.60240827]
[ 0.32274246  1.60240828]
[ 0.32274246  1.60240827]
[ 0.32274246  1.60240827]
[ 0.32274246  1.60240827]
[ 0.32274246  1.60240827]
[ 0.32274246  1.60240827]
[ 0.32274246  1.60240827]
[ 0.32274246  1.60240827]
[ 0.32274246  1.60240827]
[ 0.32274246  1.60240827]
[ 0.32274246  1.60240827]
[ 0.32274246  1.60240827]
```

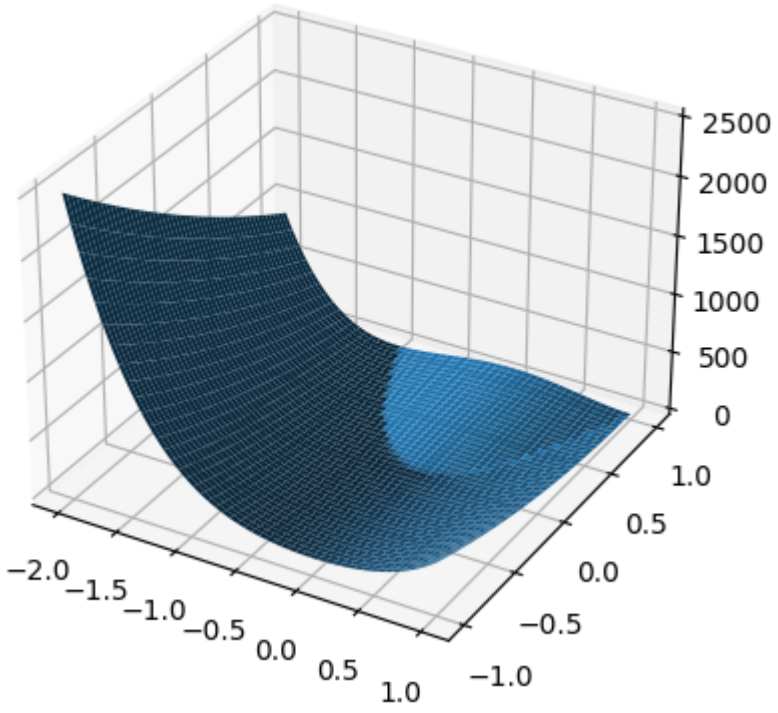
```
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240828]
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240828]
 [ 0.32274246  1.60240827]
 [ 0.32274246  1.60240827]]
Best value found: x* = [0.32274246 1.60240827] with f(x*) = -0.6574000294758536
Optimizer trajectory:
[[-1.5         0.25      ]
 [ 1.04643491  1.0952896 ]
 [ 1.04646869  1.0951876 ]
 [ 1.04637086  1.0951552 ]
 [ 1.04640459  1.09505333]
 [ 1.04630688  1.09502098]
 [ 1.04634057  1.09491923]
 [ 1.04624298  1.09488692]
 [ 1.04627663  1.0947853 ]
 [ 1.04617916  1.09475302]
 [ 1.04621276  1.09465154]
 [ 1.04611541  1.0946193 ]
 [ 1.04614897  1.09451794]
 [ 1.04605173  1.09448574]
 [ 1.04608525  1.09438451]
 [ 1.04598813  1.09435235]
 [ 1.04602161  1.09425124]
 [ 1.04592461  1.09421912]
 [ 1.04595805  1.09411815]
 [ 1.04586116  1.09408606]
 [ 1.04589456  1.09398521]]
Best value found: x* = [1.04589456 1.09398521] with f(x*) = 0.0021071167191768964
```

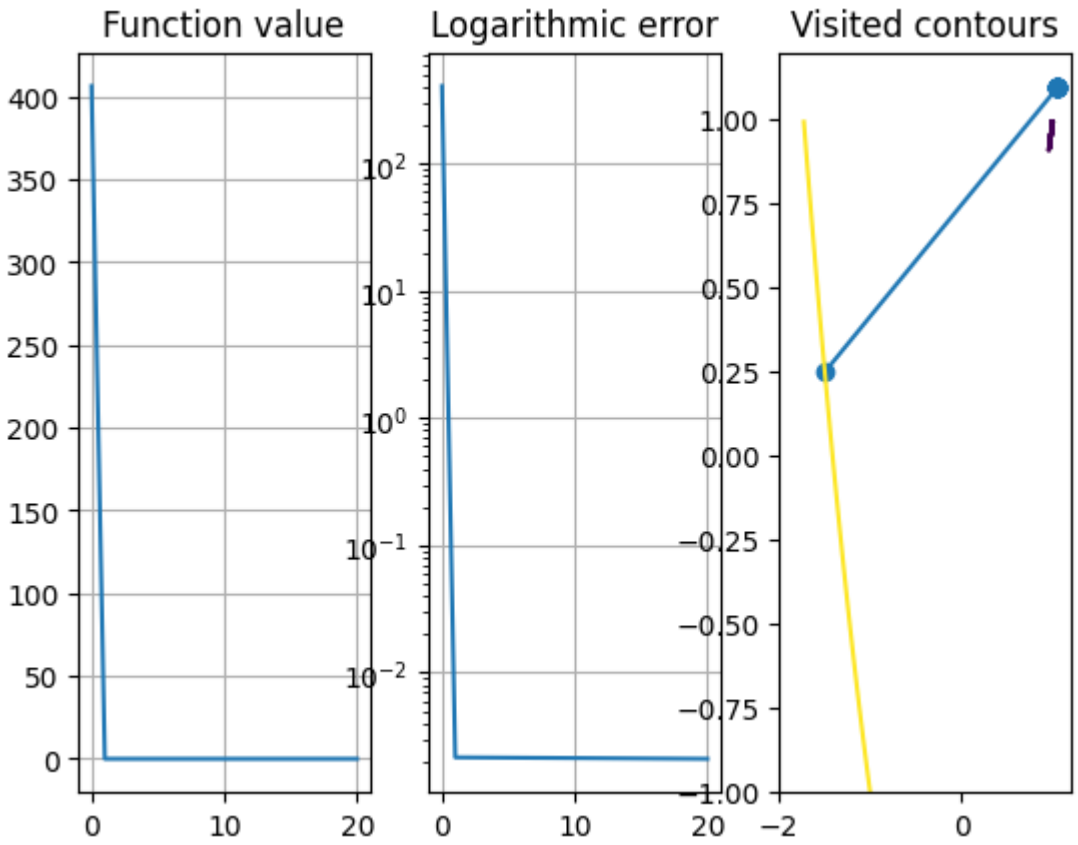

17500
15000
12500
10000
7500
5000
2500
−40
−20
0
20
40
40
20
0
−20
−40


Function value
Logarithmic error
Visited contours
4000
3500
3000
2500
2000
1500
1000
500
0
0
20
10^{3}
10^{-1}
10^{-5}
10^{-9}
10^{-13}
10^{-17}
10^{-21}
10^{-25}
0
20
40
20
0
−20
−40
−50
0
50


0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5


Function value
0.2
0.0
−0.2
−0.4
−0.6
0
25
50
75
100
Visited contours
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1
0
1





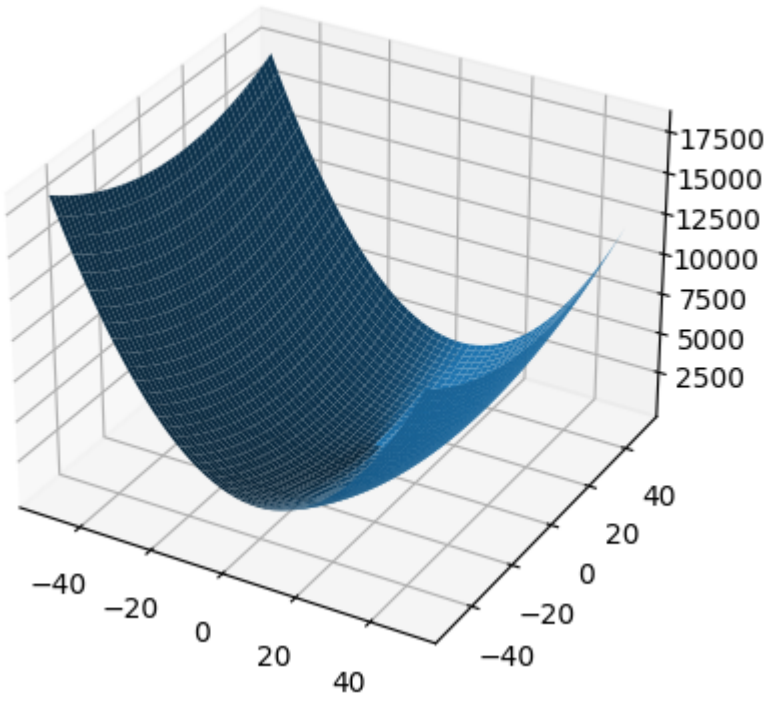
In [36]:
```
test_linear_search(wolfe_conditions_search(0.1, 0.9))
```

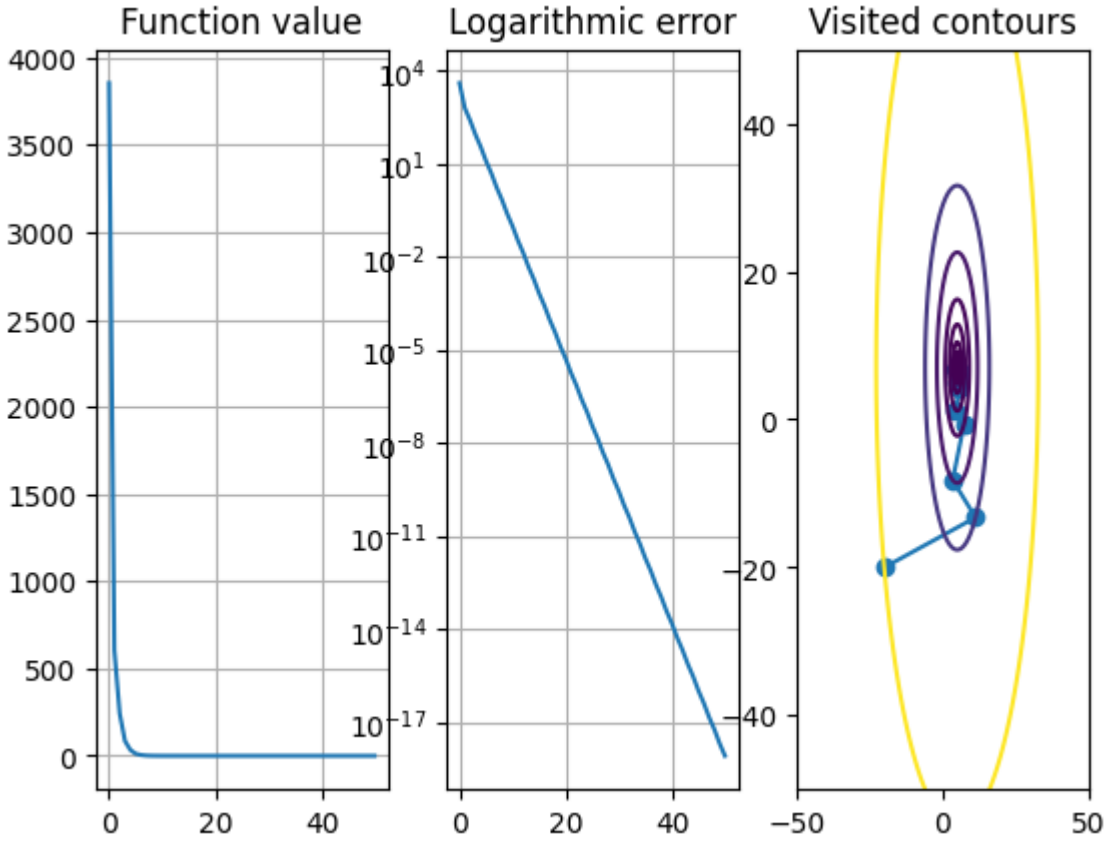
```
Optimizer trajectory:
[[-20.         -20.        ]
 [ 11.25       -13.25      ]
 [  3.4375      -8.1875    ]
 [  7.34375     -0.59375   ]
 [  4.4140625    1.3046875 ]
 [  5.87890625   4.15234375]
 [  4.78027344   4.86425781]
 [  5.32958984   5.93212891]
 [  4.91760254   6.19909668]
 [  5.12359619   6.59954834]
 [  4.96910095   6.69966125]
 [  5.04634857   6.84983063]
 [  4.98841286   6.88737297]
 [  5.01738071   6.94368649]
 [  4.99565482   6.95776486]
 [  5.00651777   6.97888243]
 [  4.99837056   6.98416182]
 [  5.00244416   6.99208091]
 [  4.99938896   6.99406068]
 [  5.00091656   6.99703034]
 [  4.99977086   6.99777276]
 [  5.00034371   6.99888638]
 [  4.99991407   6.99916478]
 [  5.00012889   6.99958239]
 [  4.99996778   6.99968679]
 [  5.00004833   6.9998434 ]
 [  4.99998792   6.99988255]
 [  5.00001813   6.99994127]
 [  4.99999547   6.99995596]
 [  5.0000068    6.99997798]
 [  4.9999983    6.99998348]
 [  5.00000255   6.99999174]
 [  4.99999936   6.99999381]
 [  5.00000096   6.9999969 ]
 [  4.99999976   6.99999768]
 [  5.00000036   6.99999884]
 [  4.99999991   6.99999913]
 [  5.00000013   6.99999956]
 [  4.99999997   6.99999967]
 [  5.00000005   6.99999984]
 [  4.99999999   6.99999988]
 [  5.00000002   6.99999994]
 [  5.           6.99999995]
 [  5.00000001   6.99999998]
 [  5.           6.99999998]
 [  5.           6.99999999]
 [  5.           6.99999999]
 [  5.           7.        ]
 [  5.           7.        ]
 [  5.           7.        ]
 [  5.           7.        ]]
Best value found: x* = [5. 7.] with f(x*) = 8.68699069152602e-19
Optimizer trajectory:
[[-0.1        -0.4       ]
 [-0.56759629  1.21301718]
```

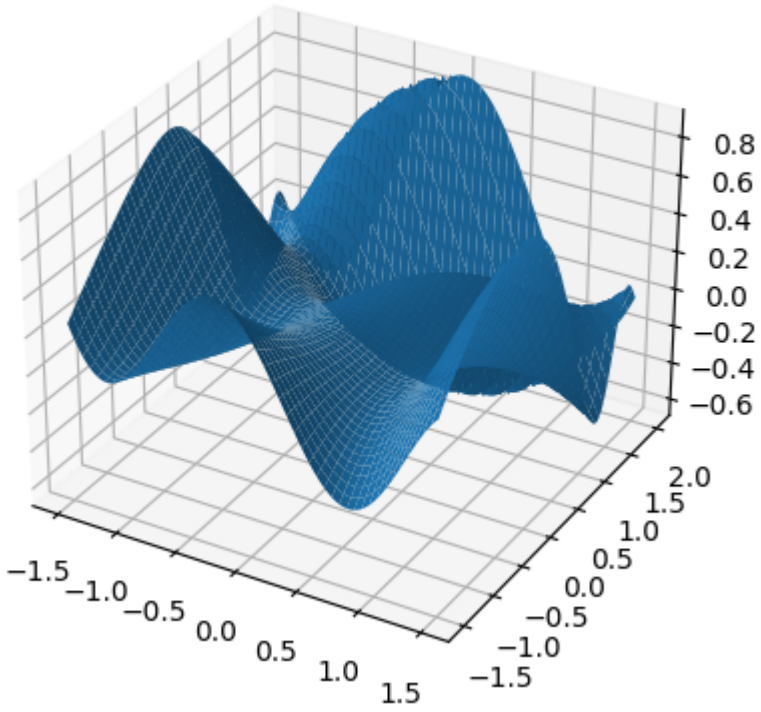
```
[-0.08753898  1.29920687]
[-0.11792528  1.43688184]
[-0.06170733  1.4103885 ]
[-0.04373678  1.4568742 ]
[ 0.03802216  1.43782035]
[ 0.03332396  1.51091008]
[ 0.05771333  1.48617769]
[ 0.19584068  1.57603569]
[ 0.21243918  1.5497162 ]
[ 0.21625641  1.56627429]
[ 0.22421644  1.55927248]
[ 0.23733251  1.57361651]
[ 0.24378702  1.56779461]
[ 0.25402186  1.57998019]
[ 0.25952692  1.57449784]
[ 0.26725852  1.5854132 ]
[ 0.27218514  1.57976722]
[ 0.2749607   1.58492333]
[ 0.28096047  1.5814074 ]
[ 0.28110781  1.5860151 ]
[ 0.2895345   1.58493915]
[ 0.28930034  1.58948074]
[ 0.29288391  1.58854892]
[ 0.29405601  1.59277019]
[ 0.29622775  1.59078113]
[ 0.29948419  1.59535766]
[ 0.30158117  1.59289968]
[ 0.30267274  1.59524738]
[ 0.30400668  1.59431869]
[ 0.30651201  1.5970479 ]
[ 0.30778048  1.59582036]
[ 0.30954953  1.59858151]
[ 0.3108154   1.59699347]
[ 0.31137418  1.59846125]
[ 0.31217711  1.59781556]
[ 0.3135254   1.5995193 ]
[ 0.3143142   1.5986495 ]
[ 0.31476681  1.59954297]
[ 0.31528724  1.59919738]
[ 0.31628882  1.60027471]
[ 0.31679191  1.59978911]
[ 0.31714119  1.60034577]
[ 0.31782732  1.59997281]
[ 0.31785079  1.60050582]
[ 0.31886926  1.60030617]
[ 0.31880643  1.60093038]
[ 0.31926304  1.60073237]
[ 0.31930078  1.60105683]
[ 0.31999406  1.60097484]
[ 0.31997511  1.60135362]
[ 0.32028221  1.60124938]
[ 0.32032039  1.60145002]
[ 0.32079535  1.60142481]
[ 0.32079607  1.60165769]
[ 0.32100419  1.60160415]
[ 0.32107109  1.60185521]
```

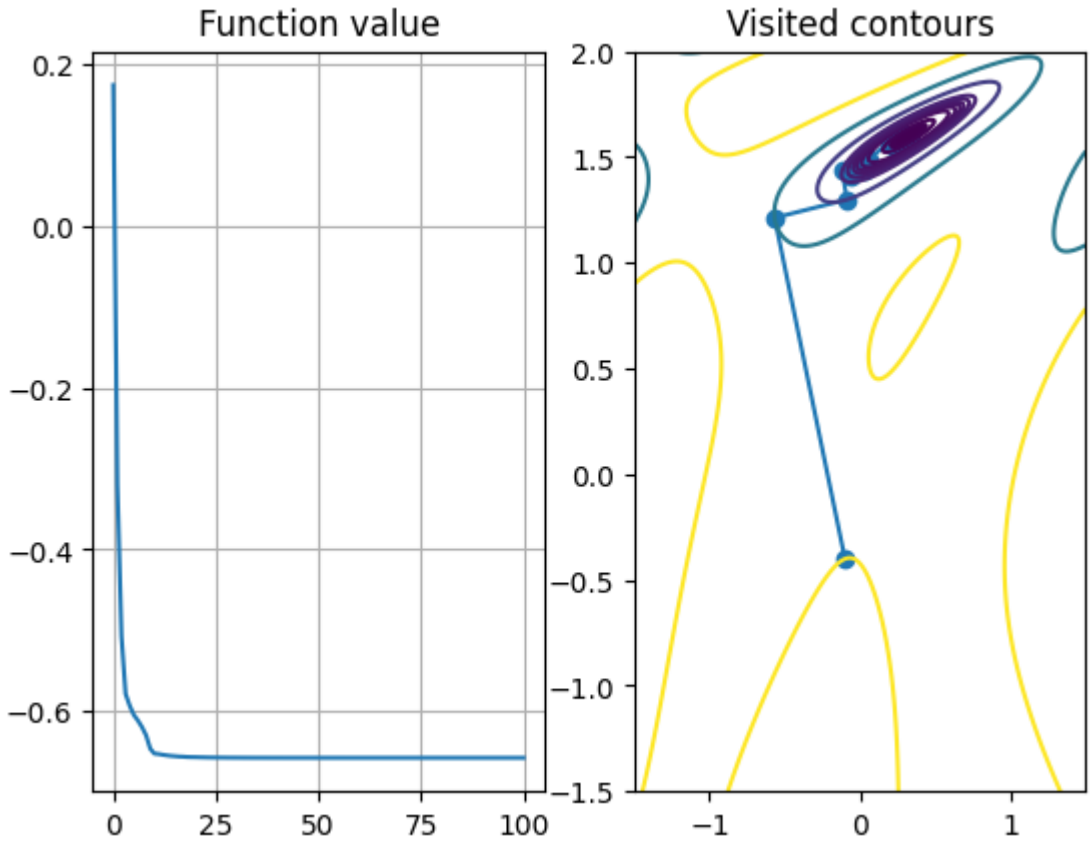
```
 [ 0.32120111  1.60173004]
 [ 0.32129134  1.60187501]
 [ 0.3214699   1.60177631]
 [ 0.32147513  1.60191661]
 [ 0.32174085  1.60186018]
 [ 0.32172242  1.60202727]
 [ 0.32184244  1.60197167]
 [ 0.32185073  1.60205938]
 [ 0.32203245  1.60203288]
 [ 0.32202521  1.60213669]
 [ 0.32210633  1.60210582]
 [ 0.32211491  1.60216098]
 [ 0.32223964  1.60215024]
 [ 0.32223799  1.60221505]
 [ 0.32229307  1.60219829]
 [ 0.32230846  1.60226802]
 [ 0.32234371  1.60223176]
 [ 0.32236609  1.6022717 ]
 [ 0.32241394  1.60224224]
 [ 0.32241399  1.60228173]
 [ 0.32244921  1.60227245]
 [ 0.32246031  1.60231527]
 [ 0.32248242  1.60229374]
 [ 0.32249753  1.6023185 ]
 [ 0.32252787  1.60230132]
 [ 0.32252857  1.60232545]
 [ 0.32255112  1.60232036]
 [ 0.322559    1.60234672]
 [ 0.32257289  1.60233393]
 [ 0.32259319  1.60236471]
 [ 0.3226074   1.60234701]
 [ 0.32261367  1.60236384]
 [ 0.32262288  1.60235622]
 [ 0.32263787  1.60237623]
 [ 0.32264715  1.60236553]
 [ 0.322652    1.60237632]
 [ 0.32265814  1.6023718 ]
 [ 0.32266913  1.60238487]
 [ 0.32267522  1.60237841]
 [ 0.32267891  1.60238536]
 [ 0.32268712  1.60238005]
 [ 0.32268702  1.60238699]
 [ 0.32269303  1.60238525]]
Best value found: x* = [0.32269303 1.60238525] with f(x*) = -0.6574000284371517
Optimizer trajectory:
[[-1.5         0.25      ]
 [ 0.85351562  1.03125   ]
 [ 1.05597167  0.91298395]
 [ 0.97250155  0.95245509]
 [ 0.97509889  0.95114731]
 [ 0.97527302  0.95108296]
 [ 0.97543233  0.95119936]
 [ 0.97532341  0.95130439]
 [ 0.97545686  0.95128539]
 [ 0.97537692  0.95137551]
 [ 0.97548482  0.9513695 ]
```

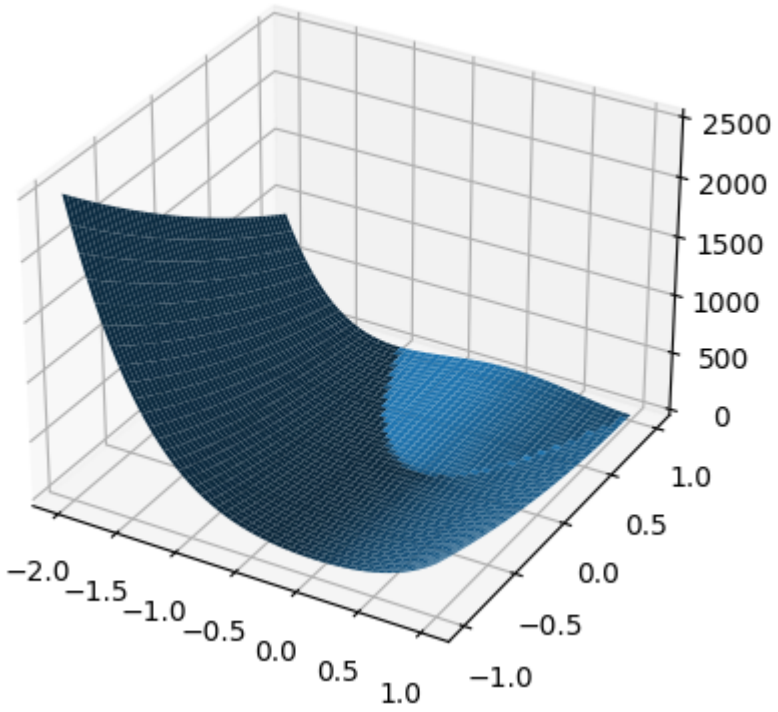
```
 [ 0.9754273   0.95144807]
 [ 0.9755154   0.95145211]
 [ 0.97547525  0.95152171]
 [ 0.975548    0.95153353]
 [ 0.97552129  0.95159617]
 [ 0.97558215  0.95161399]
 [ 0.97554954  0.95172847]
 [ 0.97566911  0.95171614]
 [ 0.97560098  0.95179976]
 [ 0.97569818  0.95179879]]
Best value found: x* = [0.97569818 0.95179879] with f(x*) = 0.0005941186318912377
```

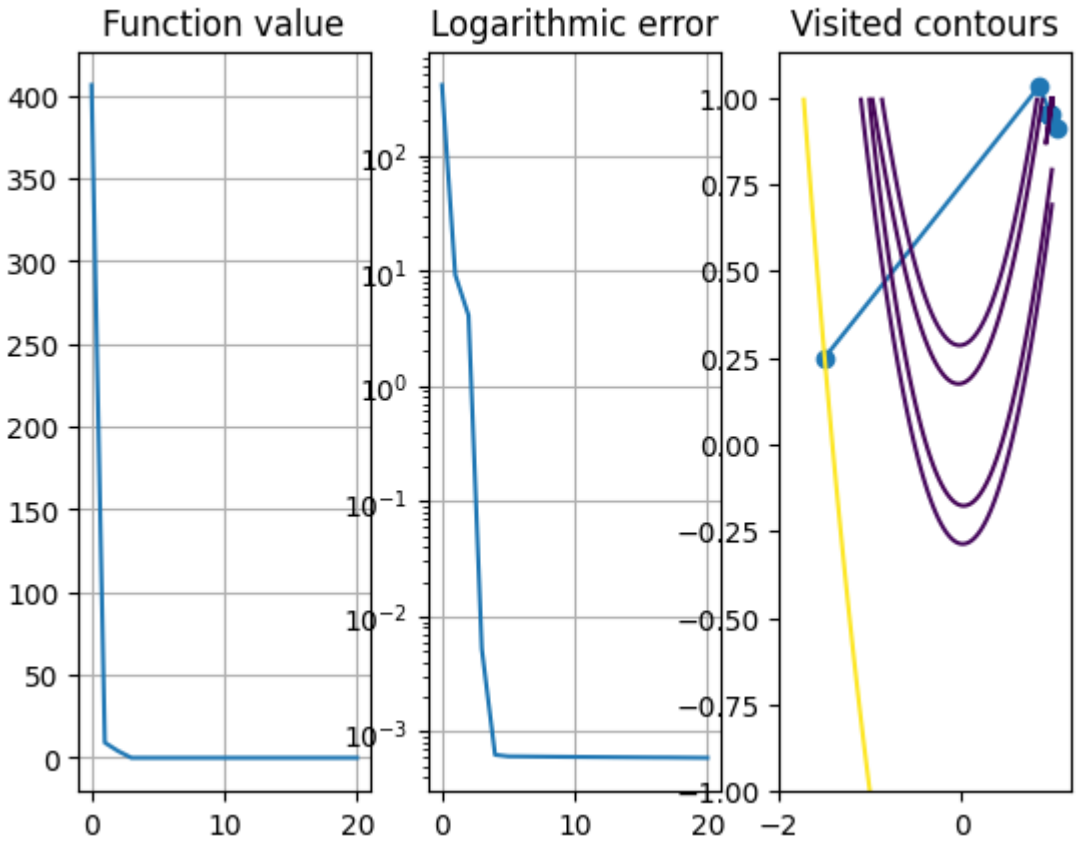



Function value
Logarithmic error
Visited contours
4000
3500
3000
2500
2000
1500
1000
500
0
10^{4}
10^{1}
10^{-2}
10^{-5}
10^{-8}
10^{-11}
10^{-14}
10^{-17}
40
20
0
−20
−40
0
20
40
−50
0
50


0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5


Function value
Visited contours
0.2
0.0
−0.2
−0.4
−0.6
0
25
50
75
100
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1
0
1


2500
2000
1500
1000
500
0
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.0
0.5
0.0
−0.5
−1.0


Function value
Logarithmic error
Visited contours
400
350
300
250
200
150
100
50
0
0
10
20
10^{2}
10^{1}
10^{0}
10^{-1}
10^{-2}
10^{-3}
0
10
20
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−2
0