Zhang_Jeffrey_Code7

May 8, 2022

Code for 10.4.5a and part 2

[95]:
```
import math
import numpy as np
from sympy import *
from scipy.integrate import quad
import scipy.special as special
from sympy.plotting import plot

def g_5a(x1,x2):

    return math.cos(x1 + x2) + math.sin(x1) + math.cos(x2)

def grad_g_5a(x1,x2):
    return [-math.sin(x1 + x2) + math.cos(x1), -math.sin(x1 + x2) - math.
 ↪sin(x2)]

def Steepest_Descent(x1,x2,TOL,N,g, grad_g):
    k = 1
    while(k <= N):
        g1 = g(x1,x2)
        z = grad_g(x1,x2)
        z0 = (z[0]**2 + z[1]**2)**(1/2)
        if z0 == 0:
            print("Zero gradient")
            return (x1,x2,g1)
        z[0] /= z0
        z[1] /= z0
        a1 = 0
        a3 = 1
        g3 = g(x1 - a3*z[0], x2 - a3*z[1])
        while(g3 >= g1):
            a3 = a3/2
            g3 = g(x1 - a3*z[0], x2 - a3*z[1])
            if a3 < TOL/2:
                print("No likely improvement")
                return (x1,x2,g1)
        a2 = a3/2
```

```
        g2 = g(x1 - a2*z[0], x2 - a2*z[1])
        h1 = (g2 - g1)/a2
        h2 = (g3 - g2)/(a3-a2)
        h3 = (h2 - h1)/a3
        a0 = 0.5*(a2 - h1/h3)
        g0 = g(x1 - a0*z[0], x2 - a0*z[1])
        a = a3 if g3 < g0 else a0
        x1 -= a*z[0]
        x2 -= a*z[1]
        print(x1,x2,g(x1,x2))
        if(abs(g(x1,x2) - g1) < TOL):
            print("Here")
            print(k)
            return (x1,x2,g(x1,x2))
        k += 1
    return(x1,x2, g(x1,x2))
```

[96]:
```
Steepest_Descent(0,0.0,0.005,100,g_5a,grad_g_5a)
```

```
2.8369211624532897 0.0 0.34603426176603536
3.731627260834447 0.21404285708964138 -0.27298556046871836
3.944688594969322 -0.7629959691074895 -0.9959369430930101
4.611585508842834 -1.5081459582143708 -1.9315864973358694
4.75107521216096 -2.472972013356772 -2.4337183081338774
5.102500087632519 -2.404673127982956 -2.5685502852321314
5.139257120830334 -2.5865137743539086 -2.591706072783908
5.20724505072657 -2.5728596073033225 -2.5967255387401327
5.215094930996798 -2.6111396030482363 -2.597781020758581
Here
9
```

[96]: (5.215094930996798, -2.6111396030482363, -2.597781020758581)

[74]:
```
g(3.3231994,0.11633359)
```

[74]: -0.14331238754164233

[81]:
```
grad_g(3.3231994, 0.11633359)
```

[81]: [-0.6900028583854367, 0.17748054212504333]

[82]:
```
grad_g(5.215094930996798, -2.6111396030482363)
```

[82]: [-0.030309596657231486, -0.0061838876972424295]

1 AMSC 460 HW 7 Part 2

Minimize the function

$$f(x,y) = -\frac{1}{\sqrt{x^2+x+2}+\sqrt{y^2+\pi y+8}}$$

by any iterative procedure with the stopping criteria $\left\|\mathbf{v}^{k+1}-\mathbf{v}^k\right\|_2 < \epsilon$, where $\mathbf{v} = (x,y)^T$.

a) Pick an initial guess $\mathbf{v}^0$. How many iterations do you need for $\epsilon = 10^{-6}, 10^{-7}, 10^{-8}$?

[98]:
```
def f(x,y):
    return -1*(math.sqrt(x**2 + x + 2) + math.sqrt(y**2 + math.pi*y + 8))**(-1)

def grad_f(x,y):
    return [(2*x+1)/((2*(math.sqrt(x**2 + x + 2) + math.sqrt(y**2 + math.pi*y +␣
 ↪8))**2)*math.sqrt(x**2 + x + 2)),
            (2*y+math.pi)/((2*(math.sqrt(x**2 + x + 2) + math.sqrt(y**2 + math.
 ↪pi*y + 8))**2)*math.sqrt(y**2 + math.pi*y + 8))]
```

[110]:
```
Steepest_Descent(0,0,10**(-6),100,f,grad_f)
```

```
-1.0213534484121316 -1.6043382451250774 -0.2649504719993911
-0.5009081350513076 -1.5840991025076654 -0.27210430438690325
-0.4993114292812974 -1.5709449063678003 -0.27210709908490355
-0.5000044011016196 -1.5708608093454606 -0.2721071126350593
Here
4
```

[110]: (-0.5000044011016196, -1.5708608093454606, -0.2721071126350593)

[103]:
```
grad_f(-0.5000044011016196, -1.5708608093454606)
```

[103]: [-2.463327527022477e-07, -2.0298194150055305e-06]

[105]:
```
Steepest_Descent(0,0,10**(-7),100,f,grad_f)
```

```
-1.0213534484121316 -1.6043382451250774 -0.2649504719993911
-0.5009081350513076 -1.5840991025076654 -0.27210430438690325
-0.4993114292812974 -1.5709449063678003 -0.27210709908490355
-0.5000044011016196 -1.5708608093454606 -0.2721071126350593
Here
4
```

[105]: (-0.5000044011016196, -1.5708608093454606, -0.2721071126350593)

[111]:
```
grad_f(-0.5000044011016196, -1.5708608093454606)
```

[111]: [-2.463327527022477e-07, -2.0298194150055305e-06]

[106]:
```
Steepest_Descent(0,0,10**(-8),100,f,grad_f)
```

```
-1.0213534484121316 -1.6043382451250774 -0.2649504719993911
-0.5009081350513076 -1.5840991025076654 -0.27210430438690325
-0.4993114292812974 -1.5709449063678003 -0.27210709908490355
-0.5000044011016196 -1.5708608093454606 -0.2721071126350593
-0.49999666307664525 -1.5707970468406602 -0.27210711270072563
Here
5
```

[106]:
```
(-0.49999666307664525, -1.5707970468406602, -0.27210711270072563)
```

[112]:
```
grad_f(-0.49999666307664525, -1.5707970468406602)
```

[112]:
```
[1.867699469312804e-07, -2.2666021416355283e-08]
```

If we set $(x, y) = (0, 0)$ for $\epsilon = 10^{-6}, 10^{-7}$, we have to iterate through the Steepest Descent algorithm 4 times. For $\epsilon = 10^{-8}$, we iterate 5 times

b) Start from different initial guesses $\mathbf{v}^0, 10\mathbf{v}^0, 10^2\mathbf{v}^0$. How many iterations do you need for $\epsilon = 10^{-6}$? Please comment on the observed in a) and b) behavior of the iterative procedure.

[113]:
```
Steepest_Descent(1,1,10**(-6),100,f,grad_f)
```

```
0.28711177947667343 0.2987223195915294 -0.22007305895134185
-0.3477959457985589 -0.47386565623117927 -0.25465019574757497
-0.6450603392618166 -1.5730591962206395 -0.27152117439114665
-0.5000637746208743 -1.5717794666174065 -0.272107097374193
-0.49995152631862094 -1.5708062677895418 -0.27210711263373283
Here
5
```

[113]:
```
(-0.49995152631862094, -1.5708062677895418, -0.27210711263373283)
```

[114]:
```
grad_f(-0.49995152631862094, -1.5708062677895418)
```

[114]:
```
[2.71310603383974e-06, -3.1292843980403156e-07]
```

[115]:
```
Steepest_Descent(10,10,10**(-6),100,f,grad_f)
```

```
9.288532609402315 9.297280886757644 -0.04762535857444489
8.57655024665862 8.59508353903985 -0.05100246809409814
7.863961787637539 7.893501264413445 -0.05488432819670585
7.150655698477648 7.192648624290415 -0.0593901748207939
6.436495489333383 6.492666341889625 -0.06467870710125849
5.72131529995059 5.793726218048968 -0.07096509942581151
5.004917865281218 5.096033796132875 -0.0785469791845324
4.28708131525463 4.399822127567973 -0.08784462420487708
3.5675929988117607 3.7053175829448257 -0.09946312498102203
```

```
2.846362959720881 3.0126219566319636 -0.11428462877544783
2.123778574710304 2.3213392311578733 -0.13358134879056388
1.4018432803711225 1.6293786609030363 -0.15902563110661255
0.6881720591087682 0.9288978346544515 -0.19192059787769888
0.012101330392060183 0.19206110255400888 -0.22946463606892598
-0.5036232949585587 -0.6646933026327773 -0.2601781875511165
-0.4965883462092926 -1.5879974973657605 -0.2721021301768726
-0.5021738550366484 -1.5721595077193145 -0.27210695120464345
-0.49988941786373525 -1.5713538379213075 -0.2721071074667636
Here
18
```

[115]: (-0.49988941786373525, -1.5713538379213075, -0.2721071074667636)

[116]:
```python
grad_f(-0.49988941786373525, -1.5713538379213075)
```

[116]: [6.189359676394398e-06, -1.7549659934689287e-05]

[121]:
```python
Steepest_Descent(100,100,10**(-6),10000,f,grad_f)
```

```
99.29282907157113 99.2929573718756 -0.0049827404616299914
98.58565724574623 98.58591564131122 -0.005018094861984858
97.87848450356805 97.87887482726993 -0.005053954459621728
97.17131082554175 97.1718349492526 -0.005090330158965298
96.4641361916158 96.4647960273171 -0.005127233180476671
95.7569605811619 95.75775808209835 -0.005164675072184072
95.04978397295419 95.0507211348291 -0.005202667721722031
94.34260634514733 94.34368520736186 -0.0052412233689053695
93.63542767525388 93.6366503221916 -0.005280354618865891
92.92824794012044 92.92961650247955 -0.005320074455781399
92.22106711590291 92.22258377207804 -0.005360396257228423
91.51388517804048 91.51555215555643 -0.005401333809191935
90.80670210122855 90.80852167822832 -0.005442901321767427
90.09951785939036 90.10149236617984 -0.00548511344559288
89.39233242564731 89.3944642462994 -0.0055279852890505205
88.68514577228795 88.68743734630876 -0.005571532436280723
87.97795787073544 87.98041169479549 -0.005615770966053195
87.27076869151355 87.27338732124711 -0.0056607174715433495
86.56357820421103 86.5663642560867 -0.005706389081064969
85.8563863774442 85.85934253071032 -0.005752803479813497
85.14919317881782 85.15232217752619 -0.005799978932677915
84.44199857488408 84.44530322999579 -0.005847934308182944
83.73480253109939 83.73828572267699 -0.005896689103627462
83.02760501177927 83.0312696912693 -0.0059462634714894085
82.32040598005086 82.32425517266138 -0.005996678247172243
81.61320539780303 81.61724220498093 -0.006047954978173138
80.90600322563401 80.91023082764714 -0.006100115954758593
80.19879942279631 80.20322108142584 -0.006153184242239141
```

79.49159394713884 79.49621300848743 -0.006207183714941207
78.78438675504592 78.78920665246788 -0.006262139091981138
78.0771778013732 78.08220205853291 -0.006318075974953916
77.36996703938013 77.37519927344562 -0.006375020887657111
76.66275442065881 76.66819834563765 -0.0064330013179794854
75.95553989505903 75.96119932528426 -0.006492045762093022
75.2483234106092 75.25420226438342 -0.006552183771097511
74.54110491343299 74.54720721683937 -0.0066134460002779255
73.8338843476613 73.84021423855063 -0.006675864261146856
73.12666165533935 73.13322338750312 -0.006739471576457445
72.41943677632845 72.42623472386839 -0.006804302238386433
71.71220964820232 71.71924831010752 -0.0068703918701024606
71.00498020613722 71.01226421108103 -0.006937777490951566
70.29774838279594 70.30528249416511 -0.007006497585510191
69.59051410820484 69.5983032293747 -0.007076592176775962
68.88327730962362 68.89132648949382 -0.007148102903788357
68.17603791140736 68.18435235021381 -0.007221073103995079
67.46879583486016 67.47738089027983 -0.00729554790070606
66.76155099807988 66.77041219164626 -0.0073715742960053735
66.05430331579338 66.06344633964183 -0.0074492012695224295
65.3470526991813 65.35648342314491 -0.007528479883498011
64.63979905569208 64.6495235347699 -0.00760946339461797
63.932542288843905 63.94256677106545 -0.00769220737312849
63.22528229801411 63.23561323272546 -0.007776769829791875
62.51801897821477 62.528663024813746 -0.007863211351291343
61.81075221985367 61.82171625700348 -0.007951595244747945
61.103481908479296 61.114773043832585 -0.008041987692072726
60.396207924508786 60.407833504976196 -0.008134457914943691
59.688930142937394 59.700897765537704 -0.008229078351270253
58.98164843302799 58.99396595635974 -0.008325924844088874
58.27436265797903 58.287038214356784 -0.008425076843923228
57.56707267456918 57.58011468287117 -0.008526617625741407
56.859778332776685 56.873195512054366 -0.008630634521752958
56.15247947537137 56.16628085927573 -0.00873721917141079
55.445175937476975 55.45937088956101 -0.008846467790119042
54.73786754610118 54.752465776063175 -0.008958481458299587
54.03055411963065 54.0455657005684 -0.009073366432638546
53.32323546728792 53.338670854040245 -0.009191234481522962
52.615911388546756 52.631781437205504 -0.009312203246888348
51.90858167250225 51.924897661185426 -0.009436396634933992
51.20124609719152 51.21801974817651 -0.009563945238427206
50.4939044288604 50.511147932185416 -0.009694986793614895
49.786556421171085 49.80428245982313 -0.009829666675094604
49.079201814345176 49.09742359116393 -0.009968138432373307
48.37184033423573 48.39057160067556 -0.010110564372266298
47.664471691321594 47.68372677822737 -0.010257116191767744
46.95709557961617 46.97688943018433 -0.010407975666566625
46.24971167548207 46.27005988059547 -0.010563335400996467

45.542319636342064 45.56323847248636 -0.010723399645905211
44.83491909927566 44.85642556926645 -0.010888385191725549
44.127509679489194 44.149621556263284 -0.011058522344931014
43.420090968646086 43.442826842397174 -0.011234055997096466
42.712662533042085 42.73604186201143 -0.01141524679696399
42.00522391160849 42.02926707687537 -0.01160237243727077
41.297774613724215 41.32250297837924 -0.011795729069652887
40.590314116815044 40.615750089942956 -0.011995632862732221
39.88284186371553 39.9090089696632 -0.012202421720562798
39.17535725976584 39.202280213226786 -0.012416457181006424
38.4678596696121 38.49556445712207 -0.012638126516381597
37.76034841367423 37.7888623821844 -0.012867845061955093
37.05282276424052 37.08217471751676 -0.013106058801602198
36.34528194114223 36.375502244832695 -0.013353247244351026
35.63772510695473 35.6688458032753 -0.01360992663066743
34.93015136166375 34.962206294774056 -0.013876653513376268
34.22255973672625 34.25558469001072 -0.014154028765229006
33.514949188444504 33.54898203507627 -0.014442702073534265
32.80731859055918 32.84239945891376 -0.014743376992231801
32.09966672595249 32.13583818165702 -0.015056816633639196
31.391992277334687 31.429299523993134 -0.015383850096237484
30.68429381676615 30.722784917697606 -0.015725379741788423
29.976569793842618 30.016295917516477 -0.016082389455415857
29.268818522341295 29.309834214599565 -0.016455954046817694
28.561038165090103 28.603401651725108 -0.016847249980484855
27.853226716779748 27.897000240599112 -0.017257567658928358
27.145381984386887 27.190632181564958 -0.017688325527029716
26.437501564814625 26.484299886121658 -0.0181410863197395
25.72958281928125 25.77800600272576 -0.01861757584203024
25.021622843896296 25.071753446445125 -0.019119704752584656
24.313618435750776 24.36554543314698 -0.019649593925479613
23.605566053710607 23.65938551904307 -0.02020960409273995
22.897461772932097 22.95327764658791 -0.02080237063248192
22.189301231907848 22.24722619794111 -0.021430844572232932
21.481079570589245 21.541236057472336 -0.022098341137946065
20.772791357804238 20.83531268512246 -0.022808597513737604
20.0644305057778 20.129462202855585 -0.02356584190915939
19.35599016904371 19.423691496968686 -0.024374876592204386
18.647462624378903 18.718008339701022 -0.025241178281724338
17.938839127554832 18.012421534446826 -0.026171020264302874
17.230109741630393 17.306941089978544 -0.027171621894523025
16.521263130137577 16.601578430508145 -0.028251332876741534
15.812286306741903 15.89634665024879 -0.02941986208669468
15.10316433067458 15.191260823516771 -0.0306885639270609
14.393879934277917 14.486338384500803 -0.032070799695589944
13.684413065185803 13.781599594834281 -0.03358239773139789
12.97474032074441 13.077068122295957 -0.03524224502765897
12.264834246010453 12.372771760630055 -0.03707305582826619

```
11.554662458827908 11.668743328895975 -0.039102381430073066
10.84418655605444 10.965021799043273 -0.04136395308526762
10.133360744536837 10.261653712135471 -0.043899491482801374
9.422130130951006 9.558694954953856 -0.04676117980598258
8.710428601761958 8.856212974152944 -0.05001509604173445
7.99817624257815 8.154289490547288 -0.053746056172717105
7.285276320010029 7.453023706124316 -0.058064570494047
6.571612064193735 6.7525357833375095 -0.06311702389233091
5.857044058448814 6.052969799912267 -0.06910086313818314
5.141410506447074 5.354493873878038 -0.07628767554448272
4.424536454013694 4.6572911824405026 -0.08505878032802762
3.7062682747391915 3.9615248260810625 -0.09596032252832604
2.9865787435978177 3.2672287950067735 -0.109786210161075
2.2658766985181447 2.573983843912302 -0.12768711080376288
1.5459547774685571 1.8799287829897033 -0.15122847044033558
0.8331599297406836 1.1785561965190046 -0.18193932284009484
0.15051574979713955 0.44780526522352804 -0.21863942309697995
-0.4104249109763142 -0.38005074586236187 -0.2523799421720576
-0.6033123351035714 -1.6696084967975473 -0.27165600766222164
-0.4887475869049985 -1.6078493628868085 -0.2720819641074483
-0.5057664944178738 -1.5763337270801112 -0.27210569952094704
-0.49936771236664956 -1.5728779261703565 -0.27210703331365044
-0.5003239434632939 -1.5711074110708068 -0.27210710824113515
Here
147
```

[121]: (-0.5003239434632939, -1.5711074110708068, -0.27210710824113515)

[122]:
```
grad_f(-0.5003239434632939, -1.5711074110708068)
```

[122]: [-1.8131342323177508e-05, -9.792492259445488e-06]

It appears that increasing $v^{(0)}$ by a factor of 10 increased the number of iterations in the Steepest Descent algorithm more than decreasing ϵ by a factor of 10