**Python History for problem 1 with Golden Section Method**

[Command: python -u C:\Users\lxyxi\OneDrive\ME441\_project\ME441\_Engineering\_Optimization\_2017Fall\Homework4\GoldenSection.py]

cG= 0.61803398875

Step#: 0 a b c d fa fb fc fd

Step#: 1 0.816311896062 1.71352549156 1.15901699437 1.37082039325 10.3383178603 8.28267744039 8.22290286525 7.73763198885

Step#: 2 1.15901699437 1.71352549156 1.37082039325 1.50172209269 8.22290286525 8.28267744039 7.73763198885 7.75174285191

Step#: 3 1.15901699437 1.50172209269 1.28991869381 1.37082039325 8.22290286525 7.75174285191 7.84885778028 7.73763198885

Step#: 4 1.28991869381 1.50172209269 1.37082039325 1.42082039325 7.84885778028 7.75174285191 7.73763198885 7.71470626413

Step#: 5 1.37082039325 1.50172209269 1.42082039325 1.45172209269 7.73763198885 7.75174285191 7.71470626413 7.71803738703

Step#: 6 1.37082039325 1.45172209269 1.40172209269 1.42082039325 7.73763198885 7.71803738703 7.71933192215 7.71470626413

Step#: 7 1.40172209269 1.45172209269 1.42082039325 1.43262379212 7.71933192215 7.71803738703 7.71470626413 7.71440066584

Step#: 8 1.42082039325 1.45172209269 1.43262379212 1.43991869381 7.71470626413 7.71803738703 7.71440066584 7.71518703728

Step#: 9 1.42082039325 1.43991869381 1.42811529494 1.43262379212 7.71470626413 7.71518703728 7.71428717069 7.71440066584

Step#: 10 1.42082039325 1.43262379212 1.42532889044 1.42811529494 7.71470626413 7.71440066584 7.71435931266 7.71428717069

Step#: 11 1.42532889044 1.43262379212 1.42811529494 1.42983738762 7.71435931266 7.71440066584 7.71428717069 7.71429693285

Step#: 12 1.42532889044 1.42983738762 1.42705098312 1.42811529494 7.71435931266 7.71429693285 7.71430189657 7.71428717069

Step#: 13 1.42705098312 1.42983738762 1.42811529494 1.42877307581 7.71430189657 7.71429693285 7.71428717069 7.71428599892

Step#: 14 1.42811529494 1.42983738762 1.42877307581 1.42917960675 7.71428717069 7.71429693285 7.71428599892 7.71428830345

Step#: 15 1.42811529494 1.42917960675 1.42852182588 1.42877307581 7.71428717069 7.71428830345 7.71428573151 7.71428599892

Step#: 16 1.42811529494 1.42877307581 1.42836654487 1.42852182588 7.71428717069 7.71428599892 7.71428600813 7.71428573151

Optimal Solution: [ 7.71428583 1.42844419]

[Finished in 1.585s]

**Python Code for problem 1 with Golden Section Method**

'''

GoldenSectionSearchAlgorithm

Date: 10/13/17

Class: ME441\_EngineeringOptimization

Author: Xiaoyi Liu

'''

import numpy as np

cG=(-1+np.sqrt(5))/2

print("cG=",cG)

delta=0.05

def fLinear(x):

return 7\*x\*x-20\*x+22

def InitialBracketing(fLinear):

xl=0

xm=xl+delta

xu=xm+(1+cG)\*delta

while((fLinear(xl)-fLinear(xm))

\*(fLinear(xm)-fLinear(xu))>0):

xl=xm

xm=xu

xu=xm+(1+cG)\*(xm-xl)

return(xl,xu)

def GoldenSection(fLinear,a,b):

# (a,b)=(InitialBracketing(fLinear))

c=a+(1-cG)\*(b-a)

d=a+cG\*(b-a)

fa=fLinear(a)

fb=fLinear(b)

fc=fLinear(c)

fd=fLinear(d)

i=0

print('Step#: ', i, 'a b c d fa fb fc fd\n')

while((np.abs(a-b))>0.001):

if(fc<fd):

b=d

fb=fd

d=c

fd=fc

c=a+(1-cG)\*(b-a)

fc=fLinear(c)

else:

a=c

fa=fc

c=d

fc=fd

d=a+cG\*(b-a)

fd=fLinear(d)

i+=1

print('Step#: ', i, a, b, c, d, fa, fb,fc,fd,'\n')

# print(a,b)

return np.array([fLinear((a+b)/2),(a+b)/2])

(a,b)=(InitialBracketing(fLinear))

print('Optimal Solution: ', GoldenSection(fLinear,a,b))

**Python History for problem 2 with Steepest Descent Method**

[Command: python -u C:\Users\lxyxi\OneDrive\ME441\_project\ME441\_Engineering\_Optimization\_2017Fall\Homework4\SteepestDesent.py]

Step # = 0

df= [4 8 6] x= [1, 1, 1] alpha= 0.154430856687 f0= 9 norm= 10.7703296143

Step # = 1

df= [ 0.29365944 -0.03040455 -0.17723427] x= [ 0.38227657 -0.23544685 0.07341486] alpha= 0.41238353925 f0= 0.0532030752104 norm= 0.344343271325

Step # = 2

df= [ 0.07653547 -0.07627486 0.14019638] x= [ 0.26117625 -0.22290852 0.14650335] alpha= 0.660003665687 f0= 0.0287654015479 norm= 0.177004399043

Step # = 3

df= [ 0.07619146 -0.16099575 -0.12924075] x= [ 0.21066256 -0.17256683 0.05397323] alpha= 0.223935688187 f0= 0.0184288369988 norm= 0.220063493694

Step # = 4

df= [ 0.11417287 0.00697429 0.0586311 ] x= [ 0.19360058 -0.13651414 0.08291485] alpha= 0.368362879187 f0= 0.0130066171457 norm= 0.128536733694

Step # = 5

df= [ 0.02492064 -0.13061113 -0.03289712] x= [ 0.15154353 -0.13908321 0.06131732] alpha= 0.240498834187 f0= 0.00996260641973 norm= 0.13697636773

Step # = 6

df= [ 0.07575752 -0.00112716 0.0615734 ] x= [ 0.14555014 -0.10767138 0.06922904] alpha= 0.366234255562 f0= 0.00770527430314 norm= 0.0976307126875

Step # = 7

df= [ 0.02109313 -0.10006652 -0.02780214] x= [ 0.11780514 -0.10725858 0.04667875] alpha= 0.240498834187 f0= 0.00596005119489 norm= 0.105977294555

Step # = 8

df= [ 0.05907915 -0.00057598 0.04707515] x= [ 0.11273227 -0.0831927 0.05336514] alpha= 0.365827724624 f0= 0.00461010752707 norm= 0.0755430139848

Step # = 9

df= [ 0.01627498 -0.07740151 -0.02138901] x= [ 0.09111948 -0.08298199 0.03614374] alpha= 0.240498834187 f0= 0.00356641026694 norm= 0.0819350911507

Step # = 10

df= [ 0.0456767 -0.00048178 0.03641706] x= [ 0.08720537 -0.06436702 0.04128777] alpha= 0.366234255562 f0= 0.00275892193413 norm= 0.0584191388164

Step # = 11

df= [ 0.01257285 -0.0599071 -0.01657875] x= [ 0.07047699 -0.06419057 0.0279506 ] alpha= 0.240498834187 f0= 0.00213409066513 norm= 0.0634176014177

Step # = 12

df= [ 0.03534051 -0.00034992 0.02818511] x= [ 0.06745324 -0.04978298 0.03193777] alpha= 0.365827724624 f0= 0.00165071065569 norm= 0.0452048053245

Step # = 13

df= [ 0.00973945 -0.04631674 -0.01280245] x= [ 0.0545247 -0.04965497 0.02162687] alpha= 0.240498834187 f0= 0.00127701024582 norm= 0.0490306063674

Step # = 14

df= [ 0.02733304 -0.00028696 0.02179169] x= [ 0.05218237 -0.03851585 0.02470585] alpha= 0.366234255562 f0= 0.000987868926663 norm= 0.0349579112552

Step # = 15

df= [ 0.00752264 -0.03584891 -0.00992158] x= [ 0.04217208 -0.03841076 0.01672498] alpha= 0.24024758425 f0= 0.000764145319843 norm= 0.0379495955081

Step # = 16

df= [ 0.02113327 -0.00024578 0.01683819] x= [ 0.04036478 -0.02979814 0.01910862] alpha= 0.366892036437 f0= 0.000591059044944 norm= 0.0270222173097

Step # = 17

df= [ 0.00580636 -0.02774794 -0.00769265] x= [ 0.03261115 -0.02970797 0.01293082] alpha= 0.24024758425 f0= 0.000457107307567 norm= 0.0293741169655

Step # = 18

df= [ 0.01634918 -0.00017608 0.01303266] x= [ 0.03121619 -0.0230416 0.01477896] alpha= 0.366234255562 f0= 0.000353512771629 norm= 0.0209087785596

Step # = 19

df= [ 0.0045029 -0.02143941 -0.00593039] x= [ 0.02522856 -0.02297711 0.01000596] alpha= 0.240498834187 f0= 0.000273438949242 norm= 0.0226956787072

Step # = 20

df= [ 0.01264932 -0.00012818 0.01008692] x= [ 0.02414561 -0.01782096 0.01143221] alpha= 0.365827724624 f0= 0.00021151278573 norm= 0.0161792366698

Step # = 21

df= [ 0.00348816 -0.01657571 -0.0045796 ] x= [ 0.01951814 -0.01777407 0.00774213] alpha= 0.240498834187 f0= 0.000163622094971 norm= 0.0175469124556

Step # = 22

df= [ 0.00978324 -0.00010497 0.00779883] x= [ 0.01867925 -0.01378763 0.00884352] alpha= 0.366234255562 f0= 0.000126579979174 norm= 0.0125117766354

Step # = 23

df= [ 0.00269421 -0.01282951 -0.00354908] x= [ 0.01509629 -0.01374918 0.00598732] alpha= 0.240498834187 f0= 9.79091914365e-05 norm= 0.0135812739114

Step # = 24

df= [ 7.56926604e-03 -7.63933320e-05 6.03608184e-03] x= [ 0.01444833 -0.0106637 0.00684087] alpha= 0.365827724624 f0= 7.57349845423e-05 norm= 0.00968162735738

Step # = 25

df= [ 0.00208706 -0.00991903 -0.00274069] x= [ 0.01167929 -0.01063575 0.0046327 ] alpha= 0.240498834187 f0= 5.85875081001e-05 norm= 0.0105002110546

Step # = 26

df= [ 5.85422332e-03 -6.25778862e-05 4.66687281e-03] x= [ 0.01117735 -0.00825024 0.00529184] alpha= 0.366234255562 f0= 4.53236556762e-05 norm= 0.007487025341

Step # = 27

df= [ 0.00161203 -0.00767728 -0.00212397] x= [ 0.00903333 -0.00822732 0.00358267] alpha= 0.240498834187 f0= 3.50579528233e-05 norm= 0.00812714190446

Step # = 28

df= [ 4.52939721e-03 -4.55302756e-05 3.61203166e-03] x= [ 0.00864564 -0.00638094 0.00409348] alpha= 0.365827724624 f0= 2.711792513e-05 norm= 0.00579346914879

Step # = 29

df= [ 0.00124875 -0.00593563 -0.00164018] x= [ 0.00698866 -0.00636429 0.0027721 ] alpha= 0.240498834187 f0= 2.09781943648e-05 norm= 0.00628340927321

Step # = 30

df= [ 3.50312728e-03 -3.73047339e-05 2.79268779e-03] x= [ 0.00668834 -0.00493678 0.00316656] alpha= 0.366234255562 f0= 1.62287415327e-05 norm= 0.00448022293045

Step # = 31

df= [ 0.00096452 -0.00459414 -0.0012711 ] x= [ 0.00540537 -0.00492311 0.00214378] alpha= 0.240498834187 f0= 1.25530610562e-05 norm= 0.00486334611275

Step # = 32

df= [ 2.71036042e-03 -2.71355128e-05 2.16146386e-03] x= [ 0.00517341 -0.00381823 0.00244948] alpha= 0.365827724624 f0= 9.709936145e-06 norm= 0.00346680197598

Step # = 33

df= [ 0.00074716 -0.00355192 -0.00098158] x= [ 0.00418188 -0.0038083 0.00165876] alpha= 0.240498834187 f0= 7.51157804383e-06 norm= 0.00376004177229

Step # = 34

df= [ 2.09624746e-03 -2.22382549e-05 1.67116299e-03] x= [ 0.00400219 -0.00295407 0.00189482] alpha= 0.366234255562 f0= 5.81091811726e-06 norm= 0.00268095760831

Step # = 35

df= [ 0.0005771 -0.00274917 -0.0007607 ] x= [ 0.00323447 -0.00294592 0.00128279] alpha= 0.240498834187 f0= 4.49482441027e-06 norm= 0.00291026485615

Step # = 36

df= [ 1.62186120e-03 -1.61721834e-05 1.29343440e-03] x= [ 0.00309568 -0.00228475 0.00146573] alpha= 0.365827724624 f0= 3.47677263474e-06 norm= 0.00207452834545

Step # = 37

df= [ 0.00044705 -0.0021255 -0.00058743] x= [ 0.00250236 -0.00227883 0.00099256] alpha= 0.240498834187 f0= 2.68964067234e-06 norm= 0.00225003872044

Step # = 38

df= [ 1.25438018e-03 -1.32565684e-05 1.00003507e-03] x= [ 0.00239484 -0.00176765 0.00113384] alpha= 0.366234255562 f0= 2.08067700876e-06 norm= 0.00160428037293

Step # = 39

df= [ 0.0003453 -0.00164512 -0.00045524] x= [ 0.00193545 -0.0017628 0.00076759] alpha= 0.240498834187 f0= 1.60944381773e-06 norm= 0.00174152555783

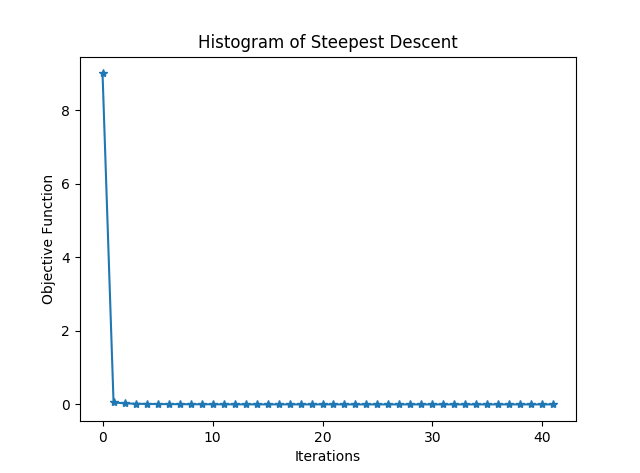
Step # = 40

df= [ 9.70510692e-04 -9.63811484e-06 7.73999782e-04] x= [ 0.0018524 -0.00136715 0.00087707] alpha= 0.365827724624 f0= 1.24490499023e-06 norm= 0.00124139420006

Step # = 41

df= [ 0.00026748 -0.00127192 -0.00035155] x= [ 0.00149736 -0.00136362 0.00059392] alpha= 0.240498834187 f0= 9.63068867191e-07 norm= 0.00134644096035

[Finished in 2.247s]



**Python Code for problem 2 with Steepest Descent Method**

'''

SteepestDescentAlgorithm

Date: 10/13/17

Class: ME441\_EngineeringOptimization

Author: Xiaoyi Liu

'''

import numpy as np

import matplotlib.pyplot as plt

from GoldenSection\_For\_Multiopt import GoldenSection

def fQuar(x):

return x[0]\*\*2+2\*x[1]\*\*2+2\*x[2]\*\*2 + 2\*x[0]\*x[1]+2\*x[1]\*x[2]

def dfQuar(x):

return np.array([2\*x[0]+2\*x[1], 2\*x[0]+4\*x[1]+2\*x[2], 2\*x[1]+4\*x[2]])

#SteepestDescentAlgorithm

x0=[1,1,1]

f0=fQuar(x0)

df0=dfQuar(x0)

epsilon=0.001

i=0

y=[]

while (np.linalg.norm(df0)>epsilon):

y.append(f0)

print('Step # = ',i)

dr=-df0

def localf(alpha):

return fQuar(x0+alpha\*dr)

[ff,alpha0]=GoldenSection(localf,-1,1)

print('df= ', df0, 'x= ', x0,'alpha= ',alpha0, 'f0= ',f0,'norm= ',np.linalg.norm(df0))

x0=x0+alpha0\*dr

f0=fQuar(x0)

df0=dfQuar(x0)

i=i+1

plt.plot(y,'-\*')

plt.xlabel('Iterations')

plt.ylabel('Objective Function')

plt.title('Histogram of Steepest Descent')

plt.show()

**Python History for problem 2 with Conjugate Method**

[Command: python -u C:\Users\lxyxi\OneDrive\ME441\_project\ME441\_Engineering\_Optimization\_2017Fall\Homework4\FRConjugate.py]

Step # = 0

df= [-4 -8 -6] x= [1, 1, 1] alpha= 0.154430856687 f0= 9 norm= 10.7703296143

c0= [4 8 6] beta= 0.00102217490092 c1= [ 0.29365944 -0.03040455 -0.17723427]

[-0.29774814 0.02222715 0.17110122]

Step # = 1

df= [-0.29774814 0.02222715 0.17110122] x= [ 0.38227657 -0.23544685 0.07341486] alpha= 0.415576474687 f0= 0.0532030752104 norm= 10.7703296143

c0= [ 0.29365944 -0.03040455 -0.17723427] beta= 0.250626550618 c1= [ 0.06465936 -0.09871919 0.12566246]

[-0.13928294 0.1042899 -0.08277995]

Step # = 2

df= [-0.13928294 0.1042899 -0.08277995] x= [ 0.25853945 -0.22620977 0.1445205 ] alpha= 1.73918327244 f0= 0.0286045204994 norm= 0.344127341699

c0= [ 0.06465936 -0.09871919 0.12566246] beta= 1.08048567787 c1= [-0.05705927 -0.14561831 -0.08745704]

[-0.09343396 0.25830205 -0.00198551]

Step # = 3

df= [-0.09343396 0.25830205 -0.00198551] x= [ 0.01630088 -0.04483052 0.000551 ] alpha= 0.172871376375 f0= 0.00277491949107 norm= 0.192687940581

c0= [-0.05705927 -0.14561831 -0.08745704] beta= 7.16122555095e-06 c1= [ -5.73195041e-05 3.23008828e-06 4.76072645e-04]

[ 5.66504025e-05 -1.38032900e-06 -4.76086863e-04]

Step # = 4

df= [ 5.66504025e-05 -1.38032900e-06 -4.76086863e-04] x= [ 0.00014883 -0.00017749 0.00020776] alpha= 0.251237921249 f0= 4.49027964009e-08 norm= 0.274688547566

c0= [ -5.73195041e-05 3.23008828e-06 4.76072645e-04] beta= 0.193647045967 c1= [ -2.95476274e-05 -2.08913765e-04 -3.06523283e-06]

[ 4.05178105e-05 2.08646468e-04 -8.91275819e-05]

Step # = 5

df= [ 4.05178105e-05 2.08646468e-04 -8.91275819e-05] x= [ 1.63059707e-04 -1.77833521e-04 8.81504522e-05] alpha= 0.264105631937 f0= 1.60318206036e-08 norm= 0.000479447468228

c0= [ -2.95476274e-05 -2.08913765e-04 -3.06523283e-06] beta= 0.242244313259 c1= [ 1.02063751e-04 -1.41711643e-05 1.29877964e-05]

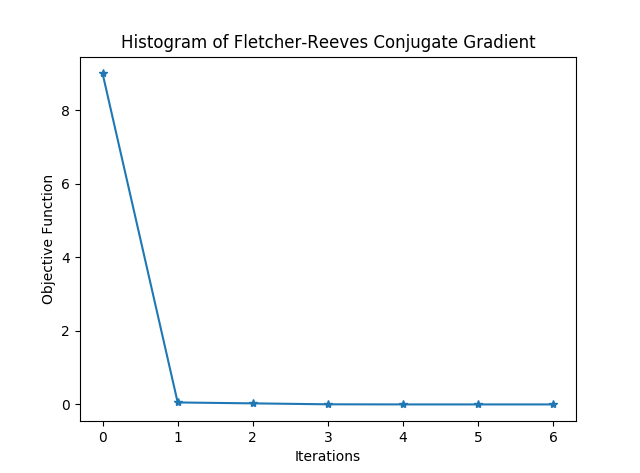
[ -9.22485419e-05 6.47145847e-05 -3.45784463e-05]

Step # = 6

df= [ -9.22485419e-05 6.47145847e-05 -3.45784463e-05] x= [ 1.73760689e-04 -1.22728814e-04 6.46113559e-05] alpha= 1.88361046344 f0= 1.01565185216e-08 norm= 0.000230475090734

c0= [ 1.02063751e-04 -1.41711643e-05 1.29877964e-05] beta= 0.00332715195415 c1= [ -1.66274892e-06 -4.36817325e-06 -3.74735904e-06]

[ 1.35582401e-06 4.58348851e-06 3.63231129e-06]

[Finished in 0.874s]

**Python Code for problem 2 with Conjugate Method**

'''

Fletcher-Reeves Conjugate Gradient Algorithm

Date: 10/13/17

Class: ME441\_EngineeringOptimization

Author: Xiaoyi Liu

'''

import numpy as np

import matplotlib.pyplot as plt

from GoldenSection\_For\_Multiopt import GoldenSection

def fQuar(x):

return x[0]\*\*2+2\*x[1]\*\*2+2\*x[2]\*\*2 + 2\*x[0]\*x[1]+2\*x[1]\*x[2]

def dfQuar(x):

return np.array([2\*x[0]+2\*x[1], 2\*x[0]+4\*x[1]+2\*x[2], 2\*x[1]+4\*x[2]])

#SteepestDescentAlgorithm

x0=[1,1,1]

f0=fQuar(x0)

c0=dfQuar(x0)

c1=c0

d0=-c0

d1=d0

epsilon=0.001

i=0

y=[]

while (np.linalg.norm(d1)>epsilon and i<10):

y.append(f0)

print('Step # = ',i)

def localf(alpha):

return fQuar(x0+alpha\*d1)

[ff,alpha0]=GoldenSection(localf,-1,1)

print('df= ', d1, 'x= ', x0,'alpha= ',alpha0, 'f0= ',f0,'norm= ',np.linalg.norm(d0))

x0=x0+alpha0\*d1

f0=fQuar(x0)

c0=c1

c1=dfQuar(x0)

d0=d1

d1=-c1+d0\*((np.linalg.norm(c1)/np.linalg.norm(c0))\*\*2)

print('c0= ',c0,'beta=',(np.linalg.norm(c1)/np.linalg.norm(c0))\*\*2,'c1=',c1)

print(d1)

i=i+1;

plt.plot(y,'-\*')

plt.xlabel('Iterations')

plt.ylabel('Objective Function')

plt.title('Histogram of Fletcher-Reeves Conjugate Gradient')

plt.show()

Comparison of Steepest Descent and Conjugate methods:

|  |  |  |
| --- | --- | --- |
|  | Steepest Descent | Conjugate method |
| Iteration numbers: | 41 | 6 |
| Optimal Solution: | 9.631e-7 | 1.018e-8 |
| Run Time: | 2.247s | 0.874s |