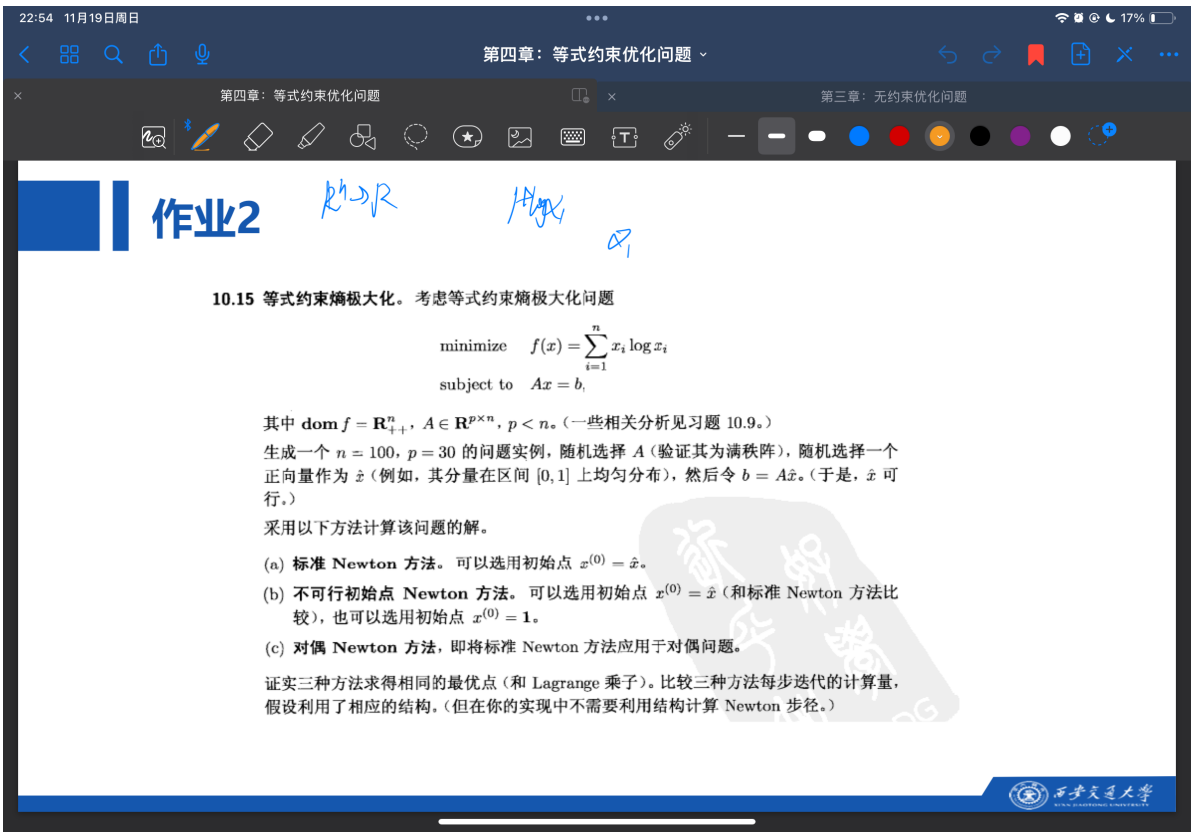
第三次大作业

1.问题描述



2.问题分析

使用numpy库和math库计算梯度和Hessian 矩阵，然后分别采取三种方法进行优化，并记录误差随迭代次数的变化，绘制图像。

首先确定一个A矩阵，其秩为p。

```
38  def main():
39
40      n=100
41      p=30
42      A=np.random.rand(p,n)
43      while np.linalg.matrix_rank(A)!=p:
44          A=np.random.rand(p,n)*10
45      x=np.random.rand(n)*0.1
46      v=np.random.rand(p)*0.1
47      b=np.matmul(A,x.reshape(n,1))
48      alpha=0.2
49      beta=0.8
```

(a) 标准Newton方法：首先确定A,b,x0的值，并进行回溯直线搜索即可，具体代码如下：

```
#标准Newton方法
x1=x.copy()
print("可行初始点Newton方法")
while True:
    errors1.append(f(x1))
    diff=df(x1)
    diff2=d2f(x1)
    d=np.matmul(np.linalg.inv(np.block([[diff2,A.transpose()],[A,np.zeros((p,p))]])),np.block([[-diff.reshape(n,1)],[np.zeros((p,1))]]))[:100]
    lambda2=np.matmul(np.matmul(d.reshape(1,n),diff2),d)
    if 0.5*lambda2<=0.0001:
        print("函数值:{}".format(f(x1)))
        print("x:{}".format(x1))
        break
    t=1
    while f(x1+t*d.reshape(n))>f(x1)-alpha*t*lambda2:
        t=beta*t
    x1+=t*d.reshape(n)

errors1 -= fmin

plt.plot(errors1, label='feasible: alpha=0.1, beta=0.7')

plt.xlabel('iteration')
plt.ylabel('error')
```

(b) 不可行点Newton方法:初始值与第一问相同，并进行回溯直线搜索，具体代码如下：

```
#不可行初始点Newton方法
x2=x.copy()
v2=v.copy()
print("不可行初始点Newton方法")
errors2 = []
while True:
    errors2.append(f(x2))
    diff=df(x2)
    diff2=d2f(x2)
    r=np.block([[diff.reshape(n,1)+np.matmul(A.transpose(),v2.reshape(p,1))],[np.matmul(A,x2.reshape(n,1))-b]])
    d=np.matmul(np.linalg.inv(np.block([[diff2,A.transpose()],[A,np.zeros((p,p))]])),-r)
    dx=d[:100]
    dv=d[100:]
    if np.linalg.norm(r)<=0.0001:
        print("函数值:{}".format(f(x2)))
        print("x:{}".format(x2))
        print("拉格朗日乘子v:{}".format(v2))
        break
    while np.linalg.norm(np.block([[df(x2+t*dx.reshape(n)).reshape(n,1)+np.matmul(A.transpose(),(v2.reshape(p,1)+t*dv))],
                                   [np.matmul(A,(x2.reshape(n,1))+t*dx)-b]]))>(1-alpha*t)*np.linalg.norm(r):
        t=beta*t
    x2+=t*dx.reshape(n)
    v2+=t*dv.reshape(p)

errors2 -= fmin
```

(c) 对偶Newton 方法：计算新的对偶函数的梯度和Hessain矩阵，并进行求解，具体代码 如下：

```
#对偶Newton方法
x3=x.copy()
v3=v.copy()
print("对偶Newton方法")
value = []
while True:
    value.append(-g(v3, A,b))
    tmp=-np.matmul(A.transpose(),v3.reshape(p,1))-1
    grad_=np.array([math.exp(i) for i in tmp])
    grad=b-np.matmul(A,grad_.reshape(n,1))
    hessian=np.matmul(np.matmul(A,np.diagflat(grad_)),A.transpose())
    d=-np.matmul(np.linalg.inv(hessian),grad)
    lambda2=np.matmul(np.matmul(grad.transpose(),np.linalg.inv(hessian)),grad)
    if 0.5*lambda2<=0.0001:
        print("函数值:{}".format(-np.matmul(b.transpose(), v3.reshape(p, 1)).item() - grad_.sum()))
        print("x:{}".format(x3))
        print("拉格朗日乘子v:{}".format(v3))
        break
    t=1
    while np.matmul(b.transpose(),v3.reshape(p,1)+t*d)+\
            sum([math.exp(i) for i in -np.matmul(A.transpose(),v3.reshape(p,1)+t*d)-1])>=\
            np.matmul(b.transpose(),v3.reshape(p,1))+sum([math.exp(i) for i in -np.matmul(A.transpose(),
                                                                                         v3.reshape(p,1))-1])+alpha*t*np.matmul(grad.reshape(1,p),d):
        t=beta*t
    v3+=t*d.reshape(p)
```

3.结果分析

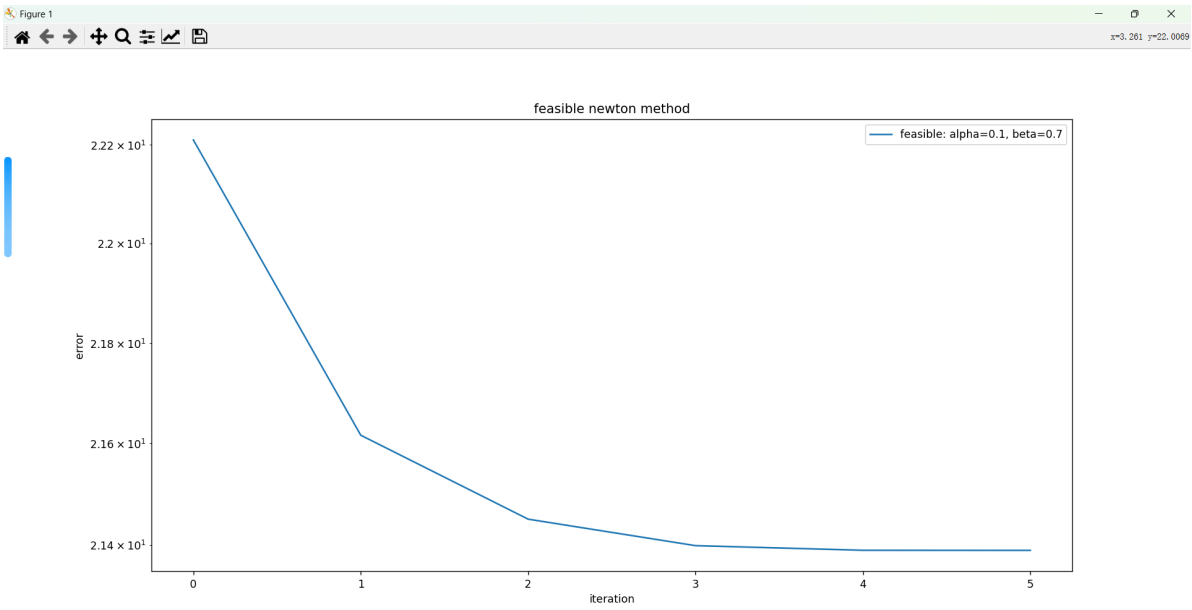


Figure 1
x=3.261 y=22.0069
feasible newton method
feasible: alpha=0.1, beta=0.7
error
iteration

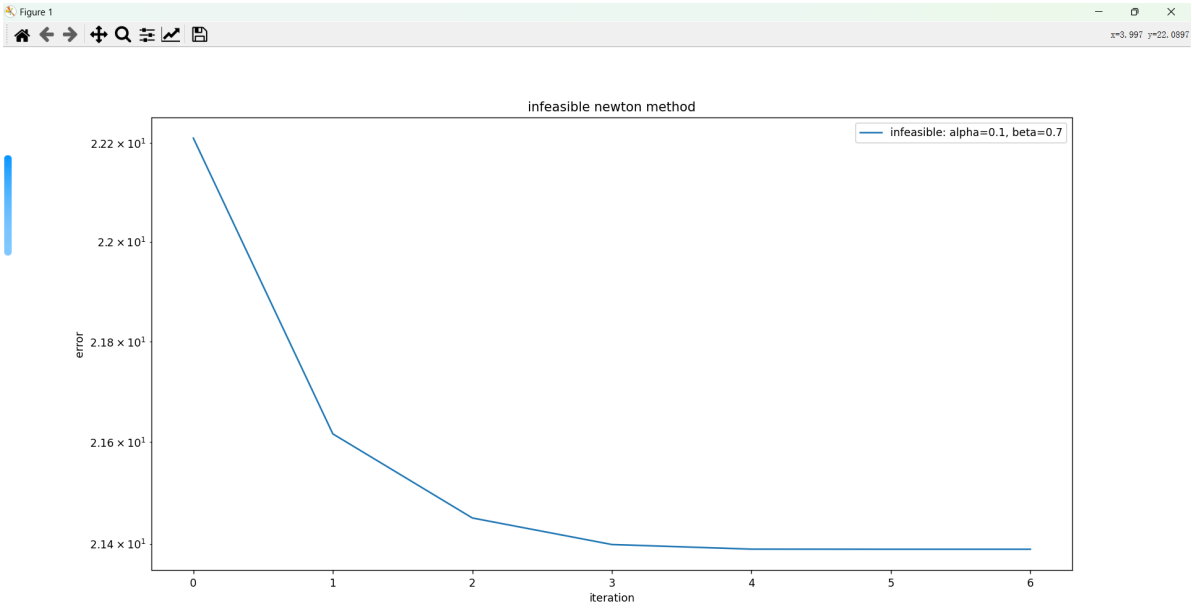


Figure 1
x=3.997 y=22.0897
infeasible newton method
infeasible: alpha=0.1, beta=0.7
error
iteration

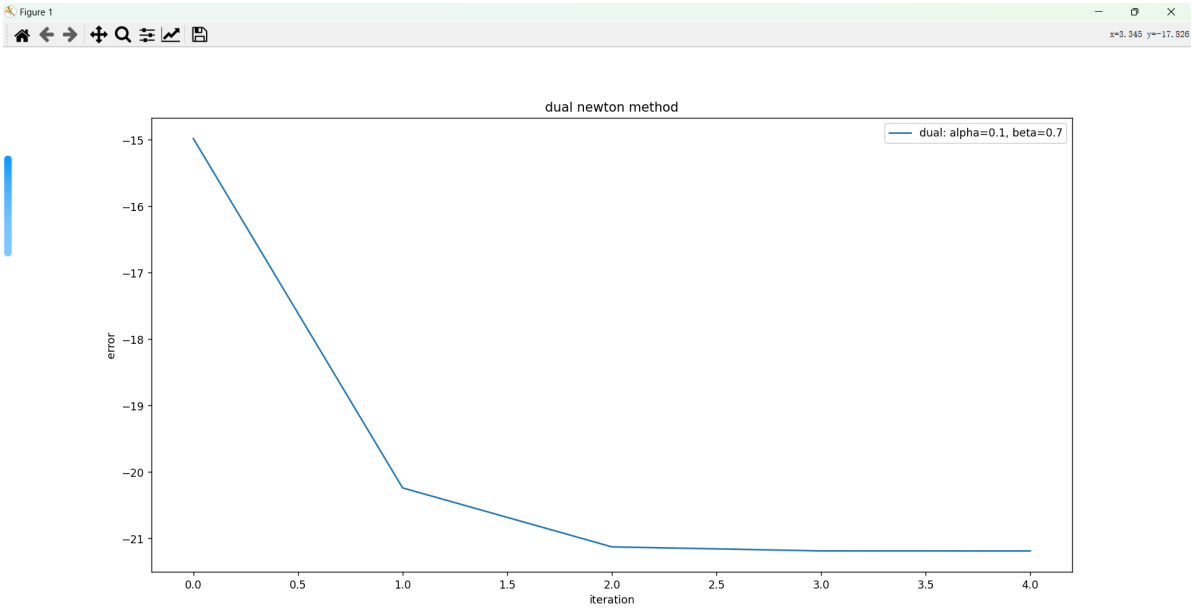


Figure 1
x=3.345 y=-17.826
dual newton method
dual: alpha=0.1, beta=0.7
error
iteration

```
可行初始点Newton方法
函数值:-15.199787636537602
x:[0.05450898 0.04505763 0.04760358 0.03436298 0.05051661 0.03885682
 0.06618052 0.02950261 0.05792658 0.0462313  0.03844978 0.01974521
 0.04932296 0.03135709 0.03711586 0.04672348 0.04388756 0.0890104
 0.04724859 0.04787639 0.0375107  0.06364131 0.04921528 0.10516022
 0.05113229 0.05941975 0.09381693 0.03563944 0.06532693 0.03942636
 0.0426281  0.05183439 0.04307489 0.0413667  0.04146803 0.03721055
 0.06165093 0.04820693 0.04944241 0.07626992 0.03570815 0.06490797
 0.04573547 0.07075127 0.08320197 0.04233392 0.0516568  0.03923292
 0.03672885 0.0404469  0.05807615 0.04897527 0.04520475 0.05018853
 0.12075081 0.09958677 0.03096714 0.03580574 0.04203884 0.02704605
 0.04425564 0.02387388 0.04227918 0.0473866  0.03950705 0.0522236
 0.03775617 0.0572656  0.05294606 0.0427892  0.07360691 0.05969181
 0.03661498 0.03021394 0.0227921  0.07379691 0.09621189 0.05134784
 0.08894424 0.04209579 0.06259779 0.03328331 0.04445472 0.05676424
 0.04109747 0.05314334 0.02975431 0.06502077 0.12290232 0.03621326
 0.12797312 0.05995201 0.03865758 0.03544833 0.09509679 0.04581517
 0.07233131 0.09422828 0.05324662 0.0444809 ]
```

```
不可行初始点Newton方法
函数值:-15.199805871173519
x:[0.05449015 0.04502606 0.04755054 0.03435846 0.05050006 0.03873945
 0.066074   0.02942294 0.05791613 0.04625116 0.03842923 0.01970211
 0.0493467  0.03134228 0.03711416 0.0466963  0.04395158 0.08899261
 0.04715058 0.04780104 0.03742034 0.06371775 0.04912903 0.10513446
 0.05119451 0.05942058 0.09385861 0.03562894 0.0653429  0.0394202
 0.04258201 0.05183543 0.04307317 0.04131468 0.04144963 0.03730267
 0.06171878 0.04829873 0.04941239 0.07641806 0.03561897 0.06489567
 0.04572734 0.07064193 0.08322775 0.04233676 0.05166641 0.03916562
 0.03662471 0.040373   0.05810477 0.04893447 0.04524943 0.05010792
 0.12089627 0.09950569 0.03091969 0.03581575 0.04208871 0.02701374
 0.04425058 0.02387532 0.04226202 0.04741876 0.03960315 0.05223942
 0.03880218 0.05722835 0.05300343 0.04281657 0.07371137 0.05968488
 0.03651014 0.0301903  0.02275185 0.07381562 0.09608246 0.05134378
 0.08891727 0.04224953 0.06242164 0.03321277 0.04451151 0.05680458
 0.04111243 0.05320608 0.02973727 0.06491742 0.12288345 0.03623245
 0.12795307 0.05987851 0.03868639 0.0353907  0.09523174 0.04574177
 0.0722096  0.09434348 0.05323475 0.04449655]
拉格朗日乘子v:[-0.42959289  0.05420025  0.17473945  0.16418734  0.10238618  0.27278064
  0.29546147  0.48806357 -0.04955737  0.47887701  0.52780544  0.33905149
 -0.02586492 -0.26552182 -0.01858462 -0.01585899  0.13637521 -0.14758676
  0.35808202 -0.18015605  0.0812936   0.37969652  0.30775443  0.38715724
  0.2943561   0.14511125 -0.19672788  0.26951037  0.15909445 -0.05189269]
```

```
对偶Newton方法
函数值:-15.199806169105566
x:[0.09365785 0.08695555 0.04675593 0.07245249 0.08667979 0.02713237
 0.07911065 0.00503892 0.051661   0.03708376 0.02943351 0.01271756
 0.05732927 0.04804429 0.03699153 0.08323931 0.05503367 0.07780739
 0.07270018 0.09076052 0.04935468 0.03393185 0.04655515 0.03907125
 0.03999455 0.08540156 0.09605476 0.05126004 0.09740517 0.0888109
 0.0131237  0.0560611  0.04894175 0.07657025 0.04140137 0.04849692
 0.08693473 0.03111868 0.00919013 0.09359911 0.03368965 0.0095742
 0.07805799 0.07788702 0.06914618 0.05392007 0.05539108 0.08741545
 0.0300913  0.08315218 0.08039648 0.02610782 0.07719689 0.06956445
 0.08976846 0.0756526  0.03310006 0.07596267 0.01225372 0.04898808
 0.00914991 0.04378818 0.02686448 0.00744126 0.0742123  0.05281836
 0.00080726 0.02511918 0.00407721 0.0062046  0.08430626 0.08355137
 0.05029474 0.00611249 0.00719994 0.05463936 0.08692918 0.02146313
 0.03534297 0.00160009 0.06001273 0.04803768 0.01404932 0.0674735
 0.02991388 0.06194564 0.0334535  0.06018795 0.09172782 0.04417247
 0.09246794 0.09687902 0.00576031 0.01030671 0.09142483 0.06442638
 0.09090313 0.03170796 0.04243645 0.01032126]
拉格朗日乘子v:[-0.42916184  0.05424378  0.17461014  0.16415029  0.10240701  0.27260812
  0.29513524  0.48795045 -0.04955704  0.47852299  0.52762905  0.33890388
 -0.02581429 -0.26509384 -0.0184235  -0.01586077  0.13637249 -0.14745458
  0.35781906 -0.1798529   0.08134433  0.37959853  0.30753711  0.3870013
  0.29427544  0.14516064 -0.19660783  0.26945975  0.15887878 -0.05175165]
```

从结果可以看出，

迭代次数: 对偶 Newton 方法<标准 Newton 方法<不可行初始点 Newton 方法，且得到相同的最优值。

同时对于不可行初始点 Newton 方法、对偶 Newton 方法打印了每一步迭代时 Lagrange 乘子v 的值，可以看出最终得到相同的 Lagrange 乘子v的值