第二次大作业

计试 001 苏悦馨 2204120515

2022 年 11 月 17 日

1 P34 页大作业

1.1 问题描述

对于两个变量的二次规划问题：$\min f(x) = \frac{1}{2}(x_1^2 + \gamma x_2^2), \gamma > 0$。初始值为 $x^0 = (\gamma, 1)$。采用精确直线搜索梯度下降，第 k 次迭代后

$$x^k = (\frac{\gamma - 1}{\gamma + 1})^k \begin{bmatrix} \gamma \\ (-1)^k \end{bmatrix}$$

编程证明：

a) $\gamma = 1$ 时，一次迭代可以取得最优解；

b) 对于 γ 离 1 不不远的情况，收敛速度很快；

c) 如果 $\gamma \gg 1$ 或 $\gamma \ll 1$ 时，收敛速度将会非常慢。

1.2 问题思路

代码思路：

a) 设置临界误差 $\epsilon = 1E - 10$，计算精确直线搜索梯度下降所需的搜索次数为

$$N = \lceil \frac{\log((f(x^0) - p*)/\epsilon)}{\log(1/c)} \rceil$$

```
######3-34.py中26-36行
# 迭代次数N
e = 1E-10
if gama[k] == 1:
N = 1
else:
if gama[k] > 1:
N = math.ceil(
math.log(0.5*(pow(gama[k], 2)+gama[k])/e)/math.log(gama[k]/(gama[k]-1)))
else:
N = math.ceil(
math.log(0.5*(pow(gama[k], 2)+gama[k])/e)/math.log(1/(1-gama[k])))
```

b) 绘制搜索空间的等高线图, 以便直观的看到迭代中 $f(x)$ 的减少量;

```
######3-34.py中9-22行
# 建立步长为0.01，即每隔0.01取一个点
step = 0.01
x1 = np.arange(-x1_bound[k], x1_bound[k], step)
x2 = np.arange(-x2_bound[k], x2_bound[k], step)

# 将原始数据变成网格数据形式
X, Y = np.meshgrid(x1, x2)
Z = 0.5*(X**2+gama[k]*(Y**2))

# 画出5条线，并将颜色设置为黑色
ax = plt.subplot(2, 2, k+1)
contour = plt.contour(X, Y, Z, 5, colors='k', linestyles='dashed')
# 等高线上标明z（即高度）的值，字体大小是10，颜色分别是黑色和红色
plt.clabel(contour, fontsize=10, colors='k')
```

c) 由所计算得到的迭代次数 N，以及问题中给出的 x^k 表达式 $x^k = (\frac{\gamma-1}{\gamma+1})^k \begin{bmatrix} \gamma \\ (-1)^k \end{bmatrix}$，在已绘制等高线的空间中绘出历次迭代的点 $x^k, (k = 0, 1, \dots, N)$。观察取定 γ 时迭代过程中点的分布，即可证明上述结论。

```
 ######3-34.py中37-39行
x_ = [((gama[k]-1)/(gama[k]+1))**i*gama[k] for i in range(N+1)]
y_ = [((gama[k]-1)/(gama[k]+1))**i*(-1)**i for i in range(N+1)]
plt.plot(x_, y_, 'go--', linewidth=1, markersize=3)
```

γ 取值：

- $\gamma = 1$；
- γ 离 1 不不远的情况，取值 $\gamma = 0.5, 2, 5$ 为例考察；
- $\gamma \gg 1$ 或 $\gamma \ll 1$ 的情况，取值 $\gamma = 0.01, 100$ 为例进行考察。

1.3 结果与分析

$\gamma = 0.5, 1, 2, 5$ 的迭代过程如下

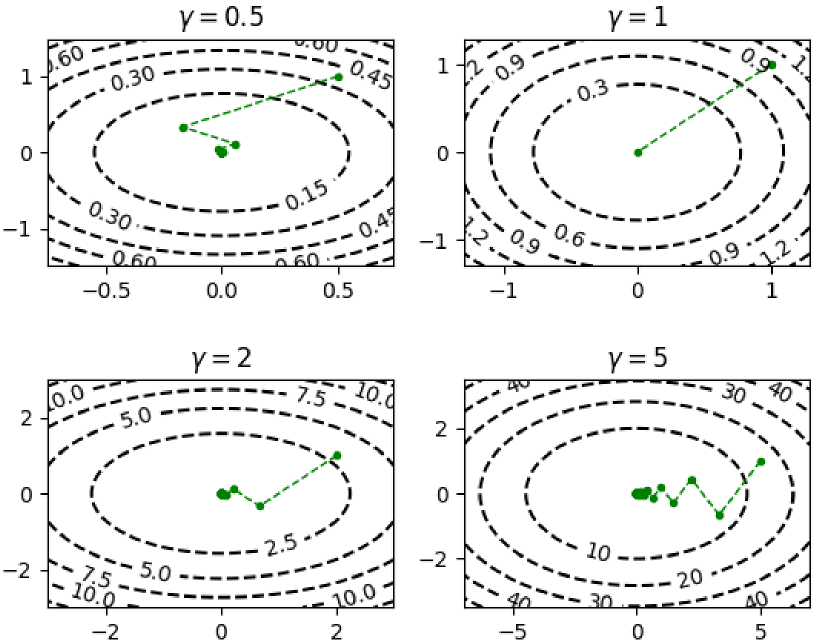

图 1: $\gamma = 0.5, 1, 2, 5$ 的迭代过程

$\gamma = 0.01, 100$ 的迭代过程如下

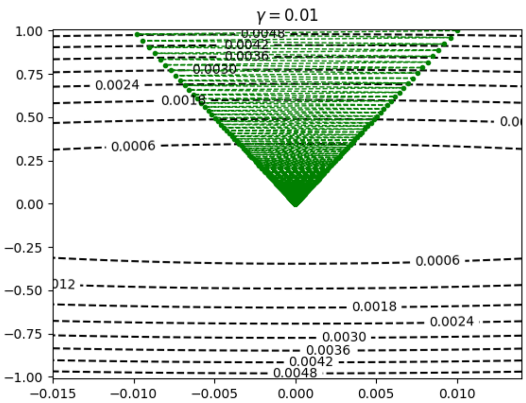

图 2: $\gamma = 0.01$

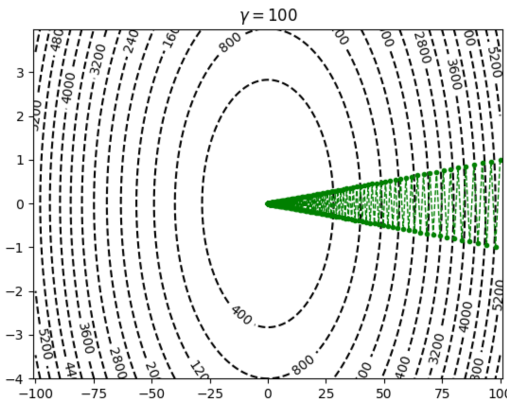

图 3: $\gamma = 100$

可以看出，$\gamma = 1$ 时，一次迭代可以取得最优解；对于 γ 离 1 不不远的情况，收敛速度很快，经过 10 次以内的迭代便可以收敛到最优解；而如果 $\gamma \gg 1$ 或 $\gamma \ll 1$ 时，迭代路径将通过震荡到达最优解，收敛速度非常慢，从而证明的了要求的结论

2 P38 页大作业

2.1 问题描述

对于 R^2 空间的非二次规划问题 $\min f(x)=e^{x_1+3x_2-0.1}+e^{x_1-3x_2-0.1}+e^{-x_1-0.1}$，分析回溯直线搜索采用不同的 α,β 值时，误差随迭代次数改变的情况。

2.2 问题思路

由梯度下降方法回溯直线搜索算法，下降方向 d^k 满足

$$d^k=-\nabla f(x)$$

易求得

$$\nabla f(x)=\begin{pmatrix} e^{x_1+3x_2-0.1}+e^{x_1-3x_2-0.1}-e^{-x_1-0.1} \\ 3e^{x_1+3x_2-0.1}-3e^{x_1-3x_2-0.1} \end{pmatrix}$$

且易求得，问题的最优解为 $\nabla f(x)=0$，解得 $(x1,x2)^T=\left(-\frac{1}{2}\ln 2,0\right)^T$。带入原函数得到问题的最优值为 $p^*=2.559266697$，从而可画出 $f(x^k)-p^*$ 随迭代次数改变的情况。

代码思路:

a) 使用匿名函数设置原函数和其梯度

```
%-----Backtracking_line_search.m中3-8行
%原函数
f=@(x1,x2) exp(x1+3*x2-0.1)+exp(x1-3*x2-0.1)+exp(-x1-0.1);
p=f(log(1/sqrt(2)),0);
%原函数的对x1的偏导
diff_f_1=@(x1,x2) exp(x1+3*x2-0.1)+exp(x1-3*x2-0.1)-exp(-x1-0.1);
%原函数的对x2的偏导
diff_f_2=@(x1,x2) 3*exp(x1+3*x2-0.1)-3*exp(x1-3*x2-0.1);
```

b) 外循环：判断 $\|\nabla f(x)\|_2<\epsilon$

```
%-----Backtracking_line_search.m中23-28行
nabla_f=[diff_f_1(x(1),x(2)),diff_f_2(x(1),x(2))]';
norm_f=norm(nabla_f);
%进行回溯直线搜索
while norm_f>e
```

c) 内循环：通过回溯直线搜索求解 t^k，使得 $f(x^k+t^kd^k)<f(x^k)+\alpha t^k\nabla f(x^k)^Td^k$

```
%-----Backtracking_line_search.m中31-34行
while f(x_(1),x_(2)) > f(x(1),x(2))-alpha(i)*t*norm_f^2
    t=beta(j)*t;
    x_=x-t*nabla_f;
end
```

d) 在每次内循环结束后，记录 $f(x^k)-p^*$，更新 $x^{k+1},\nabla f(x^{k+1}),\|\nabla f(x^{k+1})\|_2$

参数取值：

- 初始点 $x^0=(0.1,0.1)^T,(4,3)^T,(-89,-5)^T,(20,50)^T$
- $\alpha=0.01,0.05,0.1,0.3,0.45,0.49$
- $\beta=0.01,0.05,0.1,0.4,0.8,0.95,0.99$
- 停止准则 $\|\nabla f(x^k)\|_2<10^{-6}$

在对于初始点的选择上，选择在初始解附近以及离初始解较远的四个初始点，分别考察回溯直线搜索采用不同的 α,β 值时，误差随迭代次数改变的情况，以使结论具有普遍性。

对于确定的初始点 x^0，遍历 α,β 取值的所有组合，来分析回溯直线搜索采用不同的 α,β 值时，误差随迭代次数改变的情况。下图中纵坐标为'$f(x^k)-p*$'，横坐标为迭代次数，纵坐标采用对数坐标。

由于在 β 较小时，β 的改变对迭代次数的影响很大，故将不同的 α,β 组合的误差随迭代次数变化分为 3 个子图，迭代次数小于 100 次，迭代次数大于 400 次，以及迭代次数在 100 到 400 之间，三个区间分别做图。

2.3 结果与分析

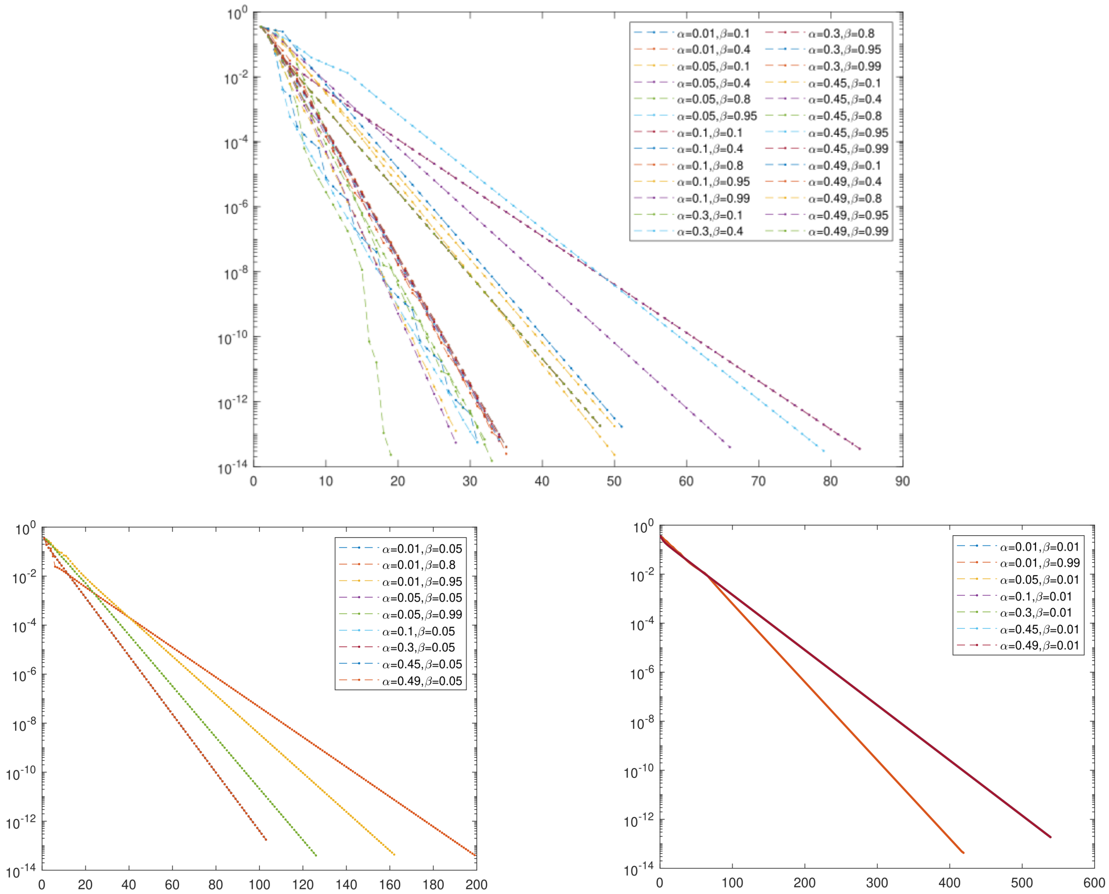

图 4: 初始点 $x_0 = (0.1, 0.1)^T$

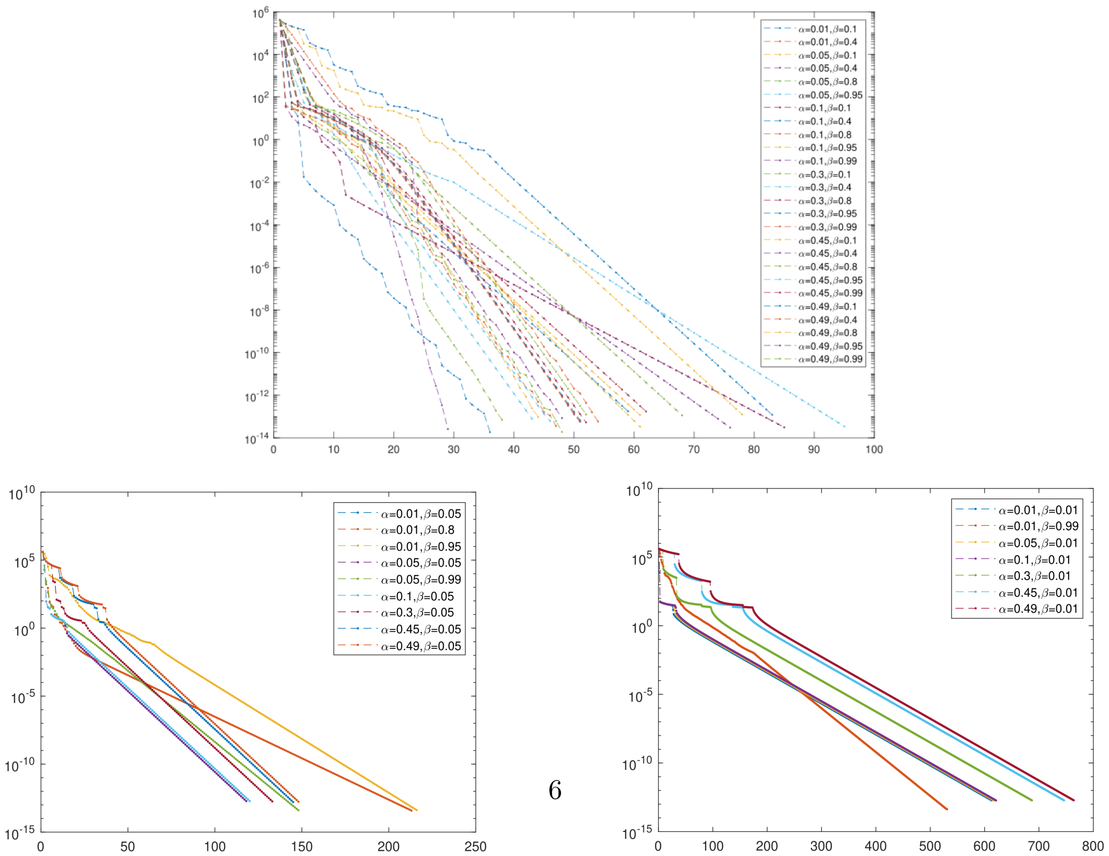

图 5: 初始点 $x_0 = (4, 3)^T$

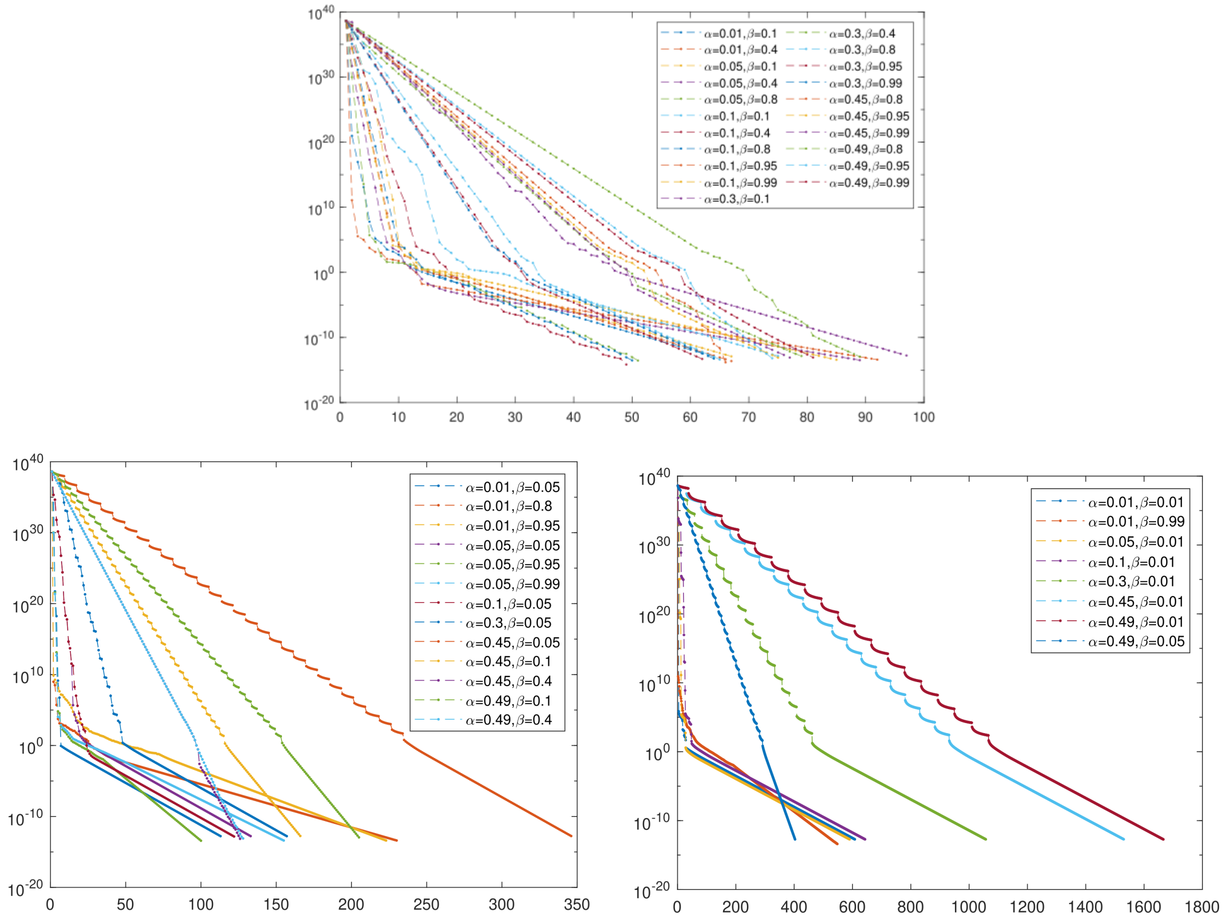

图 6: 初始点 $x_0 = (-89, 5)^T$

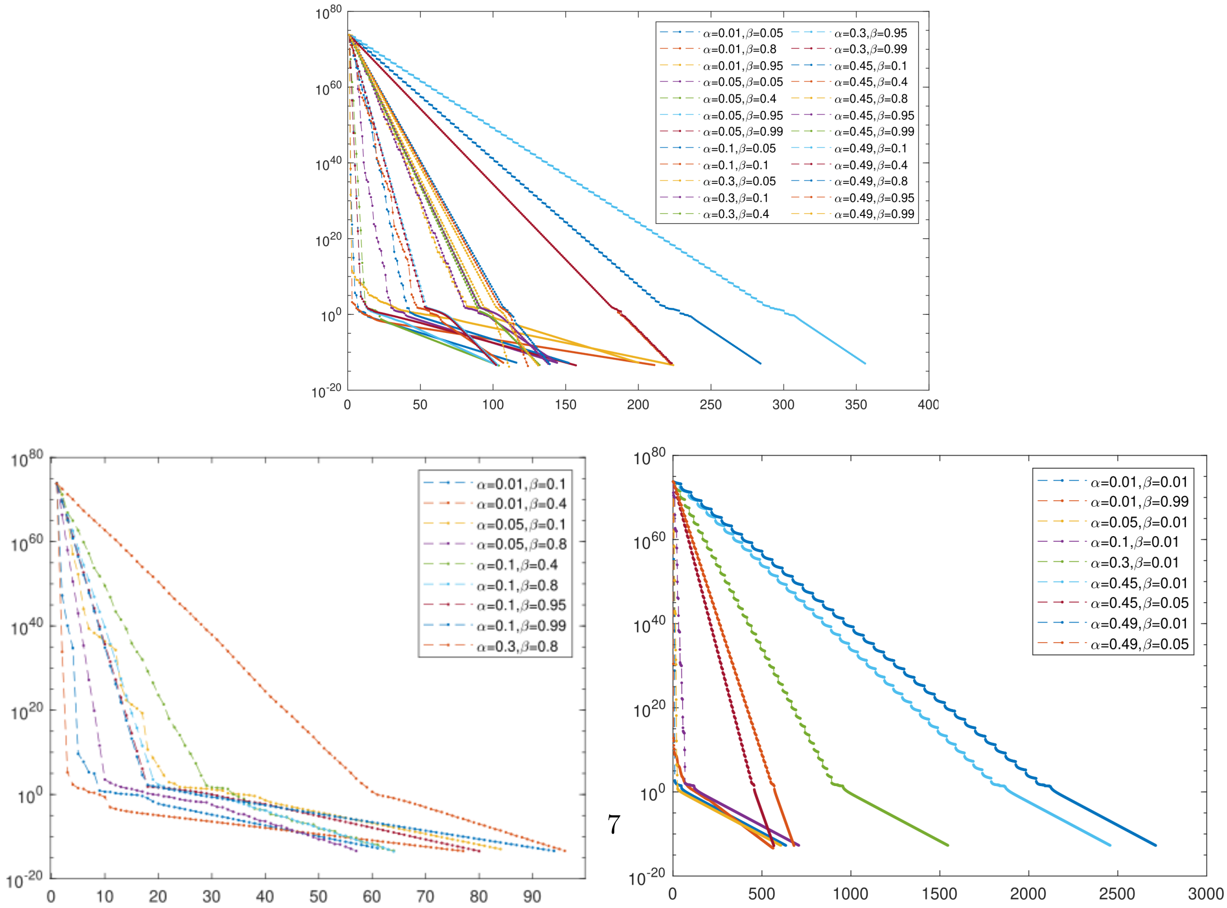

图 7: 初始点 $x_0 = (20, 50)^T$

分析如下：

- 在 β 较小时 $(\beta<0.1)$，迭代次数会随着 β 的减小而显著增加；
- 在 α 取值接近 0 或 0.5，β 取值接近 0 或 1 时，迭代次数也会出现较明显的增加；
- 在 $\alpha\in[0.1,0,4]$, $\beta\in[0.1,0.9]$ 的情况下，同一初始点在不同的 α,β 取值情况下，迭代次数不会发生显著变化。

3 P77 页大作业

3.1 问题描述

对于 R^2 空间的非二次规划问题 $\min f(x)=e^{x_1+3x_2-0.1}+e^{x_1-3x_2-0.1}+e^{-x_1-0.1}$，分析使用 $Newton$ 下降方法，回溯直线搜索时 $\alpha=0.1,\beta=0.7$，误差随迭代次数改变的情况。

3.2 问题思路

由牛顿方法，下降方向 d_{nt}^k 满足：

$$d_{nt}^k=-(\nabla^2 f(x^k))^{-1}\nabla f(x^k)$$

牛顿减少量 $\lambda^2(x^k)$ 满足

$$\lambda^2(x^k)={d_{nt}^k}^T\nabla^2 f(x)d_{nt}^k$$

由 P38 页大作业所述，该问题的最优解为 $\left(-\frac{1}{2}\ln 2,0\right)^T$, 最优值为 $p^*=2.559266697$，从而由牛顿下降方法生成点列 $\{x^0,x^1,\cdots,x^k\}$，即可画出 $f(x^k)-p^*$ 随迭代次数改变的情况。

代码思路:

a) 使用 matlab 符号函数（syms）求出原函数 $f(x)$ 的梯度和 $Hessian$ 矩阵；

```
f= exp(x1+3*x2-0.1)+exp(x1-3*x2-0.1)+exp(-x1-0.1);%原函数
grad_f=[diff(f,x1),diff(f,x2)]';%函数f的梯度
hessian_f=hessian(f,[x1,x2]);%函数f的hessian矩阵
```

b) 使用 matlab 符号函数转匿名函数功能（matlabFunction），使其转为可代入值计算的匿名函数；

```
%将上面的符号函数都转为匿名函数
f=matlabFunction(f);
grad_f=matlabFunction(grad_f);
hessian_f=matlabFunction(hessian_f);
```

c) 外循环：判断 $\frac{1}{2}\lambda^2(x^k)<10^{-6}$

d) 内循环：通过回溯直线搜索求解 t^k，使得 $f(x^k+t^kd^k)<f(x^k)-\alpha t^k\lambda^2(x^k)$

```
while (f(x_(1),x_(2)) > (f(x(1),x(2))-alpha*t*lamda_2))
    t=beta*t;
    x_=x+t*d;
end
```

e) 在每次内循环结束后，记录 $f(x^k)-p^*$，更新 $x^{k+1},\lambda^2(x^k)$

```
%更新参量
x=x+t*d;
d=-hessian_f(x(1),x(2))\grad_f(x(1),x(2));%下降方向
lamda_2= d'*hessian_f(x(1),x(2))* d; %牛顿减少量的平方
k=k+1;
gap(k)=f(x(1),x(2))-p;
```

参数取值：

- $\alpha=0.1$
- $\beta=0.7$
- 停止准则 $\frac{1}{2}\lambda^2(x^k)<10^{-6}$
- 初始点为 $x_0=(2,5)^T$

3.3 程序运行结果

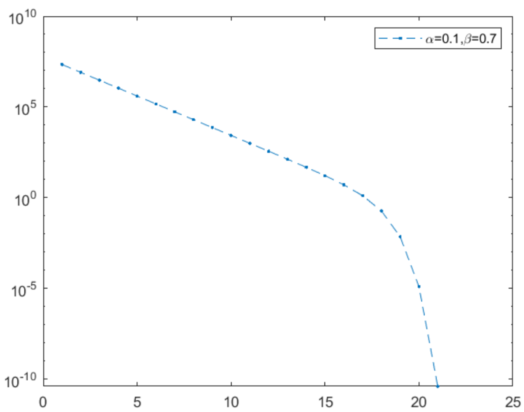

图 8: $Newton$ 下降方法，$\alpha=0.1,\beta=0.7$