数学实验第五次实验报告

计 76　张翔　2017011568

2020 年 4 月 8 日

1　实验目的

1. 掌握 Matlab 优化工具箱的基本用法, 对不同算法进行初步分析、比较。
2. 练习用无约束优化方法建立和求解实际问题的模型 (包括非线性最小二乘拟合)。
3. 掌握用 MATLAB 优化工具箱和 LINGO 解线性规划的方法。
4. 练习建立实际问题的线性规划模型。

2　Ch7-P5 原子位置

2.1　问题分析与模型建立

题目需要求出原子之间的位置关系，并且给出了某些原子对之间的距离。由于只需要求出相对位置，建立原子位置的平面直角坐标系，不失一般性，令原子 1 在原点 (0, 0) 处，其余原子的位置坐标为 (x_i, y_i) $(i = 2, 3, \cdots, 25)$。使用题给的原子距离关系可以构造非线性方程组，但方程组可能出现矛盾解，更好的方法是将问题转化为下述的无约束优化问题

$$\min_{x_2, y_2, \cdots, x_{25}, y_{25}} \sum_{(i,j)\in \mathbf{P}} ((x_i - x_j)^2 + (y_i - y_j)^2 - d_{ij}^2)^2$$

其中 $\mathbf{P}$ 是已知距离的原子对集合，d_{ij} 是原子 i 与 j 之间的距离。由此可以求解出尽量满足题给条件的原子坐标。

2.2　算法设计

上述是 48 变量的无约束优化问题，可以使用 MATLAB 的 `fminunc` 求解。求解时采用不同的搜索方向算法，如 BFG，DFP 或最速下降，可以比较不同算法的迭代次数，并取使得目标函数最小的变量值作为最终结果。计算时可以根据选择合适的迭代终止条件。

2.3　Matlab 程序

```
%% data
dist = [4, 1, 0.9607; 12, 1, 0.4399; ...];  % ignore data here


%% solver
% initial val
% x0 = -1 + 2 * rand(48, 1);
```

```matlab
x0 = ones(48, 1);
% quasi-netwon method
opt = optimset('HessUpdate', 'bfgs', 'MaxFunEvals', 1000000, ...
'MaxIter', 10000);  % 'dfp' or 'steepdesc'
[x, fval, exitflag, output] = fminunc(@(x)atom_dist(x, dist), x0, opt);

%% function that calc the distance squares
function y = atom_dist(x, data)
    y = 0;
    for k = 1 : length(data)
        i = data(k, 1);
        j = data(k, 2);
        dist = data(k, 3);
        x_i = 0;
        y_i = 0;
        x_j = 0;
        y_j = 0;
        if i ~= 1
            x_i = x(2 * i - 3);
            y_i = x(2 * i - 2);
        end
        if j ~= 1
            x_j = x(2 * j - 3);
            y_j = x(2 * j - 2);
        end
        y = y + ((x_i - x_j)^2 + (y_i - y_j)^2 - dist^2)^2;
    end
end
```

2.4 计算结果

2.4.1 初值选取

计算时，选取最大迭代次数 10^4，最大目标函数调用次数 10^6，选取不同初始值，结果如下

- 初值为 0 向量，不收敛；
- 初值为全 1 向量，不同算法的收敛速度和结果如下（表中 * 表示算法提前终止，不收敛）

表 1: 初值为 1 时不同算法的收敛情况

	迭代次数	目标函数调用次数	最终函数值
BFGS	152	7987	0.0482
DFP	2484*	122549*	0.1224
SteepDesc	710*	104664*	2.6499

这里的提前终止是因为迭代步长已经小于默认的最小步长了。

2.4.2 最终求解

可以看出初值的选取对最终的解影响较大，因此使用随机生成 $[-1, 1]$ 初值的方法，使用上述表现较好的 BFGS 算法，运行 1000 次，取最好的结果作为最终解。

1000 次采样中，目标函数调用次数与最终函数值的关系如下：

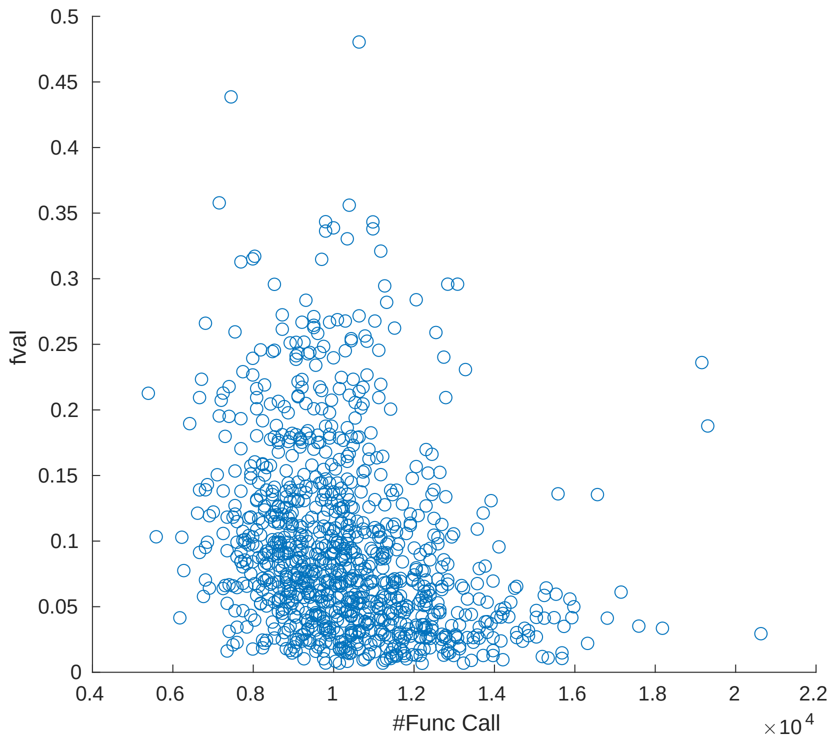

图 1: BFGS 算法得到的最终函数值与目标函数调用次数的关系

最后得到目标函数最小值为 $f_{min} = 0.0068$，此时各个原子的坐标如下表和下图所示

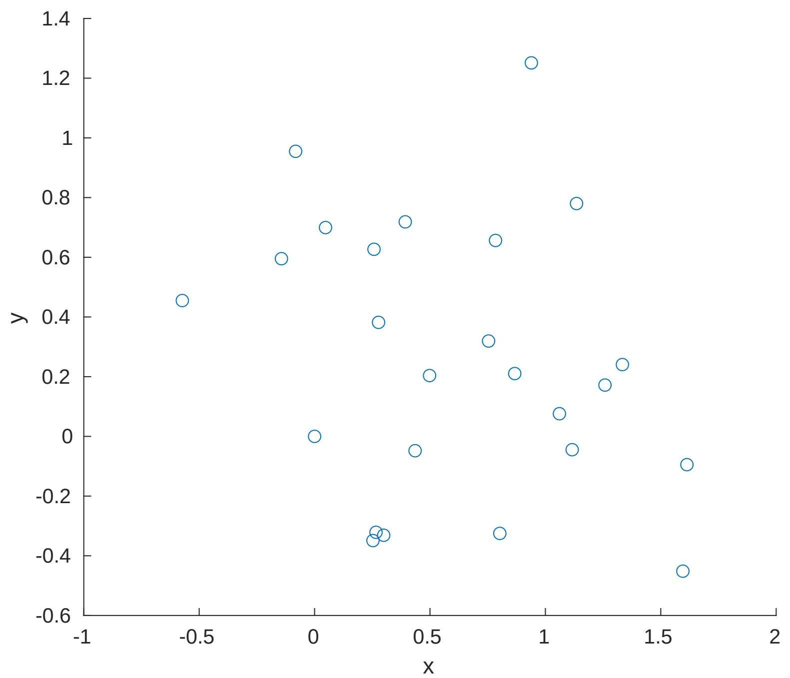

图 2: 原子的位置关系图

表 2: 各个原子的位置

原子编号	x	y	原子编号	x	y
1	0	0	14	0.2771	0.3819
2	0.7840	0.6563	15	-0.1435	0.5953
3	0.2662	-0.3213	16	0.9393	1.2513
4	-0.0822	0.9550	17	1.1350	0.7800
5	0.2575	0.6268	18	0.2529	-0.3491
6	1.1161	-0.0447	19	0.4354	-0.0481
7	0.4982	0.2037	20	0.8673	0.2104
8	0.7539	0.3195	21	1.2582	0.1719
9	0.0474	0.6995	22	0.8026	-0.3251
10	1.0609	0.0759	23	1.5958	-0.4516
11	1.6133	-0.0947	24	-0.5731	0.4550
12	0.2994	-0.3313	25	1.3337	0.2405
13	0.3928	0.7187			

2.5 结果分析

对比上述不同的初值、不同算法的选取以及相应的结果，可以看出求解本题时，使用 Quasi-Newton 方法时，采用 BFGS 公式能够用较少的迭代次数、目标函数调用次数来获得更优的目标函数值，而 DFP、SteepDesc 算法在该问题中表现不好，最后收敛很慢甚至不收敛。并且，初值的选取对该问题的解有较大的影响，初值选取不当将使得算法无法收敛，而收敛时也影响收敛结果，如图 (1) 中，目标函数调用次数少的情形可能反而比调用次数多的更优，某些不恰当的取值则使得解陷入局部最优，而整个散点图也呈现出较为随机的形态。不过总体来说，随着采样次数增加，获得最优解的概率将增大，因此上面取 1000 次采样的最好结果作为最终解输出。

2.6 结论

使用 BFGS 公式的 Quasi-Newton 方法可以求得原子的相对位置，见上表 (2)。

3 Ch7-P8 给药方案

3.1 问题分析与模型建立

题给情景可以用两室模型进行分析，设有一吸收室和中心室，它们的排出速率均与当前的浓度成正比，设二室的比例系数分别为 k_1, k，浓度分别为 c_1, c，体积为 V_1, V，那么有微分方程组

$$\begin{cases} \dfrac{\mathrm{d}c_1}{\mathrm{d}t} = -k_1 c_1,\, c_1(0) = \dfrac{d}{V_1} \\ \dfrac{\mathrm{d}c}{\mathrm{d}t} = -kc + \dfrac{V_1}{V} k_1 c_1,\, c(0) = 0 \end{cases}$$

d 为初始的服药量。由此解出

$$c(t) = \frac{d}{V}\frac{k_1}{k_1 - k}(e^{-kt} - e^{-k_1 t}) = b\frac{k_1}{k_1 - k}(e^{-kt} - e^{-k_1 t}) \tag{1}$$

用实验数据拟合 b, k_1, k 参数时，由于关系较为复杂，无法直接化为线性最小二乘的方法。可以将

它视作无约束优化问题，构造如下：

$$\min_{b,\,k,\,k_1} \sum_{i \in \mathbf{D}} \left(c_i(t_i) - b\frac{k_1}{k_1 - k}(e^{-kt_i} - e^{-k_1 t_i}) \right)^2$$

D 为实验得到的数据集。解上述问题，可以获得需要的参数值。

3.2 算法设计

上述问题可以当成普通的无约束优化问题，使用 MATLAB 的 `fminunc` 函数的 Quasi-Newton 方法求解。不过，注意到它同样是一个非线性最小二乘问题，可以使用 `lsqnonlin` 函数，它可以选择使用 Levenberg-Marquart 等方法进行求解。

3.3 Matlab 程序

下面的程序使用三种方法求解，分别是 Quasi-Newton, LM 和 Trust Region 方法（由于新版的 Matlab 已经弃用 Gauss-Newton，这里使用了新的 Trust Region 方法）。程序中的 `lsqcurvefit` 和 `lsqnonlin` 只是提供的接口不同，具体算法是相同的。

```
%% input data
data = [ 0.083, 10.9; 0.167, 21.1; 0.25, 27.3;
    0.50, 36.4; 0.75, 35.5; 1.0, 38.4;
    1.5, 34.8; 2.25, 24.2; 3.0, 23.6;
    4.0, 15.7; 6.0, 8.2; 8.0, 8.3;
    10.0, 2.2; 12.0, 1.8;];

%% solvers: fminunc
tol = 1e-8;

x0 = 10 * rand(3, 1);
opt = optimoptions(@fminunc, 'HessUpdate', 'bfgs', 'MaxFunEvals', 1000000, ...
'MaxIter', 10000, 'FunctionTolerance', tol);
[x1, resnorm1, exitflag1, output1] = fminunc(@(x)func(x, data), x0, opt);
residual1 = func_lsq(x1, data);

%% solvers: non-linear least squares
opt_lsq = optimoptions(@lsqnonlin,'Algorithm','levenberg-marquardt', 'FunctionTolerance', tol);
[x2, resnorm2, residual2, exitflag2, output2] = lsqnonlin(@(x)func_lsq(x, data), x0, [], [], opt_lsq);

% in newer Matlab we do not have G-N method
opt_curve_fit = optimoptions(@lsqcurvefit,'Algorithm','trust-region-reflective', 'FunctionTolerance', tol);

[x3, resnorm3, residual3, exitflag3, output3] = lsqcurvefit(@func_est, x0, data(:,1), data(:,2), [], [], opt_curve_fit);
```

```

%% plot
t = data(:, 1);
c_real = data(:, 2);
hold on;
% we can pickup either one of x because they are the same
fplot(@(t)func_est(x1, t), [0, max(data(:,1)) + 1], 'DisplayName', 'FittedCurve');
plot(t, c_real, 'x', 'DisplayName', 'Real Data');
legend('show');
xlabel('Time t');
ylabel('Drug Concentration c(t)');


%% params to determine
% estimated c(t) function
function F = func_est(x, t)
    b = x(1);
    k1 = x(2);
    k = x(3);
    F = b * k1 / (k1 - k) .* (exp(-k*t) - exp(-k1*t));
end

% residual, used by Least Squares
function F = func_lsq(x, data)
    F = func_est(x, data(:, 1)) - data(:, 2);
end

% L-2 norm, used by fminunc
function F = func(x, data)
    tmp = func_lsq(x, data);
    F = sum(tmp.^2);
end
```

3.4 计算结果与分析

严格来说，根据待求参数的物理含义，它们需要满足非负的条件，但求解时可以按非约束优化的方法，给定一个合适的非负初值（这里取 $[0, 10]$ 的随机数），然后验证解的合理性即可。

上述三种方法得到的结果相同，均为

$$b = 46.8275,\ k_1 = 3.6212,\ k = 0.2803$$

它们是满足物理含义的。其中 b 代表单位体积的药量，可以通过它来计算不同人需要的总药量。而 $k_1 > k$，表明单位体积吸收速率大于释放速率，这是血药浓度在一定时间内能维持较高水平的必要条件。由这些参数得到拟合函数 $\hat{c}(t)$，作图如下

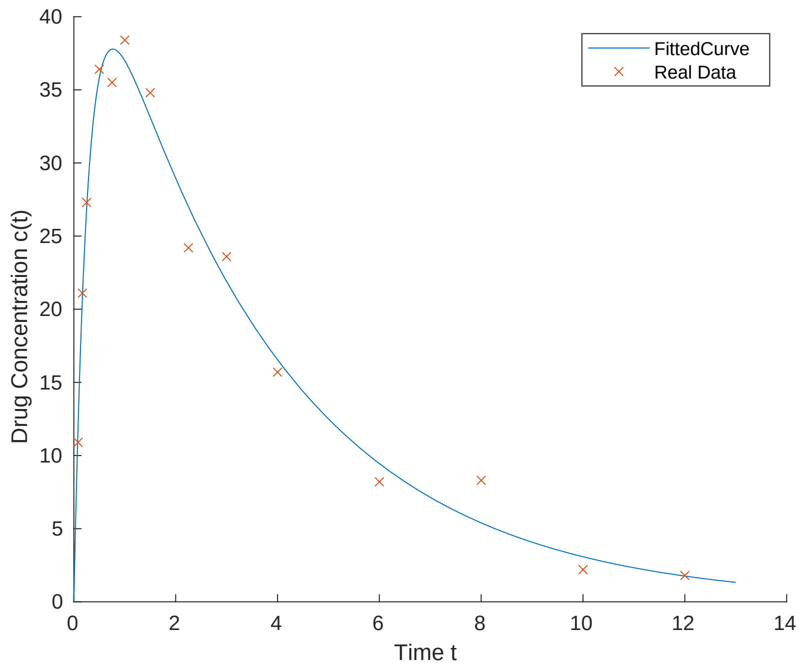

图 3: 拟合曲线与实际数据的比较

图线的形状符合常识，即药物在摄入初期引起血药浓度迅速上升，到达峰值后逐渐减少。使用拟合函数可以计算峰值血药浓度、保持有效血药浓度的最长时间，从而帮助病人更好地使用药物。

上述拟合函数在题给的各个数据点的残差如下

表 3: 在各点的残差

1.1090	-0.3885	-0.5052	-0.5820	2.2767	-1.4071	-1.6859
2.8020	-1.7058	0.8438	1.2452	-2.9075	0.8787	-0.0423

总误差的平方和 $\sum_i (c(t_i) - \hat{c}(t_i))^2$ 为 34.2317。可以看出，总误差偏大，结合上述拟合图像，可以看出测量得到的数据噪声比较大，如在浓度峰值附近，测量数据出现上下抖动的情况，一定程度上影响了最后的结果。考虑到实际测量时存在多种对结果有较大影响的因素，得到这样的测量结果是可以理解的。如果要减小误差，可以考虑增加测量次数，使得测量噪声对拟合曲线的影响更小。

比较三种算法的迭代次数、目标函数调用次数，有

表 4: 不同算法的迭代次数、函数调用次数

	#Iters	**#Func calls**
Quasi-Newton	15	169
Levenberg-Marquart	15	71
Trust Region	21	88

可以看出，在题给函数中，Quasi-Newton 与 LM 方法的迭代次数均要小于 Trust Region，收敛更快，而 LM 方法虽然迭代次数与 QN 相同，但目标函数调用次数要少得多，计算速度会更快。因此可以认为使用 LM 方法求解题给问题是最佳选择。

3.5 结论

拟合得到参数 $b = 46.8275, k_1 = 3.6212, k = 0.2803$。

4 Ch8-P6 投资

4.1 问题分析与模型建立

设经理购入 A-E 证券各为 x_A, x_B, x_C, x_D, x_E 万元，考虑最终纳税，总收益为

$$z = p_A x_A + \frac{1}{2} p_B x_B + \frac{1}{2} p_C x_C + \frac{1}{2} p_D x_D + p_E x_E$$

其中 $p_A, \cdots, p_E$ 表示各个证券的税前收益。
首先，有非负约束 $x_i \geq 0, i = A, \cdots, E$
约束条件 (1) 可以表示为

$$x_B + x_C + x_D \geq 400$$

约束条件 (2) 可以表示为

$$\frac{l_A x_A + l_B x_B + l_C x_C + l_D x_D + l_E x_E}{x_A + x_B + x_C + x_D + x_E} \leq 1.4$$

等价变形得到

$$(1.4 - l_A) x_A + (1.4 - l_B) x_B + (1.4 - l_C) x_C + (1.4 - l_D) x_D + (1.4 - l_E) x_E \geq 0$$

其中 $l_A, \cdots, l_E$ 表示各个证券的信用等级。
约束条件 (3) 可以表示为

$$\frac{y_A x_A + y_B x_B + y_C x_C + y_D x_D + y_E x_E}{x_A + x_B + x_C + x_D + x_E} \leq 5$$

等价变形得到

$$(5 - y_A) x_A + (5 - y_B) x_B + (5 - y_C) x_C + (5 - y_D) x_D + (5 - y_E) x_E \geq 0$$

其中 $y_A, \cdots, y_E$ 表示各个证券的到期年限。
对于第 (1) 问，增加约束

$$x_A + x_B + x_C + x_D + x_E \leq 1000$$

对于第 (2) 问，设借款资金 k 万元，则增加约束

$$k \leq 100$$

$$x_A + x_B + x_C + x_D + x_E - k \leq 1000$$

同时目标函数应修正为

$$z' = z - 0.0275k$$

4.2 算法设计

上述模型的所有约束均为线性不等式，目标函数也为线性，是典型的线性规划问题。因此使用 LINGO 的 Linear Programming 功能即可求解。对于第 (3) 问，需要对目标函数进行参数的敏感性分析，可以使用 LINGO 的 `Prices & Ranges` 进行处理。

4.3 LINGO 程序

LINGO 中默认变量非负，因此实际编程时无需显式写出非负约束。

第 (1)(3) 问代码如下：

```
MODEL:
TITLE Investment;
SETS:
sn/1..5/: P, L, Y, X, tax;
ENDSETS
DATA:
! PROFIT RATE;
P = 0.043, 0.054, 0.050, 0.044, 0.045;
! CONFIDENCE RATE;
L = 2, 2, 1, 1, 5;
! Exp Year;
Y = 9, 15, 4, 3, 2;
tax = 1, 0.5, 0.5, 0.5, 1;
ENDDATA
[OBJ] max = @sum(sn: tax * P * X);
[CONS1] x(2) + x(3) + x(4) >= 400;
[CONS2] @sum(sn: (1.4 - L) * X) >= 0;
[CONS3] @sum(sn: (5 - Y) * X) >= 0;
[CONS_TOT] @sum(sn: X) <= 1000;
END
```

第 (2) 问代码如下：

```
MODEL:
TITLE Investment;
SETS:
sn/1..5/: P, L, Y, X, tax;
ENDSETS
DATA:
! PROFIT RATE;
P = 0.043, 0.054, 0.050, 0.044, 0.045;
! CONFIDENCE RATE;
L = 2, 2, 1, 1, 5;
! Exp Year;
Y = 9, 15, 4, 3, 2;
tax = 1, 0.5, 0.5, 0.5, 1;
ENDDATA
[OBJ] max = @sum(sn: tax * P * X) - 0.0275 * k;
[CONS1] x(2) + x(3) + x(4) >= 400;
[CONS2] @sum(sn: (1.4 - L) * X) >= 0;
[CONS3] @sum(sn: (5 - Y) * X) >= 0;
[CONS_TOT] @sum(sn: X) <= 1000 + k;
```

```
[CONS_LEND] k <= 100;
END
```

4.4 计算结果与分析

(1) LINGO 输出显示使用了 LP 模型，符合预期。求解器迭代次数为 3，目标函数最优值为 29.836。其余变量求解如下

表 5: (1) 问求解结果

变量	**最优值**	**减少费用**
x_A	218.18	0
x_B	0	0.03
x_C	736.36	0
x_D	0	0.0006
x_E	45.45	0

其中 x_A, x_C, x_E 为基变量。条件 CONS2，CONS3，CONS _TOT 的松弛变量为 0，说明最优值的情况下这几个约束条件起作用。

(2) 类似于第 (1) 问，求解器迭代次数为 3，目标函数最优值为 30.070。其余变量求解如下

表 6: (2) 问求解结果

变量	**最优值**	**减少费用**
x_A	240.00	0
x_B	0	0.03
x_C	810.00	0
x_D	0	0.0006
x_E	50.00	0
k	100.00	0

其中 x_A, x_C, x_E, k 为基变量。条件 CONS2，CONS3，CONS_TOT，CONS_LEND 的松弛变量为 0，说明最优值的情况下这几个约束条件起作用。

(3) 使用 (1) 问的代码进行敏感度分析，得到下列结果

```
                    Objective Coefficient Ranges:

                    Current          Allowable         Allowable
    Variable      Coefficient         Increase          Decrease
     X( 1)        0.4300000E-01     0.3500000E-02     0.1300000E-01
     X( 2)        0.2700000E-01     0.3018182E-01      INFINITY
     X( 3)        0.2500000E-01     0.1733333E-01     0.5600000E-03
     X( 4)        0.2200000E-01     0.6363636E-03      INFINITY
     X( 5)        0.4500000E-01     0.5200000E-01     0.1400000E-01
```

根据上面的结果，如果证券 A 税前收益增加为 4.5%，系数增加量 0.2% 在变量 X(1) 的增加允许范围 0.35% 内，投资方案可以不变；当证券 C 的税前收益减少为 4.8%，系数减少量 $0.2\% \times \frac{1}{2}$ 超过了 X(3) 的减少允许范围 0.056%，投资方案需要改变。

另外，可以发现总收益对于证券 C 的收益率减少量最为敏感，而在投资时该种证券购买量最大；总收益对于证券 B、D 不敏感，投资时这两种证券的购买量均为 0；对于收益率增加量，总收益对证券 D 最敏感，如果它的收益率超出这个范围，则可能最优解要求购入一定量的证券 D 了。比如修改 D 的收益率为 4.53%，使其刚好超过增加允许范围，可以解出如下结果

表 7: 修改 D 的收益率后的结果

变量	最优值	减少费用
x_A	336.00	0
x_B	0	0.03
x_C	0	0.0001
x_D	648.00	0
x_E	16.00	0

显然，最优结果发生了较大变化，选择买入 A,D,E 而不是 A,C,E，且 A, E 的买入量也有了明显改变。这与敏感度分析得到的结论是一致的。

4.5 结论

1. 若经理有 1000 万资金，最佳选择投资 A, C, E 证券分别为 218.18 万元，736.36 万元，45.45 万元，其余证券不投资。最大收益 29.84 万元。

2. 若最大借贷额度 100 万元，应借贷 100 万元，投资 A, C, E 证券分别为 240 万元，810 万元，50 万元，其他证券不投资。最大收益 30.07 万元。

3. 若证券 A 税前收益增加为 4.5%，无需改变现有方案；当证券 C 税前收益减少为 4.8% 时，需要改变现有方案。

5 收获与建议

通过本次实验，我对 Matlab 以及 LINGO 编程解决非约束优化、线性规划问题更加熟练，并对相应知识有了更深刻的理解。建议：希望教材中能够增加对 Trust Region 优化方法的介绍，与时俱进。