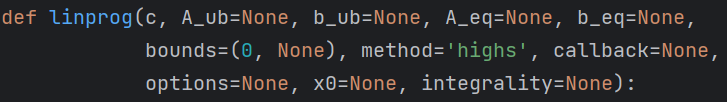
第1次作业\_python与优化

电 25 吴晨聪 2022010311

# （1） 查阅linprog函数的文档，了解linprog函数有哪些输入参数，说明上页提供给的代码求解的是一个怎样的线性规划问题

scipy.optimize.linprog 函数用于解决线性规划问题。在标准线性规划问题中，目标是最小化一个线性目标函数并且满足线性约束条件。其输入参数见下图:



其中:

**c:** 目标函数的系数向量;

**A\_ub:** 不等式约束的系数矩阵;

**b\_ub:** 不等式约束的右端项向量;

**A\_eq:** 等式约束的系数矩阵;

**b\_eq:** 等式约束的右端项向量;

**bounds:** 变量x的上下界;

**method:** 用于指定求解线性规划问题所采用的求解算法。

**Callback:** 允许在每次迭代时调用一个自定义的回调函数。

**options:** 控制求解器的行为。在本例中disp=True 表示输出求解器的迭代过程;

**x0:** 初始猜测值，用于指定优化过程中的初始点;

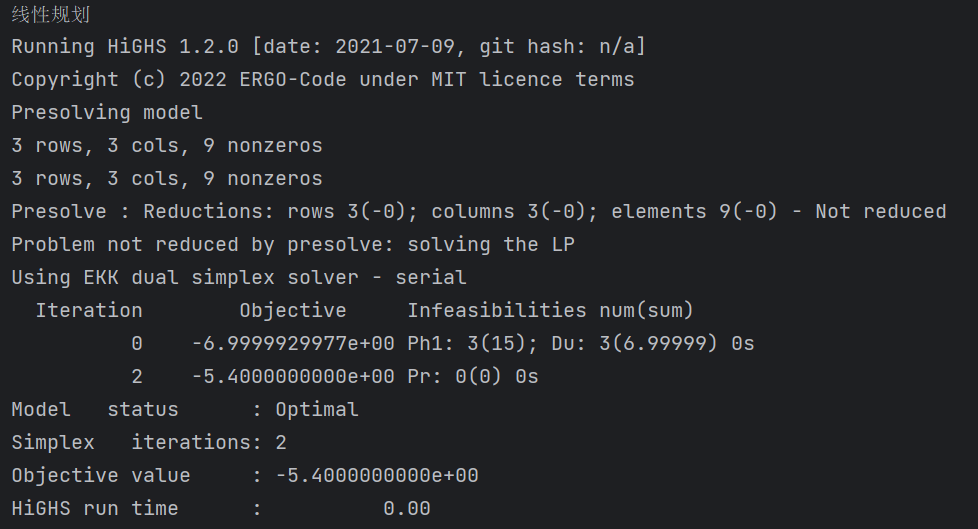
**integrality:** 用于指定整数规划问题中的变量类型。

提供代码所求解的线性规划问题具体形式为：

目标函数：

不等式约束条件：

Python运行结果:



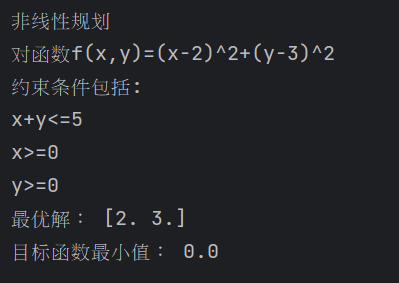
# （2） 调用了什么函数求解非线性规划问题，该函数所用的求解方法是什么

目标函数：

不等式约束条件：

代码中调用了 scipy.optimize.minimize 函数。minimize 可以求解不同类型的非线性优化问题，具体求解方法通过 method 参数指定，在本问题中调用了SLSQP（Sequential Least Squares Programming）即用于解决带约束的非线性优化问题。

Python运行结果:



# （3） 对于自己构造的非线性规划问题，判断python是否求得了其全局最优解

minimize 函数求解的是局部最优解，要判断是否为全局最优解，可以从判断目标函数是否为凸优化问考虑，因为对于凸优化问题，任何局部最优解都是全局最优解。这里的目标函数是一个二次函数，且是凸函数，因为它的二阶导数恒为正，同时其约束条件皆为线性条件，代表以上问题非线性规划问题是一个凸优化问题，因此可以判断针对以上目标函数Python 求解的解为全局最优解。

不过如果目标函数不为凸优化问题，则目标函数可能具有多个局部极小值，而minimize 函数所找到的解可能只是局部最优解。若想找出目标函数是非凸的全局最优解则需要调用Python中的其他函数如differential\_evolution 或 basinhopping等函数。