

Primal-dual 内点算法技术文档

姓名：陈泽豪 学号：SA22001009

2023 年 6 月 29 日

1 运行环境

本程序使用 python 3.9.16 进行实现，依赖如下模块进行实现：

- matplotlib version 3.6.2
- scipy version 1.10.0
- sympy version 1.11.1
- PyQt5-sip version 12.11.0
- numpy version 1.23.5

2 程序包含的文件介绍

- TestFuncClass.py

在这个文件内存储着一个 TestFunc 类，通过修改输入类的 test_func_str 来决定想要进行测试的函数，这里我提供了三个函数 test_1, test_2, test_3 来进行实验，类函数 test_func_val 返回进行测试的函数值，test_func_constraint 则提供了与这个函数挂钩的测试约束条件。

- JacobianEval.py

在这个文件内存储着 JacobianEval 类，这个类是一个符号计算类，通过调用 sympy 库进行特定函数的梯度，偏导计算，因此在此类内实现了返回一个函数的符号计算 Jacobian, hessian 的代码，可以依照类函数 n_jacobi, n_hessian 的方式将符号矩阵转变为数值矩阵。

- ui_convex.py

在这个文件内存储着 Ui_MyResult 类，这个类是 PyQt 用来设计 UI 界面的一个类，可以在类里定义 Widget，设置 UI 里的各种按钮，label 等元件。

- MyConvexSolver.py

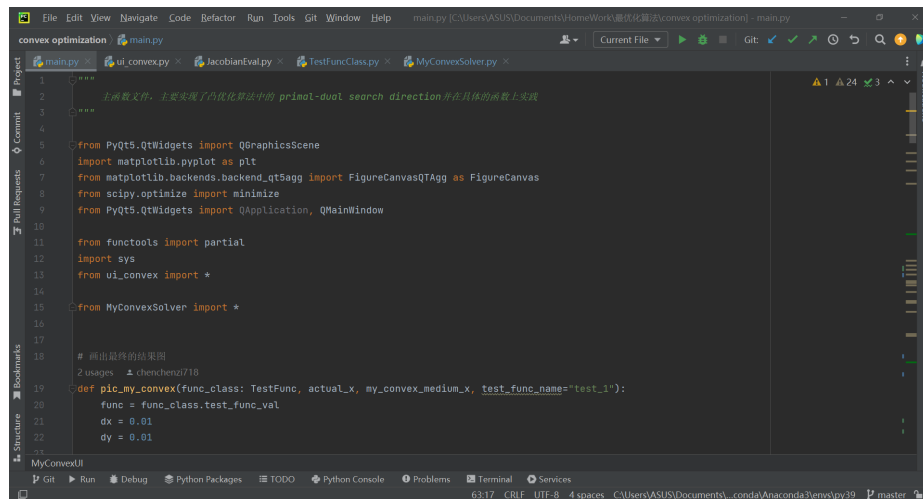
在这个文件内存储着我实现的求解凸优化问题的类，用来实现 Primal-dual 内点算法。类函数有例如计算了 dual residual, primal residual, central residual 的函数，还有实现了 Newton 迭代法的函数 newton_iteration，以及实现了一维线性搜索的函数 line_search，以及实现主体函数 primal_dual_convex_algorithm。

- main.py

在这个文件里主要完成了主程序以及编写了可视化函数 `pic_my_convex`，来得到报告中的二维区域上从初值靠近最终结果的过程；同时类 `MyConvexUI` 继承至文件 `ui_convex.py` 中的类，以实现 UI 界面上按钮的槽响应函数编写。

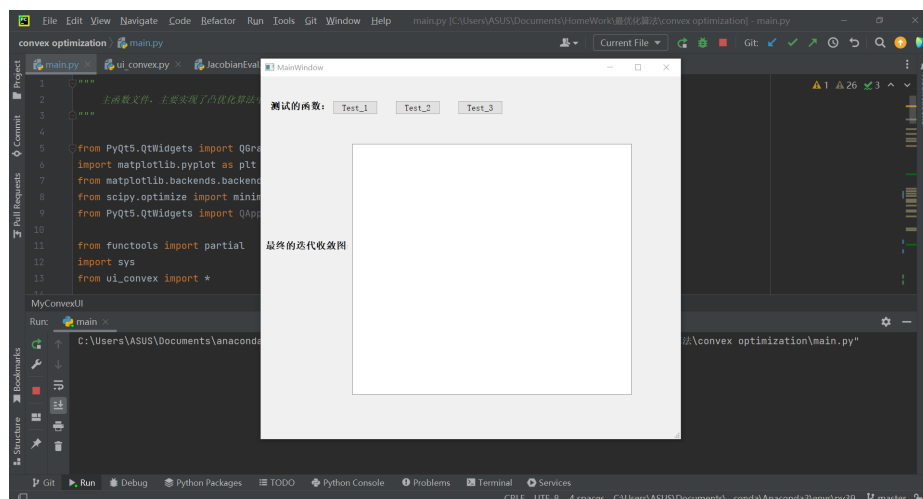
3 调用方法

下面演示一下代码运行步骤，首先打开 `main.py`，为如下的界面：

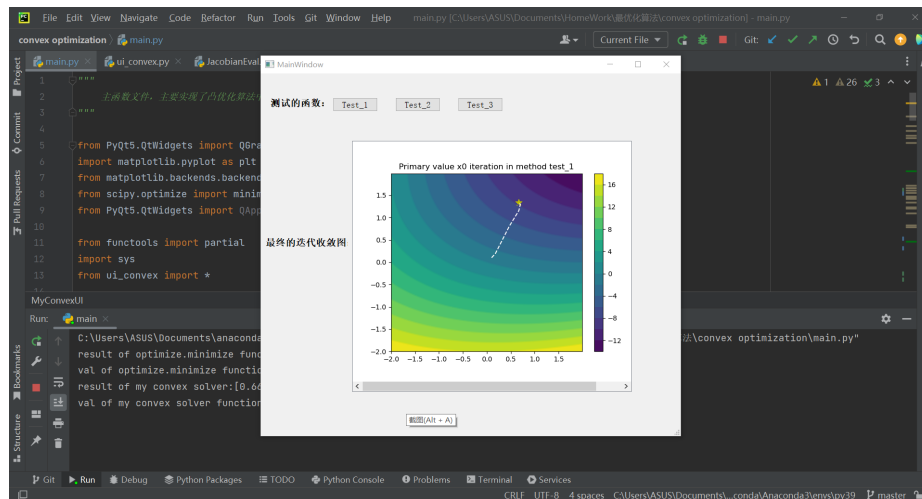


```
1 #####
2 上函数文件，主要实现了凸优化算法中的 primal-dual search direction 并在具体的函数上实践
3 #####
4
5 from PyQt5.QtWidgets import QGraphicsScene
6 import matplotlib.pyplot as plt
7 from matplotlib.backends.backend_qt5agg import FigureCanvasQTAgg as FigureCanvas
8 from scipy.optimize import minimize
9 from PyQt5.QtWidgets import QApplication, QMainWindow
10
11 from functools import partial
12 import sys
13 from ui_convex import *
14
15 from MyConvexSolver import *
16
17
18 # 画出最终的结果图
19 2 usages  chenchen0718
20 def pic_my_convex(func_class: TestFunc, actual_x, my_convex_medium_x, test_func_name="test_1"):
21     func = func_class.test_func_val
22     dx = 0.01
23     dy = 0.01
```

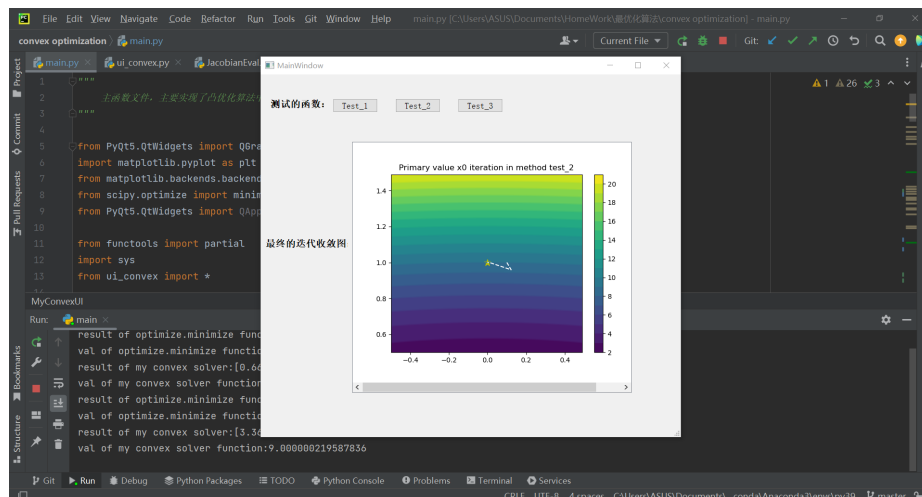
然后在保证 `pyqt` 安装的情况下直接点击运行得到如下的界面：



那么就可以通过点击上面的三个按钮得到不同的测试结果，例如我先点击 `test_1` 按钮，有如下结果：

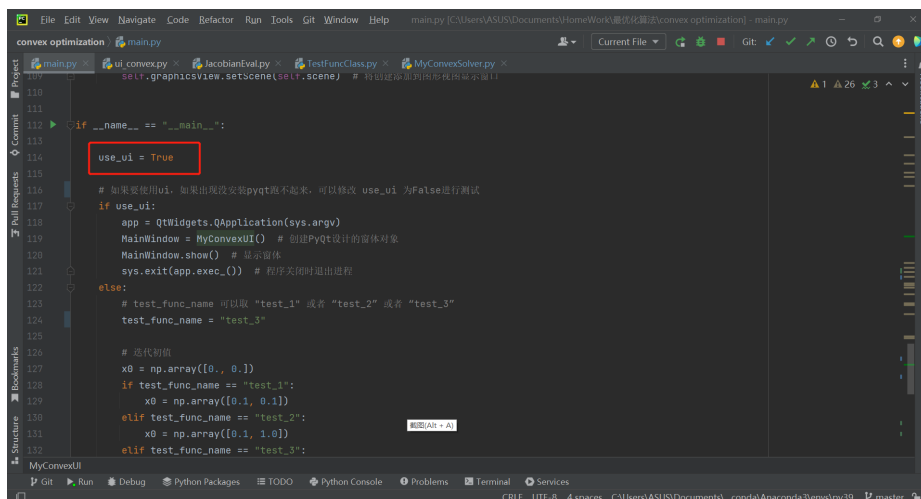


然后如果要查看 test_2 的结果，就再点击按钮 test_2 得到：



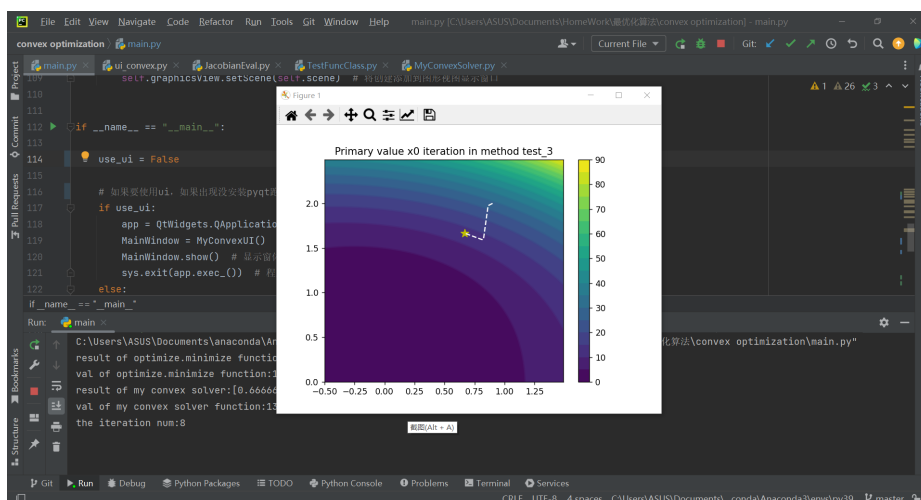
同理如果要查看 test_3 的结果，就点击按钮 test_3 即可。

如果实在无法安装 PyQt5 进行试验，那么程序也提供了直接进行实验的方法，修改 main 函数内部的参数 use_ui 为 False，如下图所示（下一页的第一张图）：



```
109 self.graphicsview.setscene(self.scene) # 将创建好的图形视图显示窗口
110
111
112 if __name__ == "__main__":
113     use_ui = True
114
115     # 如果要使用ui, 如果出现没安装pyqt跑不起来, 可以修改 use_ui 为False进行测试
116     if use_ui:
117         app = QtWidgets.QApplication(sys.argv)
118         MainWindow = MyConvexUI() # 创建PyQt设计的窗体对象
119         MainWindow.show() # 显示窗体
120         sys.exit(app.exec_()) # 程序关闭时退出进程
121     else:
122         # test_func_name 可以取 "test_1" 或者 "test_2" 或者 "test_3"
123         test_func_name = "test_3"
124
125         # 迭代初值
126         x0 = np.array([0., 0.])
127         if test_func_name == "test_1":
128             x0 = np.array([0.1, 0.1])
129         elif test_func_name == "test_2":
130             x0 = np.array([0.1, 1.0])
131         elif test_func_name == "test_3":
132             x0 = np.array([0.1, 1.0])
```

则此时直接点击运行就有：



然后关闭这张图后结束整个代码进程。如果还是无法运行，请助教私聊找我，我会进行自己代码的录屏演示。