

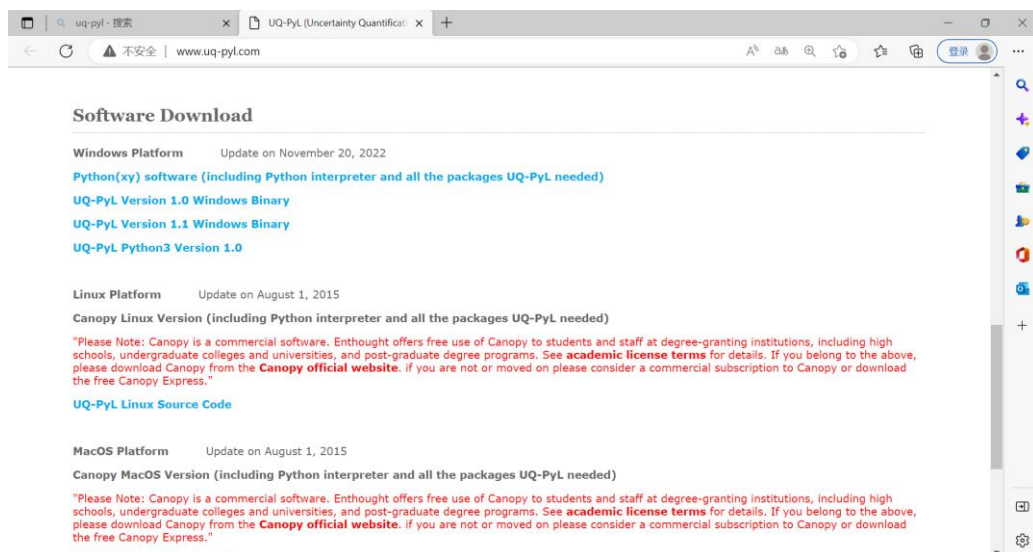
UQ-PyL 用于 SWAT 模型操作指南

UQ-PyL 是一个适用于多种模型的参数不确定性分析工具箱，包含多种不确定性分析方法，能够进行参数采样、构建替代模型、敏感性分析以及参数优化。本文主要介绍 UQ-PyL 用于 SWAT 模型参数不确定性分析的必要步骤，从 SWAT 模型构建完成后到如何利用 UQ-PyL 中的方法对模型进行参数不确定性分析。有关 UQ-PyL 的安装、方法简介以及各部分展示可在官方网站 <http://www.uq-pyl.com/> 下载 UQ-PyL User Manual version 进行查阅。

本文使用的 UQ-PyL 版本可从官网 <http://www.uq-pyl.com/> 下载 UQ-PyL Python3 Version; 相关说明可下载 UQ-PyL for SWAT model.pdf。

一、下载 UQ-PyL

- 1、打开网址 <http://www.uq-pyl.com/>，下载 UQ-PyL Python3 Version1.0 解压后安装至 D 盘（一定要是 D 盘）。



- 2、需要的 package: numpy、scipy、pandas、matplotlib、scikit-learn（python 推荐 3.7 版本）

- 3、注意：MARS 敏感性分析需要配置 py-earth，利用 Anaconda Prompt 进行配置：

①打开 Anaconda Prompt 后，激活使用的环境；

②输入 `conda install -c conda-forge sklearn-contrib-py-earth` 进行安装配置；

③将目录 `D:\UQ-PyL\UQRSmodel\pyearth` 的 `earth.py` 替换到运行环境下的 `pyearth` 中；

示例：运行环境的 `pyearth` 位置 `D:\Anaconda\envs\UQ-env\Lib\site-packages\pyearth`（该文件夹下有一个 `earth.py`；Anaconda 安装位置为 D 盘，运行环境名为 UQ-env）

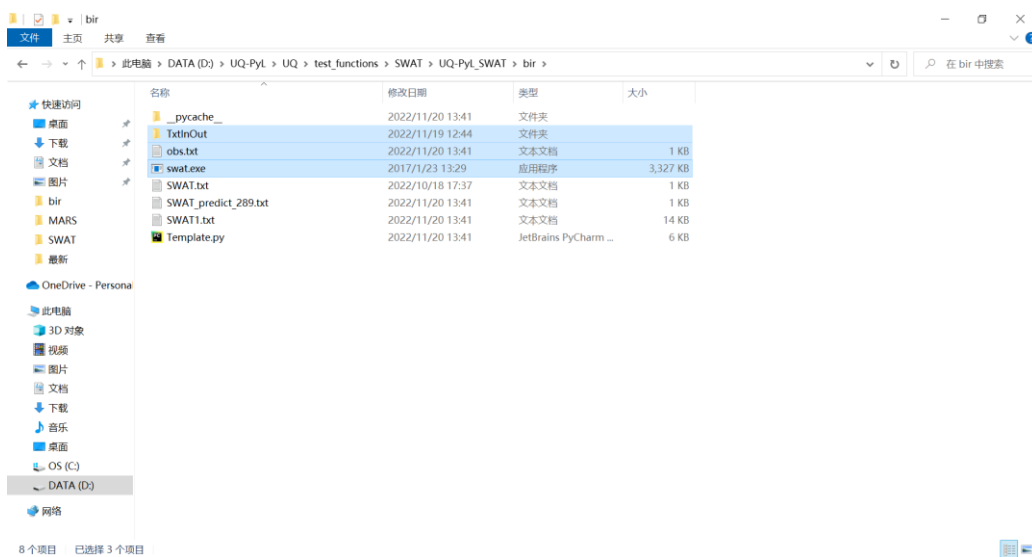
二、准备必要文件

1、对 SWAT 模型进行参数不确定性分析，必要文件有：参数文件、模型执行文件、观测数据。参数文件即构建 SWAT 模型后所得的 TxtInOut 文件；模型执行文件为 Arcswat 软件中的可执行文件，选择对应版本；观测数据保存为 obs.txt，格式如下：

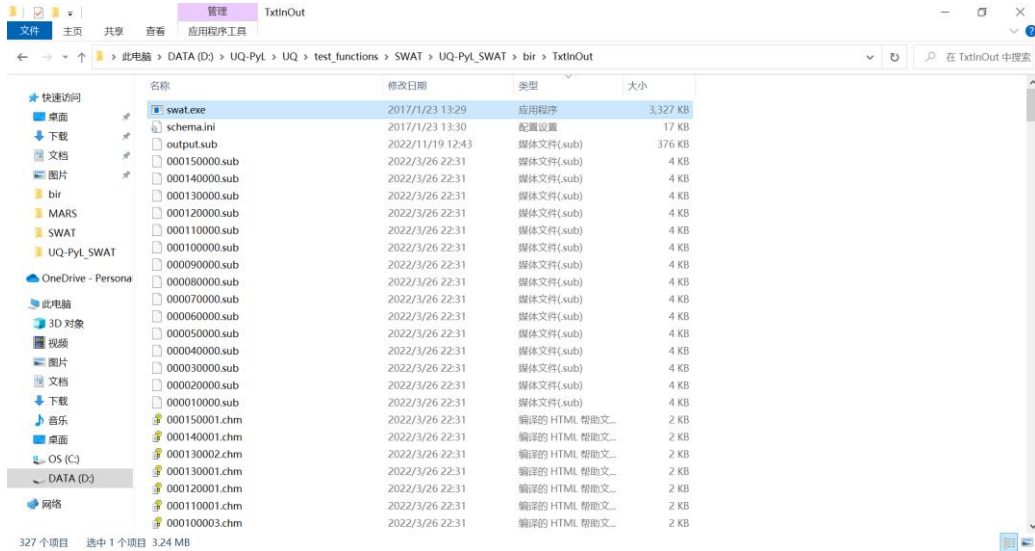


```
obs.txt - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
41.10
49.00
122.00
388.00
480.00
376.00
270.00
230.00
199.00
127.00
118.00
122.00
113.00
125.00
128.00
139.00
230.00
370.00
379.00
290.00
200.00
169.00
第 84 行, 第 8 列 100% Windows (CRLF) UTF-8
```

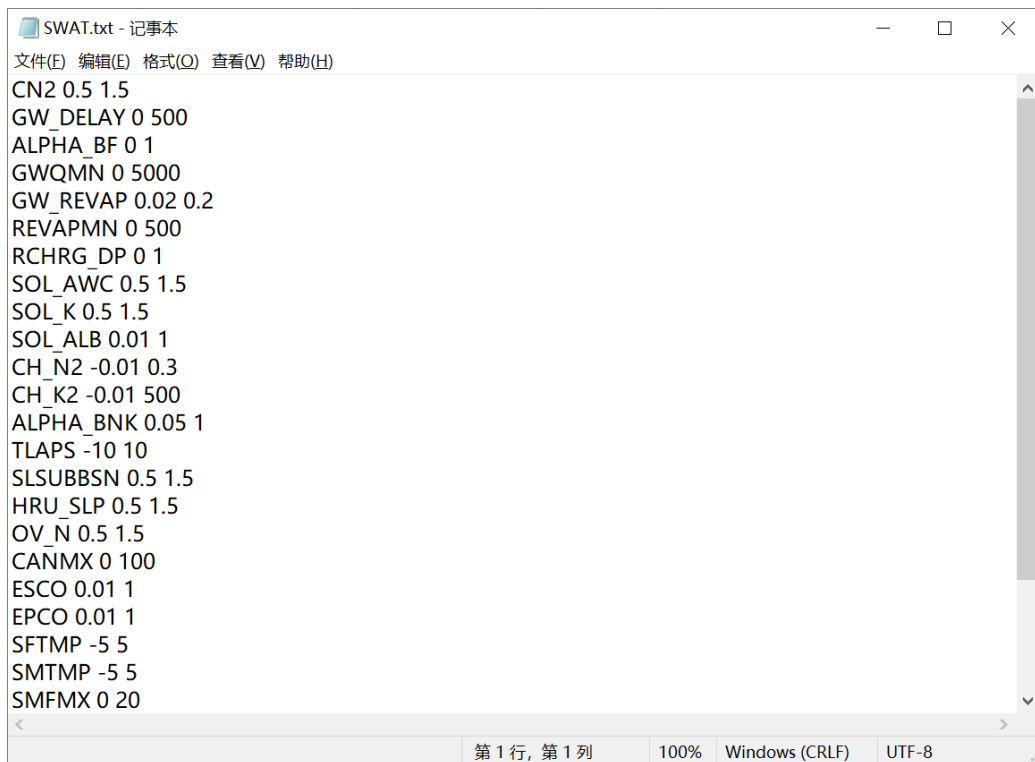
2、将三种文件复制到 D:\UQ-PyL\UQ\test_functions\SWAT\UQ-PyL_SWAT\bir 下（注意提前删除该文件夹下的对应示例文件），如下：



3、将 swat.exe 放入到 TxtInOut 中：



4、选择参数及范围，参数模板在 D:\UQ-PyL\UQ\test_functions\SWAT\UQ-PyL_SWAT\SWAT.txt，完成后，将选择后的 SWAT.txt 复制到 D:\UQ-PyL\UQ\test_functions\SWAT\UQ-PyL_SWAT\bir 下（注意严格遵守模板文件的格式“参数名 上限 下限”，不要多余空格或空行；目前可调参数数量有限，如需增加参数，请联系作者，后续会补充）

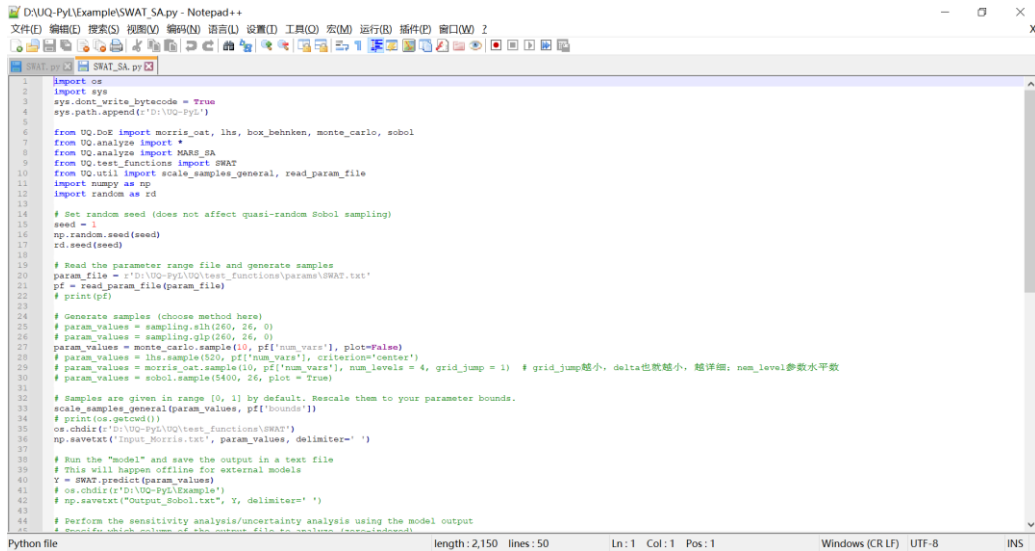


5、选择参数后，运行两次 D:\UQ-PyL\UQ\test_functions\SWAT\UQ-PyL_SWAT\main.py（该程序有两个参数，Path0 表示 main 所在的文件位置，Subbasin 表示模拟出口点所在的子流域；每一次选择参数后，需要重新运行两次该文件；一定要运行该程序两次，间隔 2s 左右）

三、选择方法进行参数不确定性分析

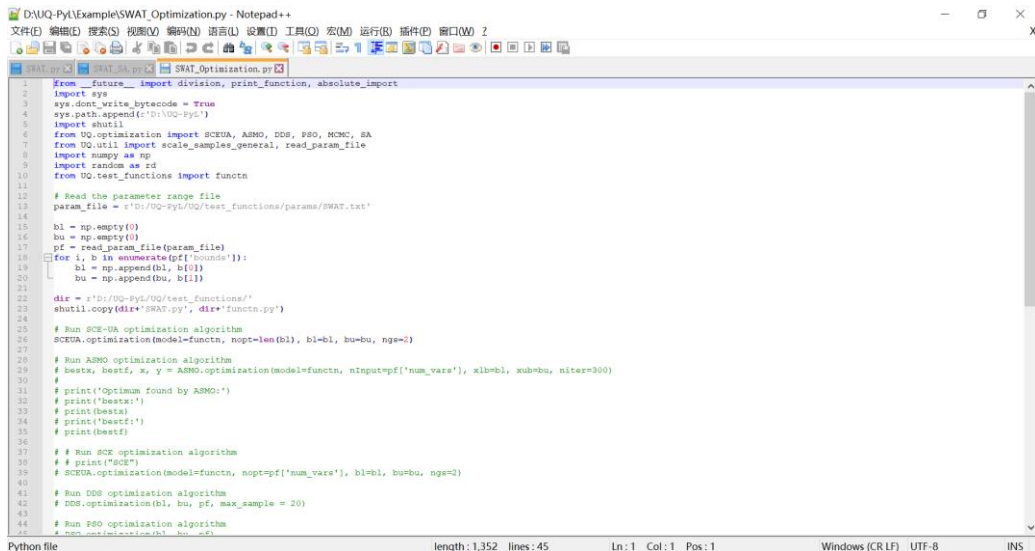
完成上述步骤后，打开 D:\UQ-Py\UQ\test_functions\SWAT，选择 SWAT_*.py 即可进行参数不确定性分析，*表示时 DoE 为参数采样、Surrogate 为构建替代模型、SA 为敏感性分析、Optimization 为优化。本文选择了 SA 做敏感性分析、Optimization 做优化。

打开 SWAT_SA.py，选择程序中的采样方法和敏感性分析方法，即可进行敏感性分析：



```
1 import os
2 import sys
3 sys.dont_write_bytecode = True
4 sys.path.append('D:\UQ-Py\')
5
6 from UQ.DoE import morris_out, lhs, box_behnken, monte_carlo, sobol
7 from UQ.analyze import *
8 from UQ.analyze import MARS_SA
9 from UQ.test_functions import SWAT
10 from UQ.util import scale_samples_general, read_param_file
11 import numpy as np
12 import random as rd
13
14 # Set random seed (does not affect quasi-random Sobol sampling)
15 seed = 1
16 np.random.seed(seed)
17 rd.seed(seed)
18
19 # Read the parameter range file and generate samples
20 param_file = 'D:\UQ-Py\UQ\test_functions\params\SWAT.txt'
21 pf = read_param_file(param_file)
22 # print(pf)
23
24 # Generate samples (choose method here)
25 # param_values = sampling_lhs(200, 26, 0)
26 # param_values = sampling_glp(200, 26, 0)
27 param_values = monte_carlo.sample(10, pf['num_vars'], plot=False)
28 # param_values = lhs.sample(520, pf['num_vars'], criterion='center')
29 # param_values = morris_out.sample(10, pf['num_vars'], num_levels = 4, grid_jump = 1) # grid_jump越小, delta也越小, 越详细; num_level参数水平数
30 # param_values = sobol.sample(5000, 26, plot = True)
31
32 # Samples are given in range [0, 1] by default. Rescale them to your parameter bounds.
33 scale_samples_general(param_values, pf['bounds'])
34 # print(os.getcwd())
35 os.chdir('D:\UQ-Py\UQ\test_functions\SWAT')
36 np.savetxt('Input_Morris.txt', param_values, delimiter=' ')
37
38 # Run the "model" and save the output in a text file
39 # This will happen offline for external models
40 Y = SWAT.predict(param_values)
41 # os.chdir('D:\UQ-Py\Example')
42 # np.savetxt('Output_Sobol.txt', Y, delimiter=' ')
43
44 # Perform the sensitivity analysis/uncertainty analysis using the model output
45 # Specify which column of the output file to analyze (zero-indexed)
```

打开 SWAT_Optimization.py，选择优化算法即可优化模型：



```
1 from __future__ import division, print_function, absolute_import
2 import sys
3 sys.dont_write_bytecode = True
4 sys.path.append('D:\UQ-Py\')
5 import shutil
6 from UQ.optimization import SCEUA, ASMO, DDS, PSO, MCMC, SA
7 from UQ.util import scale_samples_general, read_param_file
8 import numpy as np
9 import random as rd
10 from UQ.test_functions import functn
11
12 # Read the parameter range file
13 param_file = 'D:\UQ-Py\UQ\test_functions\params\SWAT.txt'
14
15 bl = np.empty(0)
16 bu = np.empty(0)
17 pf = read_param_file(param_file)
18 for i, b in enumerate(pf['bounds']):
19     bl = np.append(bl, b[0])
20     bu = np.append(bu, b[1])
21
22 dir = 'D:\UQ-Py\UQ\test_functions\'
23 shutil.copy(dir+'SWAT.py', dir+'functn.py')
24
25 # Run SCE-UA optimization algorithm
26 SCEUA.optimization(model=functn, nopt=len(bl), bl=bl, bu=bu, ngs=2)
27
28 # Run ASMO optimization algorithm
29 bestx, bestf, x, y = ASMO.optimization(model=functn, ninputs=pf['num_vars'], xlb=bl, xub=bu, niter=300)
30 #
31 # print('Optimum found by ASMO:')
32 # print('bestx:')
33 # print(bestx)
34 # print('bestf:')
35 # print(bestf)
36
37 # # Run SCE optimization algorithm
38 # # print('SCE')
39 # SCEUA.optimization(model=functn, nopt=pf['num_vars'], bl=bl, bu=bu, ngs=2)
40
41 # Run DDS optimization algorithm
42 DDS.optimization(bl, bu, pf, max_sample = 20)
43
44 # Run PSO optimization algorithm
45 nco.optimization(bl, bu, pf)
```