

EDS Lab 使用手册

EDS Lab 使用手册

UI

功能示例

部分结果

文档

证书

快速开始

0. 安装Python和相关依赖
1. 运行使用
 - * 创建虚拟环境 (非必须, 推荐)
2. 下载工程代码
3. 安装相关依赖
4. 运行项目
5. 本地查看项目
6. 运行实验脚本

管理和运行

1. 创建超级用户
2. 注册、登录后台管理系统
3. 用户管理
4. 脚本管理
5. 增加脚本
6. 运行脚本

功能特性

相关技术和工具

常见问题

- Q1. UnicodeDecodeError: 'gbk' codec can't decode byte 0x80 in position xxx: illegal multibyte sequence
- Q2. 如何为脚本增加参数控制?
- Q4. 如何生成的视频还有.gif图?
- Q5. 系统如何将python脚本执行的结果显示到前端的? 如何修改?

变更日志

实验设计和数据处理的作业，搭建了一个基于Django框架和，开源项目Wooey的可视化界面，用来展示优选法可视化的实验结果。

注意：本手册的word版本由相关软件导出，可能在格式问题，推荐查看其他格式（pdf, html）以及在线文档
<https://eds-lab.readthedocs.io/zh/latest/>。

UI

The screenshot shows the EDS Lab application interface. At the top, there is a dark header bar with the EDS Lab logo, user status (me), and navigation links for Queue, Task, Result, Simplified Chinese, and Log In/Register. Below the header is a large teal banner with the EDS Lab logo and the text "实验设计与数据分析可视化用户界面". The main content area features a search bar labeled "搜索脚本..." and a grid of script cards. Each card has a title, a brief description, and the "WOOEY SCRIPTS" label. The cards are arranged in three columns:

- sum** - 计算某个整数以下的所有整数的和
- a-plus-b** - 计算两个数的和
- mpl-demo** - Test matplotlib

- Orthogonal-Rotation** - Orthogonal Rotation Analysis
- TwoWays-Compare** - Two Ways Divide Method Final
- parallel** - Parallel Method Visualization

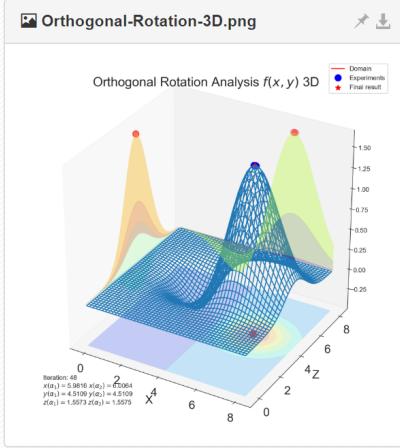
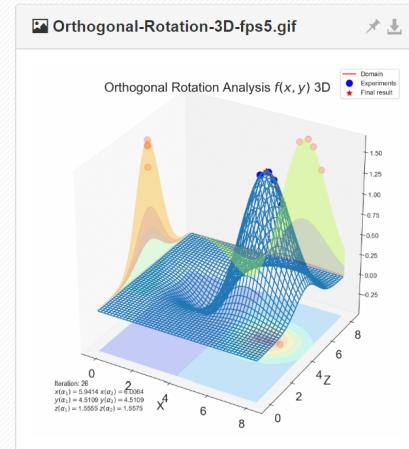
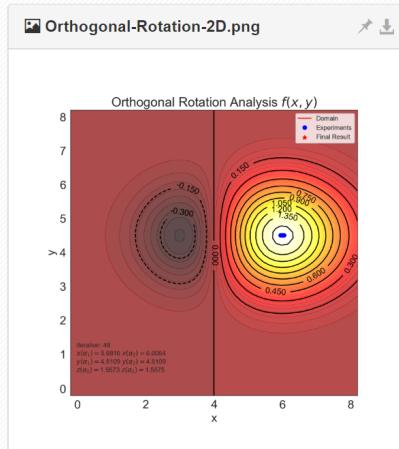
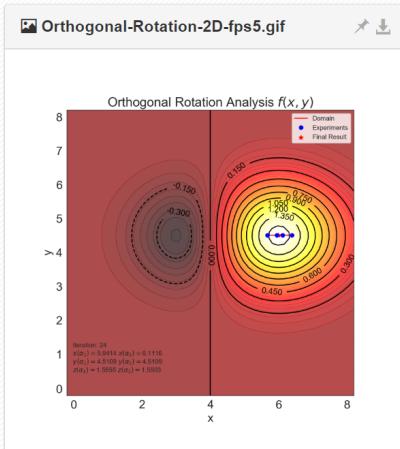
- simplex_opt**

功能示例

Orthogonal-Rotation 20181227-13:42

• 优先级 1 • me 提交于 5 日, 16 小时 前 • 更新于 5 日, 16 小时 前 ✓ 成功

[下载](#) [分享](#) [克隆](#) [重跑](#) [重提交](#) [删除](#)



控制台

```
g:\mysite\localsite\envs\wooley\scripts\python.exe E:\Homeworks\MyWooProject\MyWooProject\user_uploads\wooley_scripts\Orthogonal-Rotation_iBgZJld.py --x_min 0 --x_m
CHOOSE: -1
f(x,y) = 0.0*(x-0.0)^1.0+0.0+ 1.0*(y-0.0)^1.0+0.0

-----Iteration 001-----
-----Keep x = 1.0000 fixed-----
Experiments 02 on y = [0. 1.8886 3.0554 4.944 ], results=[-0. -0.0027 -0.0217 -0.0503]
Experiments 03 on y = [0. 1.1672 1.8882 3.0554], results=[-0. -0.0004 -0.0027 -0.0217]
Experiments 04 on y = [0. 0.7213 1.1669 1.8882], results=[-0. -0.0001 -0.0004 -0.0027]
Experiments 05 on y = [0. 0.7213 1.1669 1.8882], results=[-0. -0.0001 -0.0004 -0.0027]
-----Keep y = 0.9441 fixed-----
Experiments 02 on x = [3.056 4.9446 6.1114 8. ], results=[-0.0013 0.0034 0.0056 0.0015]
Experiments 03 on x = [3.056 4.2232 4.9442 6.1114], results=[-0.0034 0.0056 0.0044 0.0015]
Experiments 04 on x = [4.9446 5.6659 6.1115 6.8328], results=[0.0034 0.0054 0.0056 0.0044]
Experiments 05 on x = [4.9446 5.6659 6.1115 6.8328], results=[0.0034 0.0054 0.0056 0.0044]

-----Iteration 002-----
-----Keep x = 5.8887 fixed-----
Experiments 02 on y = [3.056 4.9446 6.1114 8. ], results=[0.6137 1.4199 0.4889 0.0067]
Experiments 03 on y = [3.056 4.2232 4.9442 6.1114], results=[0.6137 1.4984 1.4201 0.4889]
Experiments 04 on y = [3.056 3.7773 4.2229 4.9442], results=[0.6137 1.2291 1.4983 1.4201]
Experiments 05 on y = [3.7773 4.2231 4.4985 4.9442], results=[1.2291 1.4984 1.5503 1.4201]
Experiments 06 on y = [4.2231 4.4986 4.6687 4.9442], results=[1.4984 1.5503 1.5308 1.4201]
Experiments 07 on y = [4.2231 4.3933 4.4985 4.6687], results=[1.4984 1.5425 1.5503 1.5308]
Experiments 08 on y = [4.3933 4.4985 4.5635 4.6687], results=[1.5425 1.5503 1.5475 1.5308]
Experiments 09 on y = [4.3933 4.4583 4.4985 4.5635], results=[1.5425 1.5491 1.5503 1.5475]
Experiments 10 on y = [4.4583 4.4985 4.5234 4.5635], results=[1.5491 1.5503 1.5499 1.5475]
Experiments 11 on y = [4.4583 4.4985 4.5234 4.5635], results=[1.5491 1.5503 1.5499 1.5475]
-----Keep y = 4.5109 fixed-----
Experiments 02 on x = [3.056 4.9446 6.1114 8. ], results=[-0.367 0.9438 1.5504 0.4216]
```

```

Experiments 03 on x = [4.9446 6.1118 6.8328 8.    ], results=[0.9438 1.5503 1.2231 0.4216]
Experiments 04 on x = [4.9446 5.6659 6.1115 6.8328], results=[0.9438 1.491  1.5504 1.2231]
Experiments 05 on x = [5.6659 6.1117 6.3871 6.8328], results=[1.491  1.5503 1.4755 1.2231]
Experiments 06 on x = [5.6659 5.9414 6.1116 6.3871], results=[1.491  1.5555 1.5503 1.4755]
Experiments 07 on x = [5.6659 5.8362 5.9413 6.1116], results=[1.491  1.5416 1.5555 1.5503]
Experiments 08 on x = [5.8362 5.9414 6.0064 6.1116], results=[1.5416 1.5555 1.5575 1.5503]
Experiments 09 on x = [5.9414 6.0064 6.0466 6.1116], results=[1.5555 1.5575 1.5563 1.5503]
Experiments 10 on x = [5.9414 5.9816 6.0064 6.0466], results=[1.5555 1.5573 1.5575 1.5563]
Experiments 11 on x = [5.9414 5.9816 6.0064 6.0466], results=[1.5555 1.5573 1.5575 1.5563]

-----Iteration 003-----
-----Keep x = 5.9940 fixed-----
Experiments 02 on y = [3.056 4.9446 6.1114 8.    ], results=[0.6166 1.4266 0.4912 0.0067]
Experiments 03 on y = [3.056 4.2232 4.9442 6.1114], results=[0.6166 1.5054 1.4268 0.4912]
Experiments 04 on y = [3.056 3.7773 4.2229 4.9442], results=[0.6166 1.2349 1.5053 1.4268]
Experiments 05 on y = [3.7773 4.2231 4.4985 4.9442], results=[1.2349 1.5054 1.5576 1.4268]
Experiments 06 on y = [4.2231 4.4986 4.6687 4.9442], results=[1.5054 1.5576 1.538  1.4268]
Experiments 07 on y = [4.2231 4.3933 4.4985 4.6687], results=[1.5054 1.5497 1.5576 1.538 ]
Experiments 08 on y = [4.3933 4.4985 4.5635 4.6687], results=[1.5497 1.5576 1.5548 1.538 ]
Experiments 09 on y = [4.3933 4.4985 4.4985 4.5635], results=[1.5497 1.5564 1.5576 1.5548]
Experiments 10 on y = [4.4983 4.4985 4.5234 4.5635], results=[1.5564 1.5576 1.5572 1.5548]
Experiments 11 on y = [4.4983 4.4985 4.5234 4.5635], results=[1.5564 1.5576 1.5572 1.5548]
-----Keep y = 4.5109 fixed-----
Experiments 02 on x = [3.056 4.9446 6.1114 8.    ], results=[-0.367   0.9438 1.5504 0.4216]
Experiments 03 on x = [4.9446 6.1118 6.8328 8.    ], results=[0.9438 1.5503 1.2231 0.4216]
Experiments 04 on x = [4.9446 5.6659 6.1115 6.8328], results=[0.9438 1.491  1.5504 1.2231]
Experiments 05 on x = [5.6659 6.1117 6.3871 6.8328], results=[1.491  1.5503 1.4755 1.2231]
Experiments 06 on x = [5.6659 5.9414 6.1116 6.3871], results=[1.491  1.5555 1.5503 1.4755]
Experiments 07 on x = [5.6659 5.8362 5.9413 6.1116], results=[1.491  1.5416 1.5555 1.5503]
Experiments 08 on x = [5.8362 5.9414 6.0064 6.1116], results=[1.5416 1.5555 1.5575 1.5503]
Experiments 09 on x = [5.9414 6.0064 6.0466 6.1116], results=[1.5555 1.5575 1.5563 1.5503]
Experiments 10 on x = [5.9414 5.9816 6.0064 6.0466], results=[1.5555 1.5573 1.5575 1.5563]
Experiments 11 on x = [5.9414 5.9816 6.0064 6.0466], results=[1.5555 1.5573 1.5575 1.5563]

Find best indicator at (x=5.9940, y=4.5109), values=1.5575
SUSS
Creating 2D Animation VIDEO ...
Done.
Creating 2D Animation GIF ...
Done.
2D Animation Created.
Creating 3D Animation VIDEO ...
Done.
Creating 3D Animation GIF ...
Done.
3D Animation Created.

```

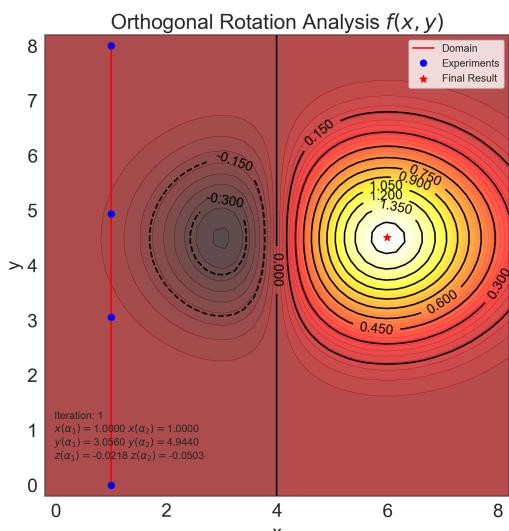
所有文件	参数	大小
文件名		
Orthogonal-Rotation_20181227-1342.tar.gz		27.4 MB
Orthogonal-Rotation_20181227-1342.zip		27.4 MB
history.pkl		5.5 KB
Orthogonal-Rotation-2D-fps5.mp4		668.1 KB
Orthogonal-Rotation-2D.pdf		53.6 KB
Orthogonal-Rotation-3D-fps5.mp4		867.0 KB
Orthogonal-Rotation-3D.pdf		91.1 KB
Orthogonal-Rotation-2D-fps5.gif		9.6 MB
Orthogonal-Rotation-2D.png		330.4 KB
Orthogonal-Rotation-3D-fps5.gif		15.3 MB
Orthogonal-Rotation-3D.png		635.6 KB

部分结果

实验方法和过程可视化结果 (2维)

2D

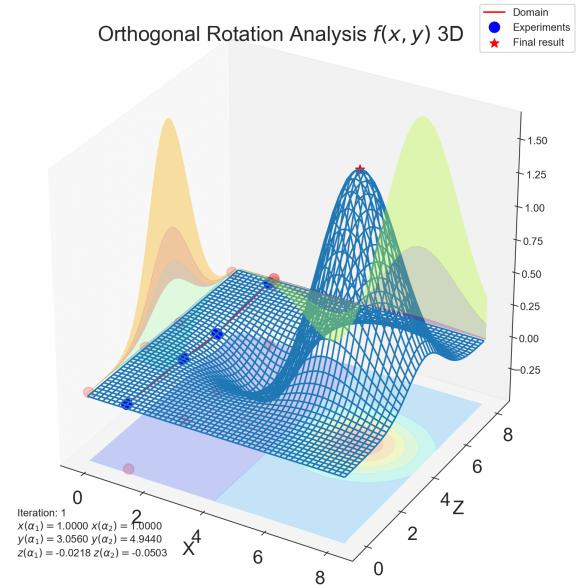
双因素选升法2D



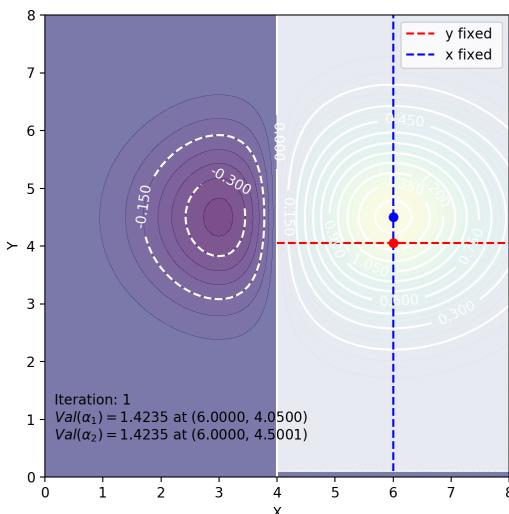
实验方法和过程可视化结果 (3维)

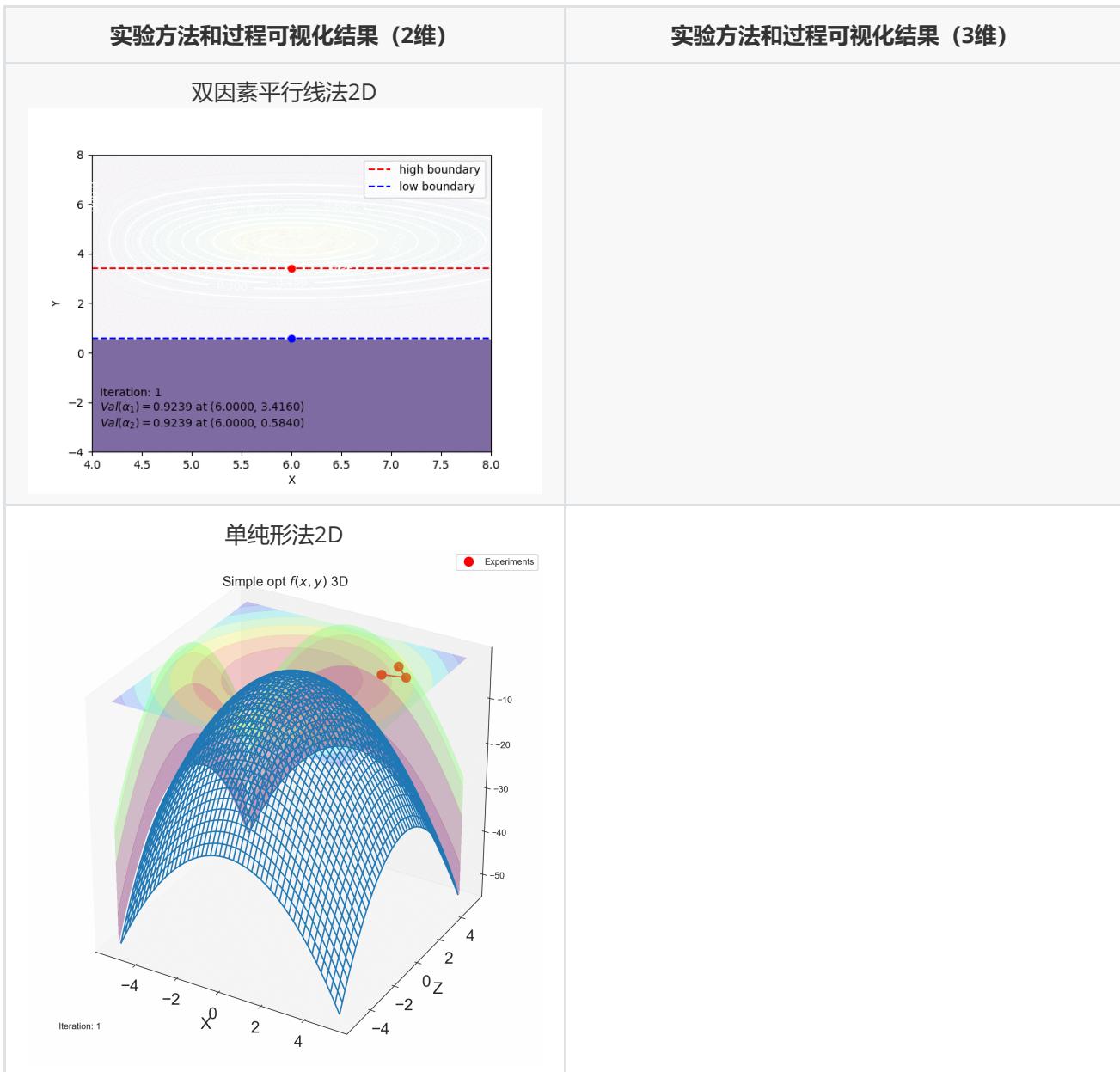
3D

双因素选升法3D



双因素对开法2D





注：图标题并非几种方法的标准英文翻译。

文档

[中文文档](#)

证书

BSD

快速开始

0. 安装Python和相关依赖

- Python - 推荐python3.6及以上
- pip - 推荐18.1及以上
- 其他：见文件 requirements.txt，内容如下，安装请见后文：

```
Django
wooey
django-sslify
django-storages
django-celery
django-autoslug
boto
waitress
psycopg2
collectfast
honcho
```

1. 运行使用

* 创建虚拟环境（非必须，推荐）

为了防止项目工程的依赖和用户自带的python环境相冲突，例如python版本不一致。建议为工程创建虚拟环境，创建方法见[virtualenv](#)，这里推荐使用[virtualenvwrapper](#)，一个更快捷的 `virtualenv` 的封装来创建python3的虚拟环境。

例如，使用 `virtualenvwrapper` 创建了名称为 `wooey` 的python3环境，然后在这个环境下运行本项目。

```
E:\Homeworks\MyWooProject\EDS-lab\docs (master -> origin)
λ workon wooey
E:\Homeworks\MyWooProject\EDS-lab\docs (master -> origin)
(wooey) λ python --version
Python 3.6.5 :: Anaconda, Inc.
```

2. 下载工程代码

Experiment design assignment. A Django site based on Wooey, to visualize the results of some algorithms. 实验设计和数据处理的作业，搭建了一个基于Django框架和开源项目Wooey的可视化界面，用来展示优选法可视化的实验结果。

Manage topics

8 commits 2 branches 0 releases 1 environment 1 contributor

Branch: master New pull request

Create new file Upload files Find file Clone or download ①

Clone with HTTPS ② Use SSH
Use Git or checkout with SVN using the web URL.
<https://github.com/liuqiddev/EDS-lab.git>

Open in Desktop Open in Visual Studio

Download ZIP

File	Description	Time
liuqiddev update db, readme, documents, so developer can do what they want	update db, readme, documents, so developer can do what they want	3 hours ago
MyWooProject	update db, readme, documents, so developer can do what they want	
assets	first commit, next step is to write a document	
docs	update db, readme, documents, so developer can do what they want	
site	update docs, add github page, the next step is to write full docs	
Orthogonal-Rotation.py	update db, readme, documents, so developer can do what they want	
README.md	update docs, add github page, the next step is to write full docs	3 hours ago
TwoWays-Compare.py	update db, readme, documents, so developer can do what they want	16 minutes ago
a_plus_b.py	update db, readme, documents, so developer can do what they want	16 minutes ago
argparser-test.py	update db, readme, documents, so developer can do what they want	16 minutes ago
manage.py	update db, readme, documents, so developer can do what they want	16 minutes ago
mkdocs.yml	update docs, add github page, the next step is to write full docs	3 hours ago

下载地址：[EDS-lab](#)

下载后解压，进入到 `EDS-lab` 路径下。

3. 安装相关依赖

使用 `pip` 安装相关依赖，[requirements.txt](#)。

```
pip install -r requirements.txt
```

4. 运行项目

对于 windows，在当前路径（即 `manage.py` 所在的路径）下，**分别启动两个终端**：

```
(wooey) λ ls
docs/ manage.py* mkdocs.yml MyWooProject/ README.md requirements.txt site/
E:\Homeworks\MyWooProject\EDS-lab (master -> origin)
(wooey) λ celery -A MyWooProject worker --pool=eventlet -l info

----- celery@Ghost v4.2.1 (windowlicker)
----- **** -----
-- * *** * -- Windows-10-10.0.17134-SP0 2019-01-09 15:02:52
-- * - **** --
- ** ----- [config]
- ** ----- .> app: MyWooProject:0x24cc0480ba8
- ** ----- .> transport: filesystem://localhost///
- ** ----- .> results:
- *** - * -- .> concurrency: 8 (eventlet)
- ***** -- .> task events: OFF (enable -E to monitor tasks in this worker)
- **** --
----- [queues]
    .> celery      exchange=celery(direct) key=celery

[tasks]
. MyWooProject.wooey_celery_app.debug_task
. wooey.tasks.cleanup_dead_jobs
. wooey.tasks.cleanup_wooey_jobs
. wooey.tasks.submit_script

[2019-01-09 15:02:52,123: INFO/MainProcess] Connected to filesystem://localhost//
[2019-01-09 15:02:52,134: WARNING/MainProcess] g:\mysite\localsite\envs\wooey\lib\site-packages\celery\fixups\django.py:200: UserWarning: Using settings.DEBUG leads to a memory leak, never use this setting in production environments!
  warnings.warn('Using settings.DEBUG leads to a memory leak, never '
[2019-01-09 15:02:52,136: INFO/MainProcess] celery@Ghost ready.
```

终端1输入：

```
celery -A MyWooProject worker --pool=eventlet -l info
```

效果入上图所示。

The image shows two terminal windows side-by-side. Both windows are titled 'Cmder' and are running on a Windows 10 system. The left terminal window displays the output of the command 'celery -A MyWooProject worker --pool=eventlet -l info'. It shows the configuration of the celery worker, including the app, transport, and results backends, and lists the available tasks. The right terminal window displays the output of the command 'python manage.py runserver'. It shows the Django development server starting at port 8000, performing system checks, and handling a 'livereload' request. Both windows show the logs from January 9, 2019, at 15:32:50.

```
(wooey) λ ls
docs/ manage.py* mkdocs.yml MyWooProject/ README.md requirements.txt site/
E:\Homeworks\MyWooProject\EDS-lab (master -> origin)
(wooey) λ celery -A MyWooProject worker --pool=eventlet -l info

----- celery@Ghost v4.2.1 (windowlicker)
----- **** -----
-- * *** * -- Windows-10-10.0.17134-SP0 2019-01-09 15:32:50
-- * - **** --
- ** ----- [config]
- ** ----- .> app: MyWooProject:0x1f4993c1b38
- ** ----- .> transport: filesystem://localhost///
- ** ----- .> results:
- *** - * -- .> concurrency: 8 (eventlet)
- ***** -- .> task events: OFF (enable -E to monitor tasks in this worker)
- **** --
----- [queues]
    .> celery      exchange=celery(direct) key=celery

[tasks]
. MyWooProject.wooey_celery_app.debug_task
. wooey.tasks.cleanup_dead_jobs
. wooey.tasks.cleanup_wooey_jobs
. wooey.tasks.submit_script

[2019-01-09 15:32:51,045: INFO/MainProcess] Connected to filesystem://localhost//
[2019-01-09 15:32:51,076: WARNING/MainProcess] g:\mysite\localsite\envs\wooey\lib\site-packages\celery\fixups\django.py:200: UserWarning: Using settings.DEBUG leads to a memory leak, never use this setting in production environments!
  warnings.warn('Using settings.DEBUG leads to a memory leak, never '
[2019-01-09 15:32:51,078: INFO/MainProcess] celery@Ghost ready.

[2019-01-09 15:33:09] "GET /livereload HTTP/1.1" 404 12320
[2019-01-09 15:34:40] "GET /livereload HTTP/1.1" 404 12320
```

终端2输入：

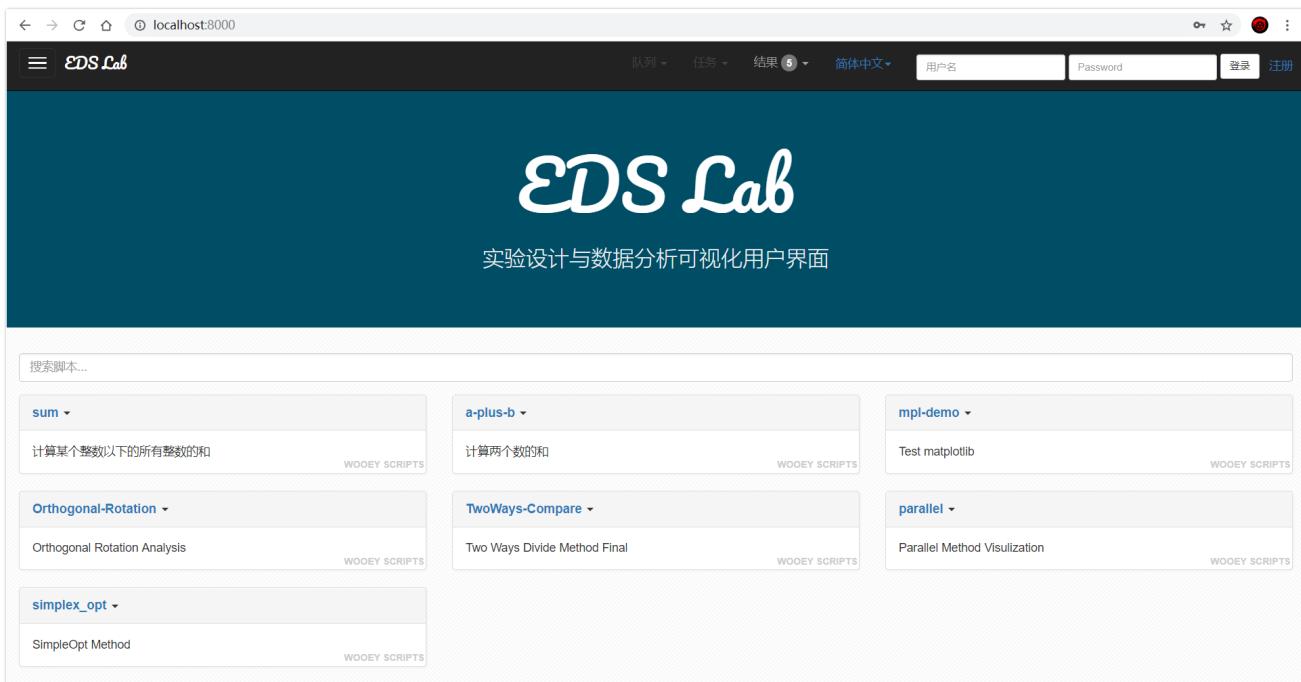
```
python manage.py runserver 0:8000
```

注：端口号可以任意指定。

对于Linux和其他操作系统，请查看[这里](#)。

5.本地查看项目

通过步骤5就构建了一个本地服务器来运行本工程。

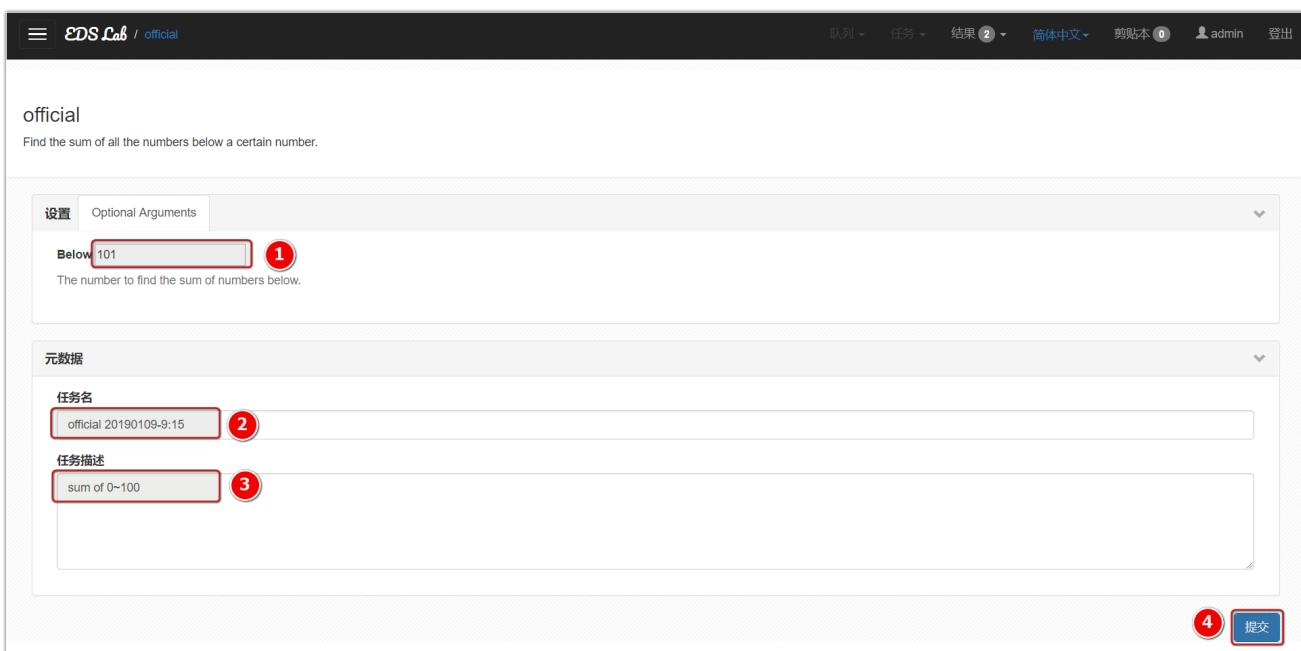


The screenshot shows the EDS Lab web interface at <http://localhost:8000>. The title bar says "EDS Lab". The main area displays a grid of script cards:

- sum**: 计算某个整数以下的所有整数的和 (WOOEY SCRIPTS)
- a-plus-b**: 计算两个数的和 (WOOEY SCRIPTS)
- mpl-demo**: Test matplotlib (WOOEY SCRIPTS)
- Orthogonal-Rotation**: Orthogonal Rotation Analysis (WOOEY SCRIPTS)
- TwoWays-Compare**: Two Ways Divide Method Final (WOOEY SCRIPTS)
- parallel**: Parallel Method Visulization (WOOEY SCRIPTS)
- simplex_opt**: SimpleOpt Method (WOOEY SCRIPTS)

使用浏览器，输入 <http://localhost:8000/>，即可查看工程。

6. 运行实验脚本



The screenshot shows the EDS Lab task configuration interface for the "official" script. The top navigation bar includes "EDS Lab / official", "队列", "任务", "结果 2", "简体中文", "剪贴本 0", "admin", and "登出".

The main area shows the "official" script details:

Find the sum of all the numbers below a certain number.

设置 Optional Arguments

Below 101 (1) The number to find the sum of numbers below.

元数据

任务名 official 20190109-9:15 (2)

任务描述 sum of 0~100 (3)

At the bottom right are buttons labeled (4) and 提交 (Submit).

A screenshot of the EDS Lab platform interface. At the top, there's a navigation bar with 'EDS Lab / official / 任务 # 44', user info ('admin'), and various buttons like '下载', '分享', '克隆', '重跑', '重提交', and '删除'. Below the navigation is a task card for 'official 20190109-9:15 sum of 0~100'. It shows a green success message: '优先级 1 • admin 提交于 0 分钟 前 · 更新于 0 分钟 前 ✓ 成功'. A control panel window is open, displaying a terminal session: 'g:\mysite\localsite\envs\woaney\scripts\python.exe E:\Homeworks\MyWooProject\EDS-lab\MyWooProject\user_uploads\woaney_scripts\official_A629Qek.py --below 101' followed by 'Sum = 5050'. Below the control panel is a file list titled '所有文件' with columns '文件名', '参数', and '大小'. It contains three files: 'official_20190109-915.tar.gz' (193 字节), 'official_20190109-915.zip' (291 字节), and 'result.txt' (9 字节).

点击脚本名称，输入相关参数，即可运行。例如上图是创建计算 $[0, 100]$ 以内所有正数和的任务，点击提交，任务会被添加到任务队列中，等待片刻，结果输出到界面，结果产生的文件以列表形式陈列于下方。

运行其他脚本过程类似。

管理和运行

本项目的后台系统是基于Django的，这里已经为您创建了一个用户：

- 用户名 admin
- 密码 ADMIN12345678

你可以使用上述用户名和密码来执行相关管理工作，其中包括创建新的用户。

1. 创建超级用户

超级用户即系统管理员，这里和所有django项目一样，在后台中输入 `createsuperuser` 进行管理员的创建。

A screenshot of a terminal window. The command 'E:\Homeworks\MyWooProject\EDS-lab (master -> origin) (woaney) λ python manage.py createsuperuser' is entered. The terminal then prompts for a username ('Username: admin'), email address ('Email address: admin@edslab.com'), and password ('Password:'). After entering the password twice ('Password (again)'), the terminal outputs 'Superuser created successfully.'

2. 注册、登录后台管理系统



两种方式登录：

1. 在用户界面右上角输入用户名和密码
2. 或者在地址栏中输入 ip:端口号/admin，输入用户名或密码登录。

系统也支持注册新用户并登录。

3. 用户管理

EDS Lab后台系统

欢迎, ADMIN 查看站点 / 修改密码 / 注销

首页 - 认证和授权 - 用户

选择用户 来修改

增加用户 +

过滤器

以 职员状态

全部 是 否

以 超级用户状态

全部 是 否

以 有效

全部 是 否

搜索框

用户名 电子邮箱地址 名字 姓氏 职员状态

用户名	电子邮件地址	名字	姓氏	职员状态
admin	admin@edslab.com			✓
<input checked="" type="checkbox"/>	nihacwork@outlook.com			✓
<input checked="" type="checkbox"/>	nihacwork@outlook.com			✗

3 用户

localhost:8002/admin/auth/user/?o=-1

如上图所示可在后台进行用户管理。

4. 脚本管理

EDS Lab后台系统

欢迎, ADMIN 查看站点 / 修改密码 / 注销

首页 - Wooey - Scripts

选择 script 来修改

增加 SCRIPT +

删除所选的 scripts

SCRIPT GROUP IS ACTIVE

SCRIPT GROUP	IS ACTIVE
Wooey Scripts	✓

7 scripts

EDS Lab后台系统

欢迎, **ADMIN** 查看站点 / 修改密码 / 注销

首页 > Wooley > Scripts > ~~删除多个对象~~

你确定吗?

请确认要删除选中的 script 吗? 以下所有对象和余它们相关的条目将都会被删除:

概览

- Scripts: 1
- Script versions: 1
- Wooley jobs: 1
- Scriptparser-scriptversion relationships: 1

对象

- Script: mpl-demo
 - Script version: mpl-demo(1: 1)
 - Wooley job: mpl-demo 20190109-8:26
 - Scriptparser-scriptversion relationship: ScriptParser_script_version object

操作

是的, 我确定 **不, 返回**

EDS Lab后台系统

欢迎, **ADMIN** 查看站点 / 修改密码 / 注销

首页 > Wooley > Scripts

✓ 成功删除了 1 个 script

选择 script 来修改

增加 SCRIPT +

0 scripts

5. 增加脚本

EDS Lab后台系统

欢迎, **ADMIN** 查看站点 / 修改密码 / 注销

首页 > Wooley > Scripts

选择 script 来修改

增加 SCRIPT +

0 scripts

Script order: 1

Is active

User groups:

按住“Control”，或者Mac上的“Command”，可以选择多个。

Execute full path

Save path:

By default save to the script name, this will change the output folder.

SCRIPT VERSIONS					
SCRIPT VERSION	SCRIPT ITERATION	SCRIPT PATH	DEFAULT VERSION	CHECKSUM	删除?
1	1	<input type="button" value="选择文件"/> mpl-demo.py	<input checked="" type="checkbox"/>		
<input type="button" value="添加另一个 Script version"/> <input type="button" value="保存并增加另一个"/> <input type="button" value="保存并继续编辑"/> <input type="button" value="保存"/>					

有两种方式来增加脚本。第一种方式通过后台来添加（不推荐，没办法添加绝对路径，checksum无法计算，从而无法添加成功）。

```
E:\Homeworks\MyWooProject\EDS-lab (master -> origin)
(wooley) λ python manage.py addscript mpl-demo.py    使用
Converting mpl-demo.py
Converted 1 scripts
```

另外一种通过终端来添加，在终端中输入：

```
python manage.py addscript [path-to-script/]script.py
```

即 `addscript` 后接脚本的路径，出现 `Concerted 1 scripts` 表明添加成功。

特别注意： 添加的脚本中不能含有中文，包括注释中都不能有中文等其他字符。

EDS Lab后台系统

欢迎, ADMIN 查看站点 / 修改密码 / 注册

首页 > Wooey > Scripts

选择 script 来修改

动作 1 个中 0 个被选

SCRIPT NAME	SCRIPT GROUP	IS ACTIVE
mpl-demo	Wooey Scripts	<input checked="" type="checkbox"/>

1 script

EDS Lab后台系统

首页 · Wooey · Scripts · mpl-demo

修改 script

Script name:

Script group: Wooey Scripts ✖️ ➕

Script description: Test matplotlib

Documentation:

Script order:

进入后台管理系统中，发现脚本被成功添加了进来，点击进入后可以修改脚本信息（非源程序）。

localhost:8002

EDS Lab

搜索脚本...

mpi-demo ★

Test matplotlib WOOEY SCRIPTS

查看站点，发现有新的脚本被添加进来。

6. 运行脚本

EDS Lab / mpl-demo / 任务 #43

队列 任务 结果 2 简体中文 剪贴本 0 admin 登出

mpl-demo 20190109-8:40 this is not the shape of my heart

优先级 1 • admin 提交于 0 分钟 前 · 更新于 0 分钟 前 ✓ 成功



控制台

```
g:\mysite\localsite\envs\wooeyscripts\python.exe E:\Homeworks\MyWooProject\EDS-lab\MyWooProject\user_uploads\wooeyscripts\mpl-demo.py
```

```
g:\mysite\localsite\envs\wooye\scripts\python.exe E:\Homeworks\MyWooProject\EDS-lab\MyWooProject\user_uploads\wooye_scripts\mpl-demo.py
Done.
```

所有文件

文件名	参数	大小
mpl-demo_20190109-840.tar.gz		57.1 KB
mpl-demo_20190109-840.zip		66.5 KB
heart.pdf		5.9 KB
heart.png		60.2 KB

运行新脚本，结果如上图。

功能特性

Orthogonal-Rotation

Orthogonal Rotation/Spin-up Method Final

设置 Optional Arguments

X Min 0
left boundary of X

X Max 8
right boundary of X

Y Min 0
left boundary of Y

Y Max 8
right boundary of Y

Delta 0.618
delta

Init X 1.0
initial x

Init Y 0.0
initial y

Tol 0.0001
tolerance

Num Divs 50
num of divides plot

Choose -1
to choose function

X P1 0.0
parameter1 of x, eg:k1*x

X P2 0.0
parameter2 of x, eg:k1*(x-k2)

X P3 1.0
parameter3 of x, eg:k1*(x-k2)^k3

X P4 0
parameter4 of x, eg:k1*(x-k2)^k3+k4

X P5 0.0
parameter5 of x, eg:k5*x**-1

Y P1 1.0
parameter1 of y, eg:k1*x

Y P2 0.0
parameter2 of y, eg:k1*(y-k2)

Y P3 1.0
parameter3 of y, eg:k1*(y-k2)^k3

Y P4 0
parameter4 of y, eg:k1*(y-k2)^k3+k4

Y P5 0.0
parameter5 of y, eg:k5*x**-1

元数据

任务名

Orthogonal-Rotation 20190109-10:24

任务描述

请输入任务描述 ...

提交

1. 可以运行任何python脚本
2. 脚本参数可调整
3. 脚本可重复运行，可查看管理历史结果
4. 良好的交互，美观的界面
5. 良好的算法可视化效果
6. 方便的后台管理
7. 基于B/S，部署后随处可访问
8. 支持多用户访问，运行结果互不干扰
9. 完全开源，方便扩展开发
10. 运行结果文件轻松下载

相关技术和工具



常见问题

Q1. UnicodeDecodeError: 'gbk' codec can't decode byte 0x80 in position xxx: illegal multibyte sequence

上面表示脚本中含有 gbk 编码的字符，这里需要注意，添加的脚本中不能有中文（这是一个bug，目前还没修复），中文不能存在于python程序的任何地方，包括注释中。所以需要检查代码，将相应的字符删除或者替换成英文。

Q2. 如何为脚本增加参数控制？

程序通过 `argparse` 来进行参数调节和解析，下方是一个示例：

```
// a_plus_b.py
import argparse
import sys

# parser and arguments
parser = argparse.ArgumentParser(description='a plus b problem')
parser.add_argument('--a', help='first num', type=int, default=0)
parser.add_argument('--b', help='second num', type=int, default=0)

def main():
    args = parser.parse_args()
    a = args.a
    b = args.b
    sum = a + b
    print('{0} + {1} = {2}'.format(a, b, sum))
    return 0

if __name__ == "__main__":
    sys.exit(main())
#main()
```

你可以通过命令行来运行，

```
python a_plus_b.py
>>> 0

python a_plus_b.py --a 1 --b 2
>>> 3
```

项目中更改脚本参数的逻辑和上面一致。

关于 `argparse`，[查看更多](#)。

Q3. 算法可视化中的动画是如何做到的？

动画使用的是[Matplotlib](#)中的[animation](#)。

例如本项目中 双因素旋升法，二维动画生成，核心代码如下，

```
def init2D():
    line.set_data([], [])
    point.set_data([], [])
    value_display.set_text('')
    return line, point, value_display

def animate2D(i):
    line.set_data(track_x[i], track_y[i])

    point.set_data(track_x[i], track_y[i])
    # point.set_data(track_x[i], track_y[i], track_z[i])

    value_display.set_text('Iteration: ' + str(i + 1) +
                          '\n'
                          r'$x(\alpha_1)= ${0:.4f}'.format(
                              track_x[i][1]) + r'$x(\alpha_2)= ${0:.4f}'.format(
                                  track_x[i][2]) +
                          '\n'
                          r'$y(\alpha_1)= ${0:.4f}'.format(
                              track_y[i][1]) + r'$y(\alpha_2)= ${0:.4f}'.format(
                                  track_y[i][2]) +
                          '\n'
                          r'$z(\alpha_1)= ${0:.4f}'.format(
                              track_z[i][1]) + r'$z(\alpha_2)= ${0:.4f}'.format(
                                  track_z[i][2])
    )

    return line, point, value_display
```

```
# 2D
fig = plt.figure(figsize=(8, 8))
# ax1 = Axes3D(fig)
ax1 = plt.gca()
ax1 = snapshot2D(ax1)

line, = ax1.plot([], [], 'r', label='Domain', lw=1.5)
point, = ax1.plot([], [], 'bo', c='b', label='Experiments')
value_display = ax1.text(0.02, 0.08, '', transform=ax1.transAxes)

ax1.legend(loc=1)
ax1.grid(False)

anim2D = animation.FuncAnimation(fig, animate2D, init_func=init2D,
                                 frames=len(track_x),
                                 interval=500,
                                 repeat_delay=80, blit=True)

writer2D = animation.writers['ffmpeg']
FPS_2D = 5
```

```

writer2D = Writer2D(fps=FPS_2D, metadata=dict(artist='L'), bitrate=1800)
# anim.save('Orthogonal-Rotation-004.gif',writer='imagemagick', fps=60)
print('Creating 2D Animation VIDEO ...')
anim2D.save('Orthogonal-Rotation-2D-fps{0}.mp4'.format(FPS_2D), writer=writer2D)
print('Done.\nCreating 2D Animation GIF ...')
anim2D.save('Orthogonal-Rotation-2D-fps{0}.gif'.format(FPS_2D), writer='imagemagick',
fps=FPS_2D)
print('Done\n2D Animation Created.')
plt.savefig('Orthogonal-Rotation-2D.pdf')
plt.savefig('orthogonal-Rotation-2D.png')

```

其中 `snapshot2D` 为生成函数的等高线图，将 `ax` (matplotlib中的坐标轴类) 的对象传回，再在该轴上绘制其他图像，也可以同时绘制。

```

def snapshot2D(ax):
    a = np.linspace(scope_x[0] - 0.2, scope_x[1] + 0.2, num_divs)
    b = np.linspace(scope_y[0] - 0.2, scope_y[1] + 0.2, num_divs)
    x, y = np.meshgrid(a, b)
    # z=func(x,y)
    # ax.contour(x, y,z, levels=np.logspace(-3,3,25), cmap='jet')
    ax.contourf(x, y, func(x, y), 50, cmap=cm.hot, alpha=0.7)
    CS = ax.contour(x, y, func(x, y), 15, colors='k')
    ax.clabel(CS, inline=True, fontsize=13)
    ax.set_xlabel(r'x', fontdict={'fontsize': 18, 'fontweight': 'medium'})
    ax.set_ylabel(r'y', fontdict={'fontsize': 18, 'fontweight': 'medium'})

    ax.set_title(r'Orthogonal Rotation Analysis $f(x,y)$', fontdict={'fontsize': 20,
    'fontweight': 'medium'})
    ax.xaxis.set_tick_params(labelsize=18)
    ax.yaxis.set_tick_params(labelsize=18)

    # ax.plot(final_x, final_y, marker='*', c='r', markersize=15)
    ax.scatter(final_x, final_y, marker='*', c='r', s=50, label='Final Result')

    return ax

```

3D的动画生成逻辑相同。

```

def init3D():
    line.set_data([], [])
    line.set_3d_properties([])
    point.set_data([], [])
    point.set_3d_properties([])

    point1.set_data([], [])
    point1.set_3d_properties([])

    point2.set_data([], [])

```

```

point2.set_3d_properties([])

point3.set_data([], [])
point3.set_3d_properties([])

display_value.set_text('')

# return line, point, display_value
return line, point, point1, point2, point3, display_value


def animate3D(i):
    line.set_data(track_x[i], track_y[i])
    line.set_3d_properties(track_z[i])

    point.set_data(track_x[i], track_y[i])
    point.set_3d_properties(track_z[i])

    point1.set_data(track_x[i], track_y[i])
    point1.set_3d_properties([-0.5] * len(track_x[i]))

    # point2.set_data(scope_x[0]* len(track_x[i]), track_y[i]) # a bug here
    # point2.set_3d_properties(track_z[i])
    point2.set_data([scope_x[0] - 0.3] * len(track_x[i]), track_y[i])
    point2.set_3d_properties(track_z[i])

    point3.set_data(track_x[i], [scope_y[1] + 0.3] * len(track_x[i]))
    point3.set_3d_properties(track_z[i])

    display_value.set_text('Iteration: ' + str(i + 1) +
                          '\n'
                          r'$x(\alpha_1)= ${} \'.format(track_x[i][1]) + r'
$'x(\alpha_2)= ${} \'{0:.4f}'.format(
    track_x[i][2]) +
                          '\n'
                          r'$y(\alpha_1)= ${} \'{0:.4f}'.format(track_y[i][1]) + r'
$'y(\alpha_2)= ${} \'{0:.4f}'.format(
    track_y[i][2]) +
                          '\n'
                          r'$z(\alpha_1)= ${} \'{0:.4f}'.format(track_z[i][1]) + r'
$'z(\alpha_2)= ${} \'{0:.4f}'.format(
    track_z[i][2])))

    # return line, point, display_value
    return line, point, point1, point2, point3, display_value

```

```

# 3D
fig1 = plt.figure(figsize=(8, 8))
ax2 = Axes3D(fig1)
ax2 = snapshot3D(ax2)

line, = ax2.plot([], [], [], 'r-', label='Domain', lw=1.5)

```

```

point, = ax2.plot([], [], [], 'bo', markersize=10, label='Experiments') # on the
# surface
point1, = ax2.plot([], [], [], 'ro', markersize=10, alpha=0.2) # projection to z
point2, = ax2.plot([], [], [], 'ro', markersize=10, alpha=0.2) # projection to x
point3, = ax2.plot([], [], [], 'ro', markersize=10, alpha=0.2) # projection to y

display_value = ax2.text(scope_x[1], scope_y[1], final_z + 0.5, '',
transform=ax2.transAxes)
ax2.grid(False)
ax2.legend(loc=1)

anim = animation.FuncAnimation(fig1, animate3D, init_func=init3D,
                               frames=len(track_x),
                               interval=1000,
                               repeat_delay=80, blit=True)

writer3D = animation.writers['ffmpeg']
FPS_3D = 5
writer3D = writer3D(fps=FPS_3D, metadata=dict(artist='L'), bitrate=1800)
# anim.save('Orthogonal-Rotation-004.gif',writer='imagemagick', fps=6)
print('Creating 3D Animation VIDEO ...')
anim.save('Orthogonal-Rotation-3D-fps{0}.mp4'.format(FPS_3D), writer=writer3D)
print('Done.\nCreating 3D Animation GIF ...')
anim.save('Orthogonal-Rotation-3D-fps{0}.gif'.format(FPS_3D), writer='imagemagick',
fps=FPS_3D)
print('Done\n3D Animation Created.')
plt.savefig('Orthogonal-Rotation-3D.pdf')
plt.savefig('Orthogonal-Rotation-3D.png')
# plt.show()

```

其中 `snapshot3D` :

```

def snapshot3D(ax):
    a = np.linspace(scope_x[0] - 0.2, scope_x[1] + 0.2, num_divs)
    b = np.linspace(scope_y[0] - 0.2, scope_y[1] + 0.2, num_divs)
    X, Y = np.meshgrid(a, b)
    Z = func(X, Y)

    # ax.plot_surface(X, Y, Z, rstride=1, cstride=1, edgecolor='none', cmap='jet')
    ax.plot_wireframe(X, Y, Z, rstride=1, cstride=1, cmap='jet')
    cset = ax.contourf(X, Y, Z, zdir='z', offset=-0.5, cmap='jet', alpha=0.2)
    cset = ax.contourf(X, Y, Z, zdir='x', offset=scope_x[0] - 0.3, cmap='jet',
alpha=0.2)
    cset = ax.contourf(X, Y, Z, zdir='y', offset=scope_y[1] + 0.3, cmap='jet',
alpha=0.2)

    # CS = ax.contour(x, y, func(x, y), colors='k')
    # ax.clabel(CS, inline=True, fontsize=13)
    ax.set_xlabel(r'X', fontdict={'fontsize': 18, 'fontweight': 'medium'})
    ax.set_ylabel(r'Y', fontdict={'fontsize': 18, 'fontweight': 'medium'})
    ax.set_zlabel(r'Z', fontdict={'fontsize': 18, 'fontweight': 'medium'})

```

```

    ax.set_title(r'Orthogonal Rotation Analysis $f(x,y)$ 3D', fontdict={'fontsize': 20,
    'fontweight': 'medium'})
    ax.xaxis.set_tick_params(labelsize=18)
    ax.yaxis.set_tick_params(labelsize=18)

    # ax.plot([final_x], [final_y], [final_z], marker='*', c='r', markersize=20,
    label='Final result')
    ax.scatter([final_x], [final_y], [final_z], marker='*', c='r', s=80, label='Final
    result')

    # ax.plot(track_x[0], track_y[0], c='b', marker='o')

    return ax

```

其余算法可视化的逻辑类似，总体为：

1. 弄清要解决的问题
2. 设计实现相应算法
3. 将数据过程中的任意点进行可视化
4. 将算法执行过程以动画形式表现出来

更多的关于代码，后续会以[博客文章](#)的形式进行讲解，不在本项目的讨论范围之中。

Q4. 如何生成的视频还有.gif图？

`gif` 和 `mp3` 分别用到了 `imagemagick` 和 `ffmpeg`。

关于 `imagemagick`，点击[这里](#)获取更多帮助。

关于 `ffmpeg`，点击[这里](#)获取更多信息。

Q5. 系统如何将python脚本执行的结果显示到前端的？如何修改？

前端采用HTML+JavaScript+CSS来控制，后端是SQLite数据库。将结果文件显示到前端利用的是Django框架的ORM（对象关系映射）机制和MVT（模型，视图，模板）设计模式（即经典的MVC设计模式），`view` 层通过从 `Model` 中取相应的结果文件的地址，以字典的形式传入到 `Template` 层，经过前端的渲染，从而得到显示效果。故要单修改前端风格，布局什么的，只需要修改 `template` 即可，修改其他则需要更具 `view` 的功能来进行相应的修改。

变更日志

变更日期	版本	新特性	修订者
2019-01-09	V0.2 (代号:炎)	实现了对于优选法的可视化，构建了基于python web框架的可视化用户界面。	魔法师L