

如何运行当前版本 CSST-IFS 仿真代码

目录

如何运行当前版本 CSST-IFS 仿真代码	1
代码获取.....	2
安装	2
安装后的文件说明:	3
.config 配置文件说明	4
仿真用到的数据文件.....	7
IFS 仪器仿真软件使用样例	9

代码获取

从国台 gitlab 服务器上拉取 CSST 仿真代码：链接: <https://csst-tb.bao.ac.cn/code/shaosim/ifs>

打包下载: `csst_ifs_sim-develop.tar.gz`

程序运行需要用到的数据库获取链接（百度云盘）: https://pan.baidu.com/s/1kvubVuJ2ceVklg_8UY_q0A，提取码：7676

安装

软件硬件环境要求：

内存：推荐 16GB 以上

存储：依赖数据文件 1GB 左右（可以和本体放在不同目录）

软件环境要求：Ubuntu 18.04 以上，Python 3.11

第三方包依赖参见 requirements.txt

安装说明：

推荐先安装 anaconda

拷贝安装包（包括本体和数据文件），进入目录：`cd ~/your_dir/`

检查软件依赖是否满足：`pip install -r requirements.txt`

galsim 的安装建议使用如下命令：`conda install -c conda-forge`

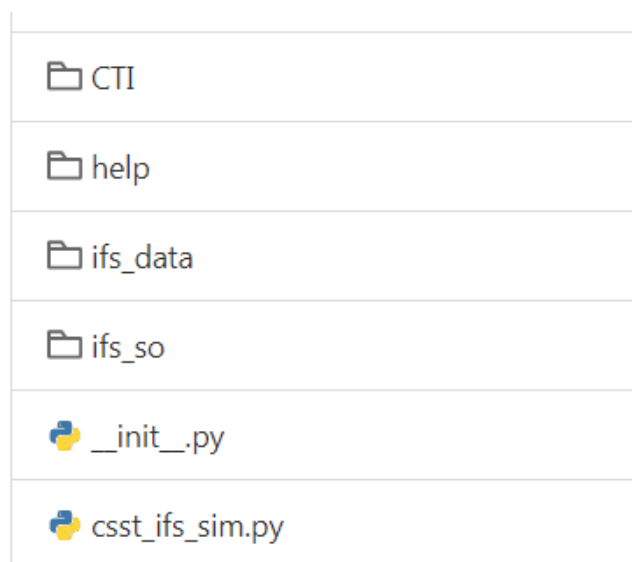
galsim

(推荐) 新建安装目录: `mkdir /the/dir/your/want/to/install/`

执行如下的安装命令: `pip install csst_ifs_sim-develop.tar.gz`

安装后的文件说明:

安装包文件夹下的子文件夹如下图所示,



- ✧ **CTI**: CTI 效应模块
- ✧ **ifs_data**: 提供默认.config 配置文件存放目录
- ✧ **ifs_so**: 仿真中需要用到的*.so 文件
- ✧ **csst_ifs_sim.py**: 仿真主程序

.config 配置文件说明

.config 文件中的参数列表及说明如下:

- **sky_fitsin**: 用于存放 GeHong 软件生成的光谱 datacube 的 fits 文件, 默认输入 datacube 为: IFS_inputdata/FengshuaiData/NGC3359_S4301.fits
- **bianpai_file**: 运行编排文件, 默认为: NGC3359_sequence_300x20_bkg.csv
- **sim_ver**: 仿真软件版本控制
- **bluesize**: 蓝端图像大小, 默认值为 4000, 请勿修改
- **redsize**: 红端图像大小, 默认值为 6000, 请勿修改
- **prescan**: prescan 大小, 请勿修改
- **overscan**: overscan 大小, 请勿修改
- **fullwellcapacity**: 满阱电子数, 请勿修改。
- **dark1_b**: 蓝端探测器第一个读出门的暗电流, 单位:e/s/pixel
- **dark2_b**: 蓝端探测器第二个读出门的暗电流, 单位:e/s/pixel
- **dark3_b**: 蓝端探测器第三个读出门的暗电流, 单位:e/s/pixel
- **dark4_b**: 蓝端探测器第四个读出门的暗电流, 单位:e/s/pixel
- **dark1_r**: 红端探测器第一个读出门的暗电流, 单位:e/s/pixel
- **dark2_r**: 红端探测器第二个读出门的暗电流, 单位:e/s/pixel
- **dark3_r**: 红端探测器第三个读出门的暗电流, 单位:e/s/pixel
- **dark4_r**: 红端探测器第四个读出门的暗电流, 单位:e/s/pixel

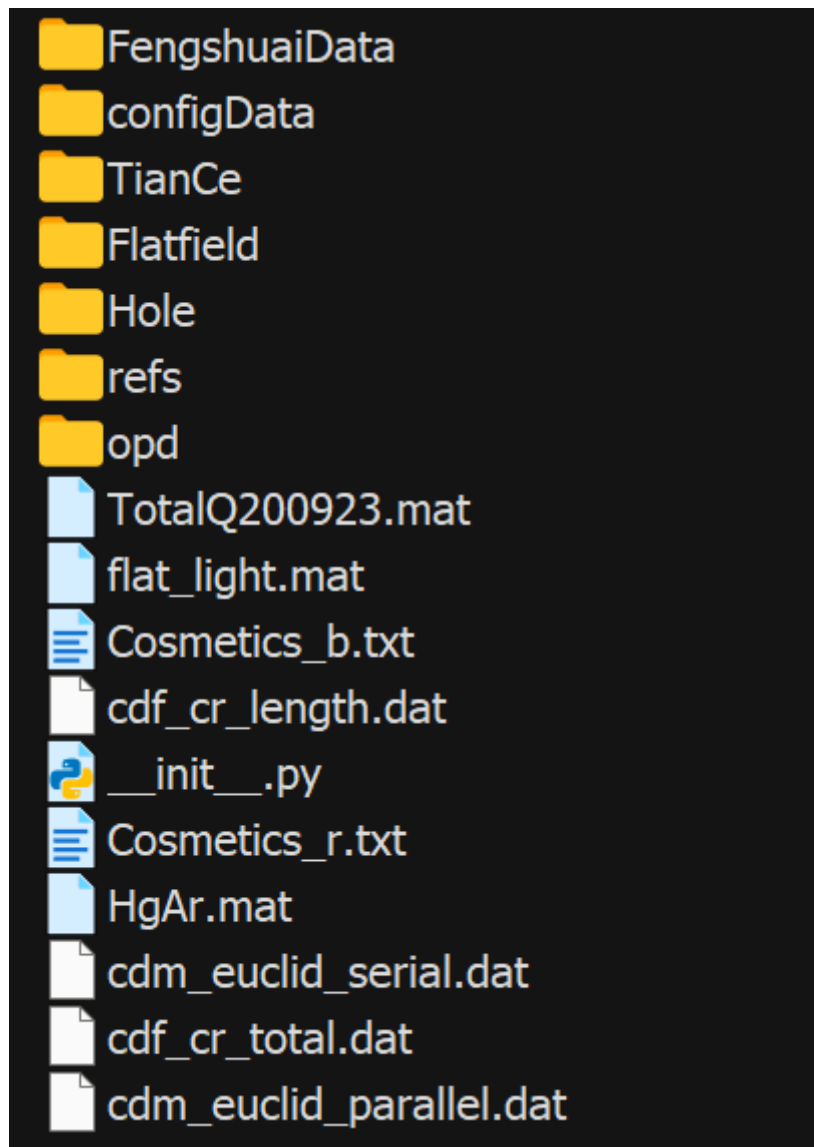
- **rn1_b**: 蓝端探测器第一个读出门的读出噪声, 单位:e/pixel
- **rn2_b**: 蓝端探测器第二个读出门的读出噪声, 单位:e/pixel
- **rn3_b**: 蓝端探测器第三个读出门的读出噪声, 单位:e/pixel
- **rn4_b**: 蓝端探测器第四个读出门的读出噪声, 单位:e/pixel
- **rn1_r**: 红端探测器第一个读出门的读出噪声, 单位:e/pixel
- **rn2_r**: 红端探测器第二个读出门的读出噪声, 单位:e/pixel
- **rn3_r**: 红端探测器第三个读出门的读出噪声, 单位:e/pixel
- **rn4_r**: 红端探测器第四个读出门的读出噪声, 单位:e/pixel
- **bias1_b**: 蓝端探测器第一个读出门的 bias, 单位:ADU/pixel
- **bias2_b**: 蓝端探测器第二个读出门的 bias, 单位:ADU/pixel
- **bias3_b**: 蓝端探测器第三个读出门的 bias, 单位:ADU/pixel
- **bias4_b**: 蓝端探测器第四个读出门的 bias, 单位:ADU/pixel
- **bias1_r**: 红端探测器第一个读出门的 bias, 单位:ADU/pixel
- **bias2_r**: 红端探测器第二个读出门的 bias, 单位:ADU/pixel
- **bias3_r**: 红端探测器第三个读出门的 bias, 单位:ADU/pixel
- **bias4_r**: 红端探测器第四个读出门的 bias, 单位:ADU/pixel
- **gain1_b**: 蓝端探测器第一个读出门的 gain, 单位: e/ ADU
- **gain2_b**: 蓝端探测器第二个读出门的 gain, 单位: e/ ADU
- **gain3_b**: 蓝端探测器第三个读出门的 gain, 单位: e/ ADU
- **gain4_b**: 蓝端探测器第四个读出门的 gain, 单位: e/ ADU
- **gain1_r**: 红端探测器第一个读出门的 gain, 单位: e/ ADU

- **gain2_r**: 红端探测器第二个读出门的 gain, 单位: e/ ADU
- **gain3_r**: 红端探测器第三个读出门的 gain, 单位: e/ ADU
- **gain4_r**: 红端探测器第四个读出门的 gain, 单位: e/ ADU
- **exptime**: 曝光时间, 默认为 300 秒
- **exposuretimes**: 单次观测曝光次数, 默认值 3
- **rdose**: 电子云密度, CTI 效应参数
- **parallelTrapfile**: CTI 效应输入数据文件路径
- **serialTrapfile**: CTI 效应输入数据文件路径
- **cosmeticsFile_b**: 蓝端探测器热像素、坏像素输入数据文件路径
- **cosmeticsFile_r**: 蓝端探测器热像素、坏像素输入数据文件路径
- **cosmicraylengths**: 宇宙线数据文件路径
- **cosmicraydistance**: 宇宙线数据文件路径
- **flatfieldM**: 探测器大尺度平场效应开关
- **sigma**: 探测器大尺度平场效应参数
- **darknoise**: 暗电流效应开关
- **sky_noise**: 天光背景、杂散光效应开关
- **cosmetics**: 热像素、坏像素效应开关
- **radiationDamage**: CTI 效应开关
- **cosmicRays**: 宇宙线效应
- **coveringFraction**: 宇宙线覆盖率百分比
- **bleeding**: 饱和溢出效应开关

- **nonlinearity**: 非线性效应开关
- **readoutnoise**: 读出噪声开关
- **save_cosmicrays**: 保存宇宙线仿真数据开关
- **appbianpai**: 运行编排开关

仿真用到的数据文件:

仿真软件需要用的数据存放在 IFS_inputdata 文件夹，其下的文件如下图所示。



- ✧ **FengshuaiData:** Gehong 软件生成的光谱输入 datacube
- ✧ **configData:** 存放.config 配置文件
- ✧ **TianCe:** 存放的望远镜轨道参数模拟数据
- ✧ **Flatfield:** 存放的大尺度平场矩阵
- ✧ **Hole:** 存放的模拟打孔板的模拟数据
- ✧ **refs:** 存放的模拟天光背景用到的数据
- ✧ **opd:** 存放的光程差模拟数据
- cdf_cr_length.dat 和 cdf_cr_total.dat: 模拟宇宙线用到的数据
- cdm_euclid_parallel.dat: CTI 效应用到的数据
- cdm_euclid_serial.dat: CTI 效应用到的数据
- Cosmetics_b.txt: 蓝端探测器的热像素和坏像素文件
- Cosmetics_r.txt: 红端探测器的热像素和坏像素文件
- TotalQ200923.mat: IFS 光学系统效率曲线数据

IFS 仪器仿真软件使用样例

```
1# -*- coding: utf-8 -*-
2"""
3This is a example for csst_ifs_sim running.
4"""
5from csst_ifs_sim import csst_ifs_sim
6import os
7# Set the parent path where the IFS_inputdata folder is stored.
8dir_path = '/home/yan/ifs_sim')
9# Set the path to the called config file.
10configfile = './csst_ifs_sim/ifs_data/IFS_sim_C10.config'
11# Set the path to the simulation data storage file.
12result_path = os.path.join(dir_path, 'ifs_sim_result')
13# Set the type of emulation, SCI, or DARK, or FLAT, or LAMP.
14sourcein = 'SCI'
15# Specifies whether the debug mode is used. If debug=True,
16# this mode is used and the program will only simulate a few wavelength slices
17# to save time. If debug=False, the full simulation will be run.
18debug = True
19
20# set the simulaiton number. With modifications, parallel simulations
21# can be executed.
22iLoop=1
23
24# run the main function
25# When the last parameter is yes and sourcein is LAMP,
26# the Hole simulation is executed.
27csst_ifs_sim.runIFSsim(sourcein, configfile,
28                        dir_path, result_path, iLoop, debug, 'no')
```