Aeolus 风场诊断模型

用户手册

项目名称：Aeolus 风场诊断模型

完成单位：

项目负责人：

开发周期：2021.02—2021.04

2021 年 4 月 27 日

目　录

一、 Aeolus 简介

Aeolus（风神）是一种基于有限点观测或廓线观测风场数据，重构区域风场的诊断模式。Aeolus 通过平流时间调整、距离-平方反比插值、近地面层稳定度-指数风廓线插值、艾克曼层地转风过渡、全局质量守恒约束等技术填充到wrfout格点模板文件，用于驱动HYSPLIT、FLEXPART 等质点扩散模型。

二、 模式原理和方法

2.1 垂直廓线插值

对同一观测帧内获取的三维风场观测数据，包括地面观测与探空观测、风廓线仪观测等，Aeolus 首先进行垂直方向的廓线估计。

对于近地面层风场，首先采用探空或者风廓线仪实测的风速廓线；若不可得，自地面至 200 米左右的各模式 η 层采用风速的幂次律外推，风向不变，即

$$U = U_0(Z/Z_0)^P \tag{2-1}$$

$$V = V_0(Z/Z_0)^P \tag{2-2}$$

其中 U 和 V 为 Z 高度处风速，U_0 和 V_0 为实际观测点高度的风速值，幂指数 P 由大气稳定度和地面粗糙度决定，见表 2.1。

表 2.1 幂指数 P 与稳定度和地面粗糙度的关系

稳定度 粗糙度	A	B	C	D	E	F
0.03	0.03	0.05	0.09	0.14	0.2	0.27
0.1	0.05	0.07	0.12	0.18	0.25	0.33
0.3	0.07	0.1	0.16	0.25	0.35	0.45
1	0.1	0.15	0.25	0.35	0.45	0.55

对于埃克曼层（距离地面高度约 200 米至 2000 米）与自由大气的风廓线估计，首先采用探空或者风廓线仪实测的风速廓线，若不可得，则根据纬度位置的中高层季节盛行风确定边界层顶（约 2000 米）自由大气地转风，再采用边界层顶地转风与近地面层顶（约 200 米）风场线性插值获得艾克曼层的垂直风廓线。

2.2 水平分布插值

水平方向上，模式采用加权插值法将 η 平面内各测站的风速实测值按其分量分别插值到模式网格点上，得到模式域 η 平面内的初猜风场：

$$U_{i,j} = \sum_{L=1}^{n} U_L \, W_L(r_{i,j}) / \sum_{L=1}^{n} W_L(r_{i,j}) \tag{2-3}$$

$$V_{i,j} = \sum_{L=1}^{n} V_L \, W_L(r_{i,j}) / \sum_{L=1}^{n} W_L(r_{i,j}) \tag{2-4}$$

其中：

$$W_L(r_{i,j}) = \begin{cases} {r_{i,j}}^{-2} & r_{i,j} \le R_x \\ 0 & r_{i,j} > R_x \end{cases} \tag{2-5}$$

$$r_{i,j} = 2R_e \arcsin\left(\sqrt{\sin^2\left(\frac{\varphi_L - \varphi_{i,j}}{2}\right) + \cos\varphi_L \cos\varphi_{i,j} \sin^2\left(\frac{\lambda_L - \lambda_{i,j}}{2}\right)}\right) \tag{2-6}$$

W_L 为权重函数（Aeolus 中采用距离平方反比），$r_{i,j}$ 为第 L 个观测点到格点(i, j)的水平距离，通过（2-6）半正矢公式求得。n 为观测点的个数，U_L 和 V_L 为第 L 个观测点的观测值，R_x 为观测点影响半径，当观测点数目大于 20 且相对均匀分散，该模式采用一个可变的影响半径使每个格点都选择距其最近的三个观测点的观测值进行内插。半正矢公式中，R_e 为地球半径，$\varphi_{i,j}$ 和 $\lambda_{i,j}$ 为格点（i，j）的纬度和经度，φ_L 和 λ_L 为第 L 个观测点的纬度和经度。

2.3 平流时间调整与层结衰减调整

水平分布插值器会遍历模式边界层内全部的 η 平面，逐一平面进行水平距离平方反比插值。若当前插值时间帧内无法检测到有效观测或有效观测点少于 3 处，则在当前时间帧的前后有效时间窗内筛择有效观测，并根据平流时间对其进行权重调整：

$$r = r_0 + |V| \cdot (t - t_0) \tag{2-7}$$

其中，r_0 为观测点到格点(i, j)的实际水平距离，$|V|$为观测点所测风速大小，t_0 为需要计算插值的当前时间帧，t 为观测点实际所在时间帧。

若实际观测点不在当前 η 平面上，则采用如下方案进行垂直方向的层结衰减调整权重：

$$r = \left[r_0 + A_0 \cdot \frac{(z - z_0)}{H_0}\right] \cdot e^{\varepsilon \frac{(z - z_0)}{H_0}} \tag{2-8}$$

其中，A_0 为中尺度重力波特征振幅，通常取 500-2500 米，H_0 为模式网格水平分辨率，z 为观测点距离地面铅直高度，z_0 为当前 η 平面距离地面铅直高度，ε 为无量纲层结衰减增益，通常取 0.5-1.5。该方法考虑大气垂直方向的层结特征，当实际观测所在高度距离所插值

η 平面较远时，其权重将受到较大惩罚。

2.4 质量守恒调整

模式随后再根据连续方程和变分法对初估插值风场进行调整，使得插值流场在边界层内满足域内总空气质量守恒（运动连续性）条件的约束，从而能够体现地形起伏对近地面流场的强迫作用，同时确保调整后的风场与初始场的总体偏差为最小。连续方程及其变分约束如下：

$$\frac{\partial U}{\partial x}+\frac{\partial V}{\partial y}+\frac{\partial W}{\partial z}=0 \tag{2-9}$$

$$I=\iiint\left[\begin{array}{c}\alpha_x^2\left(U-\tilde{U}\right)^2+\alpha_y^2\left(V-\tilde{V}\right)^2+\alpha_z^2\left(W-\tilde{W}\right)^2\\ +2\lambda\left(\frac{\partial U}{\partial x}+\frac{\partial V}{\partial y}+\frac{\partial W}{\partial z}\right)\end{array}\right]dxdydz \tag{2-10}$$

其中，其中 U、V 和 W 为调整后的风场，$\tilde{U},\tilde{V},\tilde{W}$为初估的插值风场，$\alpha_x$, α_y, α_z 为水平与垂直方向的高斯误差估计模数，通常取 $\alpha_x=\alpha_y=0.4\,\alpha_z$。

为了计算复杂地形状态的风场，风场诊断模式采用地形随动坐标系，η 坐标与笛卡尔坐标变换关系如下：

$$\eta(x,\mathrm{y},\mathrm{z})=s\frac{\left[z-z_g(x,y)\right]}{\left[s-z_g(x,y)\right]} \tag{2-11}$$

其中，s 为模式顶高，z_g 为气柱所在地形海拔高度，坐标系统在近地面层高度追随地形变化，而在模式顶为水平面。在 η 坐标系下，三维速度可以写为：

$$U^*=U$$

$$V^*=V$$

$$W^*=U\frac{\partial z_g}{\partial x}\left(\frac{\eta-s}{s-z_g}\right)+V\frac{\partial z_g}{\partial y}\left(\frac{\eta-s}{s-z_g}\right)+W\left(\frac{s}{s-z_g}\right) \tag{2-12}$$

其中，其中 U*、V*和 W*为地形追随坐标系下的三维风场。将（2-12）带入（2-9），在地形追随坐标系下可写为：

$$I = \iiint \left[\begin{array}{l} \alpha_x^2 \left(U - \tilde{U}\right)^2 + \alpha_y^2 \left(V - \tilde{V}\right)^2 + \alpha_z^2 \left(W^* - \widetilde{W^*}\right)^2 \\ +2\lambda \left(\frac{\partial U}{\partial x} + \frac{\partial V}{\partial y} + \frac{\partial W^*}{\partial z} - \frac{1}{s - z_g} \left(U \frac{\partial z_g}{\partial x} + V \frac{\partial z_g}{\partial y} \right) \right) \end{array} \right] dxdydz \quad (2\text{-}13)$$

整理(2-13)并代入连续方程，并定义：

$$\mathrm{Z}_x = \frac{1}{s - z_g} \frac{\partial z_g}{\partial x} \quad (2\text{-}14)$$

$$\mathrm{Z}_y = \frac{1}{s - z_g} \frac{\partial z_g}{\partial y} \quad (2\text{-}15)$$

可得：

$$\frac{1}{\alpha_x^2} \frac{\partial^2 \lambda}{\partial x^2} + \frac{1}{\alpha_y^2} \frac{\partial^2 \lambda}{\partial y^2} + \frac{1}{\alpha_z^2} \frac{\partial^2 \lambda}{\partial \eta^2} + \frac{1}{\alpha_x^2} \frac{\partial^2 [Z_x \lambda]}{\partial x} + \frac{1}{\alpha_y^2} \frac{\partial^2 [Z_y \lambda]}{\partial y} - Z_x \frac{1}{\alpha_x^2} \left(\frac{\partial \lambda}{\partial x} + Z_x \lambda \right) - Z_y \frac{1}{\alpha_y^2} \left(\frac{\partial \lambda}{\partial y} + Z_y \lambda \right)$$

$$= -\left[\frac{\partial U}{\partial x} + \frac{\partial V}{\partial y} + \frac{\partial W^*}{\partial z} - Z_x U - Z_y V \right] \quad (2\text{-}16)$$

(2-16)是关于 λ 的线性方程系统，求解该线性方程系统后，可以将 λ 带入如下方程组求得经质量守恒调整后的 U，V 和 W*：

$$U = \tilde{U} + \frac{1}{\alpha_x^2} \frac{d\lambda}{dx} + \frac{1}{\alpha_x^2} Z_x \lambda \quad (2\text{-}17)$$

$$V = \tilde{V} + \frac{1}{\alpha_y^2} \frac{d\lambda}{dy} + \frac{1}{\alpha_y^2} Z_y \lambda \quad (2\text{-}18)$$

$$W^* = \widetilde{W}^* + \frac{1}{\alpha_z^2} \frac{d\lambda}{d\eta} \quad (2\text{-}19)$$

对于（2-16）方程系统，其上边界、侧边界均采用开放边界方案，即$\lambda = 0$，对于模式底边界，采用闭边界条件，亦即不存在穿越模式面（地面）的运动：

$$\frac{d\lambda}{dn} = 0 \quad (2\text{-}20)$$

假设某实例中，Aeolus 诊断模式边界层内垂直分为 10 层，水平

方向为 100x100 个格点，对于（2-16）的方程系统，其系数矩阵 A 的大小为 10^5x10^5，共有 10^{10} 元素，若采用 8 字节双精度浮点数存储，则需要至少 74.5GB 内存，单机直接求解该线性系统较为困难。开发过程中进行小规模的计算测试，发现该系数矩阵填充率往往在 10^{-6} 至 10^{-5}，因此 Aeolus 采用 scipy.sparse 库中稀疏矩阵表示及求解算法, 对大规模线性系统进行高效的共轭梯度法求解，改善模式性能。模式最后会根据质量守恒约束调整后的三维风场，采用区域平均的风廓线指数插值获得近地面 10-m 高度风场。

三、 运行环境依赖

模型开发环境系统为 CentOS7.8.2003，控制脚本运行环境为 bash，模型内核代码采用基于 Anaconda 进行包管理的 python3.8 开发。模型在 CentOS6-7 操作系统、以及 python3.5 版本以上的 Anaconda 环境均进行了测试，可稳定运行。以开发环境为例，具体依赖介绍如下：

3.1 操作系统信息

Linux Kernel:

Linux hqlx27.ust.hk 3.10.0-1127.13.1.el7.x86_64 #1 SMP Tue Jun 23 15:46:38 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux

LSB Version: :core-4.1-amd64:core-4.1-noarch

Distributor ID: CentOS

Description: CentOS Linux release 7.8.2003 (Core)

Release: 7.8.2003

Codename: Core

3.2 基于 Anaconda 管理的 Python3.8 环境信息

conda 4.9.2 build with Python 3.8.5

表 3.1 Anaconda 核心依赖库

```
# Name                    Version                   Build  Channel
_libgcc_mutex             0.1                         main
ca-certificates           2021.1.19            h06a4308_0
certifi                   2020.12.5         py36h06a4308_0
cftime                    1.4.1                     pypi_0    pypi
ld_impl_linux-64          2.33.1               h53a641e_7
libedit                   3.1.20191231         h14c3975_1
libffi                    3.3                  he6710b0_2
libgcc-ng                 9.1.0                hdf63c60_0
libstdcxx-ng              9.1.0                hdf63c60_0
ncurses                   6.2                  he6710b0_1
netcdf4                   1.5.6                     pypi_0    pypi
numpy                     1.19.5                    pypi_0    pypi
openssl                   1.1.1j               h27cfd23_0
pandas                    1.1.5                     pypi_0    pypi
pip                       21.0.1            py36h06a4308_0
python                    3.6.13               hdb3f193_0
python-dateutil           2.8.1                     pypi_0    pypi
pytz                      2021.1                    pypi_0    pypi
readline                  8.1                  h27cfd23_0
scipy                     1.5.4                     pypi_0    pypi
setuptools                52.0.0            py36h06a4308_0
six                       1.15.0                    pypi_0    pypi
sqlite                    3.33.0               h62c20be_0
tk                        8.6.10               hbc83047_0
wheel                     0.36.2             pyhd3eb1b0_0
wrapt                     1.12.1                    pypi_0    pypi
wrf-python                1.3.1                     pypi_0    pypi
xarray                    0.16.2                    pypi_0    pypi
xz                        5.2.5                h7b6447c_0
zlib                      1.2.11               h7b6447c_3
```

上述包依赖最低配置环境已经写入 requirements.txt，在连接互联网的计算机上，可以通过如下命令安装：

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

命令执行回显信息如下：

```
Requirement already satisfied: certifi==2020.12.5 in /disk/r127/metctm1/
soft/anaconda3/envs/test_aeolus/lib/python3.6/site-packages (from -r req
uirements.txt (line 1)) (2020.12.5)
Collecting cftime>=1.4.1
  Using cached cftime-1.4.1-cp36-cp36m-manylinux2014_x86_64.whl (316 kB)
Collecting netCDF4>=1.5.6
  Using cached netCDF4-1.5.6-cp36-cp36m-manylinux2014_x86_64.whl (4.7 M
B)
Collecting numpy>=1.19.5
  Using cached numpy-1.19.5-cp36-cp36m-manylinux2010_x86_64.whl (14.8 M
B)
Collecting pandas>=1.1.5
  Using cached pandas-1.1.5-cp36-cp36m-manylinux1_x86_64.whl (9.5 MB)
Collecting python-dateutil>=2.8.1
```

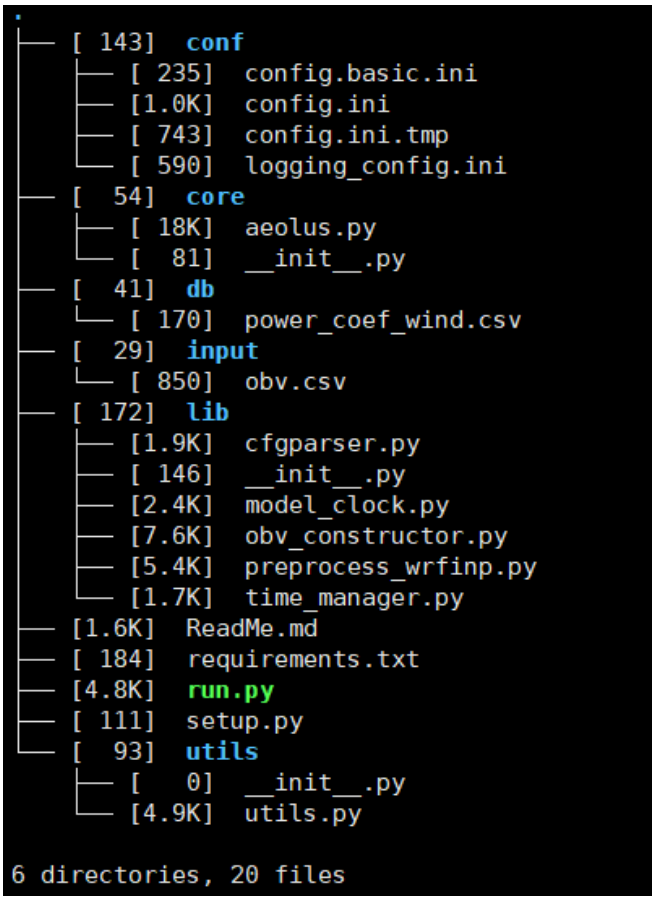
```
  Using cached python_dateutil-2.8.1-py2.py3-none-any.whl (227 kB)
Collecting pytz>=2021.1
  Using cached pytz-2021.1-py2.py3-none-any.whl (510 kB)
Collecting scipy>=1.5.4
  Using cached scipy-1.5.4-cp36-cp36m-manylinux1_x86_64.whl (25.9 MB)
Collecting six>=1.15.0
  Using cached six-1.15.0-py2.py3-none-any.whl (10 kB)
Collecting wrapt>=1.12.1
  Using cached wrapt-1.12.1-cp36-cp36m-linux_x86_64.whl
Collecting wrf-python>=1.2.0
  Using cached wrf_python-1.3.1-cp36-cp36m-linux_x86_64.whl
Collecting xarray>=0.16.2
  Using cached xarray-0.16.2-py3-none-any.whl (736 kB)
Requirement already satisfied: setuptools in /disk/r127/metctm1/soft/ana
conda3/envs/test_aeolus/lib/python3.6/site-packages (from wrf-python>=1.
2.0->-r requirements.txt (line 11)) (52.0.0.post20210125)
Installing collected packages: six, pytz, python-dateutil, numpy, wrapt,
 pandas, cftime, xarray, wrf-python, scipy, netCDF4
Successfully installed cftime-1.4.1 netCDF4-1.5.6 numpy-1.19.5 pandas-1.
1.5 python-dateutil-2.8.1 pytz-2021.1 scipy-1.5.4 six-1.15.0 wrapt-1.12.
1 wrf-python-1.3.1 xarray-0.16.2
```

不能接入互联网的计算机需要首先安装 Anaconda 后，通过异地下载依赖包并逐一安装导入 Anaconda 环境：

```
$pip3 install xxx  --user --no-index --find-links=~/pippackage_python3
```

四、 项目代码架构

Aeolus 项目基本架构图如下：



4.1 配置文件

config.basic.ini

./conf/config.basic.ini：模型基本配置文件，可以在此文件中设置基本的 IO 选项、模式插值起止时间、插值步长等属性。

config.ini

./conf/config.ini：模型详细配置文件。可以在此文件中设置模式内置参数，如缺省稳定度条件、边界层高度、层结衰减增益、高斯误差估计模数等。

logging_config.ini

./conf/logging_config.ini：日志配置文件，控制 logging 模块输出样式，输出流重定向等。

4.2 静态数据文件

power_coef_wind.csv

./db/power_coef_wind.csv：由地表粗糙度长度-稳定性级别关系确定的近地面层风指数系数表。

wrfout_d0?

./db/wrfout_d0?：WRFOUT 插值模板文件，通过 conf.basic.ini 指定。

4.3 观测输入文件

obv.csv

./input/obv.csv：观测数据文件，可以在 config.ini 中分配文件名。文件格式如下，请参考样板文件提供观测输入：

```
yyyymmddhhMM，lat，lon，height，wind_speed，wind_dir，temp，rh，pres，attr1，attr2
```

4.4 模块文件

主程序

./run.py：运行 Aeolus 的主脚本程序。

库模块

./lib/cfgparser.py：包含 config.ini 的 IO 及检测功能的模块文件

./lib/preprocess_wrfinp.py：用于构造 field_hdl 对象的类模板，其中包含用于插值的模板网格字段数据，例如 lat，lon，U，V

./lib/obv_constructor.py：用于构建站点观测对象的类模板

./lib/time_manager.py：用于构造运行时间管理器对象的类模板

./lib/model_clock.py：用于构造模式插值时钟的类模板

核心模块

./core/aeolus.py：核心模块，Aeolus 插值器，将点观测插到 WRF 网格的实现模块

实用程序

./utils/utils.py：常用的实用程序杂锦，例如风速风向转换函数，网格定位器，计算大圆距离的 Haversine 公式，logging 日志处理等。

4.5 文档

./doc/ 与模型相关的文件。

五、 运行实例及代码调用栈

5.1 运行流程

模型按第三章完成环境依赖配置后，即可测试运行实例。首先拷贝目标 wrfout 模板到 db 目录下，在 conf.basic.ini 中指定 `input_wrf` 参数指向 wrfout 模板、设置插值开始时间 `interp_strt_t`、插值小时数 `interp_t_length`、插值补偿（分钟间隔）`interp_interval`。在 input 目录下按观测情况给定 obv.csv 文件，确保时间与插值时段对应，观测点经纬度、距离地面高度、风速、风向、温度变量不可缺测（温度可以按误差±5℃估计量给定）。随后运行 run.py 文件：

```
$python3 run.py
```

运行成功后，标准输出流末尾部分返回 time_manager 统计信息形式如下：

```
----------------TIME MANAGER PROFILE----------------
       INPUT MODULE:    0.0497s     1x   0.2%
 CONSTRUCTION MODULE:    1.1059s     1x   5.0%
        CAST MODULE:    0.9977s     7x  31.8%
      OUTPUT MODULE:    1.9771s     7x  63.0%
```

```
  TOTAL ELAPSED TIME:   21.9802s     1x 100.0%

----------------TIME MANAGER PROFILE----------------
```

输出文件位于 output 目录下，完全依照 wrfout 模板和插值时间戳生成相应的输出文件：

```
-rw-r--r-- 1 metctm1 hq  18553976 Apr 18 12:34 wrfout_d07_2021-02-08_12:
00:00

-rw-r--r-- 1 metctm1 hq  18553976 Apr 18 12:34 wrfout_d07_2021-02-08_13:
00:00
```

5.2 代码调用栈

run.py 作为诊断模式的主控程序，其调用栈如下：

1. 初始化流程

run.py 首先进行如下模式初始化流程：

a. 初始化 time_manager 模块

b. 载入 logging 配置文件 logging_config.ini

c. 载入基本配置文件 config.basic.ini 和详细配置文件 config.ini 并合并配置

d. 读入 obv.csv 并调用 lib.obv_constructor.obv_examiner()模块函数对输入数据进行质量控制检查

e. 读入风廓线指数-稳定度静态数据库

f. 调用 lib.preprocess_wrfinp.wrf_mesh()类构建 wrfout 模板对象 fields_hdl

g. 调用 lib.obv_constructor.obv()类构建观测对象，返回所有观测对象（对象列表）

h. 调用 fields_hdl.get_area_pvalue()方法获得模式区域平均风廓线指数

i. 调用 lib.obv_constructor.set_upper_wind()模块函数插值观

测点风廓线

j. 调用 lib.model_clock.model_clock()构建模式插值时钟

k. 调用 core.aeolus.aeolus()类构建 Aeolus 插值器 estimator

2. Aeolus 插值管线

run.py 控制插值器调用 estimator.cast()方法，运行 Aeolus 核心插值管线如下：

a. 插值器 estimator 调用 core.aeolus.select_obv()模块函数，筛选当前插值帧的有效观测

b. 插值器 estimator 按有效观测构建观测-格点距离矩阵 dis_mtx_u，dis_mtx_v, dis_mtx_t，写入插值器属性。

c. 对 dis_mtx 进行平流时间调整和层结衰减调整

d. 决定每个观测点的有效半径 R_x

e. 调用 core.aeolus.inv_dis_wgt_2d()模块函数，对所有 η 层的 UV 风进行距离平方反比插值，获得初猜风场

f. 调用模块函数 core.aeolus.diag_vert_vel()诊断初猜垂直速度

g. 调用 estimator.adjust_mass()方法，使用 scipy.sparse 库求解关于 λ 的大规模线性方程系统，获得经质量守恒调整后的三维风场。

h. 插值近地面 10-m 高度风场与 2-m 气温，结束插值器 estimator.cast()方法调用

3. 输出及终止

a. run.py 调用 core.aeolus.output_fields()模块方法输出 wrfout 文件

b. 调用 clock.advance()方法，模式时钟步进插值下一时间帧。若 clock.advance()方法判定插值时段结束，time_manager 打印运行时间信息，回收资源，结束 Aeolus。

六、 项目版本迭代

v0.01 (Feb 19, 2021)

1. MVP 版本：完成规格制定，明确模块边界、架构、核心管线，完成最小化可行的 IDW 插值器。

v0.02 (Feb 24, 2021)

1. MVP 版本：完成最小化可行廓线插值序列，固定模块设计，完成基本单元测试。

v0.03 (Mar 05, 2021)

2. 采用 total_seconds()方法修正了 datetime.timedelta 对象 seconds 属性造成的观测数据时间帧判定 bug
3. 通过对每个模式时钟周期重构 obv 对象，修正了 obv_lst 观测对象列表在 aeolus.cast 方法中传引用后产生的 obv.u_prof 和 obv.v_prof 属性被重写的 bug（表现为部分站点观测数据不起作用）
4. 通过 effect_win（有效观测窗口）参数调节非诊断时刻汇入的观测数据，完全接受任意时间戳获取的观测数据，并采用平流时间进行权重调整
5. 风向输入同时支持 16 方位输入和 0-360 度输入
6. 确保采用全域范围内最邻近插值格点的三个观测对象进行

IDW 空间插值

v0.04 (Mar 08, 2021)

1. 引入输入数据质量控制检测 lib.obv_constructor.obv_examiner() 和异常处理 utils.throw_error()，强制要求输入观测时间戳、经纬度、风速观测高度、风速值、风向、2m 气温（允许估计值）
2. 额外插值并输出 U10、V10 以及 T2 变量。
3. obv 对象中通过 copy 传值，修正了 obv_lst 观测对象列表在 aeolus.cast 方法中传引用后产生的 obv.u_prof 和 obv.v_prof 属性被重写的 bug（完全修正 v0.03-2）
4. 增加 time_manager 对象对循环内执行事件的统计
5. 要求静态输入数据 wrfout 模板放置于 db 数据库目录下
6. 将模式基本配置文件（config.basic.ini）和全局配置文件（config.ini）分离

v0.90 (Mar 30, 2021)

7. **核心功能实现：**解决质量守恒约束下的变分优化问题，对大规模线性系统进行高效的共轭梯度法求解，使得插值流场在边界层内满足域内总空气质量守恒，从而体现地形起伏对近地面流场的强迫作用。（在部分情形下存在求解不稳定的 bug）
8. 根据质量守恒约束调整后的三维风场，采用区域平均的风廓线指数 pval 插值 10-m 高度 UV。

v0.95 (Mar 31, 2021)

1. 纳入探空数据，从而对理想的风廓线模型进行合理调整。完全接受位于任意垂直层次上的观测数据，采用对流扩散尺度和层

次差指数衰减进行权重调整

2. 垂直方向采用层结递减率（γ=0.65°C/100m）对无廓线观测的气温观测数据进行调整

V1.00 (Apr 01, 2021)

1. 采用逐步试探法修正质量守恒约束调整在部分情形下求解不稳定的 bug。根据质量守恒约束调整后的三维风场，采用区域平均的风廓线指数 pval 插值 10-m 高度 UV。

V1.10 (Apr 21, 2021)

1. 引入 logging 模块，进行规范化的事件日志分级记录，重定向保存至 aeolus.log
2. 对 config.basic.ini 及 obv.csv 进行不同规则的输入有效性检查，包括输入格式，可靠值域等，提高程序的健壮性和可靠性。
3. 对同一时间帧可获得观测点聚集且较少（<20 观测点）的情景，采用全站点 IDW 加权替代最邻近 3 站点加权法，以获得更平滑与合理的风场估计。