

Программа для ЭВМ
“Библиотека подпрограмм для модуля анализа и расчета рассеяния загрязняющих веществ
в атмосфере”

Руководство пользователя

ООО “СитиЭйр”, 2020

Содержание:

- 1. Введение.**
- 2. Установка библиотеки на компьютер.**
- 3. Системные требования для работы библиотеки.**
- 4. Описание файловой структуры дисперсионной модели рассеивания загрязнения в атмосфере FLEXPART.**
- 5. Описание основных функций подготовки расчета модели в библиотеке.**
 - 5.1. Подготовка контрольного файла запуска prep_command.**
 - 5.2. Подготовка файла с параметрами выходной сетки модели prep_outgrid.**
 - 5.3. Подготовка метеорологических данных для расчета prep_available.**
 - 5.4. Подготовка файла с параметрами источников выброса загрязняющих веществ prep_releases.**
 - 5.4.1. Описание структуры csv файла с параметрами источников выброса.**
- 6. Функция для обработки результатов расчетов модели read_output.**

1. Введение.

Данная библиотека подпрограмм предназначена для работы с дисперсионной моделью рассеивания загрязнения в атмосфере FLEXPART. В нее входит набор инструментов, необходимый для автоматизации запусков расчетов модели. Основная проблема, которую помогает решить разработанная библиотека – это упрощение работы с контрольными файлами запуска модели, которые имеют текстовый формат (ASCII) фиксированной структуры.

Основные цели создания библиотеки:

- упростить и автоматизировать процесс создания текстовых файлов в необходимом для запуска модели формате, содержащих в себе характеристики источников выбросов загрязнения в атмосферу (файл RELEASES);
- автоматизировать процесс создания или обновления контрольного файла запуска модели (файл COMMAND);
- автоматизировать процесс создания файла с описанием параметров трехмерной сетки, на которой будут выводиться результаты расчетов модели (файл OUTGRID);
- автоматизировать процесс скачивания метеорологических данных из внешних источников;
- упростить обработку выходных бинарных файлов модели и сведение результатов расчетов к одному (6-ти мерному) массиву для удобства последующей обработки.

2. Установка библиотеки на компьютер.

Библиотеку можно установить при помощи менеджера установки пакетов Python – pip, требуемая версия Python 3.7 или выше. Необходимо выполнить следующую команду:

```
pip install https://cityair.io/software/libraries/plume-master.zip
```

Также возможна установка в виде распаковки архива, содержащего программный код. Скачать соответствующий архив можно с FTP сервера:

FTP:

После скачивания и распаковки архива необходимо проверить файл ~/plume/requirements.txt, а также проверить наличие всех описанных в нем библиотек на своем компьютере. При необходимости установить недостающие библиотеки.

Для корректной работы библиотеки необходимо обеспечить доступ в интернет на ftp сервера NOAA (National Oceanic and Atmospheric Administration, USA), поскольку оттуда идет скачивание метеорологической информации, необходимой для расчетов.

3. Системные требования для работы библиотеки.

Библиотека подпрограмм написана на языке Python. Для ее работы необходимо, чтобы были установлены дополнительные библиотеки:

pandas — программная библиотека на языке Python для обработки и анализа данных;

requests – программная библиотека для составления HTTP-запросов;

progressbar2 – программная библиотека для визуального отображения прогресса длительных процессов;

PyYAML – синтаксический анализатор для YAML. YAML - это формат сериализации данных, разработанный для удобства чтения и взаимодействия с языками сценариев;

attrs – библиотека, которая позволяет обеспечивать структурированные типы данных;

Jinja2 – это текстовый шаблонизатор для языка программирования Python, может быть использован для создания любого вида разметки.

4. Описание файловой структуры дисперсионной модели рассеивания загрязнения в атмосфере FLEXPART.

Модель FLEXPART написана на языке программирования FORTRAN и имеет следующую структуру (по умолчанию) и расположение ключевых файлов.

Исходим из того, что при установке модели была создана папка, содержащая всю модель, с названием, например, *~/FLEXPART/*. Тогда внутренняя структура этой папки будет выглядеть следующим образом:

~/FLEXPART/Options/ – папка с настроечными файлами для запуска расчетов, а также описанием характеристик веществ, которые могут использоваться в расчетах

~/FLEXPART/Src/ – папка с исходным кодом модели

~/FLEXPART/Output/ – папка с результатами расчетов

~/FLEXPART/flexpart_gfs – скомпилированный исполняемый файл для запуска модели

~/FLEXPART/pathnames – файл с путями для файлов с входными данными, настройками расчета и файлов с результатами расчетов. Этот файл содержит в себе 4 строчки с основными директориями, например следующего содержания:

«

~/FLEXPART/Options/

~/FLEXPART/Output/

~/FNL/FNL/

~/FNL/FNL/AVAILABLE

»

3-я строка представляет собой адрес, где лежат файлы с необходимыми метеорологическими полями (задается вручную). 4-я строка это путь к файлу, в котором содержится описание файлов, которые есть на жестком диске с описанием даты и времени, которым они соответствуют.

Основные файлы для автоматизации запусков RELEASES, COMMAND и OUTGRID лежат в папке *~/FLEXPART/Options/*.

5. Описание основных функций подготовки расчета модели в библиотеке.

5.1. Подготовка контрольного файла запуска `prep_command`.

Описание входных параметров данной функции выглядит следующим образом:

```
def prep_command(start: datetime,
                 finish: datetime,
                 is_forward: bool = True,
                 forecast_step: int = 1200,
                 average_step: int = 1200,
                 out_step: int = 1200,
                 dir: str = "", **kwargs)
```

где:

- **start**: время начала расчета прогноза в формате `datetime`;
- **finish**: время окончания расчета прогноза в формате `datetime`;
- **is_forward**: параметр режима расчета модели (в прямом или обратном относительно времени), логическая переменная [`True`, `False`]
- **forecast_step**: временной шаг выходного файла в [секундах]
- **average_step**: время усреднения выходных данных в [секундах]
- **out_step**: частота выдачи в [секундах], \leq шаг выходного файла
- **dir**: путь к файлу `COMMAND`, в который сохранять информацию.

В результате работы данной функции получаем файл `COMMAND`, находящийся в папке `~/FLEXPART/Options/`, содержащий следующую информацию (пример):

```
+++++ HEADER +++++
+++++ HEADER +++++
+++++ HEADER +++++
+++++ HEADER +++++
+++++ HEADER +++++
+++++ HEADER +++++
+++++ HEADER +++++
1          (тип запуска 1 – прямой расчет, -1 - обратный)
20170101 000000 (дата YYYYMMDD и время HHMMSS начала расчета)
20171230 000000 (дата YYYYMMDD и время HHMMSS окончания расчета)
  10800      (временной шаг выходного файла в [секундах])
  10800      (время усреднения выходных данных в [секундах])
  10800      (частота выдачи в [секундах],  $\leq$  шаг выходного файла)
9999999     ()
60  SYNC     (модельный шаг по времени в [секундах])
-5.0  CTL     ()
4     IFINE   ()
1     IOU     ()
0     IPOU    ()
0     LSUBGRID ()
0     LCONVECTION ()
```

```

1     LAGESPECTRA    ()
0     IPIN           ()
1     IOFR           ()
0     IFLUX          ()
0     MDOMAINFILL   ()
1     IND_SOURCE     ()
1     IND_RECEPTOR ()
0     MQUASILAG     ()
0     NESTED_OUTPUT ()
0     LINIT_COND    ()

```

Параметры без описания не корректируются в процессе выполнения функции `prep_command`, поскольку относятся к настройке внутренних процессов расчета модели и должны настраиваться специалистом по моделированию. Настройка производится в шаблоне, который расположен (директория, где расположена библиотека)/*plume/templates/COMMAND.jinja*.

5.2. Подготовка файла с параметрами выходной сетки модели `prep_outgrid`.

Описание входных параметров данной функции выглядит следующим образом:

```

def prep_outgrid(long,
                  lat,
                  levels: List[float],
                  x_points: int = 101,
                  y_points: int = 101,
                  x_distance: int = 0.002,
                  y_distance: int = 0.002,
                  dir: str = "", **kwargs):

```

где

- **long**: географическая долгота левого нижнего угла интересующего домена (в градусах)
- **lat**: географическая широта левого нижнего угла интересующего домена (в градусах)
- **levels**: список высот верхних границ уровней для вывода (в метрах)
- **x_points**: количество узлов вдоль круга широты
- **y_points**: количество узлов вдоль меридиана
- **x_distance**: расстояние между узлами сетки (в градусах) вдоль круга широты
- **y_distance**: расстояние между узлами сетки (в градусах) вдоль меридиана
- **dir**: путь к файлу `OUTGRID`, в который сохранять информацию.

В результате работы данной функции получаем файл `OUTGRID`, находящийся в папке `~/FLEXPART/Options/`, содержащий следующую информацию (пример):

```

*****
*****

```

```

*
*
*      Input file for the Lagrangian particle dispersion model
FLEXPART      *
*
*      Please specify your output grid
*
*
*
*****
*****

1.  ----- .-----          4X, F11.4
      8.5321      (долгота в градусах левого нижнего угла выходной сетки)
OUTLONLEFT      (левая граница первой ячейки – не ее центра)

2.  ----- .-----          4X, F11.4
      48.0316     (широта в градусах левого нижнего угла выходной сетки)
OUTLATLOWER     (нижняя граница первой ячейки – не ее центра)

3.  -----          4X, I5
      138          (количество точек вдоль широты (=количество ячеек + 1))
NUMXGRID

4.  -----          4X, I5
      91           (количество точек вдоль долготы (=количество ячеек + 1))
NUMYGRID

5.  ----- .---          4X, F10.3
      0.010        (расстояние между точками в градусах)
DXOUTLON

6.  ----- .---          4X, F10.3
      0.010        (расстояние между точками в градусах)
DYOUTLAT

10. ----- .-          4X, F7.1
      50.0
LEVEL 4          (высота верхнего уровня (верхняя граница сетки))
10. ----- .-          4X, F7.1
      100.0
LEVEL 4          (высота верхнего уровня (верхняя граница сетки))
10. ----- .-          4X, F7.1
      150.0
LEVEL 4          (высота верхнего уровня (верхняя граница сетки))

```

От этого файла зависят размеры и характеристики выходных файлов. Шаблонный файл для подготовки файла `OUTGRID` находится в *(директория, где расположена библиотека)/plume/templates/OUTGRID.jinja*.

5.3. Подготовка метеорологических данных для расчета `prep_available`.

```
def prep_available(start: datetime,
                  finish: datetime,
                  ahead_reserve: int,
                  forecast_step_delta: int,
                  host_url,
                  dir: str = "",
                  meteodata_dir="meteo_data", **kwargs):
```

где

- **start**: время начала расчета прогноза в формате datetime;
- **finish**: время окончания расчета прогноза в формате datetime;
- **ahead_reserve**: количество дополнительных файлов, по метеорологии, которое необходимо скачать после даты finish
- **forecast_step_delta**: временной шаг для скачивания метеорологических файлов (в часах)
- **host_url**: адрес, с которого производить скачивание метеорологических файлов
- **dir**: директория, в которой содержится файл AVAILABLE
- **meteodata_dir**: директория, в которую скачиваются файлы с метеорологическими полями.

В результате работы данной функции должны быть закачаны в директорию *meteodata_dir* все необходимые файлы для запуска расчета и информация о них должны быть занесена в соответствующий файл AVAILABLE. Вид файла AVAILABLE:

```
«
DATE          TIME          FILENAME          SPECIFICATIONS
YYYYMMDD     HHMMSS
-----
20170101     000000           fnl_20170101_00_00.grib2           ON DISC
20170101     060000           fnl_20170101_06_00.grib2           ON DISC
20170101     120000           fnl_20170101_12_00.grib2           ON DISC
20170101     180000           fnl_20170101_18_00.grib2           ON DISC
»
```

где первый столбец – это год месяц день, содержащийся в файле; второй столбец – час минута и секунда, содержащиеся в файле; третий столбец – название файла, соответствующего первым двум столбцам и лежащего в *meteodata_dir*; четвертый столбец – параметр, указывающий на то, что данный метеорологический файл находится на диске устройства.

Шаблонный файл для подготовки файла AVAILABLE находится в (*директория, где расположена библиотека*)/*plume/templates/AVAILABLE.jinja*.

5.4. Подготовка файла с параметрами источников выброса загрязняющих веществ prep_releases.

Описание входных параметров данной функции выглядит следующим образом:


```
def prep_releases(csv_path: str,
                 start: datetime,
                 finish: datetime,
                 releases_dir: str,
                 dir="", **kwargs):
```

где

- `csv_path`: путь к файлу в формате csv, в котором содержатся параметры источников выбросов;
- `start`: время начала расчета прогноза в формате `datetime`;
- `finish`: время окончания расчета прогноза в формате `datetime`;
- `releases_dir`: папка, в которую будут выгружены все файлы RELEASES для запуска расчета
- `dir`: путь к папке, в которую выгружаются файлы RELEASES.

В результате работы данной функции получаем набор файлов RELEASES в необходимом формате, готовых для запуска расчета. Для примера файл RELEASES выглядит следующим образом:

```
+++++ HEADER +++++
+++++ HEADER +++++
+++++ HEADER +++++
+++++ HEADER +++++
+++++ HEADER +++++
+++++ HEADER +++++
+++++ HEADER +++++
+++++ HEADER +++++
+++++ HEADER +++++
+++++ HEADER +++++
+++++ HEADER +++++
+++++ HEADER +++++
+++++ HEADER +++++
+++++ HEADER +++++
2 (общее количество веществ, выбрасываемых в этом запуске модели)
1 (номер вещества №1, см. ~/FLEXPART/Options/SPECIES/)
4 (номер вещества №2, см. ~/FLEXPART/Options/SPECIES/)

20170101 091500 (дата YYYYMMDD и время HHMMSS начала выброса)
20171230 235500 (дата YYYYMMDD и время HHMMSS окончания выброса)
   8.5905151 (долгота в градусах левого нижнего угла источника)
  48.090351 (широта в градусах левого нижнего угла источника)
   8.5905151 (долгота в градусах правого верхнего угла источника)
  48.090351 (широта в градусах правого верхнего угла источника)
1 (флаг высоты выброса: 1 – в [м] над землей, 2 – в [м] над уровнем моря)
   0.000 (нижняя граница выброса в [м])
  50.000 (верхняя граница выброса в [м])
2000000 (общее количество частиц, выбрасываемое в расчете)
1.0 (общая масса вещества №1, выбрасываемая в расчете)
3.0 (общая масса вещества №2, выбрасываемая в расчете)
ТЕСТОВЫЙ_ИСТОЧНИК_№1 (название источника)
```

```
20170101 000000
20171230 000000
    9.7234192
    48.27084
    9.7234192
    48.27084
1
    100.000
    150.000
2000000
4.0
5.0
ТЕСТОВЫЙ_ИСТОЧНИК_№2
```

```
20170101 000000
20171230 000000
    9.23703
    48.478832
    9.23703
    48.478832
1
    300.000
    450.000
2000000
1.0
1.0
ТЕСТОВЫЙ_ИСТОЧНИК_№3
```

Здесь представлена ситуация, в которой есть 3 стационарных источника выброса загрязнения в атмосферу со своими физическими параметрами и каждый из них выбрасывает по два (одинаковых) химических вещества.

Шаблонный файл для подготовки файла RELEASES находится в *(директория, где расположена библиотека)/plume/templates/RELEASES.jinja*.

5.4.1. Описание структуры csv файла с параметрами источников выброса.

Файл может содержать различные параметры в столбцах, но при этом обязательно должны быть столбцы со следующими названиями:

- «ИНН» - любой идентификатор, позволяющий определить принадлежность данного источника к какой-то группе источников, например по собственнику;
- «Наименование загрязняющего вещества» - название химического вещества, которое выбрасывается на данном источнике;
- «Выброс» - масса выбрасываемого вещества;
- «Координаты» - координаты источника выброса в формате [долгота, широта]

В качестве разделителя в csv файле используется символ «|» (вертикальная полоса).

6. Функция для обработки результатов расчетов модели read_output.

Описание входных параметров данной функции выглядит следующим образом:

```
def read_output(filename: str,  
                time_steps_count: int,  
                species_count: int,  
                sources_count: int,  
                levels_count: int,  
                y_points: int,  
                x_points: int) -> np.ndarray:
```

где

- filename: путь к выходному файлу модели;
- time_steps_count: количество шагов по времени, содержащееся в файле;
- species_count: количество химических веществ, участвующих в расчете;
- sources_count: количество источников, участвующих в расчете;
- levels_count: количество уровней по высоте, на которых выводятся значения концентраций;
- y_points: количество ячеек вдоль меридиана;
- x_points: количество ячеек вдоль круга широты.

В результате работы этой процедуры должен быть получен шестимерный массив, соответствующий типу библиотеки NumPy – [numpy.ndarray].

Далее полученный массив легко использовать для последующей обработки, загрузки в базу данных или визуализации (рис.1).

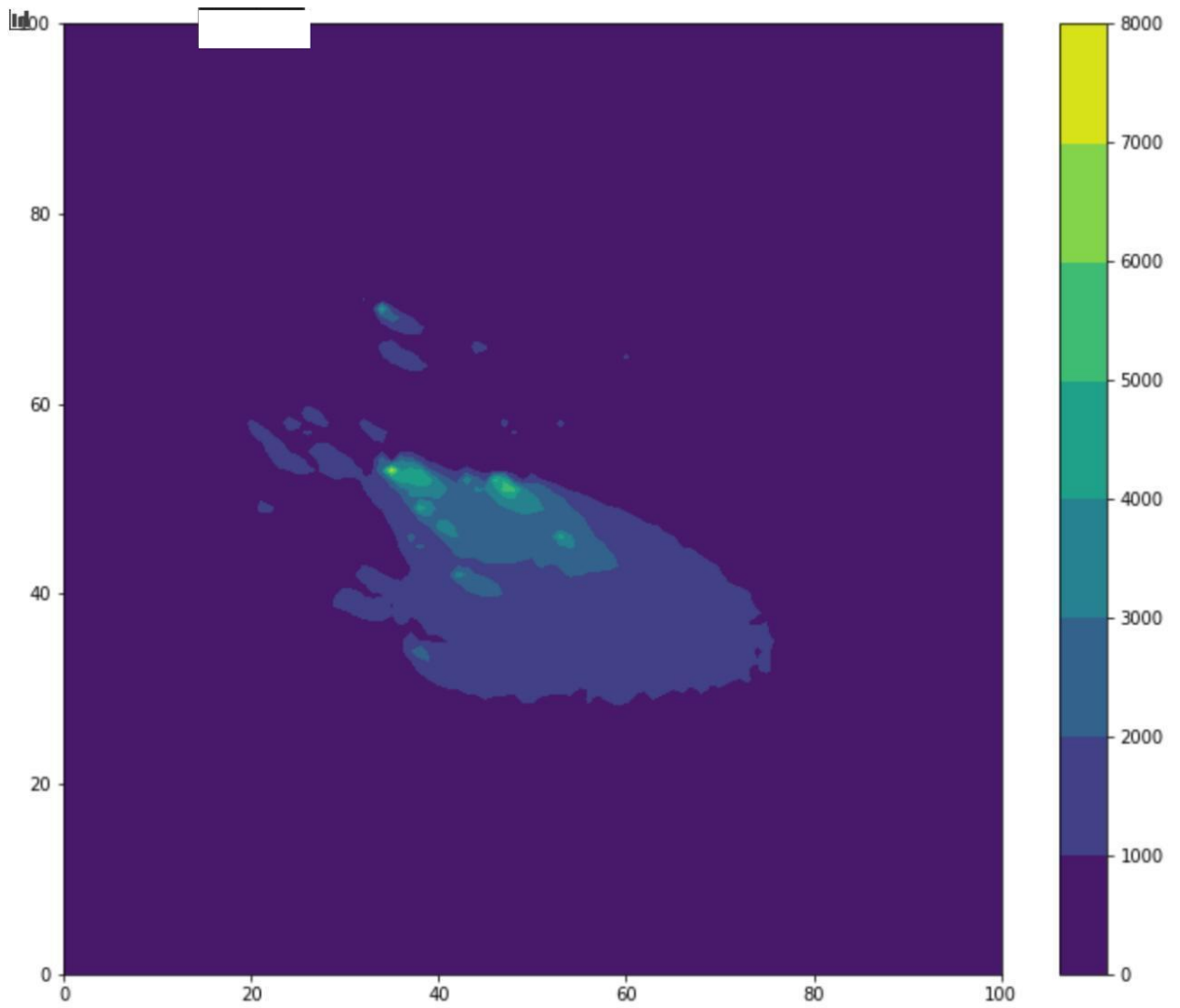


Рис.1. Сводные результаты тестовых расчетов по модели рассеивания примесей от группы источников при использовании функционала разработанной библиотеки.