flex_extract Documentation

Release 7.1 alpha

Anne Philipp and Leopold Haimberger

Mar 07, 2019

TABLE OF CONTENTS:

flex_extract is a software to retrieve meteorological fields from the European Centre for Medium-Range Weather Forecasts (ECMWF) Mars archive to server as input files for the FLEXTRA/FLEXPART Atmospheric Transport Modelling system.

All third-party software and libraries required by flex_extract are open source and free of charge.

Note: ![][/documentation/Sphinx/source/_static/cc-by-40.png] This work is licensed under the Creative Commons Attribution 4.0 International License. To view a copy of this license, visit http://creativecommons.org/licenses/by/4.0/ or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.

CHAPTER

ONE

DOWNLOADS

Describe download options of flex_extract ....

UNDER CONSTRUCTION

1.1 Download tar-balls

UNDER CONSTRUCTION

1.2 Git Repository

UNDER CONSTRUCTION

1.3 History of Changes

UNDER CONSTRUCTION

CHAPTER

TWO

INSTALLATION

UNDER CONSTRUCTION

2.1 Requirements

UNDER CONSTRUCTION

2.2 Operating modes

UNDER CONSTRUCTION

2.3 Installation test

UNDER CONSTRUCTION

CHAPTER

THREE

PROGRAM STRUCTURE

Describe structure of flex_extract ....

UNDER CONSTRUCTION

3.1 Program Overview

UNDER CONSTRUCTION

3.2 Program Components

UNDER CONSTRUCTION

3.3 Program Flow

UNDER CONSTRUCTION

CHAPTER

FOUR

USER GUIDE

UNDER CONSTRUCTION

4.1 How to use flex_extract

UNDER CONSTRUCTION

4.2 CONTROL-file templates

UNDER CONSTRUCTION

CHAPTER

FIVE

AUTO GENERATED DOCUMENTATION

5.1 Porgrams

5.1.1 install

5.1.2 submit

5.2 Classes

5.2.1 ControlFile

5.2.2 EcFlexpart

5.2.3 GribUtil

5.2.4 MarsRetrieval

class MarsRetrieval.**MarsRetrieval**(*server*, *public*, *marsclass='EA'*, *dataset=''*, *type=''*, *levtype=''*, *levelist=''*, *repres=''*, *date=''*, *resol=''*, *stream=''*, *area=''*, *time=''*, *step=''*, *expver='1'*, *number=''*, *accuracy=''*, *grid=''*, *gaussian=''*, *target=''*, *param=''*)

Specific syntax and content for submission of MARS retrievals.

A MARS revtrieval has a specific syntax with a selection of keywords and their corresponding values. This class provides the necessary functions by displaying the selected parameters and their values and the actual retrievement of the data through a mars request or a Python web api interface. The initialization already expects all the keyword values.

A description of MARS keywords/arguments and examples of their values can be found here: https://software.ecmwf.int/wiki/display/UDOC/ Identification+keywords#Identificationkeywords-class

server
: This is the connection to the ECMWF data servers.

  Type ECMWFService or ECMWFDataServer

public
: Decides which Web API Server version is used.

  Type int

marsclass
: Characterisation of dataset.

  Type str, optional

dataset
: For public datasets there is the specific naming and parameter dataset which has to be used to characterize the type of data.

  Type str, optional

type
: Determines the type of fields to be retrieved.

  Type str, optional

levtype
: Denotes type of level.

Type str, optional

levelist
Specifies the required levels.

Type str, optional

repres
Selects the representation of the archived data.

Type str, optional

date
Specifies the Analysis date, the Forecast base date or Observations date.

Type str, optional

resol
Specifies the desired triangular truncation of retrieved data, before carrying out any other selected post-processing.

Type str, optional

stream
Identifies the forecasting system used to generate the data.

Type str, optional

area
Specifies the desired sub-area of data to be extracted.

Type str, optional

time
Specifies the time of the data in hours and minutes.

Type str, optional

step
Specifies the forecast time step from forecast base time.

Type str, optional

expver
The version of the dataset.

Type str, optional

number
Selects the member in ensemble forecast run.

Type str, optional

accuracy
Specifies the number of bits per value to be used in the generated GRIB coded fields.

Type str, optional

grid
Specifies the output grid which can be either a Gaussian grid or a Latitude/Longitude grid.

Type str, optional

gaussian
This parameter is deprecated and should no longer be used. Specifies the desired type of Gaussian grid for the output.

Type str, optional

target
Specifies a file into which data is to be written after retrieval or manipulation.

Type str, optional

param
Specifies the meteorological parameter.

Type str, optional

data_retrieve()
Submits a MARS retrieval. Depending on the existence of ECMWF Web-API or CDS API it is submitted via Python or a subprocess in the Shell. The parameter for the mars retrieval are taken from the defined class attributes.

display_info()
Prints all class attributes and their values to the standard output.

print_infodata_csv(*inputdir*, *request_number*)
Write all request parameter in alpabetical order into a "csv" file.

Parameters

- **inputdir** (*str*) – The path where all data from the retrievals are stored.
- **request_number** (*int*) – Number of mars requests for flux and non-flux data.

5.2.5 UioFiles

5.3 Modules

5.3.1 get_mars_data

5.3.2 prepare_flexpart

5.3.3 tools

5.3.4 disaggregation

Disaggregation of deaccumulated flux data from an ECMWF model FG field.

Initially the flux data to be concerned are:

- large-scale precipitation
- convective precipitation
- surface sensible heat flux
- surface solar radiation
- u stress
- v stress

Different versions of disaggregation is provided for rainfall data (darain, modified linear) and the surface fluxes and stress data (dapoly, cubic polynomial).

disaggregation.**IA3**(*g*)

Interpolation with a non-negative geometric mean based algorithm.

The original grid is reconstructed by adding two sampling points in each data series interval. This subgrid is used to keep all information during the interpolation within the associated interval. Additionally, an advanced monotonicity filter is applied to improve the monotonicity properties of the series.

Note: (C) Copyright 2017-2019 Sabine Hittmeir, Anne Philipp, Petra Seibert

This work is licensed under the Creative Commons Attribution 4.0 International License. To view a copy of this license, visit http://creativecommons.org/licenses/by/4.0/ or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.

Parameters **g** (*list of float*) – Complete data series that will be interpolated having the dimension of the original raw series.

Returns **f** – The interpolated data series with additional subgrid points. Its dimension is equal to the length of the input data series times three.

Return type list of float

References

For more information see article: Hittmeir, S.; Philipp, A.; Seibert, P. (2017): A conservative interpolation scheme for extensive quantities with application to the Lagrangian particle dispersion model FLEXPART., Geoscientific Model Development

disaggregation.**dapoly**(*alist*)

Cubic polynomial interpolation of deaccumulated fluxes.

Interpolation of deaccumulated fluxes of an ECMWF model FG field using a cubic polynomial solution which conserves the integrals of the fluxes within each timespan. Disaggregation is done for 4 accumluated timespans which generates a new, disaggregated value which is output at the central point of the 4 accumulation timespans. This new point is used for linear interpolation of the complete timeseries afterwards.

Parameters **alist** (*list of array of float*) – List of 4 timespans as 2-dimensional, horizontal fields. E.g. [[array_t1], [array_t2], [array_t3], [array_t4]]

Returns **nfield** – Interpolated flux at central point of accumulation timespan.

Return type array of float

Note:

March 2000 [P. JAMES] Original author

June 2003 [A. BECK] Adaptations

November 2015 [Leopold Haimberger (University of Vienna)] Migration from Fortran to Python

---

disaggregation.**darain**(*alist*)

Linear interpolation of deaccumulated fluxes.

Interpolation of deaccumulated fluxes of an ECMWF model FG rainfall field using a modified linear solution which conserves the integrals of the fluxes within each timespan. Disaggregation is done for 4 accumluated timespans which generates a new, disaggregated value which is output at the central point of the 4 accumulation timespans. This new point is used for linear interpolation of the complete timeseries afterwards.

Parameters **alist** (*list of array of float*) – List of 4 timespans as 2-dimensional, horizontal fields. E.g. [[array_t1], [array_t2], [array_t3], [array_t4]]

Returns **nfield** – Interpolated flux at central point of accumulation timespan.

Return type array of float

Note:

March 2000 [P. JAMES] Original author

June 2003 [A. BECK] Adaptations

November 2015 [Leopold Haimberger (University of Vienna)] Migration from Fortran to Python

---

CHAPTER

SIX

SUPPORT

UNDER CONSTRUCTION

6.1 Ticket System

UNDER CONSTRUCTION

6.2 Mailing Lists

UNDER CONSTRUCTION

6.3 Known Bugs and Issues

UNDER CONSTRUCTION

6.4 FAQ - Frequently asked questions

UNDER CONSTRUCTION

CHAPTER

SEVEN

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

d

m

INDEX