Pargopy documentation

version

ROULLET Guillaume

May 28, 2018

Contents

Contents:

pargopy

atlas_generation module

Created on Thu May 24 07:11:16 2018

@author: therry

Module contenant la série d'outils pour les tâches maitre/esclaves telles que :

- **Génération d'atlas**

  -> Interpolation à l'aide des outils d'interpolation (interpolation_tools.py) -> Génération des statistiques (zmean, zstd, zdz, …) -> Propagation inverse des flag après traitement des données Argo -> Collage des "dalles" de l'atlas

Ce module contient également les outils permettant la génération d'atlas sur un domaine restreint :

- Génération sous-domaine d'atlas avec 4 points du globe (longitude, latitude)
- Génération sous-domaine dans une zone géographique définie (Mer Méditerranée, Pacifique-Sud, …)

`atlas_generation.`**`compute_stats_at_zref`**(`atlas_infos`, `itile=None`, `coord=None`)

Parameters:

- **atlas_infos** – Dictionnaire contenant les champs suivants : (mode, date, reso, timeflag, typestat) permettant de connaitre les informations sur les statistiques à calculer
- **itile** – Numéro de la dalle sur laquelle on calcule les statistiques
- **coord** – Dictionnaire contenant les champs suivants : ((lon, lat), deltalon, deltalat) définissant la zone géographique sur laquelle on souhaite calculer les statistiques

Calcul des statistiques sur un domaine défini. Le type de statistique calculé dépend du dictionnaire atlas_info passé en argument qui délivre toute l'information à la génération de celles-ci. Si itile et coord ne sont pas renseignés, la fonction retournera un message d'avertissement et produira un résultat sur une dalle géographique prédéfinie. La valeur de retour de cette fonction est un dictionnaire contenant les variables statistiques calculées.

Return type: dict

`atlas_generation.`**`interpolation_on_tiles`**(`itile`)

Parameters: **itile** – Numéro de la dalle sur laquelle on réalise l'interpolation

Interpolation des profiles ARGO appartenant à la dalle (itile) sur les niveaux de référence zref.
Les résultats de l'interpolation sont sauvés dans un fichier pickle.

Return type: DataFrame

atlas_tools module

Created on Thu May 24 08:46:51 2018

@author: therry

Module contenant la série d'outils utilisée pour l'analyse manuelle des atlas.

general_tools module

Created on Thu May 24 09:50:19 2018

@author: therry

Module contenant les outils utilisés dans la plupart des modules:

- Lecture/Ecriture au sein de fichiers pickle

- Outil de navigation au sein d'une DataFrame
- …

general_tools.**conversion_gregd_juld**(year, month, day)

Parameters: **date** – Dictionnaire (année, mois, jour) contenant une date en calendrier grégorien

Fonction convertissant une date du calendrier grégorien en un julian day

Return type: float

general_tools.**conversion_juld_gregd**(juld)

Parameters: **juld** – Date en calendrier julien à convertir

Fonction convertissant un julian day en gregorian day

Return type: list of int

general_tools.**get_profile_file_path**(dac, wmo)

Parameters:
- **dac** – DAC du profil recherché
- **wmo** – WMO du profil recherché

Fonction utilisée pour générer le chemin pour accéder au profil appartenant à la dac et au wmo donné en argument

Return type: string

general_tools.**get_tag**(kdac, wmo, kprof)

Parameters:
- **kdac** – Index of the dac (aoml = 1, bodc = 2, coriolis = 3, …)
- **wmo** – WMO number
- **kprof** – Index of the profile

Compute the tag number of a profile
The inverse of get_tag() is retrieve_infos_from_tag()

Return type: int

general_tools.**ij2tile**(i, j)

Parameters:
- **i** – Longitude du point étudié
- **j** – Latitude du point étudié

Fonction retournant le numéro de la dalle à laquelle appartient un point dont les coordonnées sont i et j (longitude, latitude) TO DO: Changer le 20 par une valeur (nlat) pour rendre le code plus flexible

Return type: int

general_tools.**retrieve_infos_from_tag**(tag)

Parameters: **tag** – tag on which you are looking for informations

Retrieve idac, wmo and iprof from tag (array of int)
It is the inverse of get_tag()

Return type: dic

general_tools.**tile_definition**()

Define the tiles coordinates, in the form of a vector of lon and lat + their margins
The tile indexing is
|—–+—–+—–+—–+—–| | 280 | 281 | 282 | … | 299 | |—–+—–+—–+—–+—–| | … | … | … | … | … |
|—–+—–+—–+—–+—–| | 20 | 21 | 22 | … | 39 | |—–+—–+—–+—–+—–| | 0 | 1 | 2 | … | 19 |
|—–+—–+—–+—–+—–|

The dictionnary returned is :

-> lat (vector 1D) -> lon (vector 1D) -> nlat (Quantity of latitude in the tile) -> nlon (Quantity of longitude in the tile) -> marginlat (vector 1D) -> marginlon (int)

Return type: dict

interpolation_tools module

Created on Fri May 25 07:32:54 2018

@author: therry

Module contenant la série d'outils utilisée pour l'interpolation des profils ARGO :

- Transformation des variables in-situ en variables TEOS-10
- Interpolation sur les niveaux zref des variables in-situ et TEOS-10

netCDF_form module

Created on Thu May 24 08:49:04 2018

@author: therry

Module contenant la série d'outil permettant la gestion d'un fichier netCDF :

- Création du fichier, des variables, des dimensions, des attributs
- Ecriture au sein du fichier de ces variables
- Lecture du fichier

`netCDF_form.`**`create_dim`**(`filename`, `zref`, `nlat`, `nlon`, `atlas_infos`)

Parameters:

- **filename** – Nom du fichier netCDF
- **zref** – Paliers de référence pour la profondeur
- **nlon** – Nombre de longitudes pour cette dalle
- **nlat** – Nombre de latitudes pour cette dalle
- **atlas_infos** – Dictionnaire contenant les champs suivants : (mode, date, reso, timeflag, typestat) permettant de connaitre les informations sur les statistiques à calculer (on ne prendra ici que mode et date)

Fonction utilisée pour créer les dimensions et les attributs d'un fichier netCDF :rtype: None

`netCDF_form.`**`create_var`**(`filename`, `var_list`)

Parameters:

- **filename** – Nom du fichier netCDF
- **var_list** – Liste des variables à générer

Create the netcdf file 'filename' for the list of variables 'var_list' the dimensions '(zref, lat, lon)' and the attributes of each variable is retrieved from the json file.

Return type: None

`netCDF_form.`**`read_var`**(`filename`, `var_list`)

Parameters:

- **filename** – Nom du fichier netCDF
- **var_list** – Liste des variables à générer

Read the list of variables 'var_list' from netcdf file 'filename' Return the result as a dictionnary of numpy arrays

Return type: dict

`netCDF_form.`**`write_var`**(`filename`, `var_list`, `var_dic`)

Parameters:

- **filename** – Nom du fichier netCDF
- **var_list** – Liste des variables à générer
- **var_dic** – Dictionnaire contenant les variables à écrire

Write the list of variables ‘var_list’ in the netcdf file ‘filename’. Variables are transfered in the form of a dictionnary ‘var_dic’, where each entry is a numpy array.

Return type: None

param module

Created on Thu May 24 08:04:37 2018

@author: therry

Module contenant tout les paramètres utiles au fonctionnement du programme :

- **Chemins vers les différents répertoires utilisés**

  -> Vers la base de données ARGO -> Vers le répertoire du projet -> Vers les fichiers de statistique -> Vers les fichiers d’atlas -> …

- **Variables définissant la/les tâche(s) à réaliser**

  -> Type de statistique (zmean, zstd, zdz, …) -> Domaine d’étude (atlas global, dalles, coordonnées, régions géographiques) -> Précision de la grille (reso_deg) -> Date du snapchot choisi pour étudier ARGO -> Statut des données que l’on souhaite traiter (R, A, D) -> Filtre (saisonnier, mensuel, …) sur les données traitées

- **Variables utilisées par tout les modules**

  -> zref -> filename -> …

`param.`**`atlas_filename`** (`diratlas`, `atlas_infos`)

Parameters:

- **diratlas** – Chemin vers le répertoire contenant les atlas
- **atlas_infos** – Dictionnaire contenant les champs suivants : (mode, date, reso, timeflag, typestat) permettant de connaitre les informations sur les statistiques à calculer

Fonction permettant de générer le nom d’un fichier atlas en fonction de ses paramètres contenus dans atlas_infos.

Return type: String

task_dispatcher module

Created on Thu May 24 09:39:58 2018

@author: therry

Module contenant les fonctions utilisées pour distribuer les tâches entre les différents esclaves

`task_dispatcher.`**`getavailableslave`** (`slavestate`)

Parameters: **slavestate** – Give the state of the slave

Return the index of a slave that is awaiting a task. A busy slave has a state == 0. If all slaves are busy then wait until a msg is received, the msg is sent upon task completion by a slave. Then determin who sent the msg. The msg is collected in the answer array. By scanning it, we determine who sent the message.

Return type: int

`task_dispatcher.`**`master_work_nonblocking`** (`nslaves`)

Parameters: **nslaves** – Number of slavbes under master control

Master basically supervises things but does no work

Return type: None

`task_dispatcher.`**`ordering_tasks`** (`tasks`)

Parameters: **tasks** – List containing the tasks to do

Sort the tasks according to their workload workload is proportional to size of the tile file

Return type: list of int

task_dispatcher.**slave_work_nonblocking**(islave)

Parameters: **islave** – Number given to the slave

Slaves enter an infinite loop: keep receiving messages from the master until reception of ‘done’. Each messages describes the task to be done. When a new task is received slave treats it. At the end of it the slave sends a message to the master saying that he is over, and that he is available for a new task.

Return type: None

tile module

Created on Thu May 24 09:37:57 2018

@author: therry

Module contenant la série d’outils utilisé pour le découpage de l’atlas en dalles :

- Découpage en 300 dalles en fonction des longitudes/latitudes
-

database module

Created on Mon May 28 12:39:10 2018

@author: therry

Tools to generate and maintains the processed database

they are high-level routines that rely on smaller modules

database.**synchronize_headers**()

Propagate the header of each tile onto the global header

Indices and tables

Index

Python Module Index