Scint\_UM100 Rough Documentation

# Grid\_coords

## SA\_grid\_overlap

### SA\_grid\_overlap.py

Creates the file SA\_UM100\_grid\_percentages.csv using the polygones created by create\_grid\_polygons

### SA\_UM100\_grid\_percentages.csv

CSV file containing the observation SA weighting % in the UM100 gridboxes for each timestep

### Plot\_SA\_grid\_selection.py

Plots observation SA data and UM100 grid box polygons

A graph showing a heat wave

Description automatically generated with medium confidence

A blue and red dot on a grid

Description automatically generated

## Grid\_coord\_lookup

### Get\_example\_points.py

For each model resolution, get a set of example coordinates (9 sets) of grid-box locations around a given point (atm, this is set to BTT’s coord) and write these as a csv file

### Rotation\_tests.csv

CSV file of example coords from get\_example\_points for a roughly chosen point around IMU. This is currently not used.

### Rotation\_bests\_BTT.csv

CSV file of example coords from get\_example\_points for the site BTT.

### Grid\_polygons

#### UM100\_shapes

Folder for all 25600 UM100 grid polygons

##### UM100\_shapes\_reduced

Folder for a reduced number of UM100 grid polygons (only 100)

#### Create\_grid\_polygons.py

Does a repeat of the steps in create\_grid\_coord.csv but saves a polygon instead ??????

### Create\_grid\_coords\_csv.py

* Reads in the example points in Rotation\_tests\_BTT
* Expands the grid by a user-given number of steps (real run = 80 steps to left, right, up and down – so 160 steps x 100 m = 16,000 m grid)
* Calculates the midpoint of the gridboxes
* Combines all into a df
* Saves the df to a csv file: grid\_coords.csv

### Grid\_coords.csv

CSV file containing the 5 coords (BL, TL, TR, BR, MID) of each UM100 gridbox.

## Plot\_SA\_on\_grid.py

Plot observation SA extent on top of model data:

A map of a river

Description automatically generated

# Data\_retreval

## Retrieve\_data\_funs.py

Functions called in Retrieve\_data\_from\_target\_grids

## Retrieve\_data\_from\_target\_grids.py

* Locates model file
* Reads model file
* Reads premade csv file SA\_UM100\_grid\_percentages from grid\_coords to get SA overlap on the UM100 grid
* Reads cords of these selected grids from premade csv file grid\_coords from gird\_coords
* Reads in cube into iris to get the index of the target grid coords after rotation
* Grabs the data from the selected grids and fills an array
* Makes plot of grid boundaries, grid data, and SA file for the target time

A screen shot of a screen

Description automatically generated

# Cloud\_content.py

Reads cloud\_volume\_fraction\_in\_atmosphere\_layer variable from all 3 resolutions and makes plots like:

A blue and yellow cloud

Description automatically generated

# Plot\_resolution\_maps\_of\_kdown.py

Reads surface\_downwelling\_shortwave\_flux\_in\_air variable from all 3 resolutions and makes plots like:

A blue and green gradients

Description automatically generated with medium confidence

# Plot\_resolution\_maps\_of\_QH.py

Reads surface\_upward\_sensible\_heat\_flux variable from all 3 resolutions and makes plots like:

A map of a river

Description automatically generated

# Pp\_to\_netcdf.py

Script which turns pp files files into netCDF. The file that is in use is a copy on the RACC

# Investigate\_stash.py

Test file to load in iris cubes of any variable

# Lewis\_scripts

## SEB.py

Lewis python scripts

## Tsurf.py

Lewis python scripts

# Plots

Where plots are saved

# Radiation\_line\_plots

## Line\_time\_series\_funs\_UM100.py

Makes time series plots of UKV data and obs with each hour in the day highlighted:

A graph of a graph

Description automatically generated with medium confidence

## Main\_line\_series\_UM100.py

Locates pre-grabbed UKV model data from scint\_plots and obs data from scint\_flux and runs line\_time\_series\_funs\_UM100.py

# SA\_134

Local copy of al the SAs for the first day