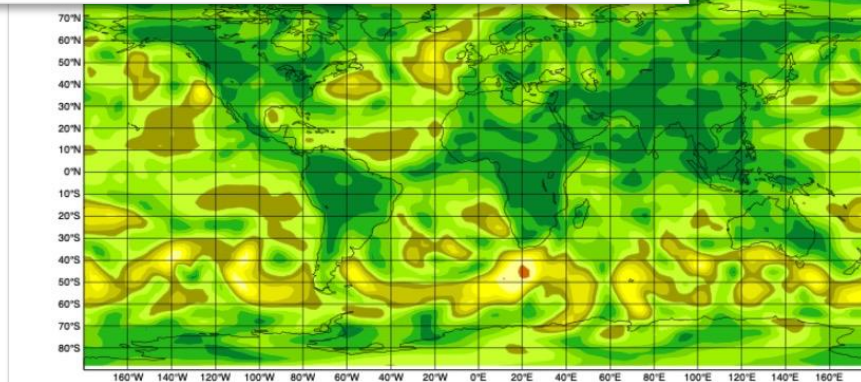
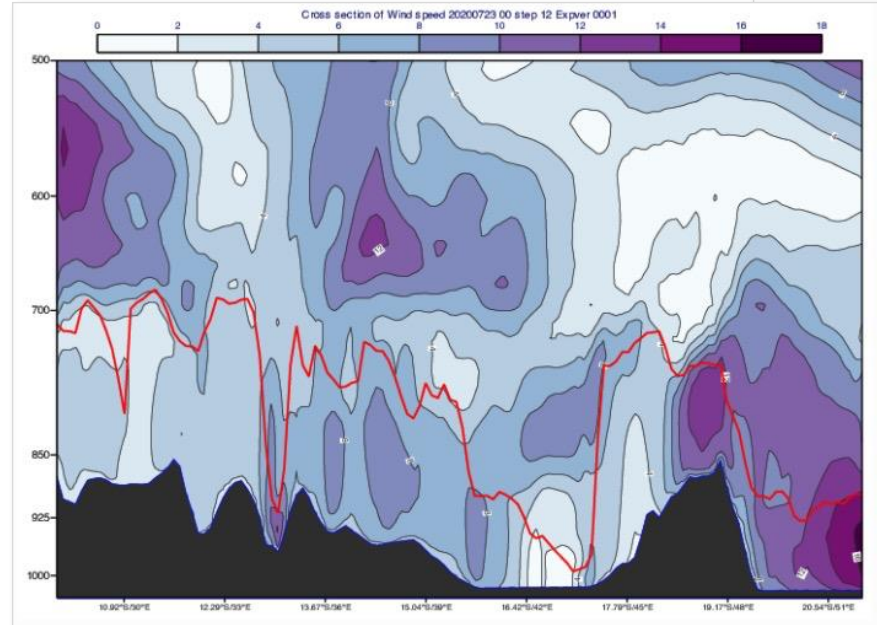


NWP : An Introduction to Metview for Data Analysis in Python

November 14, 2023

Iain Russell
Sándor Kertész

Development Section, ECMWF



Finding a range of extreme values

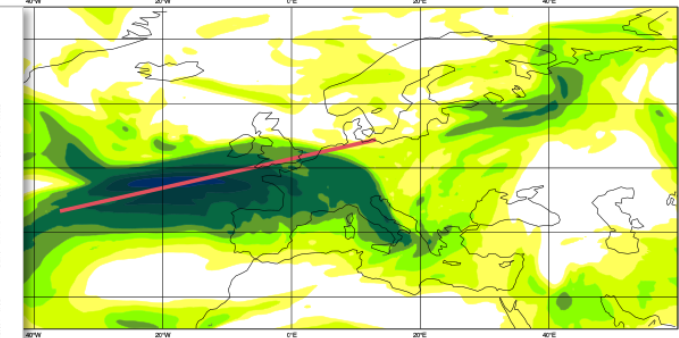
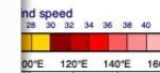
Locations where the value is within 95% of the maximum by supplying a range of values:

```
find(wg0, [max0*0.95, max0])
```

```
8.5, -5.0], [48.5, -4.5], [47.0, -7.0], [47.0, -3.5], [47.0, -3.0]]
```

To work with these points in Metview, the easiest way is to use the `gfind()` function to create a `Geopoints` variable:

```
max_points = mv.gfind(wg0, max0, max0*0.05)
print(len(max_points), 'points')
print('first point:')
print(max_points[0])
```



View along an area of interest

```
view(
    1000.0,
    100,
    line
```

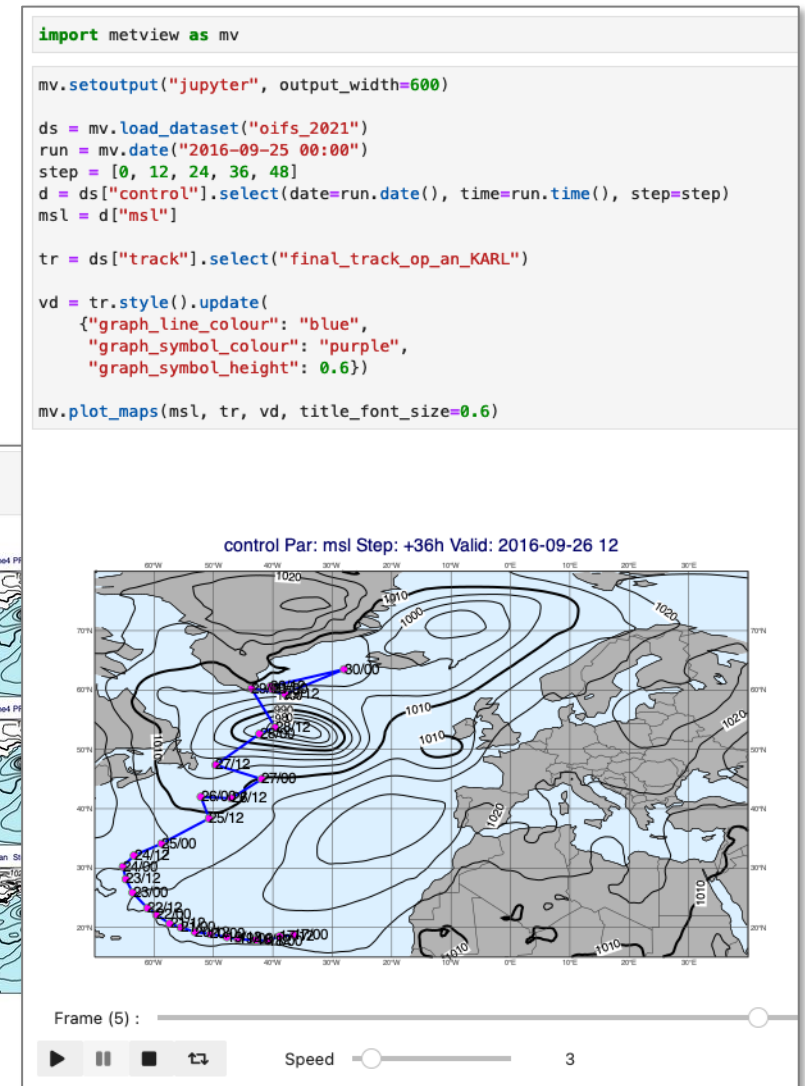
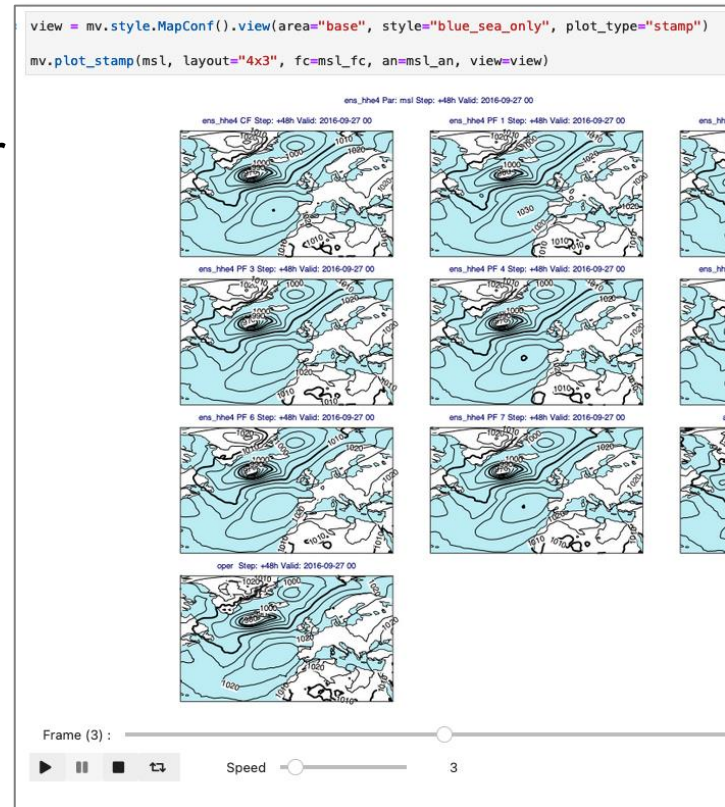
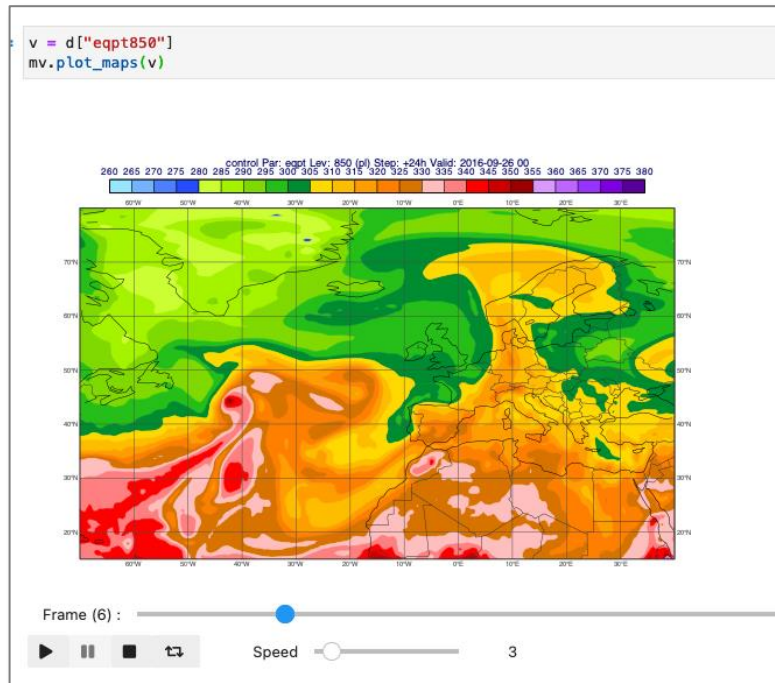
2 of the data with

= "on".

Frame (9) : Speed 3

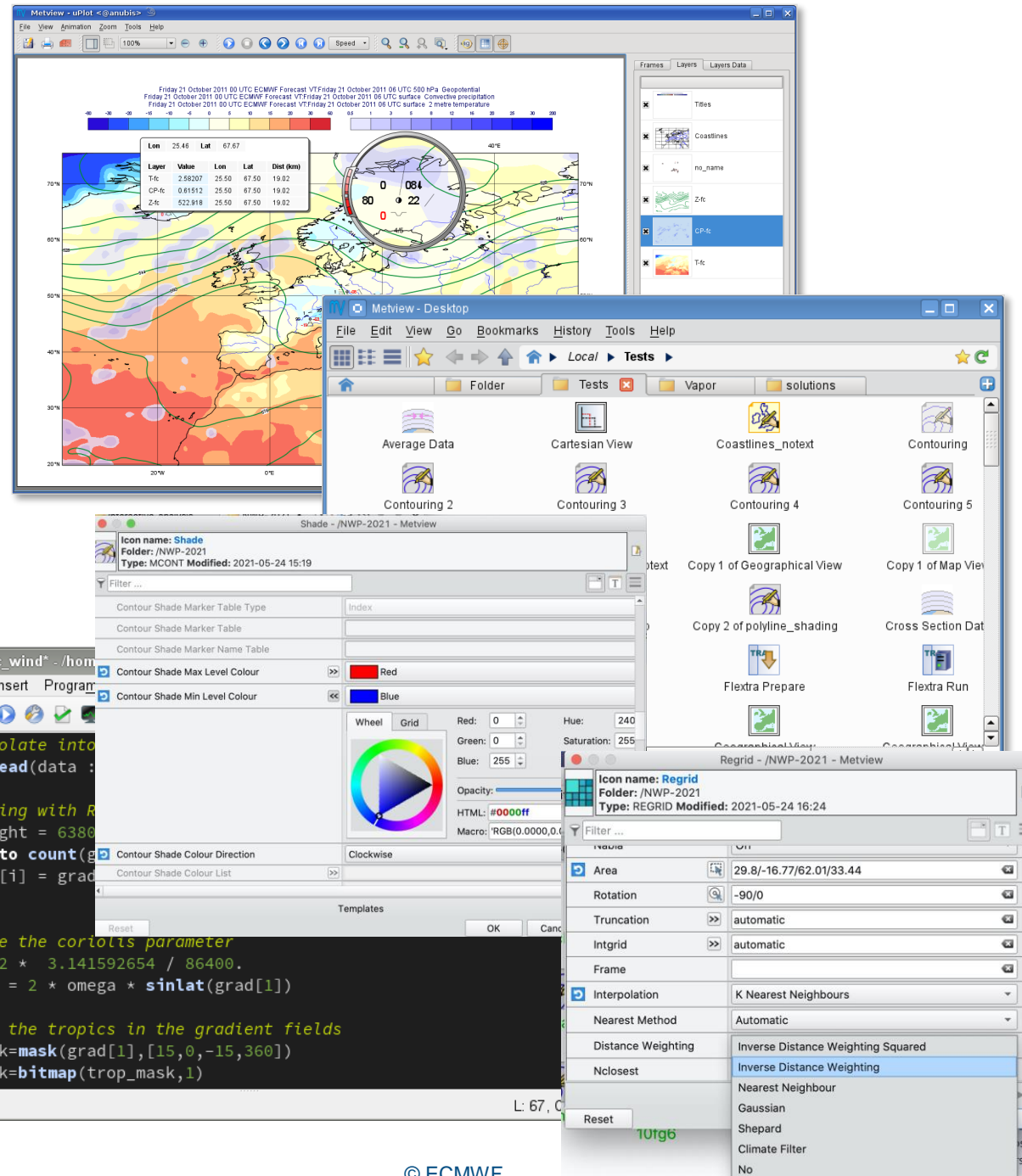
Outline

- What is Metview
- User interface
- Python interface
- How to obtain / install
- Practical exercise using Jupyter notebooks



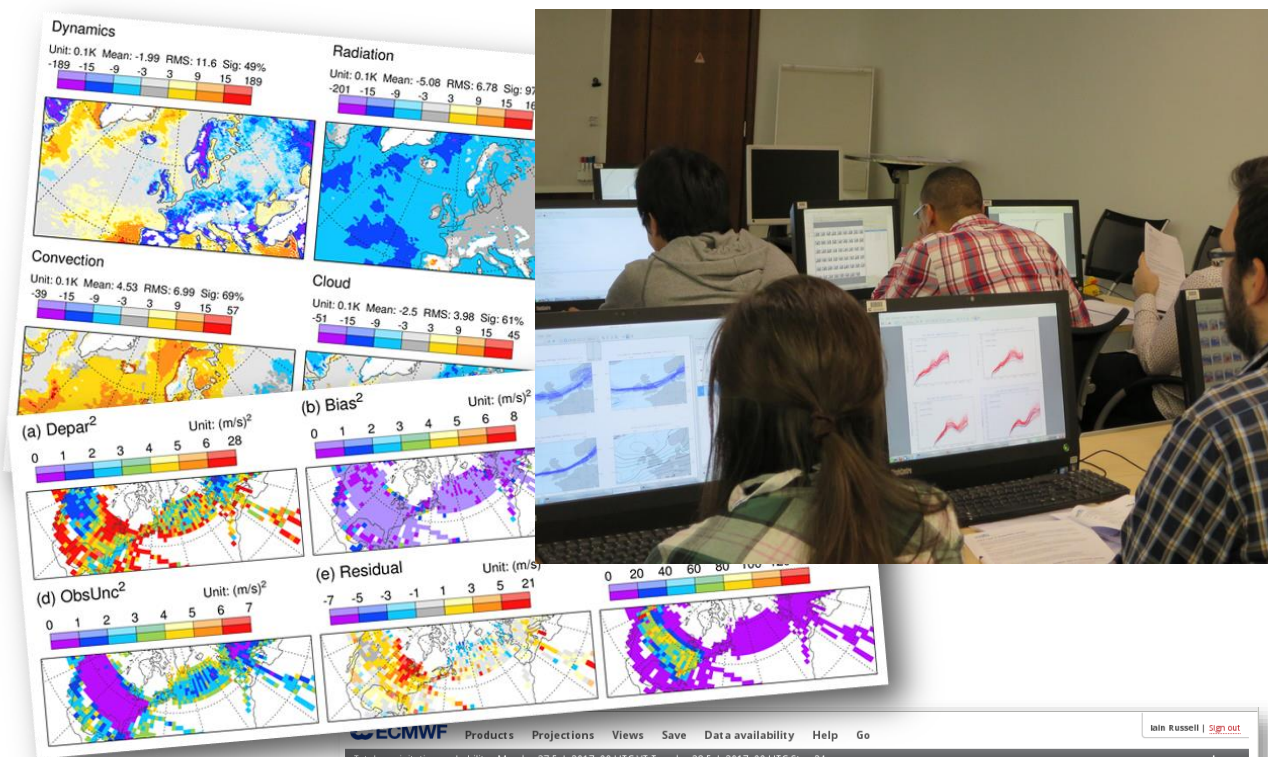
What is Metview?

- Workstation software, runs on UNIX, from laptops to supercomputers (including macOS)
- Developed at ECMWF, built on other ECMWF libraries
- Open source, Apache 2.0 license
- Data access
- Data processing
- Data visualisation
- Icon based user interface
- Powerful scripting languages - Macro and Python

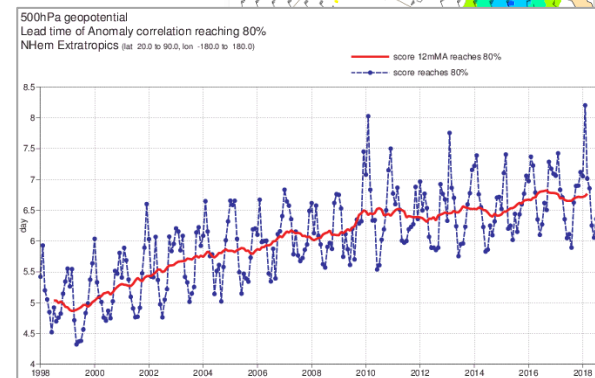


30 years of Metview so far

- Serving users of ECMWF data since 1993
- Used daily by many analysts and researchers
 - inside and outside ECMWF
 - also by commercial users of our data



- Some large developments, e.g. the Diagnostics Toolbox, OpenIFS workshops, Quaver (verification package) are based on top of Metview
- ecCharts is based on Metview's architecture and takes it onto the web



ECMWF Products Projections Views Save Data availability Help Go

Total precipitation probability - Monday 27 Feb 2017, 00 UTC VT Tuesday 28 Feb 2017, 00 UTC Step 24

ECMWF Products

Select from these ECMWF Products to add to your personal list

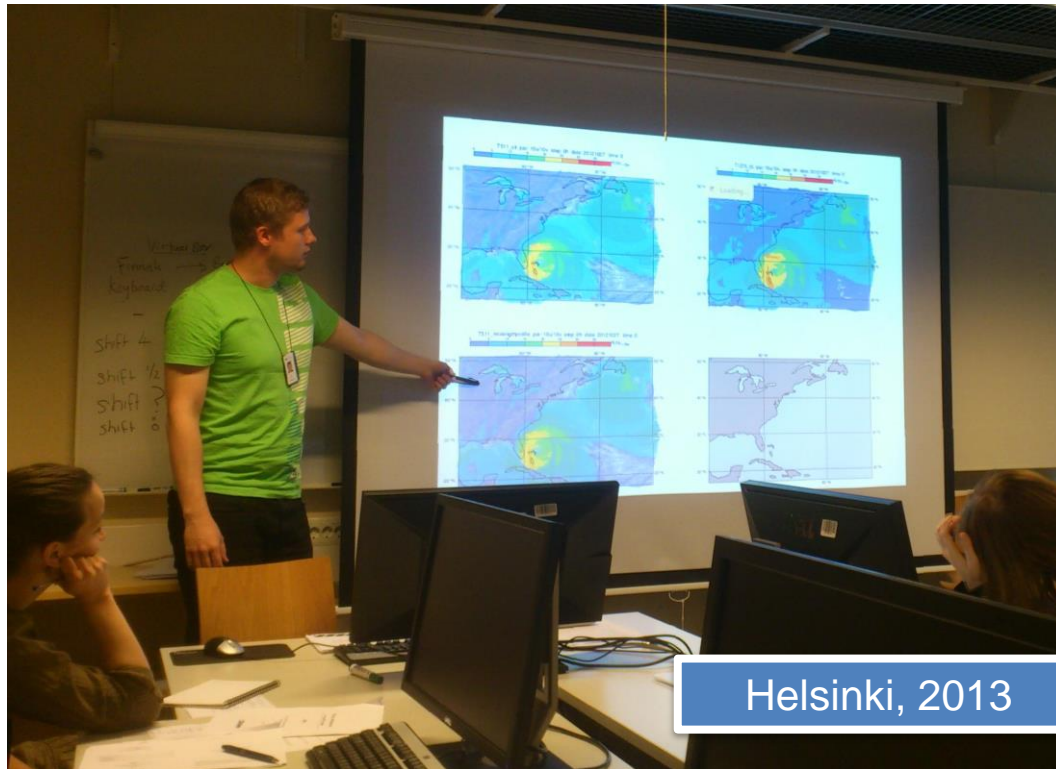
Filter: Show All

- Mean wave period probability
Mean wave period probability from perturbed forecast.
[Add to product list](#) [View in map](#)
- ENS event probabilities
- Tropical cyclone genesis
EXPERIMENTAL PRODUCT. Best viewed in global projection. Tropical cyclone (genesis) strike probabili...
[Add to product list](#) [View in map](#)
- ENS percentiles
- 10m wind percentile
10m wind for a percentile (value of 10m wind speed below which a certain percent of the ENS members fa...
[Add to product list](#) [View in map](#)
- Significant wave height percentile and mean wave direction
Significant wave height for a percentile (value of signi...
 Hide Products you already have

© ECMWF Chart updated (Network: 0.2s - Plot: 2.5s)

Metview and OpenIFS

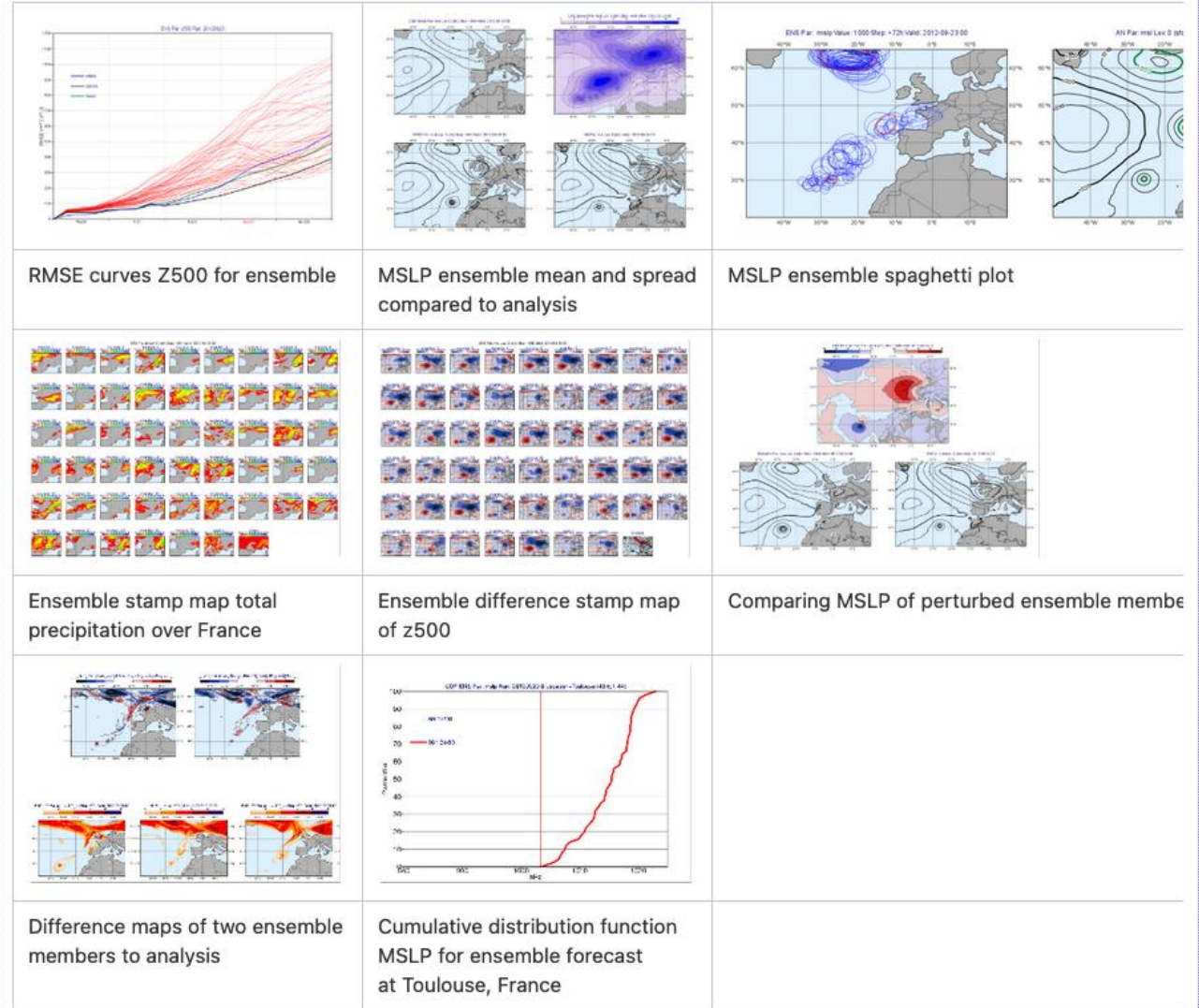
- In previous OpenIFS user workshops case studies were based on custom Metview Macro libraries; now we use Metview Python



Helsinki, 2013

Toulouse, 2016

Ensembles



User Interface

The screenshot displays the Metview user interface with several tool windows open:

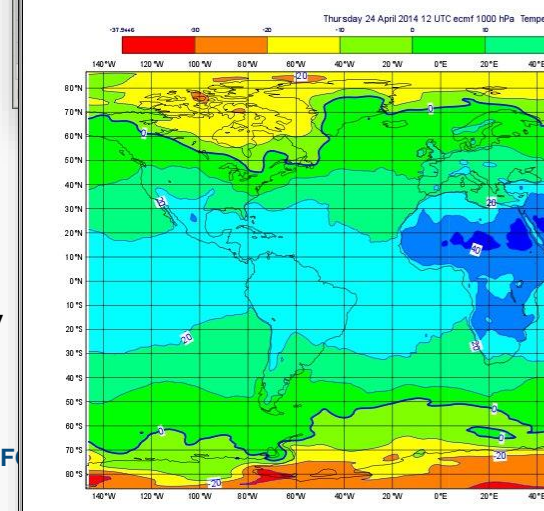
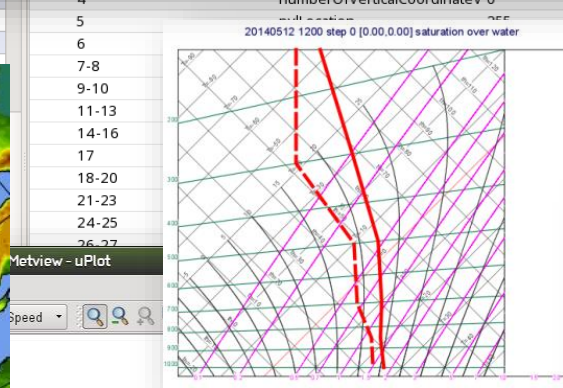
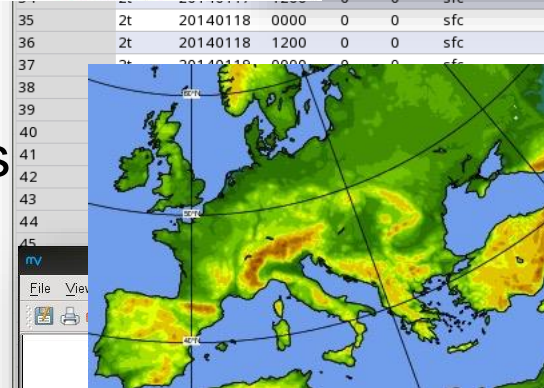
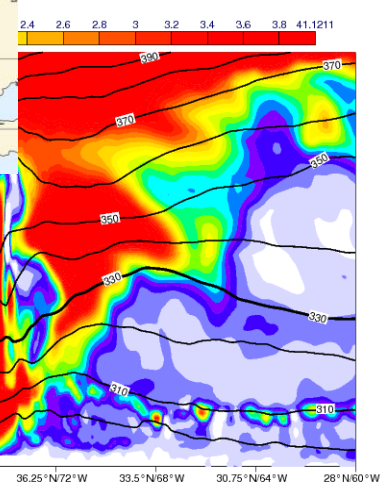
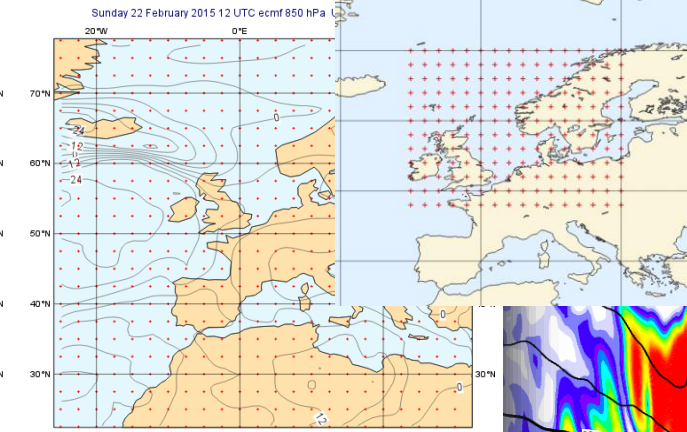
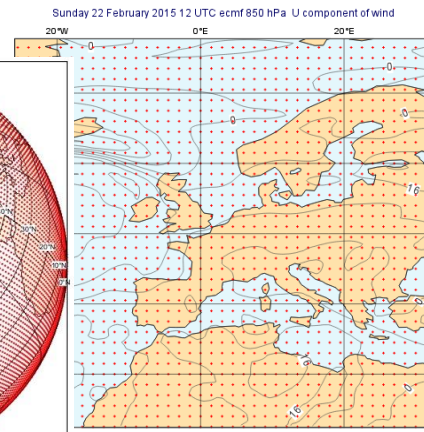
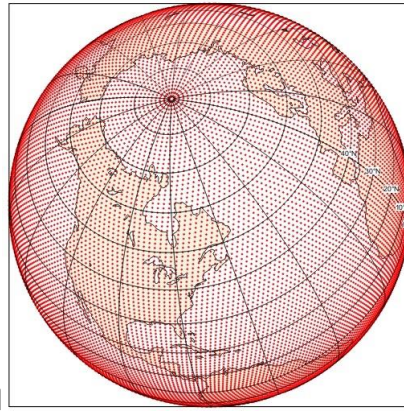
- polygons - /Tests/Macros/masking - Metview**: The main application window with a menu bar (File, View, Go, Bookmarks, History, Editors, Tools, Help) and a breadcrumb trail (Tests > Macros > masking > polygons).
- Bookmarks**: A sidebar showing a tree view of folders including UserTests, Tests, checks, uplot, and Defaults.
- Cross Section View - /Tests/Macros/masking/polygon-masks - Metview**: A window for configuring a cross-section view with fields for Bottom Level (1100.0), Top Level (0.01), Line (0/-180/0/180), and various wind parameters.
- Metview - Geography Tool**: A window showing a world map with a grid and a cross-section line. The SW/N/E coordinates are 39.82/-64.03/50.88/59.62.
- t_shade_K - /Tests/Macros/masking/polygon-masks - Metview**: A configuration window for the t_shade_K macro, showing settings for Contour Shade Method (Area Fill), Max Level Colour (Red), Min Level Colour (Blue), and Colour Direction (Clockwise).
- test-with-magics-shapefiles.py* - /home/graphics/cgi/metview/Tests/Macros/masking/polygon-n**: A code editor window showing Python code that imports the t_shade_K macro and configures its settings.

The desktop background features several icons: late-grib.grib, Mars Retrieval, ecCharts, Copy 1 of ecCharts, test.py, data.mv, gridvals_1x1, and t2m_hires_area.grib.

```
77
78 # Importing : /Tests/Macros/masking/polygon-masks/t_shade_K
79
80 t_shade_k = mv.mcont(
81     legend = "on",
82     contour_highlight = "off",
83     contour_level_selection_type = "level_list",
84     contour_level_list = [193,243,263,268,273,
85     contour_shade = "on",
86     contour_shade_method = "area_fill",
87     contour_shade_max_level_colour = "red",
88     contour_shade_min_level_colour = "blue",
89     contour_shade_colour_direction = "clockwise",
90     grib_scaling_of_retrieved_fields = "off"
91 )
```

Metview + GRIB

- Plot
- Examine
- Filter
- Spectral transform
- Regridding, cropping
- Missing values, masks
- Maths, Boolean
- Specialised:
 - Cross section
 - Thermodynamics
 - Gradient
 - Vertical integration
 - Model to pressure level
 - Etc



GRIB Filter - /tutorials/nwp-primer-2021 - Metview

Icon name: GRIB Filter
Folder: /tutorials/nwp-primer-2021
Type: READ Modified: 2021-05-21 16:10

Filter ...

Type	>> ANY
Model	ANY
Levtype	Any
Levelist	10p0/850
Param	>> T/UM/Q
Date	20210516
Verify	OFF
Refdate	OFF
Fcmonth	OFF

```
statistics - /home/graphics/cgi/metview/Tests/Macros/statistics
File Edit View Insert Program Settings

# retrieve some data
f1 = retrieve (date : -1, levels : 1000, grid : [1.5, 1.5])
f2 = retrieve (date : -2, levels : 1000, grid : [1.5, 1.5])

# perform some calculations for comparison
cv_f1f2 = covar_a (f1, f2)
ovar_a (f1, f1)
ovar_a (f2, f2)
var_a (f1)
var_a (f2)

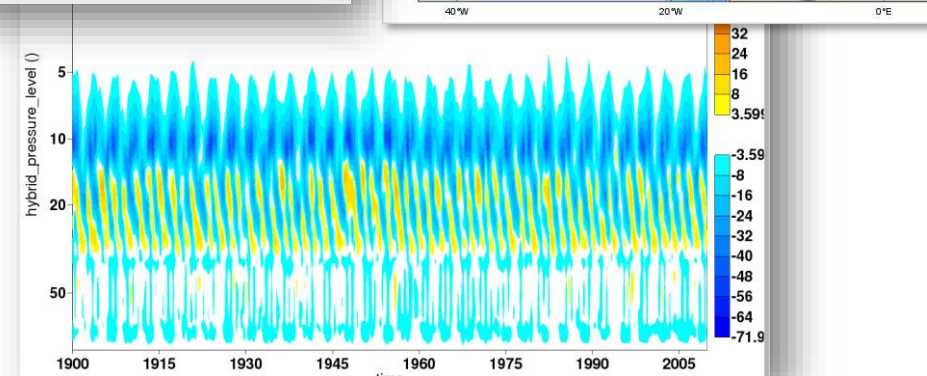
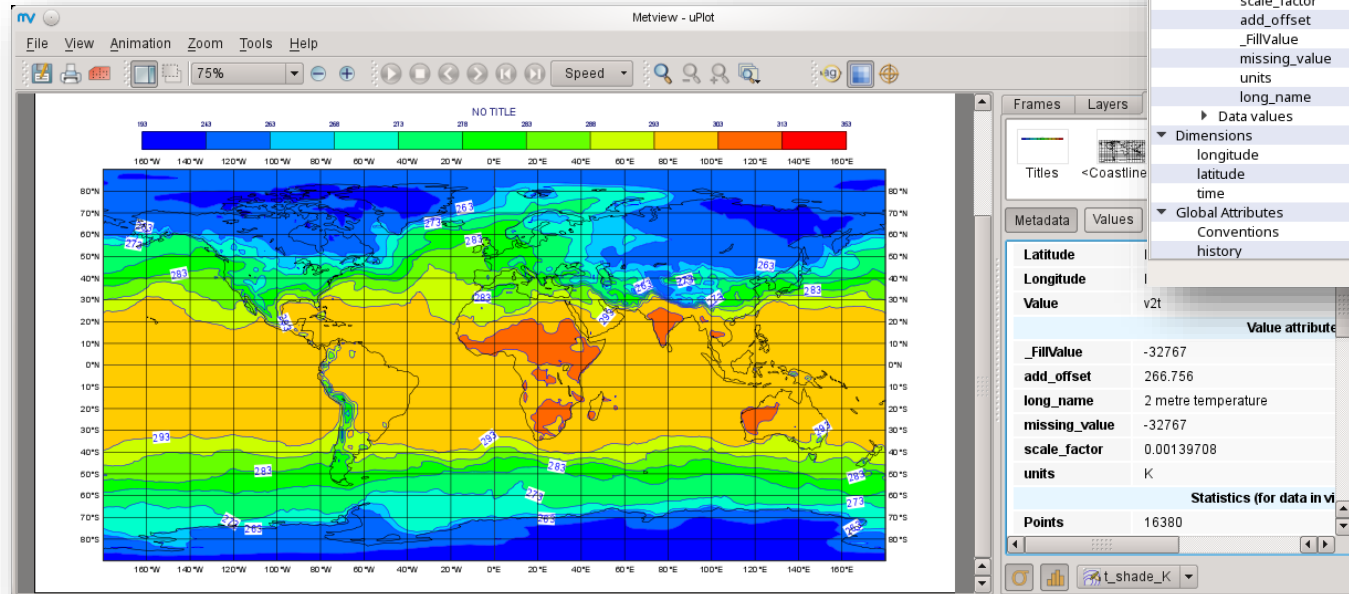
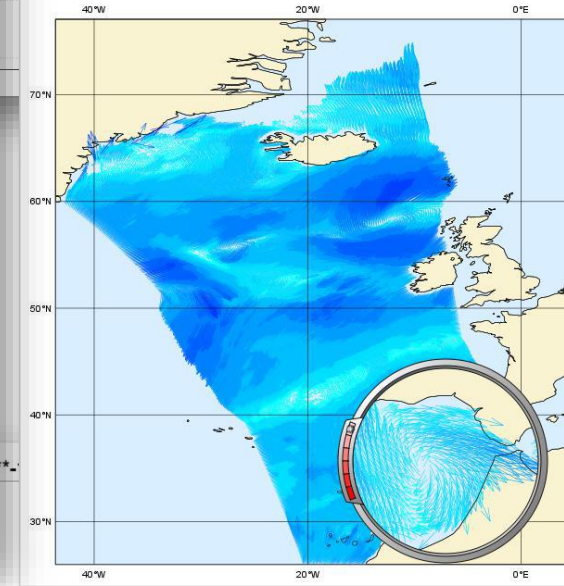
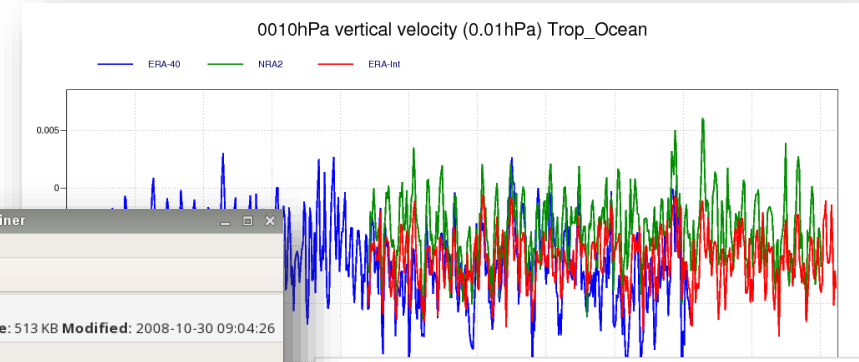
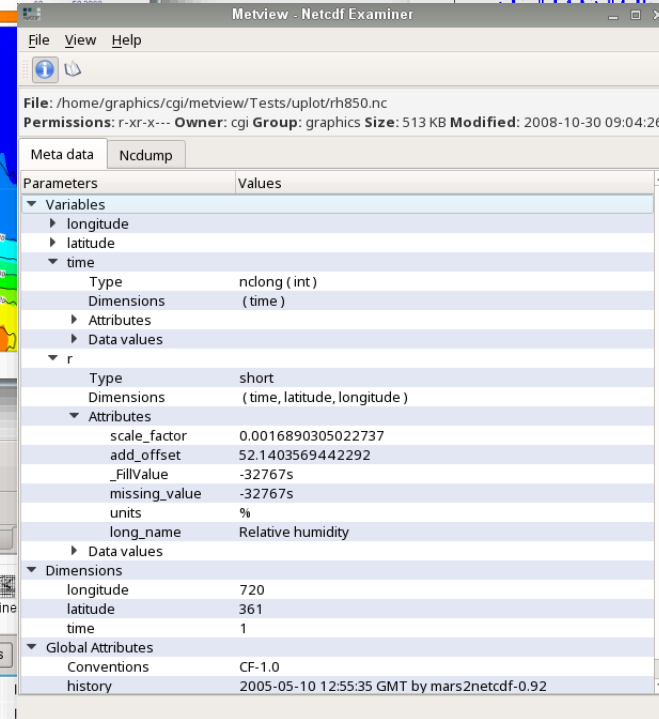
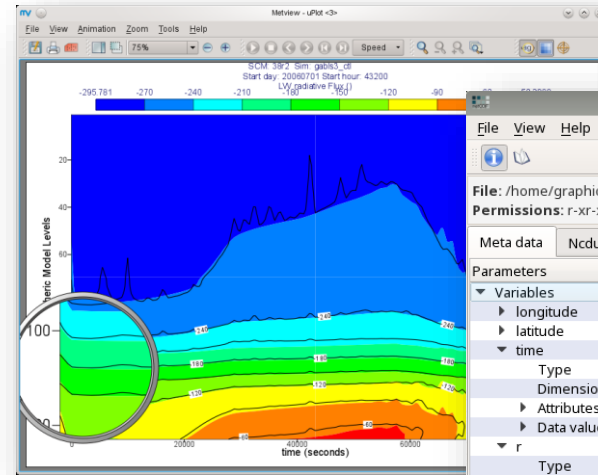
= cv_f1f2 / (sqrt(cv_f1f1) * sqrt(cv_f2f2))
2 = cv_f1f2 / (sqrt(var_f1) * sqrt(var_f2))
1 = corr_a (f1, f2)

RIEVE (MARS)
and f2 = 707195.562425
= 0.876684930973
2 = 0.876684930973
1 = 0.876684930973

OK! : 4.078 s [Finished at 14:05:55] | L: 14, C: 27
```

Metview + NetCDF

- Plot
- Examine
- Maths, Boolean



Metview + BUFR

- Plot
- Examine
- Filter
- Extract values
- Convert to Geopoints
- Thermodynamics

The image displays several screenshots from the Metview software interface:

- Top Left:** A window titled "synop.buf - Bufr Examiner (Metview)" showing a list of BUFR messages. The selected message (58) is from station 13 at 1200 UTC on 20150223. The "Data tree" on the right shows the structure of the selected message, with "3HourPressureChange" selected.
- Top Right:** A global map showing a color-coded pressure change field over the North Atlantic and Europe.
- Middle Left:** A thermodynamic chart for station 13 on 20150223 at 1200 UTC. The y-axis represents pressure in hPa (from 1000 to 200), and the x-axis represents temperature in °C (from -10 to 30). The chart shows a red line for the observed pressure change and various isotherms and mixing ratio lines.
- Bottom Left:** A detailed station plot for station 13, showing a circular plot of wind speed and direction, and a small plot of temperature and humidity.
- Bottom Right:** A global map showing station locations as colored dots. A pop-up window provides details for a specific station:

Layer	Value	Lon	Lat	Dist (km)
filter_o	301.800000	36.90	-1.32	118.92

Metview + Geopoints CSV & ASCII

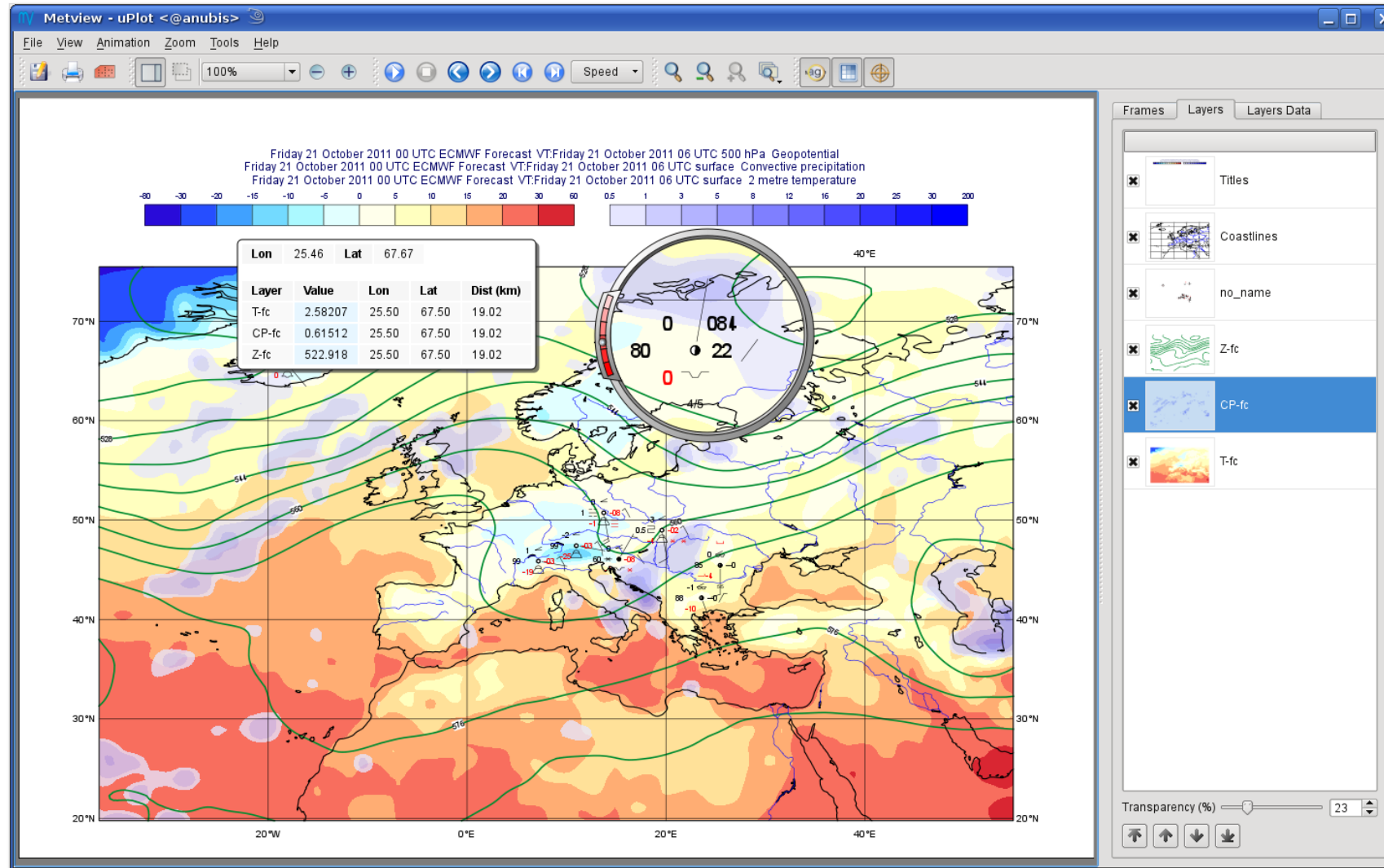
- Geopoints – geo-located values
- Plot
- Examine
- Filter
- Maths, Boolean
- Geo functions
- Convert between GRIB, BUFR and Geopoints
- Can also read CSV

CSV (and similar)

1	degrees_lat	degrees_lon	fg_depar
2	43.37942	-15.32557	7.7474003900818
3	37.25943	-22.93482	-1.9798161031505
4	30.94732	-42.5021	10.609738886033
5	43.94829	-76.45218	-3.3929916442406
6	28.94935	-89.62981	7.1024207535072
7	77233	-93.30165	4.5346224539512
8	10768	-119.4183	3.6048699999283
9	96311	-113.71029	-2.5591580715308
10	06225	-144.48569	2.0515008637495
11	38039	-179.80404	16.225524892237
12	70696	-174.86691	8.5849734979496
13	190788	-0.50169	5.1074401690909

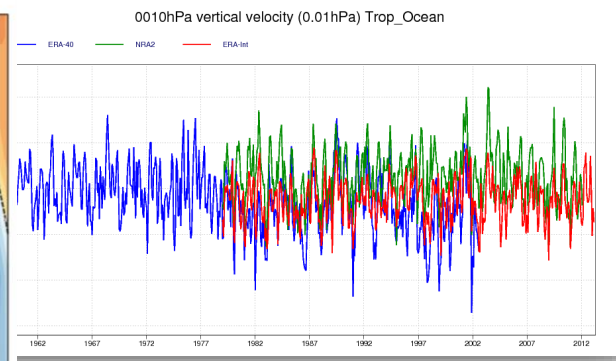
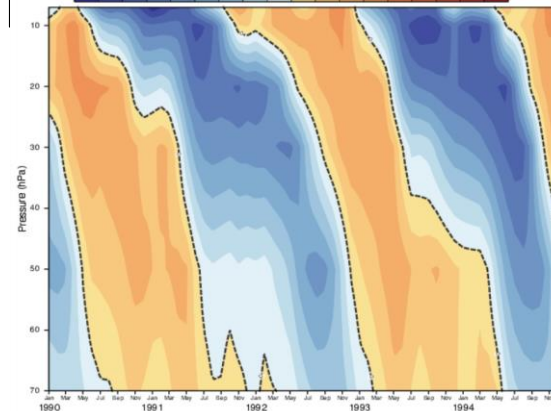
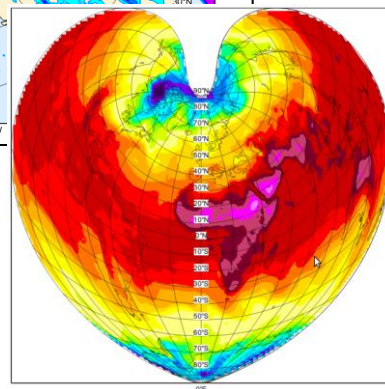
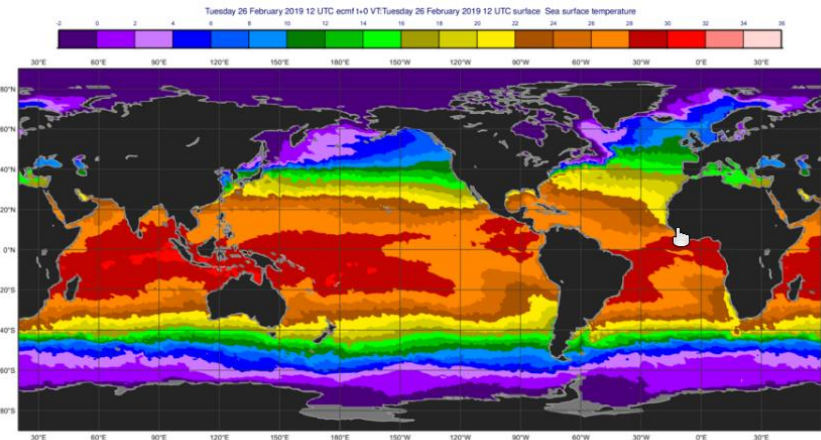
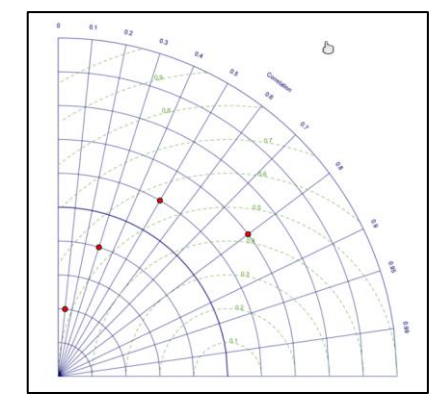
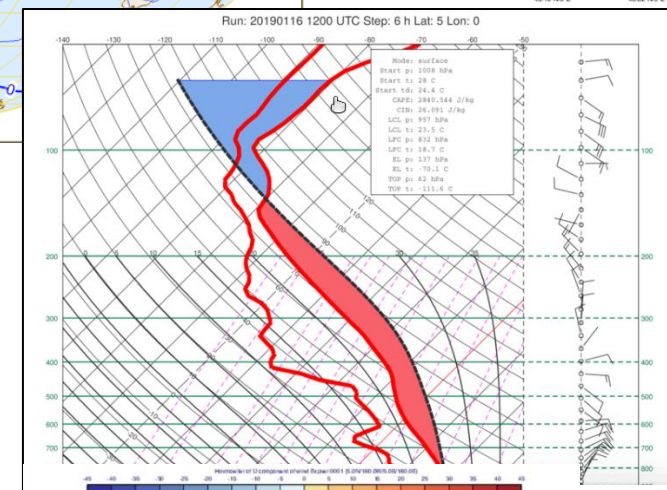
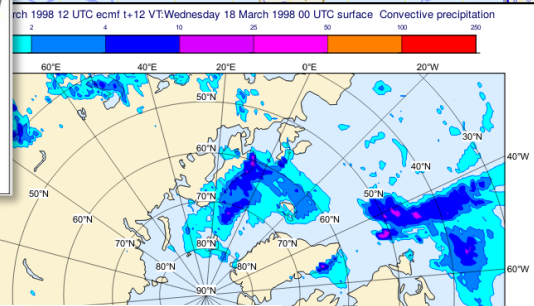
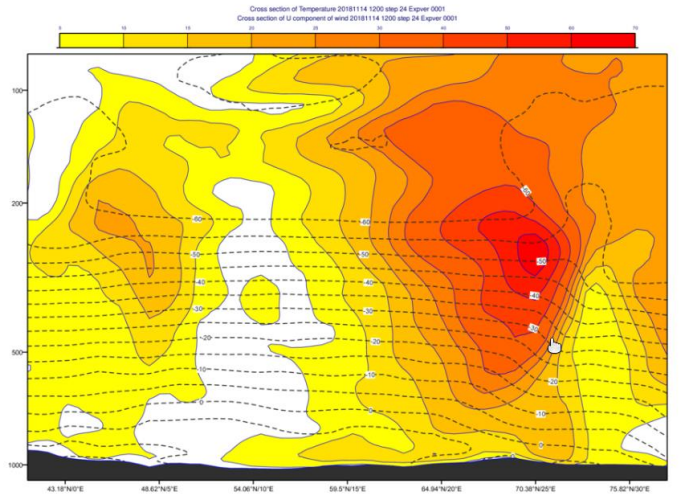
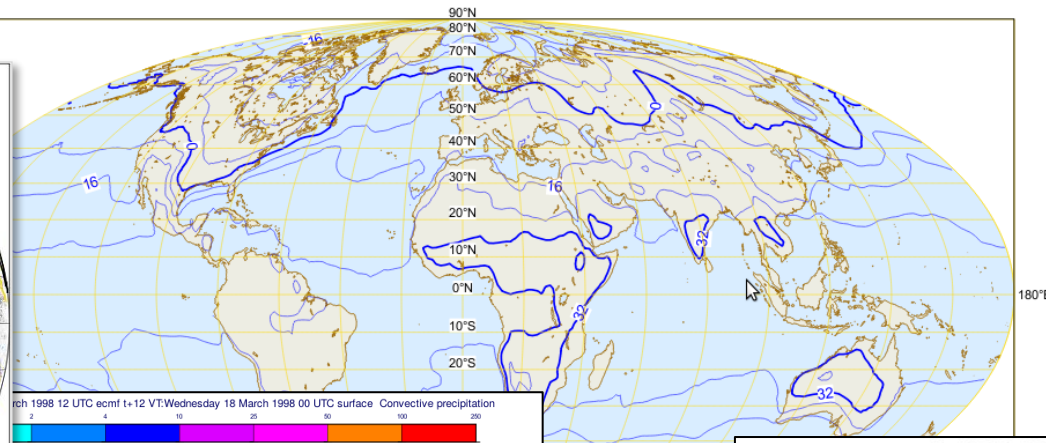
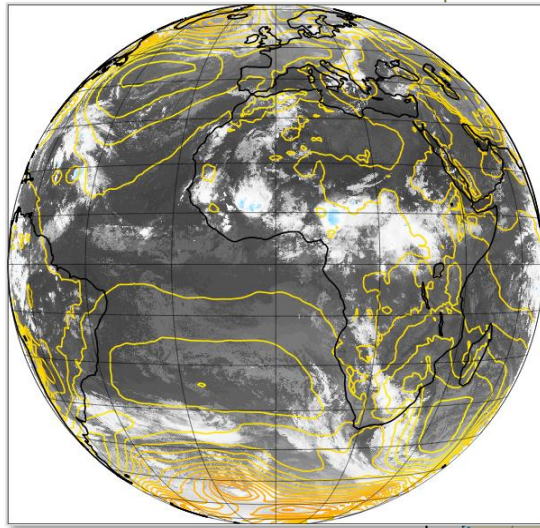
Index	latitude	longitude	level	date	time	value
1	69.6523	18.9057	0	20150126	1000	276.45
2	63.4882	10.8795	0	20150126	1000	275.25
3	63.5657	10.694	0	20150126	1000	276.25
4	61.2928	5.0443	0	20150126	1000	276.85
5	61.122	9.063	0	20150126	1000	265.35
6	60.7002	10.8695	0	20150126	1000	270.15
7	60.7733	10.8055	0	20150126	1000	270.45
8	61.455	10.1857	0	20150126	1000	267.55
9	58.7605	5.6505	0	20150126	1000	277.45
10	58.34	8.5225	0	20150126	1000	275.95
11	59.6193	10.215	0	20150126	1000	275.45
12	90	0	1000	20150220	1200	251.568
13	90	1.5	1000	20150220	1200	251.568
14	90	3	1000	20150220	1200	251.568
15	90	4.5	1000	20150220	1200	251.568
16	90	6	1000	20150220	1200	251.568
17	90	7.5	1000	20150220	1200	251.568
18	90	9	1000	20150220	1200	251.568
19	90	10.5	1000	20150220	1200	251.568
20	90	12	1000	20150220	1200	251.568
21	90	13.5	1000	20150220	1200	251.568

Visualisation - Overlay



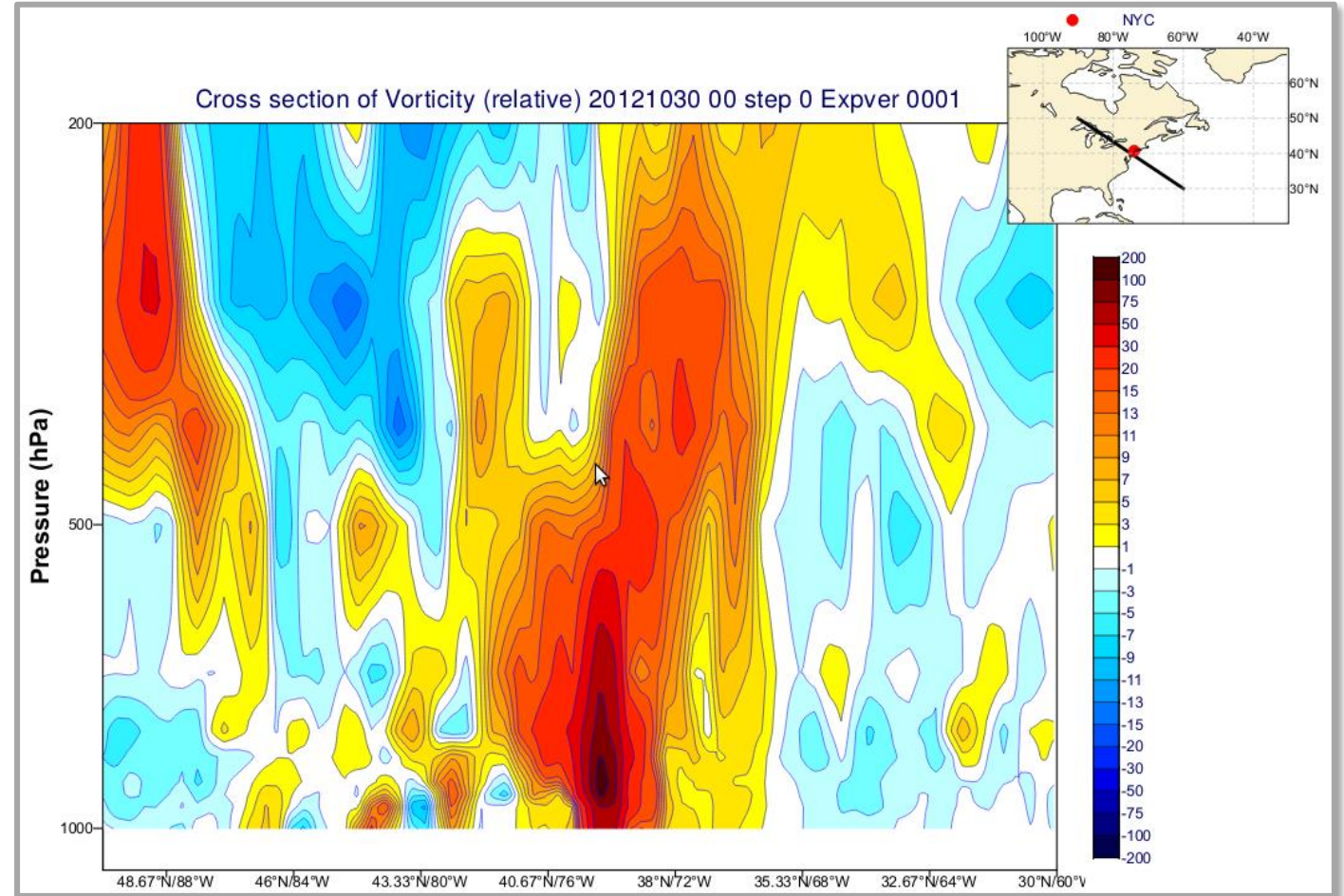
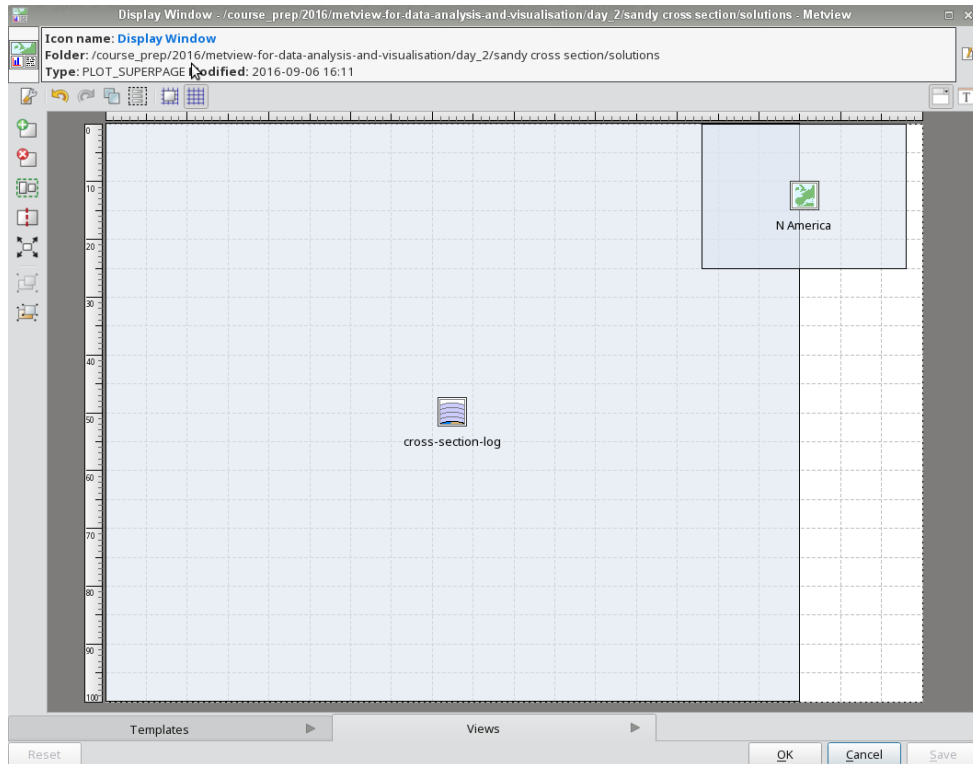
Views

Tuesday 05 March 2019 12 UTC ecmf t+0 VT: Tuesday 05 March 2019 12 UTC 1000 hPa Temperature



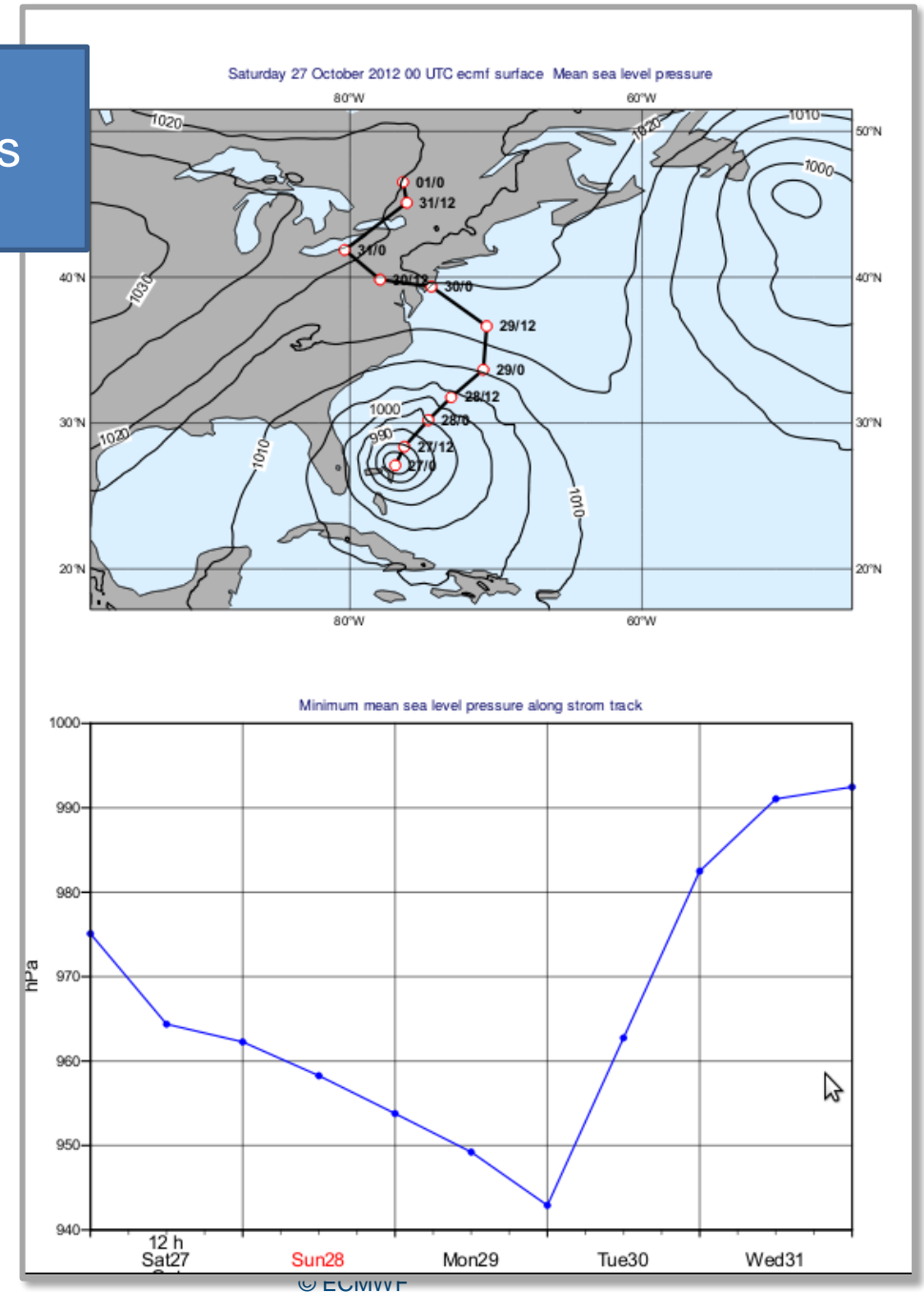
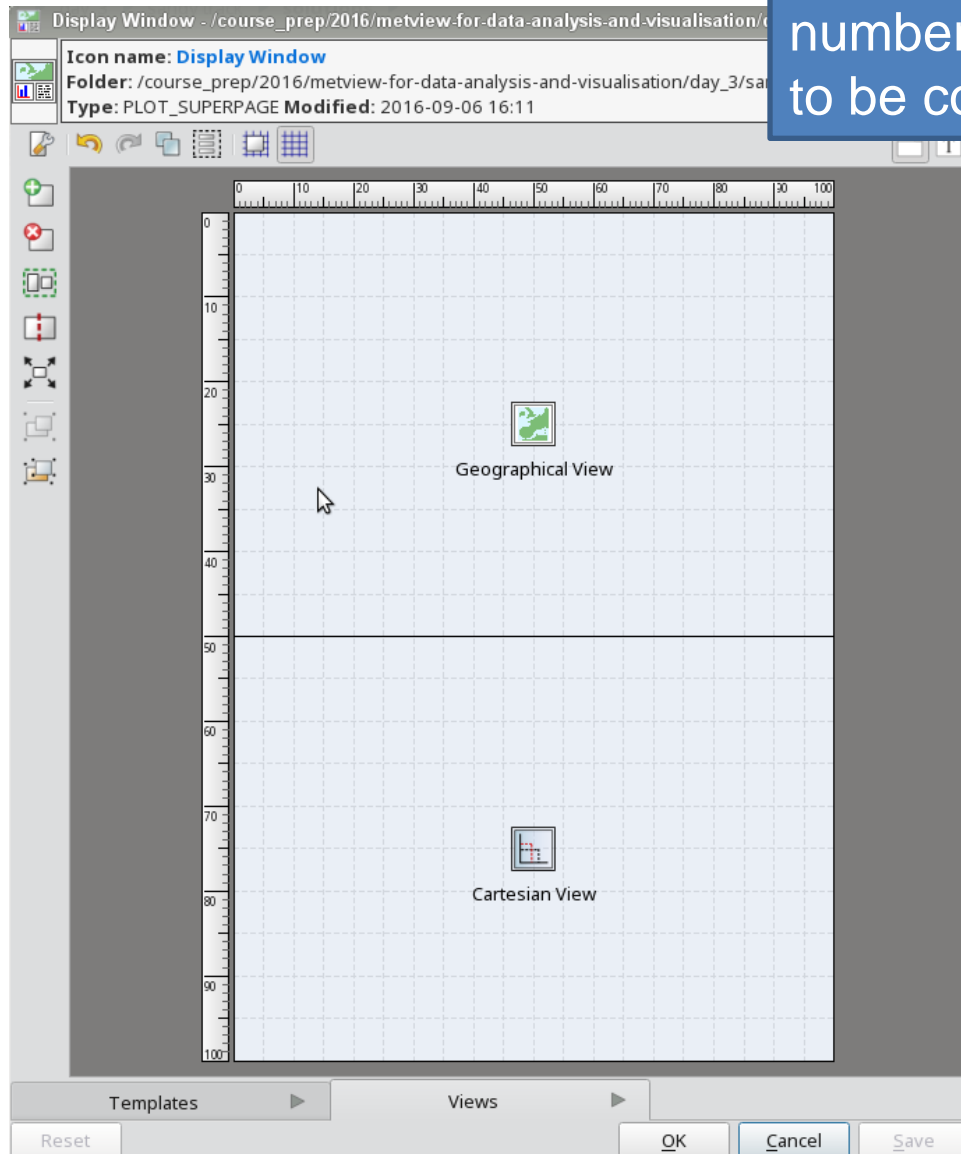
Visualisation - Layout

Layout editor allows any number of different views to be combined



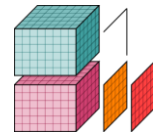
Visualisation - Layout

Layout editor allows any number of different views to be combined



Metview' Python interface

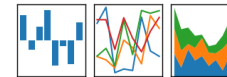
- Gives access to Metview's data retrieval, processing and visualisation capabilities in Python
- GRIB data is loaded as a *Fieldset*
- Can also return data as numpy, pandas and xarray
- Works with the user interface or standalone (UI can even generate Python code for you)
- New features include an interactive plotting widget and data overview functions
- We will use some new helper functions designed to give one-line access to useful plot layouts and styles ; also *datasets* – combination of data and pre-prepared styling



xarray

pandas

$$y_{it} = \beta^T x_{it} + \mu_i + \epsilon_{it}$$

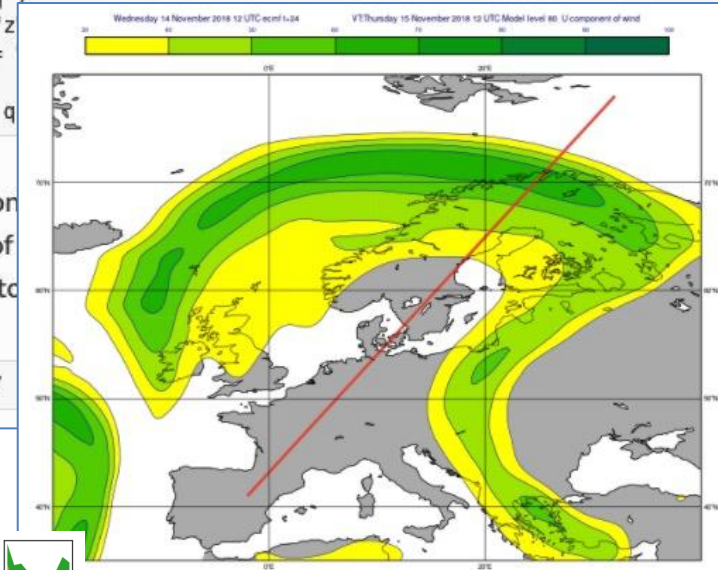


```
[6]: t = mv.read(data = fs, param = "t")
      q = mv.read(data = fs, param = "q")
      zs = mv.read(data = fs, param = "z")
      lnsp = mv.read(data = fs, param = "lnsp")

      z = mv.mvl_geopotential_on_ml(t, q)
```

Finally, the actual windshear computation is done using xarray fieldset operations. Here we made use of the `diff` function (and we have 137 model levels in total)

```
[7]: shear = (sp[0:135] - sp[1:136]) /
```



```
[11]: data1.describe()
```

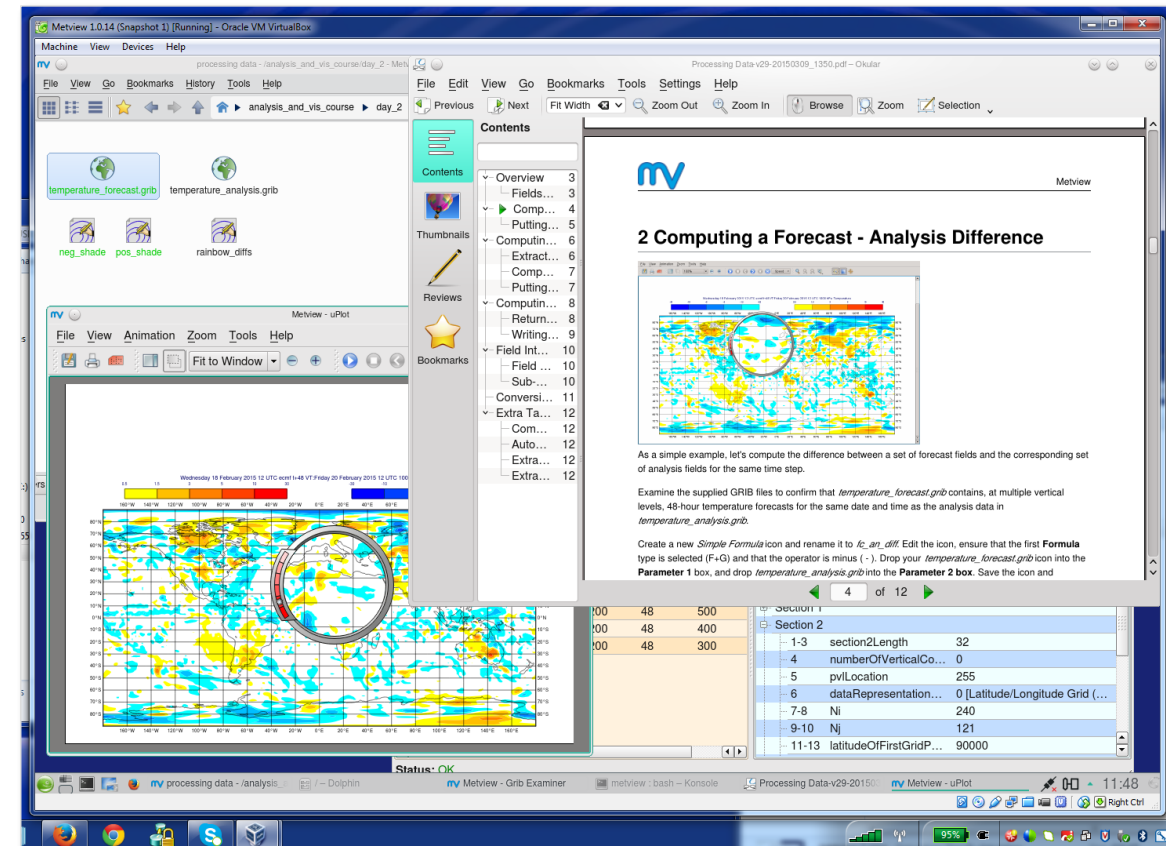
parameter	typeOfLevel	level	date	time	step	paramId	class
q	isobaricInhPa	100,250,...	20111215	0	0,6,...	133	od
t	isobaricInhPa	100,250,...	20111215	0	0,6,...	130	od

oper Par: msl Step: +72h Valid: 2016-09-28 00

an Par: msl Step: +0h Valid: 2016-09-28 00

Metview availability

- Available for Linux and macOS
- Inside ECMWF
 - `module load ecmwf-toolbox ; metview`
- Install from binaries
 - openSUSE, Fedora, Ubuntu
- Conda
 - `conda install metview` `-c conda-forge`
 - `conda install metview-batch` `-c conda-forge`
 - `conda install metview-python` `-c conda-forge`
- Homebrew
 - `brew install metview`
- Build from source
- The Metview Python interface can be installed separately if not in conda:
 - `pip install metview`



For more information...

- Ask for help:
 - <https://www.ecmwf.int/en/support>
- Visit our web pages:
 - <https://metview.readthedocs.io/en/latest/index.html>

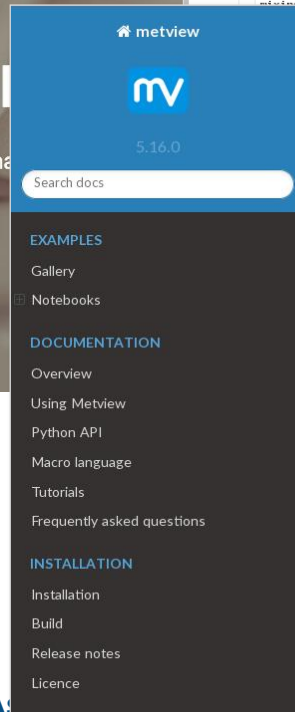


Welcome to ECMWF Support Portal

Support for users of ECMWF, Copernicus Atmosphere Monitoring Service (CAMS) and Copernicus Climate Change Service (C3S)

Search: What are you searching for?

Questions?



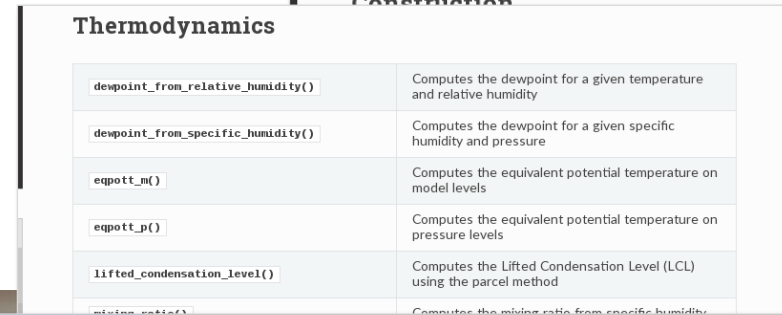
metview
mv
5.16.0

Search docs

EXAMPLES
Gallery
Notebooks

DOCUMENTATION
Overview
Using Metview
Python API
Macro language
Tutorials
Frequently asked questions

INSTALLATION
Installation
Build
Release notes
Licence



Function	Description
<code>dewpoint_from_relative_humidity()</code>	Computes the dewpoint for a given temperature and relative humidity
<code>dewpoint_from_specific_humidity()</code>	Computes the dewpoint for a given specific humidity and pressure
<code>eqpott_m()</code>	Computes the equivalent potential temperature on model levels
<code>eqpott_p()</code>	Computes the equivalent potential temperature on pressure levels
<code>lifted_condensation_level()</code>	Computes the Lifted Condensation Level (LCL) using the parcel method
<code>mixing_ratio_from_specific_humidity()</code>	Computes the mixing ratio from specific humidity



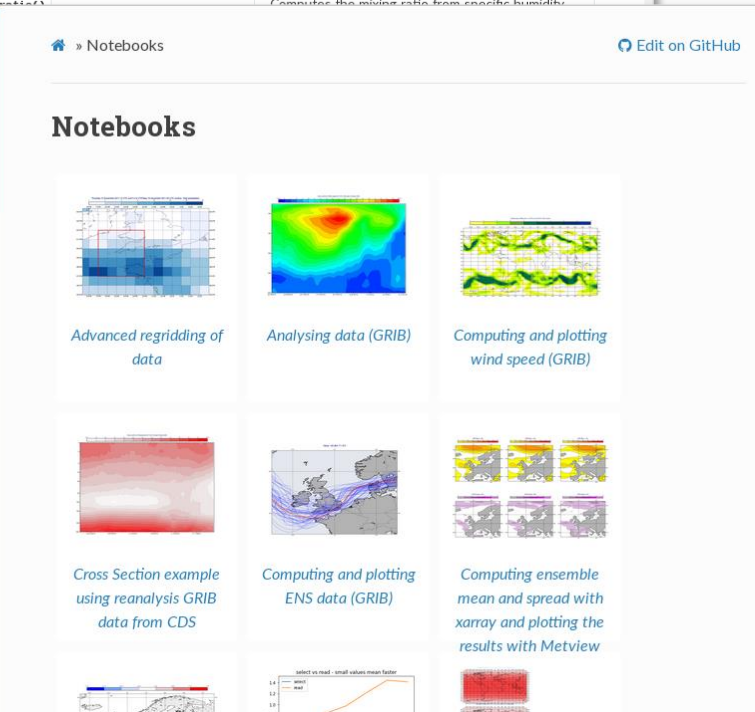
Python API » Data types » Fieldset object

Fieldset object

`class Fieldset`

Metview's Fieldset object represents GRIB data. It is a container-like object with each entry representing a GRIB message.

Construction



Notebooks

- Advanced regridding of data
- Analysing data (GRIB)
- Computing and plotting wind speed (GRIB)
- Cross Section example using reanalysis GRIB data from CDS
- Computing and plotting ENS data (GRIB)
- Computing ensemble mean and spread with xarray and plotting the results with Metview

Practical session

Inside Jupyterlab browser:
`home/Metview_hands_on/Metview_Introduction.ipynb`