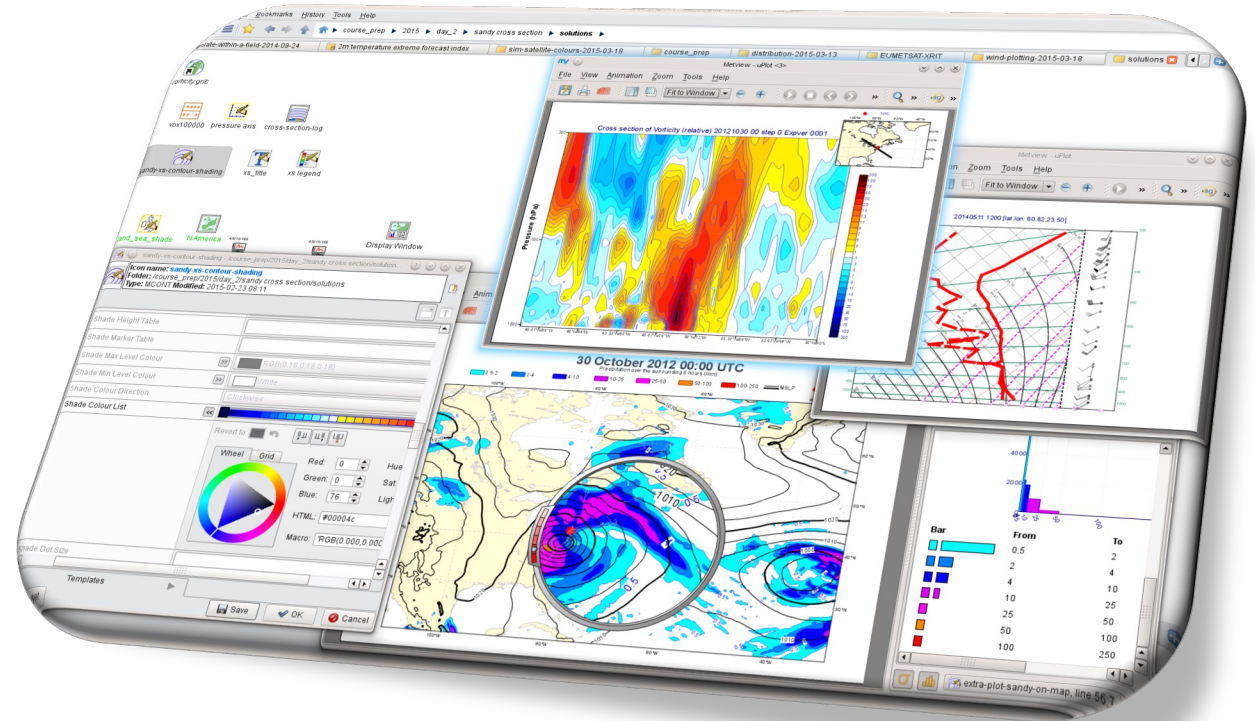


Metview Introduction

ECMWF

Iain Russell,
Sándor Kertész

Development Section, ECMWF



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
- Metview is a co-operation project with INPE (Brazil)



EUROPEAN CENTRE FOR MEDIUM-RANGE WEATHER FORECASTS

The screenshot displays the Metview software interface. At the top, a map shows a weather system over Europe with a color scale from 1 to 30. Below the map is a file browser window titled 'Metview - Desktop' showing a directory structure with folders like 'Tests', 'Vapor', and 'solutions'. A table below the browser shows properties for a file named 'ERA5-ens-z500'. In the foreground, a script editor shows Python code for data retrieval and calculation. The bottom right shows a plot area with a map and a cross-section view of wind speed and pressure.

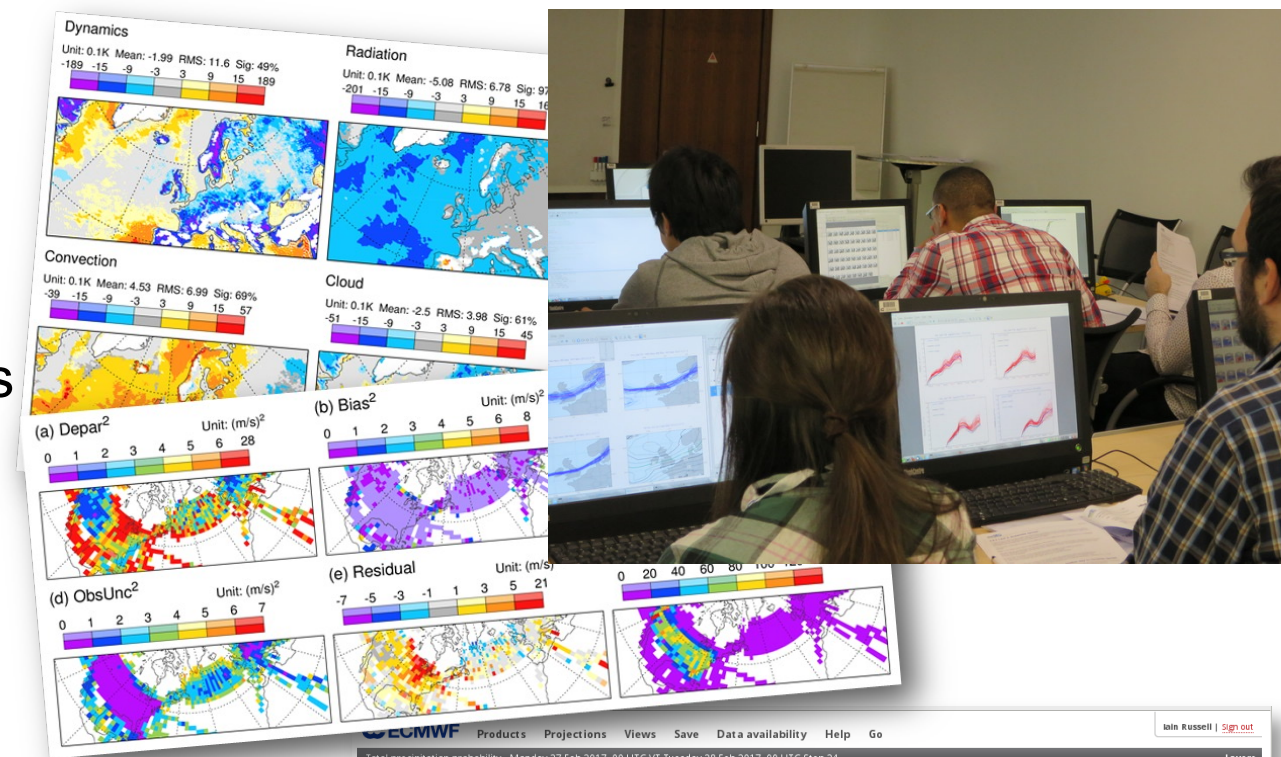
```
6 # retrieve some data
7
8 f1 = retrieve (date : -1, levels : 1000, grid : [1.5, 1.5])
9 f2 = retrieve (date : -2, levels : 1000, grid : [1.5, 1.5])
10
11
12 # perform some calculation
13
14 cv_f1f2 = covar_a (f1, f2)
15 cv_f1f1 = covar_a (f1, f1)
16 cv_f2f2 = covar_a (f2, f2)
17 var_f1 = var_a (f1)
18 var_f2 = var_a (f2)
19
20 corr_manual = cv_f1f2
21 corr_manual2 = cv_f1f1
22 corr_builtin = corr_a (cv_f1f2, cv_f1f1)
23
```

covar of f1 and f2 = 6152
corr_manual = 0.8702346
corr_manual2 = 0.8702346
corr_builtin = 0.8702346

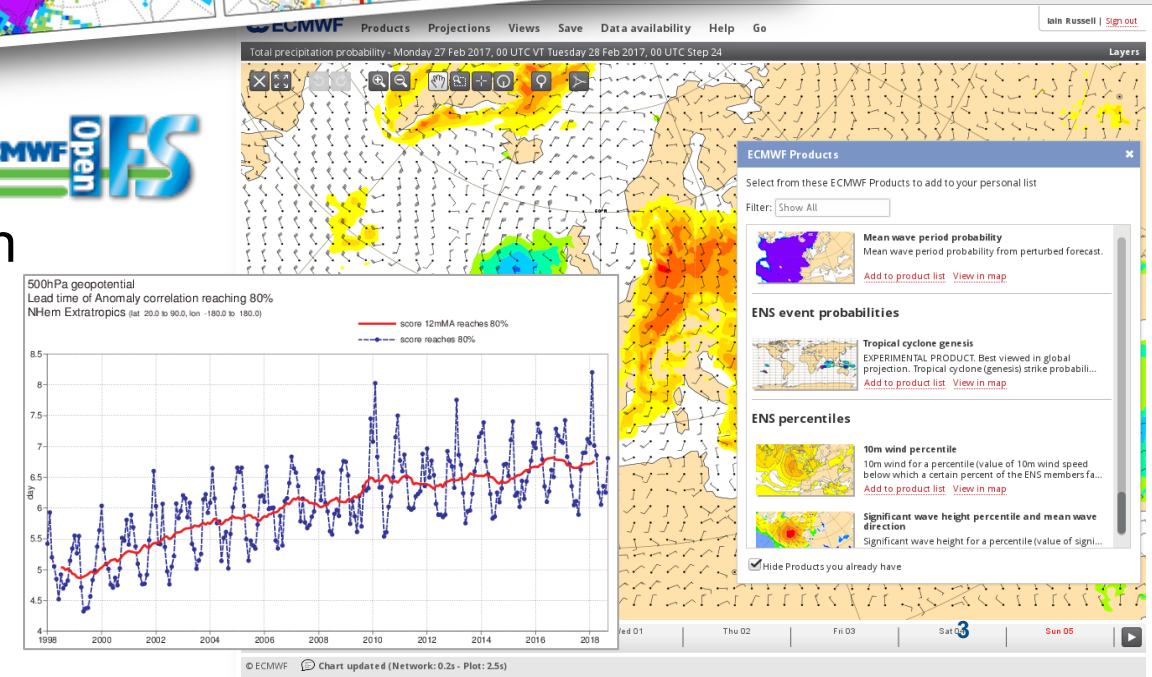
Program finished (OK) : 611 ms [Finished at 11:18:47]

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



Built on top of ECMWF software packages

Metview

Data decoding

Regridding

Plotting

Data Access

ecCodes
(GRIB,
BUFR)

ODC

Other
(NetCDF,
Geopoints
, CSV)

MIR,
Atlas

Magics

MARS

CDS

Files

WMS

URL

Metview + GRIB

- Plot
- Examine
- Filter, regrid
- Crop, mask
- Maths, Boolean
- Specialised:
 - Cross section
 - Thermodynamics
 - Gradient
 - Vertical integration
 - Model to pressure lev
 - Etc

The image displays several screenshots from the Metview software interface:

- Top Left:** Two side-by-side maps showing the U component of wind for Sunday 22 February 2015 12 UTC. The maps cover a region from 20°W to 20°E and 30°N to 70°N, with wind vectors and contours overlaid on a geographical map of Europe and the Atlantic.
- Top Right:** A vertical cross-section plot titled "20121027 0z +72". It shows a color-coded vertical profile of a meteorological variable, with a color scale at the top ranging from 0.4 to 4.1211. Contours are overlaid on the color plot.
- Middle Left:** A data table window showing a list of GRIB messages with columns for time (Zt), position, and other metadata.
- Middle Right:** A window titled "20140512 1200 step 0 [0.00,0.00] saturation over water" showing a cross-section plot with a grid and a red line indicating a specific path or boundary.
- Bottom Left:** A large map window titled "Thursday 24 April 2014 12 UTC ecmf 1000 hPa Temperature" showing a global temperature distribution at 1000 hPa with a color scale from -40 to 40.
- Bottom Center:** A histogram window titled "Histogram (for data in visible area)" showing a bar chart of data distribution with a color-coded legend.
- Bottom Right:** A statistics window titled "statistics - /home/graphics/cgi/metview/Tests/Macros/statistics" containing a script with commands like `retrieve`, `covar_a`, and `corr_manual`, along with their output values.

Metview + BUFR

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

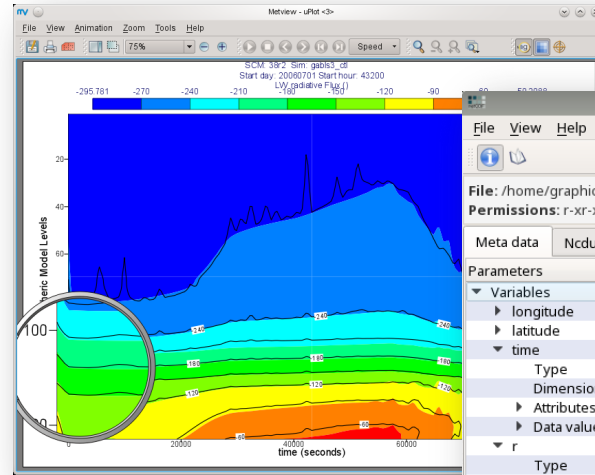
The image displays several screenshots from the Metview software interface:

- Top Left:** A window titled "synop.bufr - Bufr Examiner (Metview)". It shows a menu bar (File, View, Profiles, Filter, Help) and a key profile set to "System:Default". Below this, it displays file information: "File: /home/graphics/cgi/metview/course_prep/2016/metview-for-data-analysis-and-visualisation/day_2/data_1/synop.bufr", "Permissions: rwxr-x--- Owner: cgi Group: graphics Size: 357 KB Modified: 2016-09-06 16:11:04", and "Total number of messages: 1660". A message list table is visible with columns for Message, Typ, Sut, C, Mv, Lv, Ssc, 2, and D. The selected message (58) is highlighted.
- Top Right:** A global map showing a color-coded pressure field or wind vectors, with a color scale at the bottom ranging from 20 to 3152.
- Middle Left:** A thermodynamic chart titled "20140510 1200 [lat,lon: 60.82,23.50]". The y-axis represents pressure in hPa (from 1000 to 200), and the x-axis represents temperature in °C (from -30 to 30). The chart shows a red line representing the parcel's path, with various thermodynamic variables plotted.
- Bottom Left:** A detailed map of the Indian Ocean region, showing a circular inset with a weather symbol and numerical values (10, 114, 2, 24, 72, 5, 08, 60).
- Bottom Right:** A world map showing a global distribution of data points (green and brown dots). A circular inset shows a zoomed-in view of a specific region. A popup window displays coordinates and a table of values:

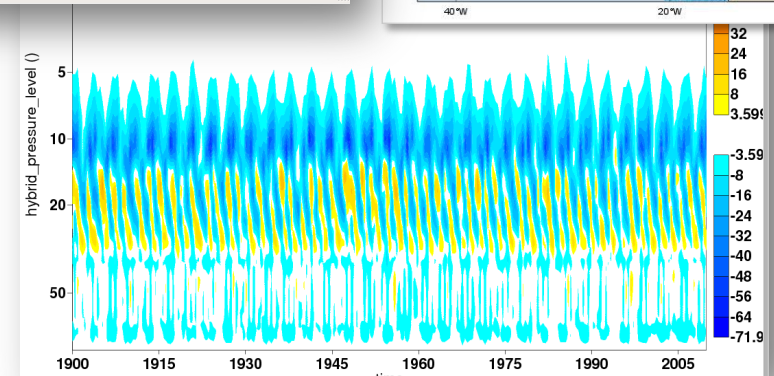
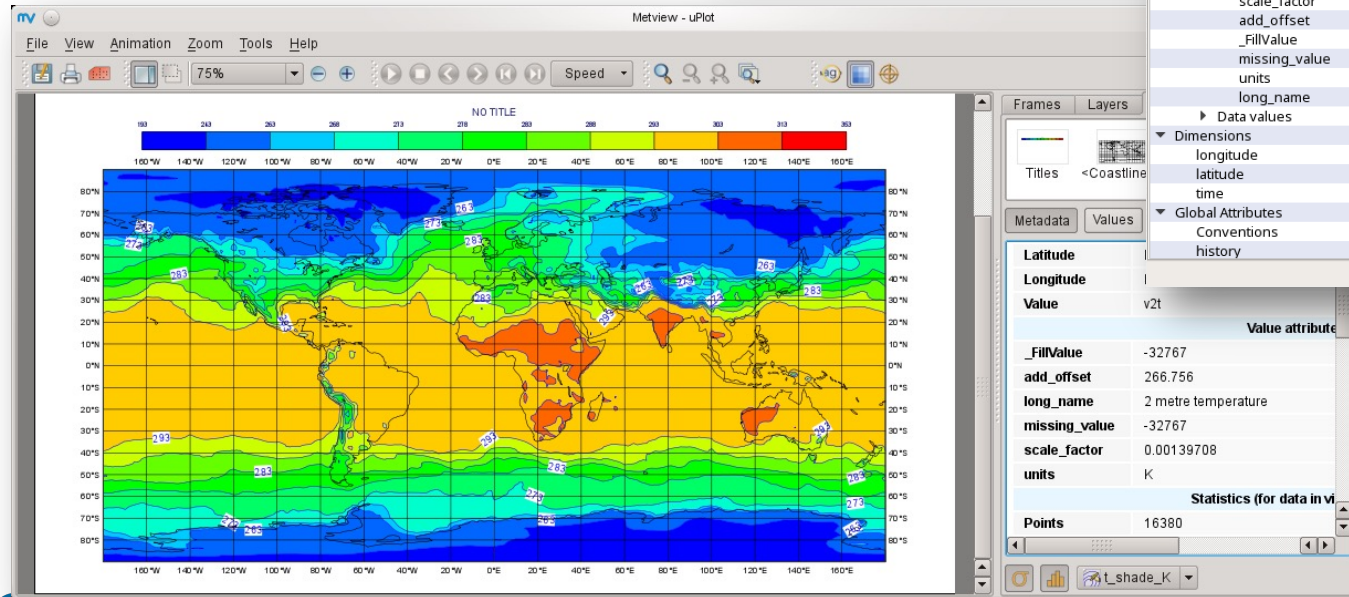
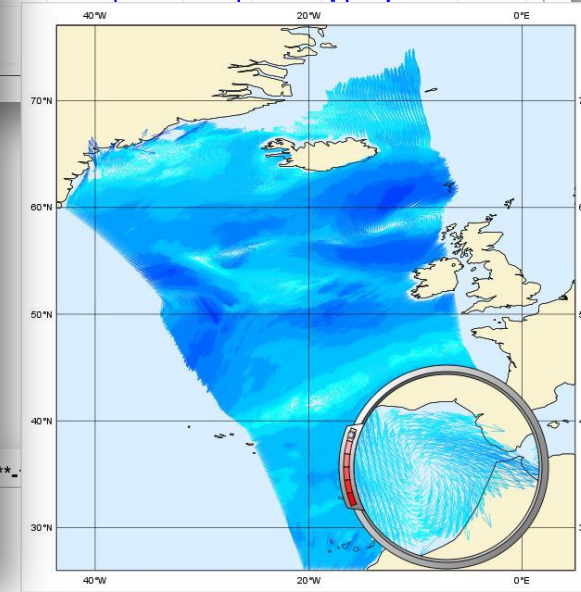
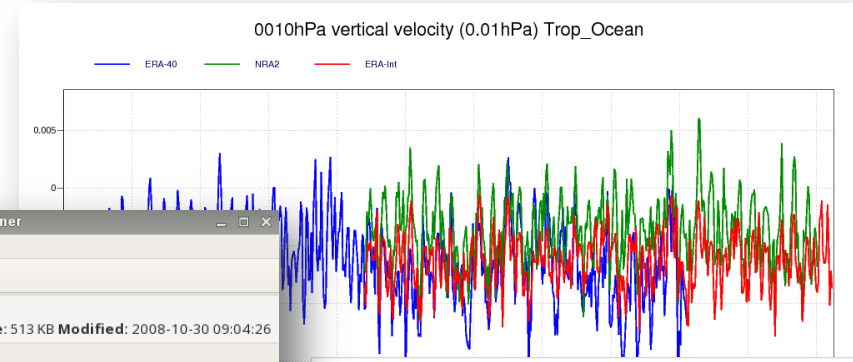
Layer	Value	Lon	Lat	Dist (km)
filter_o	301.800000	36.90	-1.32	118.92

Metview + NetCDF

- Plot
- Examine
- Maths, Boolean



Parameters	Values
Variables	
▶ longitude	
▶ latitude	
▼ time	
Type	nlong (int)
Dimensions	(time)
▶ Attributes	
▶ Data values	
▼ r	
Type	short
Dimensions	(time, latitude, longitude)
▼ Attributes	
scale_factor	0.0016890305022737
add_offset	52.1403569442292
_FillValue	-32767s
missing_value	-32767s
units	%
long_name	Relative humidity
▶ Data values	
▼ Dimensions	
longitude	720
latitude	361
time	1
▼ Global Attributes	
Conventions	CF-1.0
history	2005-05-10 12:55:35 GMT by mars2netcdf-0.92



Metview + ODB

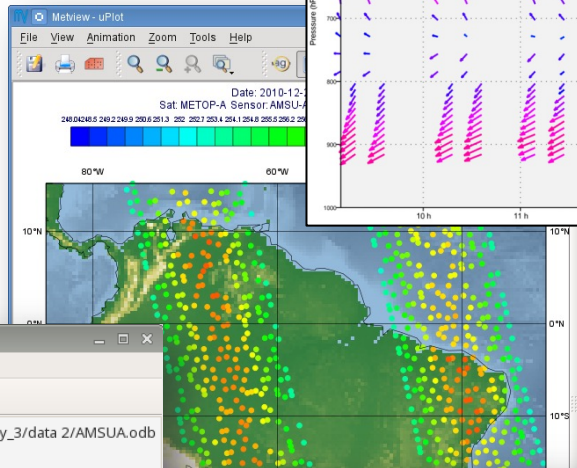
- Plot
- Examine
- Filter
- Convert to Geopoints

Metview - ODB Examiner

File View Settings Help

File: /home/graphics/cgi/metview/course_prep/2016/metview-for-data-analysis-and-visualisation/day_3/data 2/AMSUA.odb
 Symlink target: /scratch/graphics/cgr/odb_data/AMSUA.odb
 Permissions: rw-r--r-- Owner: cgr Group: graphics Size: 17 MB Modified: 2017-01-06 09:20:58

Name	Type	Constant	Min	Max	Table
an_depar@body	float	n	-3.47203	3.95354	body
an_sens_obs@body	float	y	0	0	body
andate@desc	int	y	20101222	20101222	desc
antime@desc	int	y	0	0	desc
biascorr@body	float	n	-0.739071	4.33658	body
biasctrl@body	float	n	-0.737179	4.35366	body
bufrtype@hdr	int	y	3	3	hdr
class@desc	string	y	N/A	N/A	desc
codetype@hdr	int	y	210	210	hdr
datastream@sat	int	n	0	1	sat
date@hdr	int	n	20101221	20101222	hdr
datum_anflag@body	bitfield	n	N/A	N/A	body
datum_event1@body	bitfield	y	N/A	N/A	body
datum_rdbflag@body	bitfield	y	N/A	N/A	body
datum_status@body	bitfield	n	N/A	N/A	body
entryno@body	int	n	3	14	body
expver@desc	string	y	N/A	N/A	desc
fc_sens_obs@body	float	y	0	0	body
fg_depar@body	float	n	-3.34557	3.28031	body
fg_error@errstat	float	n	0.0603554	3.8495	errstat
final@update_1	float	n	-2.14748e+09	-2.14748e+09	update_1



Statistics (for data in visible area)

Points	1132
Minimum	248.04
Maximum	264.03
Average	257.625
Stdev	3.23466
Skewness	-0.296992
Kurtosis	-0.6587

Odb Nb Rows: -1

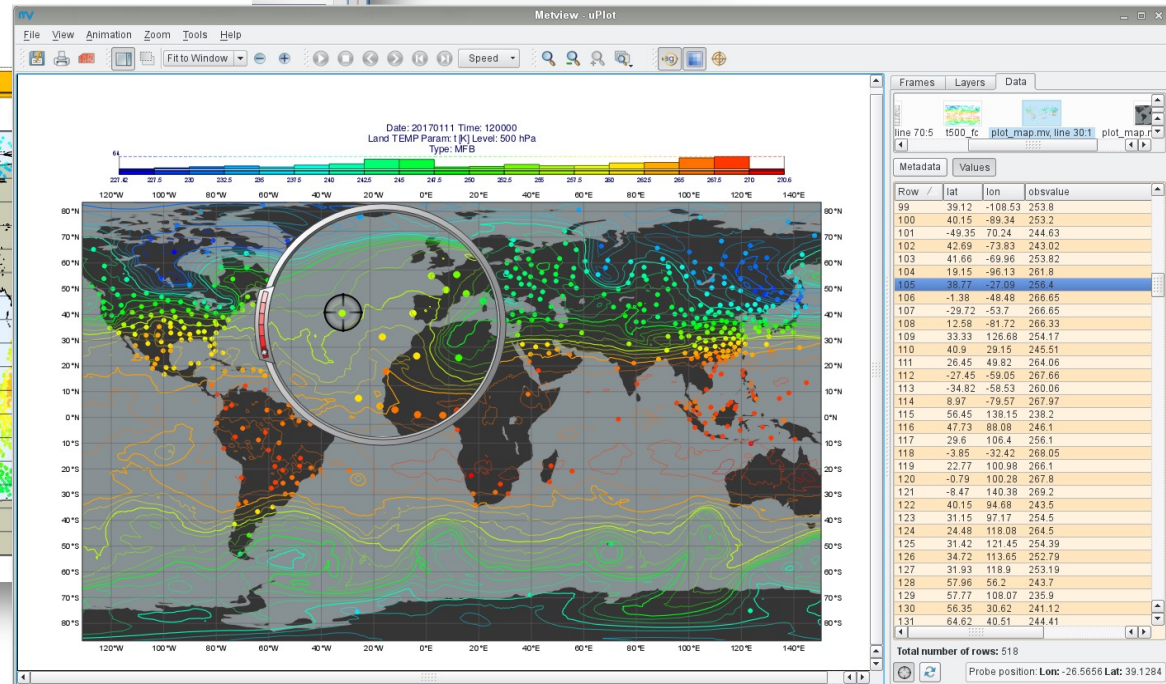
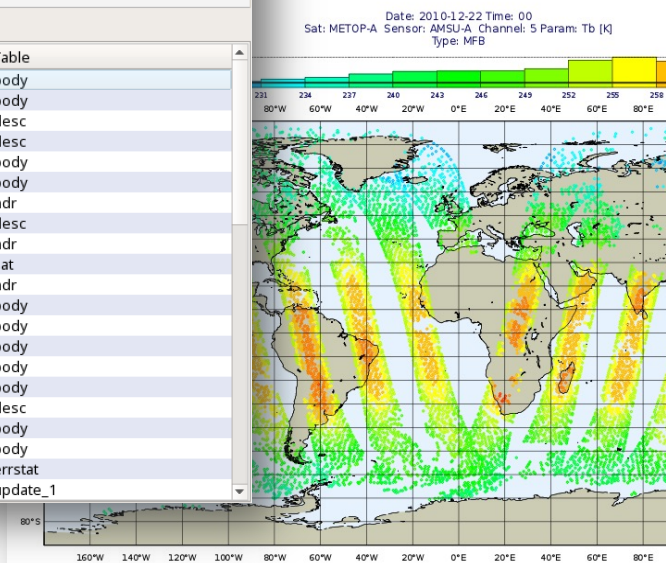
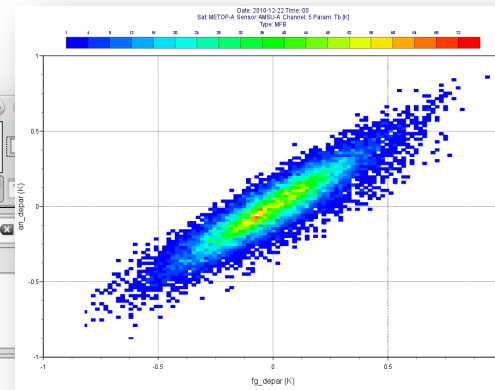
ODB Filter

Accepted icons: AMSUA.odb

Odb Query

```
select
  lat@hdr as lat,
  lon@hdr as lon,
  fg_depar@body as fg_dep,
  an_depar@body as an_dep
where
  vertco_reference_1@body = 5
```

Buttons: Reset, Save, OK, Cancel



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

The screenshot displays the Metview software interface. On the left, a map of Europe shows geopotential height contours. A color scale at the top indicates values from -13.1175 to 0. In the center, a 'Geopoints Examiner' window shows a table of 11 data points. On the right, a 'CSV (and similar)' window displays a list of coordinates and values. At the bottom, a 'geopoints.gpt' window shows the ASCII format of the data.

Geopoints Examiner Table:

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

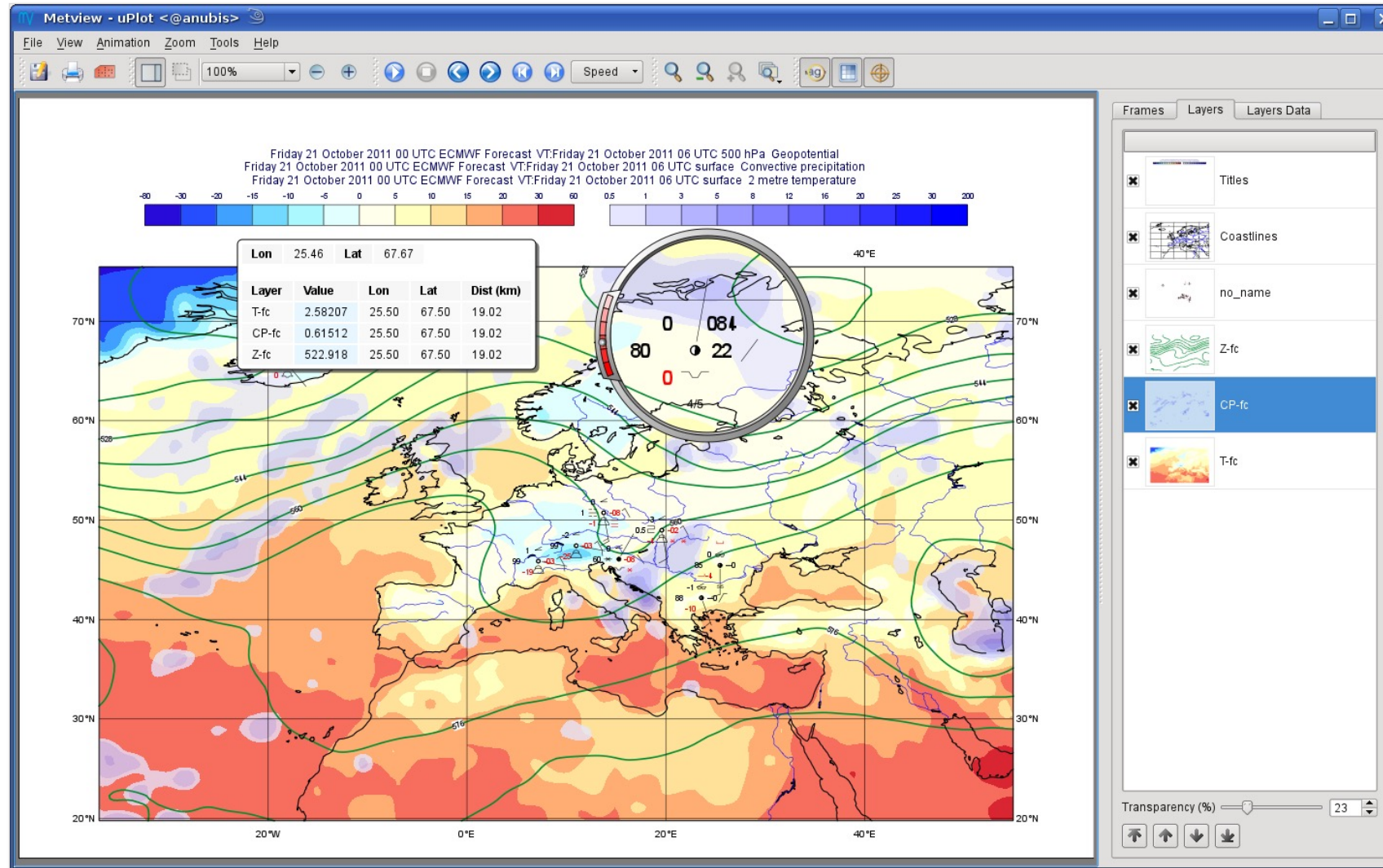
CSV (and similar) Table:

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	.90788	-0.50169	5.1074401690909

geopoints.gpt ASCII Table:

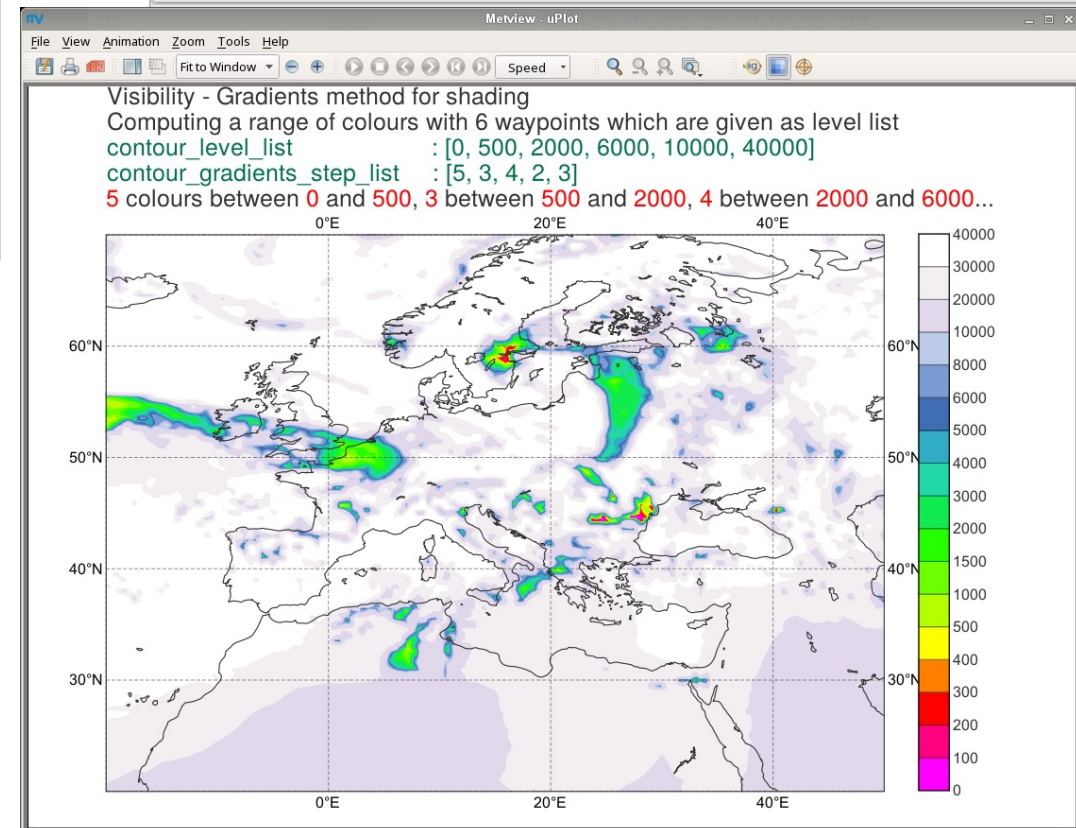
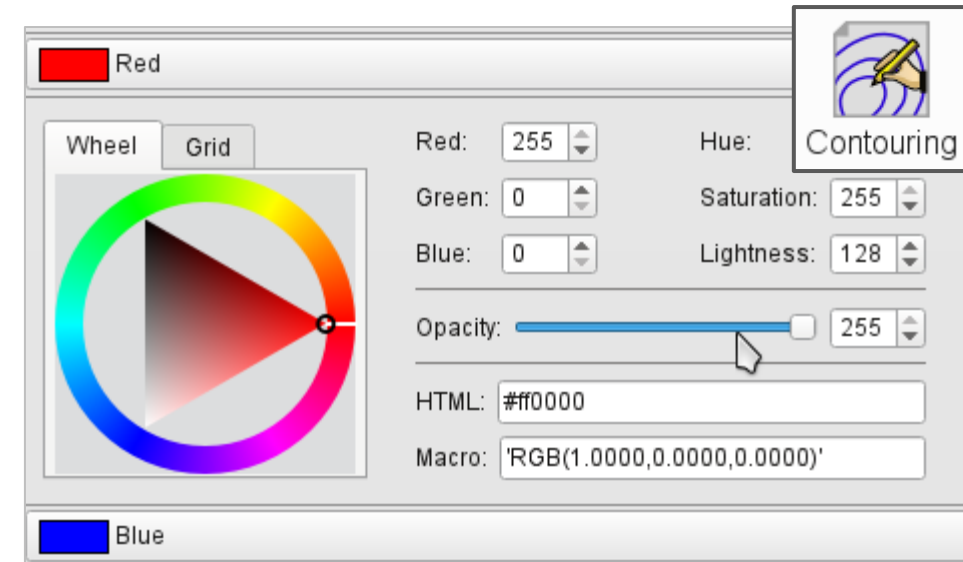
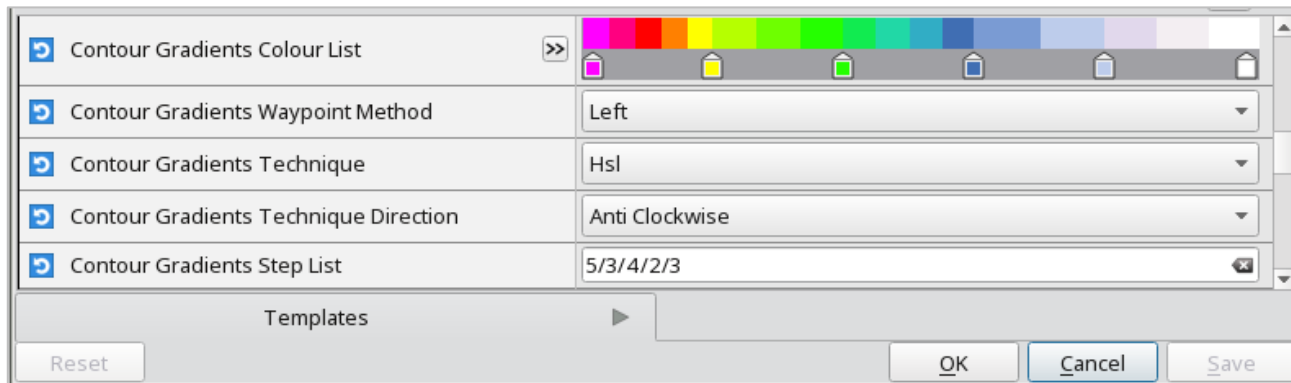
Index	latitude	longitude	level	date	time	value
6	90	0	1000	20150220	1200	251.568
7	90	1.5	1000	20150220	1200	251.568
8	90	3	1000	20150220	1200	251.568
9	90	4.5	1000	20150220	1200	251.568
10	90	6	1000	20150220	1200	251.568
11	90	7.5	1000	20150220	1200	251.568
12	90	9	1000	20150220	1200	251.568
13	90	10.5	1000	20150220	1200	251.568
14	90	12	1000	20150220	1200	251.568
15	90	13.5	1000	20150220	1200	251.568

Visualisation - Overlay



Contouring schemes

- Plenty of options for complete customisation of palettes



Contouring schemes



- A set of pre-defined palettes is also available
 - But you still have to supply the mapping between values and colours

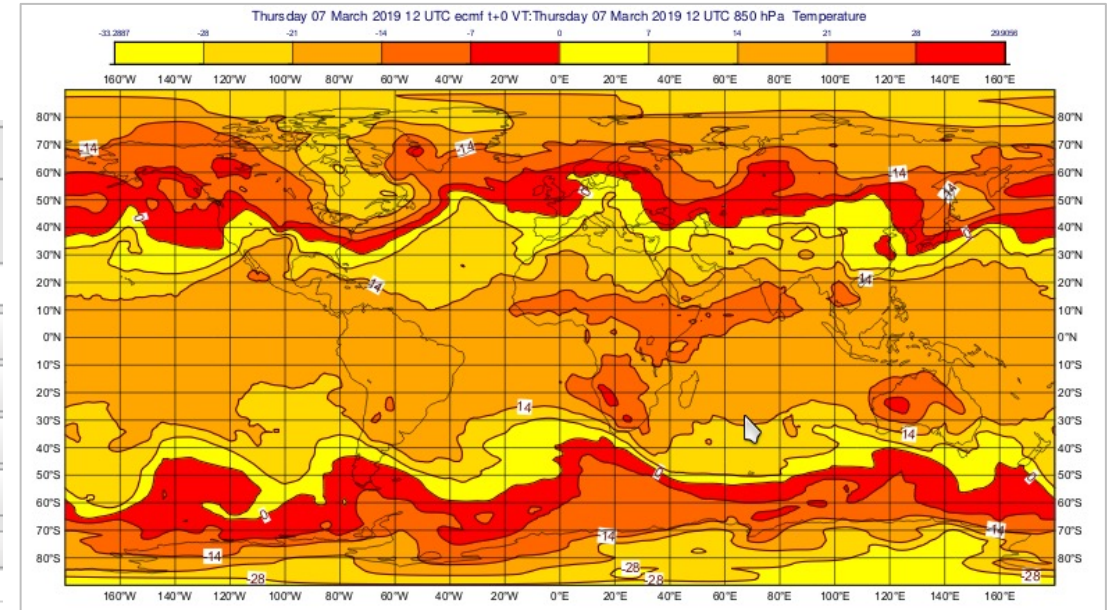
Contour Shade Palette Name: eccharts_yellow_red_5

Clear all filters

Name: ANY
Origin: ANY
Colour: ANY
Count: ANY
Parameter: ANY

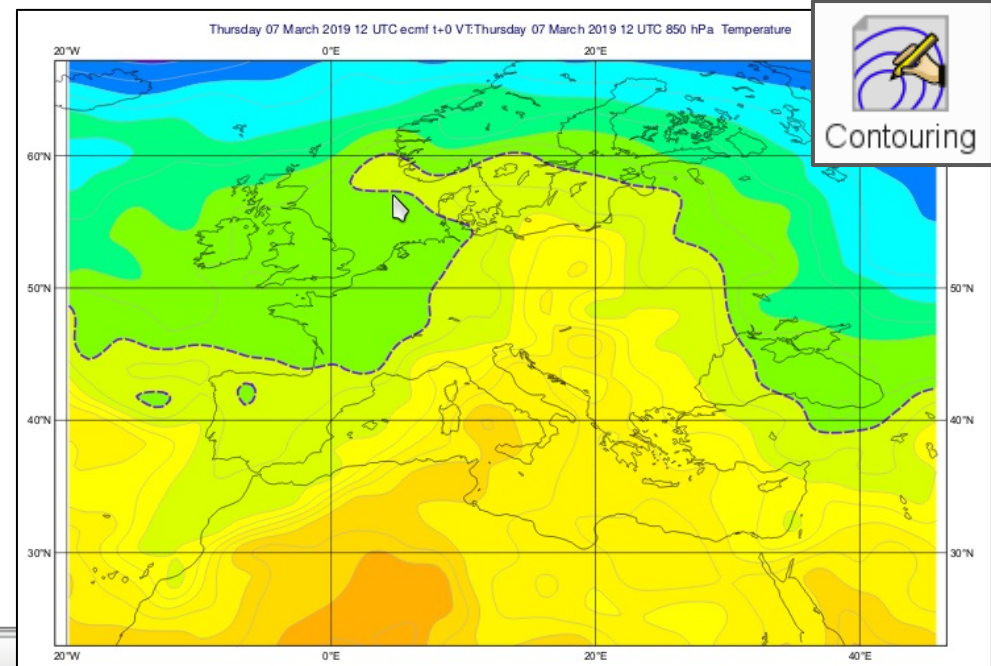
Palette	Name
	eccharts_white_transparent_12
	eccharts_yellow_blue_14
	eccharts_yellow_blue_dark_9
	eccharts_yellow_darkred_13
	eccharts_yellow_navy_8
	eccharts_yellow_purple_6
	eccharts_yellow_red_5
	eccharts_yellow_red_9
	m_alt_rainbow2_10
	m_alt_rainbow2_11

Contour Shade Palette Policy: Cycle



Contouring schemes

- Can select from pre-defined styles
 - the styles come from ecCharts
 - everything is done for you
 - or choose “Contour Automatic Setting = ECMWF” – style will be chosen based on meta-data



Contour Automatic Setting

Style Name

Contour Style Name << sh_all_fm48t56i4_ct_wh

tempera

Matching styles

- sh_all_fm48t56i4
- sh_all_fm48t56i4_ct_wh**
- sh_all_fm50t58i2
- sh_all_fm52t48i4
- sh_all_fm52t48i4_light
- sh_all_fm64t52i4
- sh_all_fm80t56i4_v2
- sh_anomaly_rb_m20t20
- sh_blured_fm1t1st
- sh_efi2t_fm1t1st

Style sh_all_fm48t56i4_ct_wh

Img

Method Method : Area fill & grey contours Level range : -48 to 56 Interval : 2 Thickness : 1 Colour : All colours Used for temperature

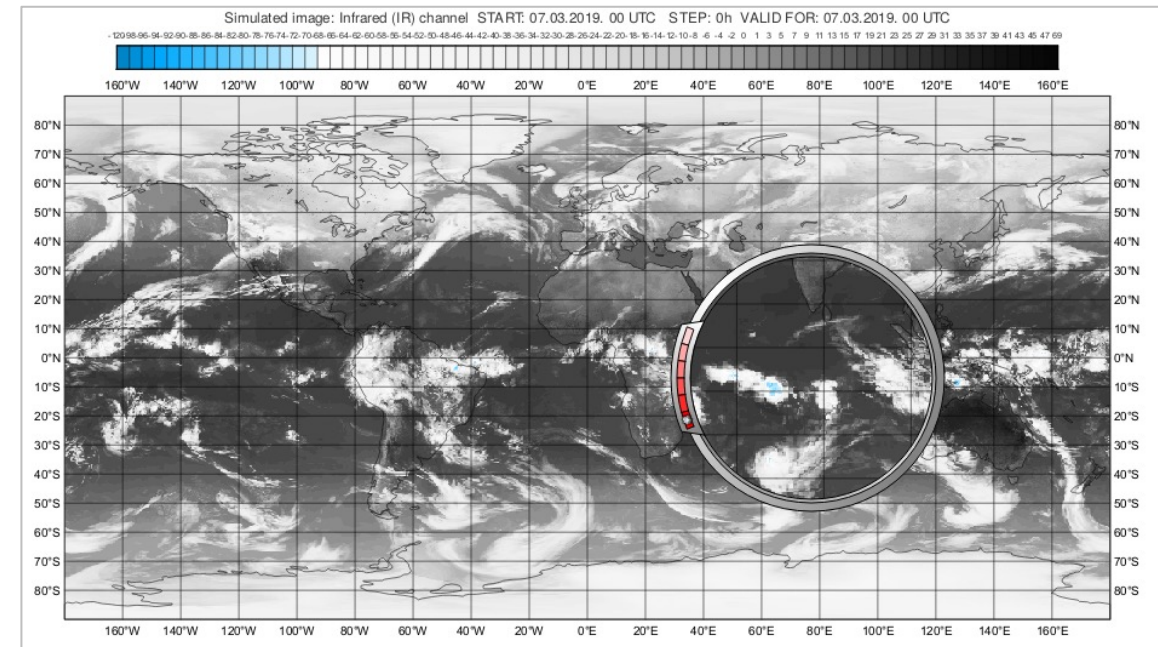
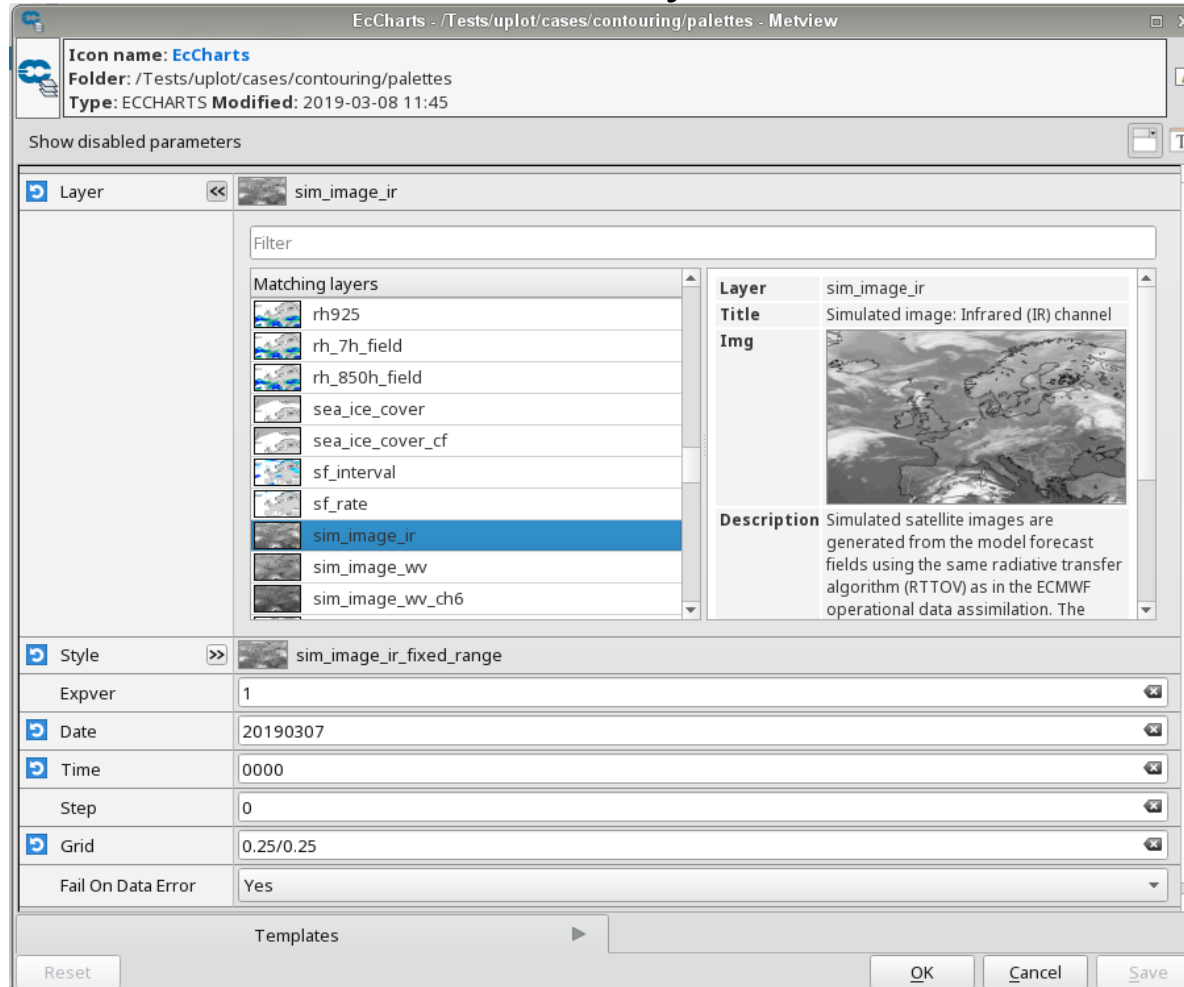
Layers 2t, mn2t, mx2t, 2t_dewpoint

Keywords temperature, T2m, rainbow

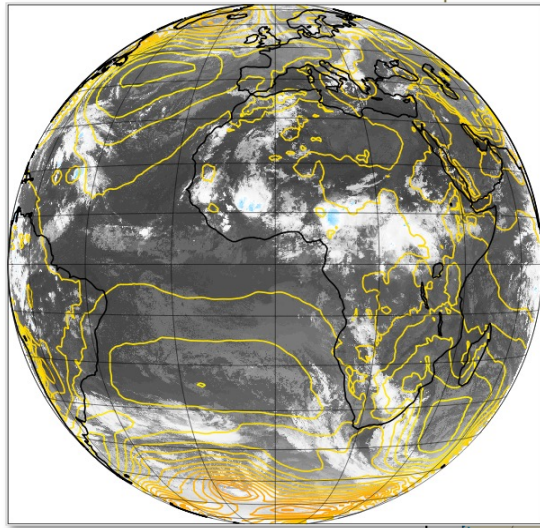
Colours blue, magenta

Complete ecCharts Layers

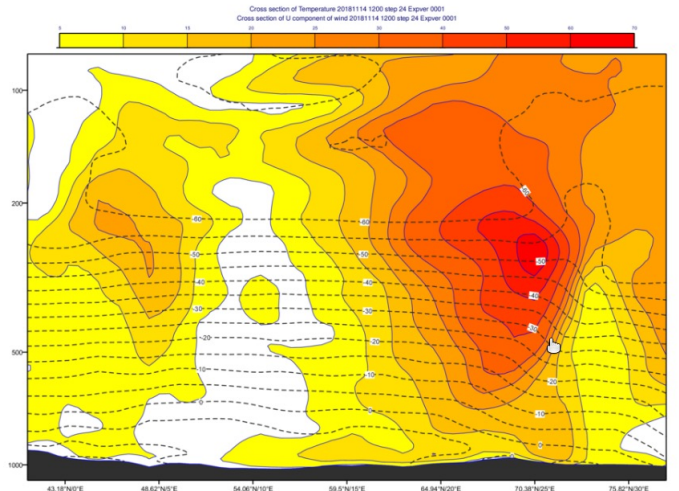
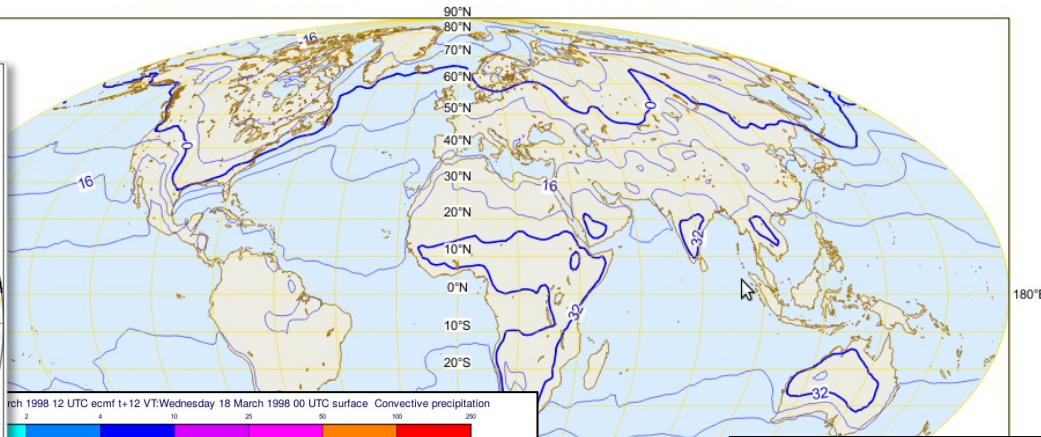
- The ecCharts icon goes further – retrieves data from MARS and styles it



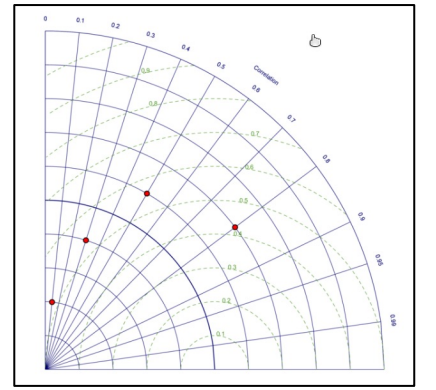
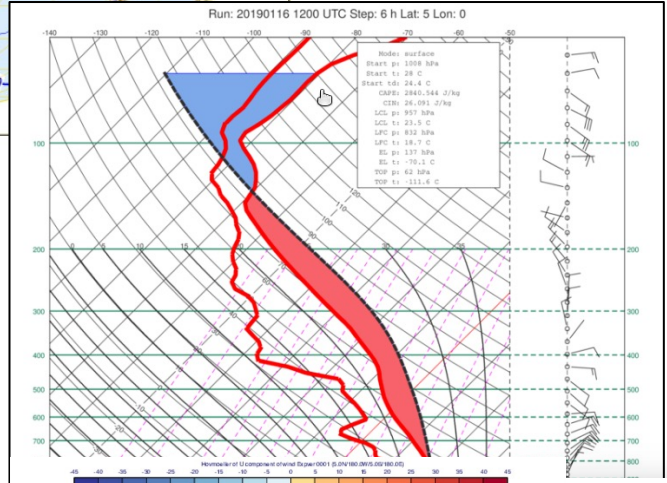
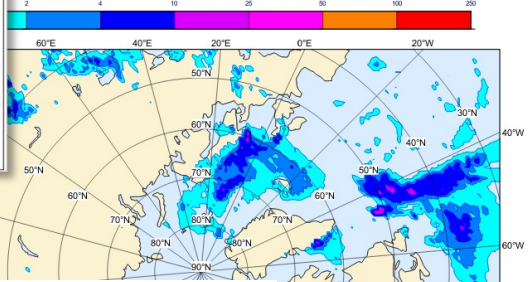
Views



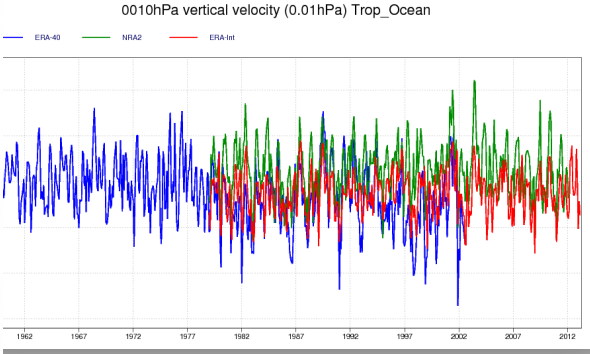
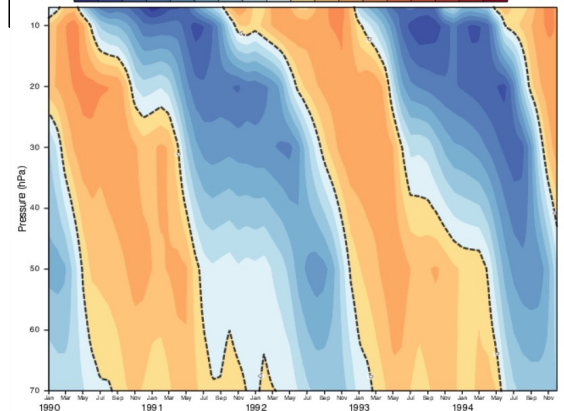
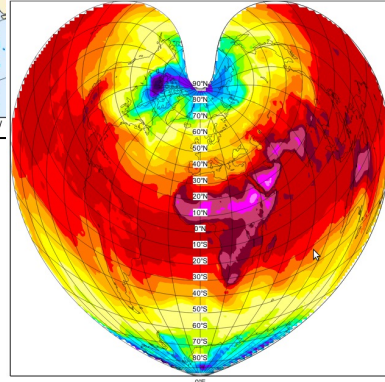
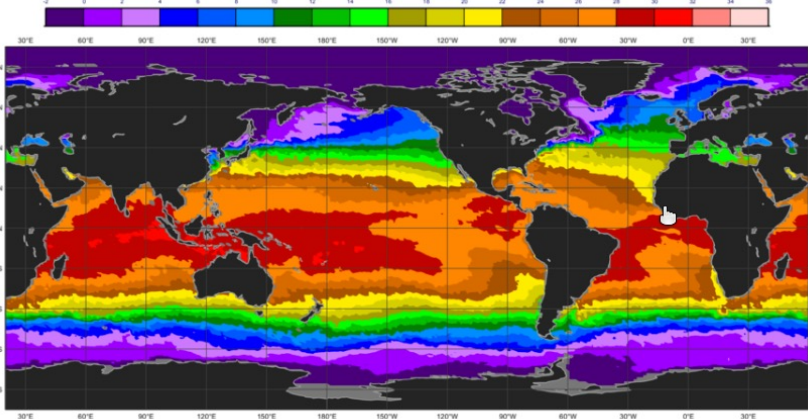
Tuesday 05 March 2019 12 UTC ecmf t+0 VT: Tuesday 05 March 2019 12 UTC 1000 hPa Temperature



Wed 1998 12 UTC ecmf t+12 VT: Wednesday 18 March 1998 00 UTC surface Convective precipitation

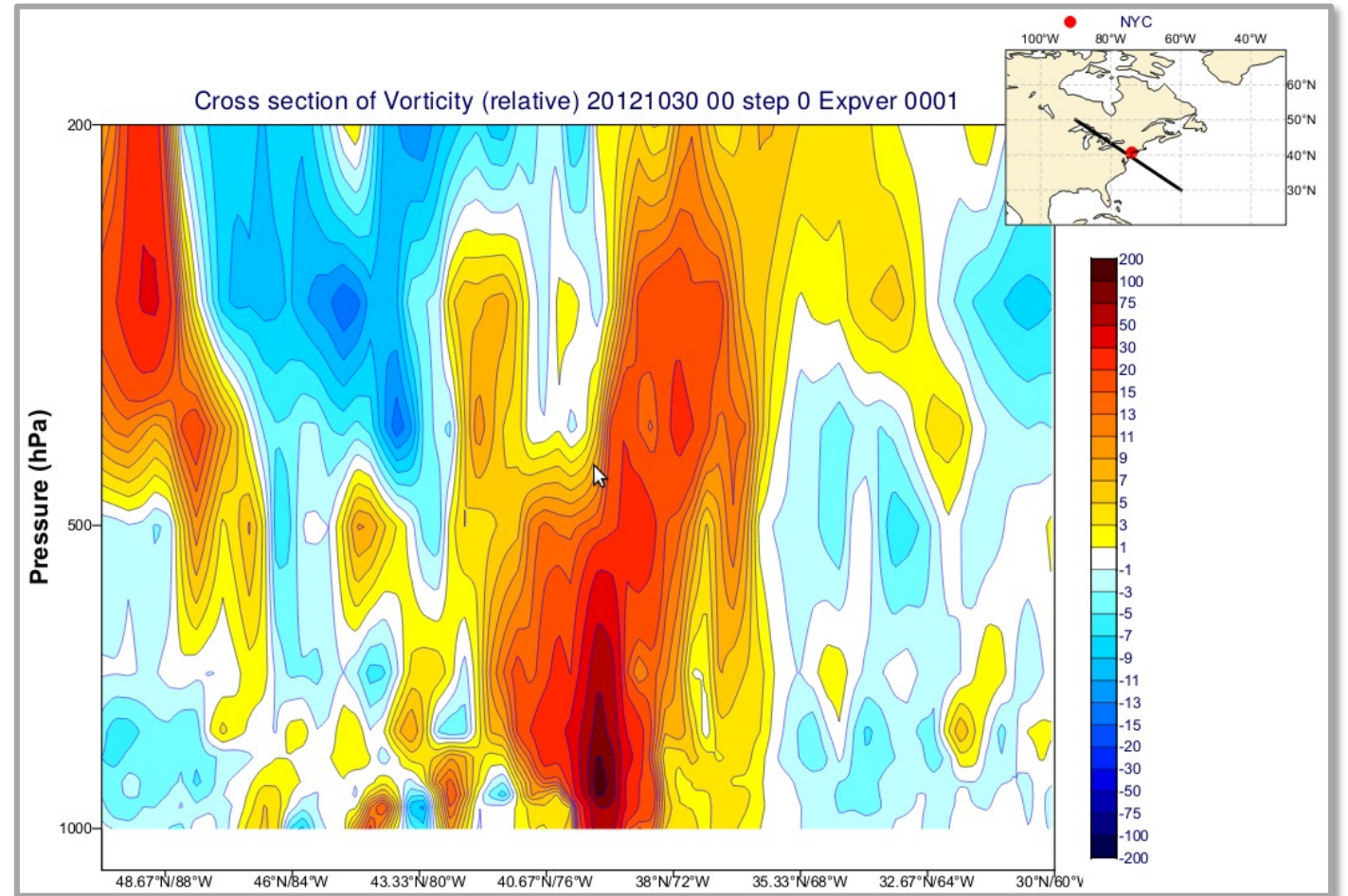
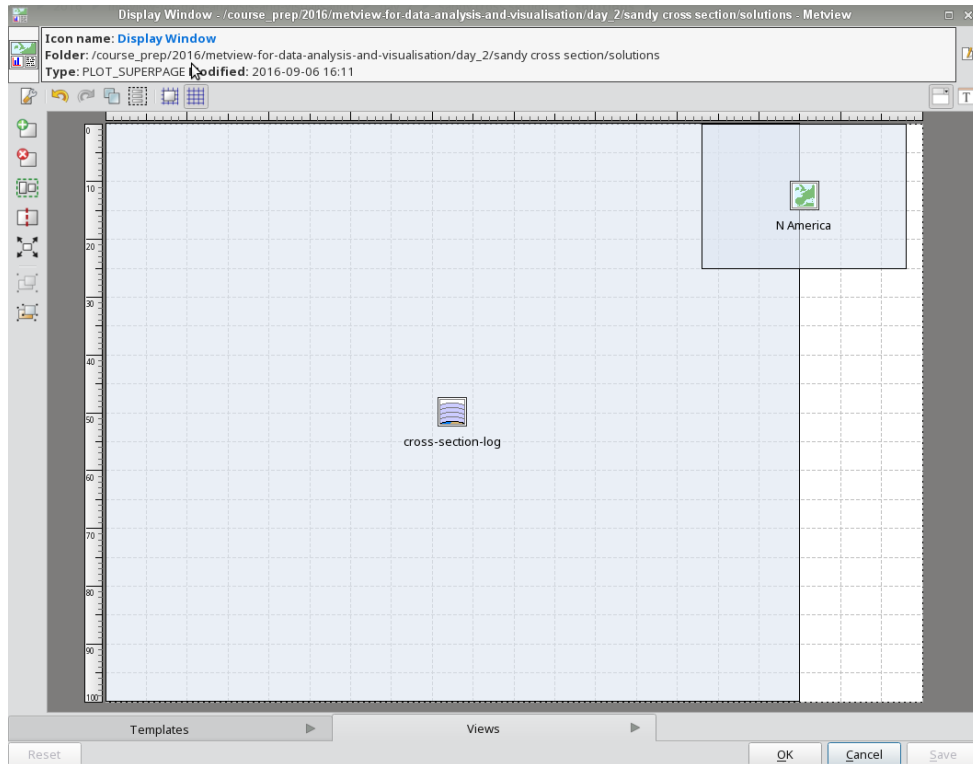


Tuesday 26 February 2019 12 UTC ecmf t+0 VT: Tuesday 26 February 2019 12 UTC surface Sea surface temperature



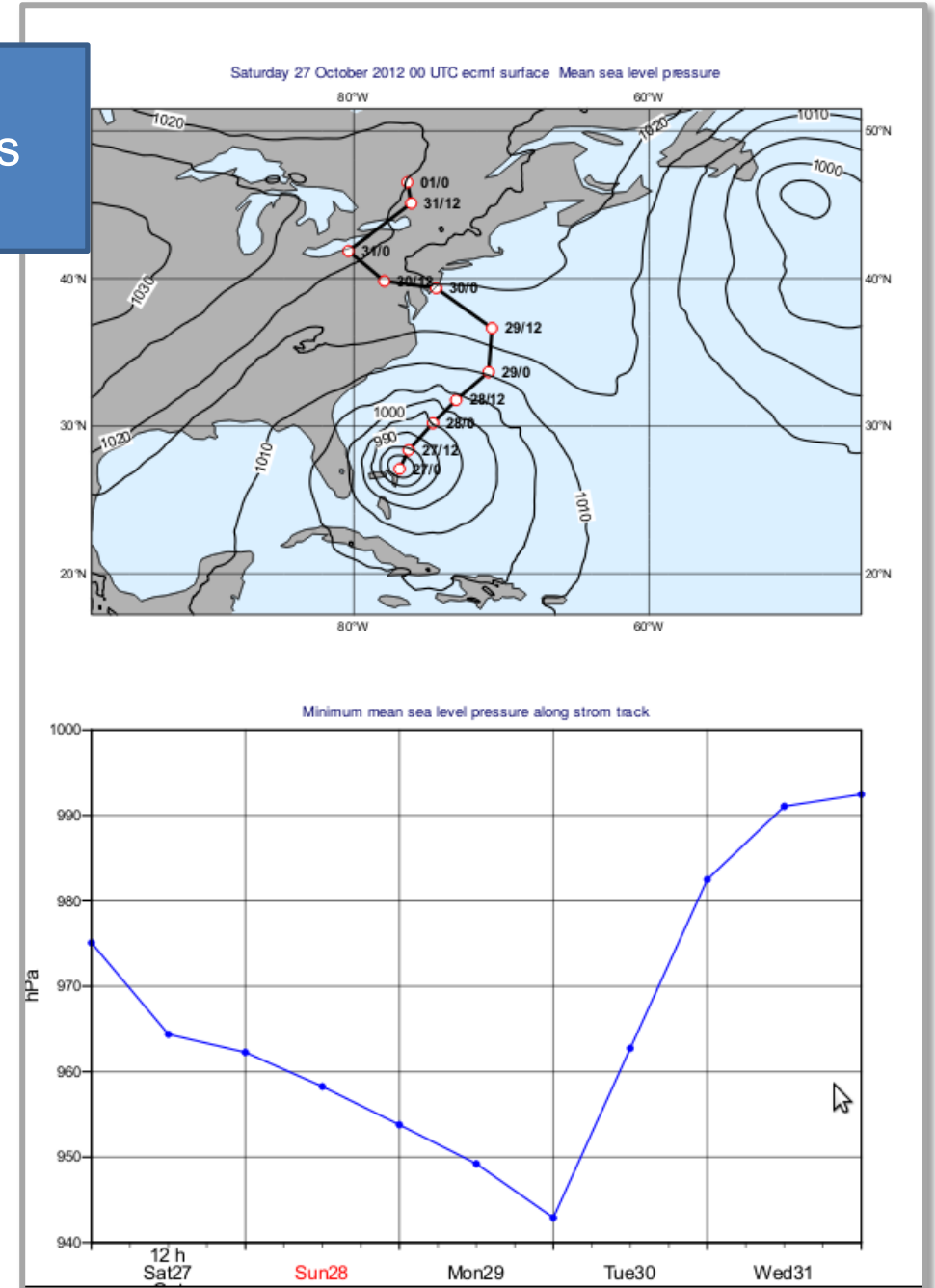
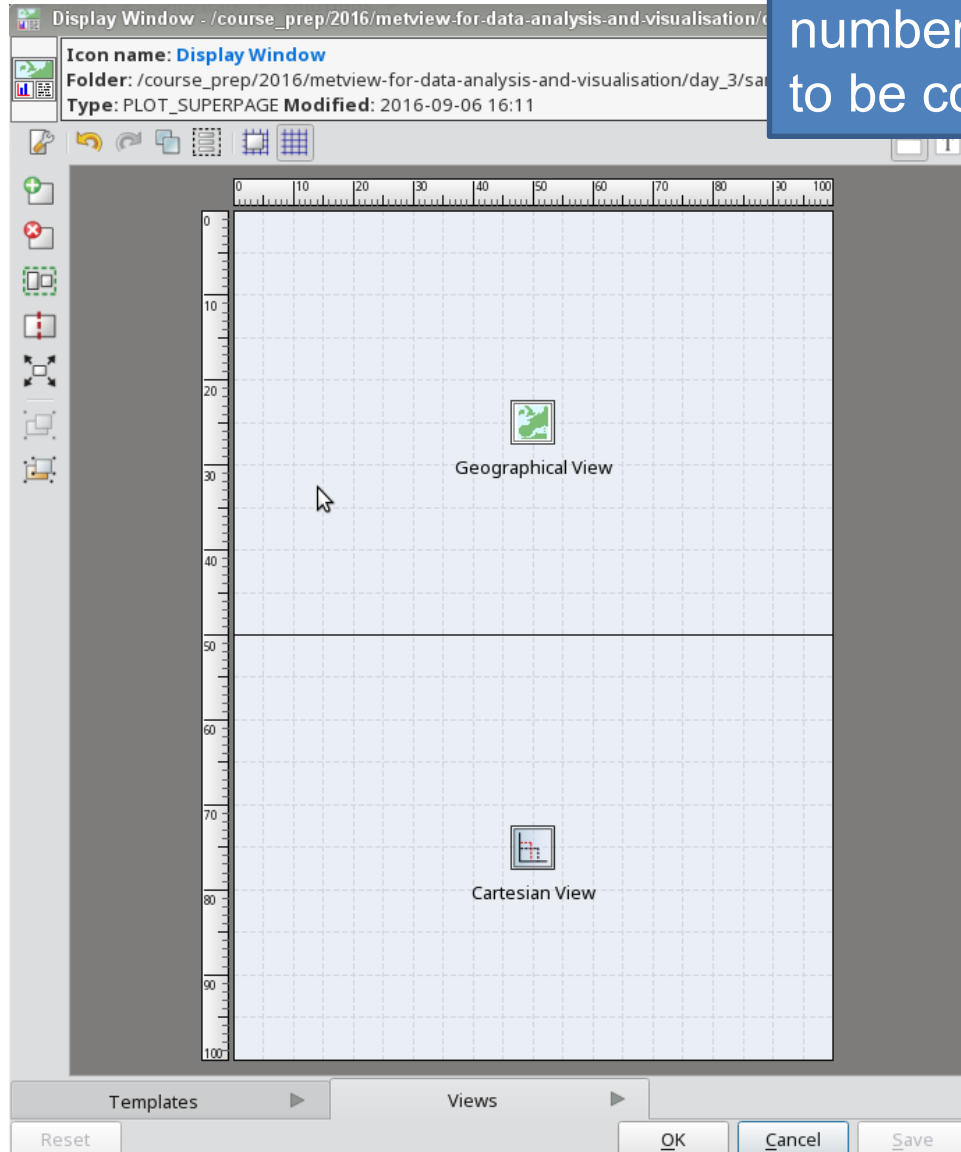
Visualisation - Layout

Layout editor allows any number of different views to be combined



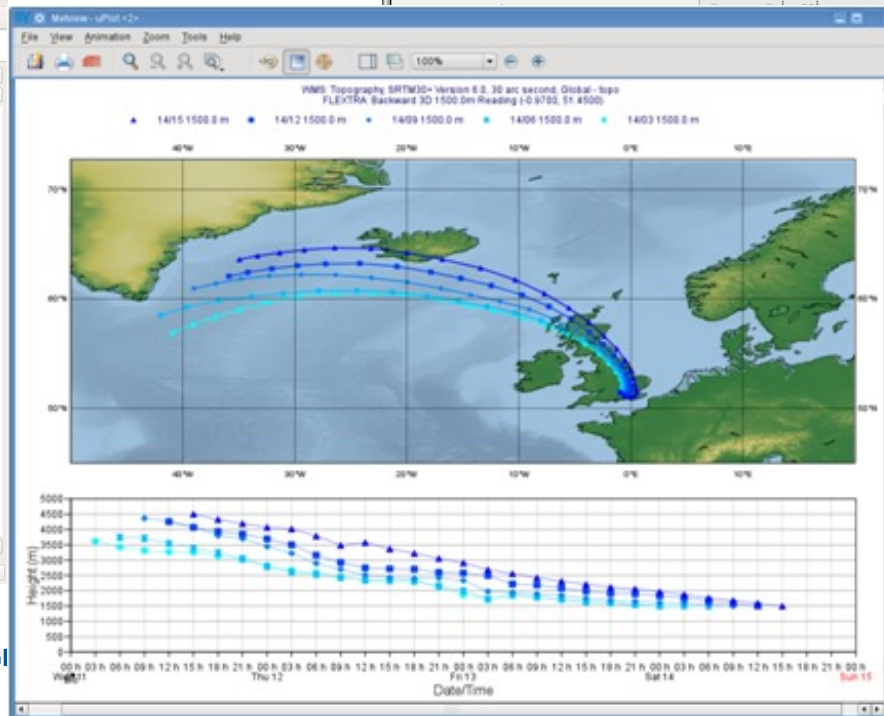
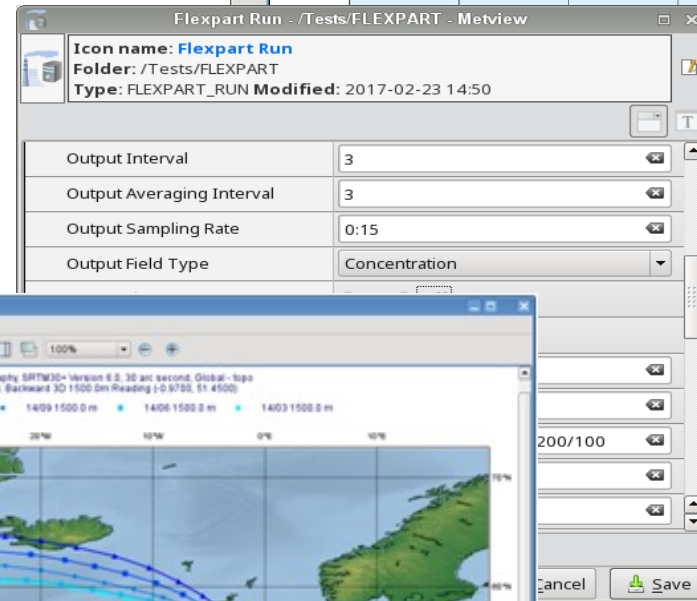
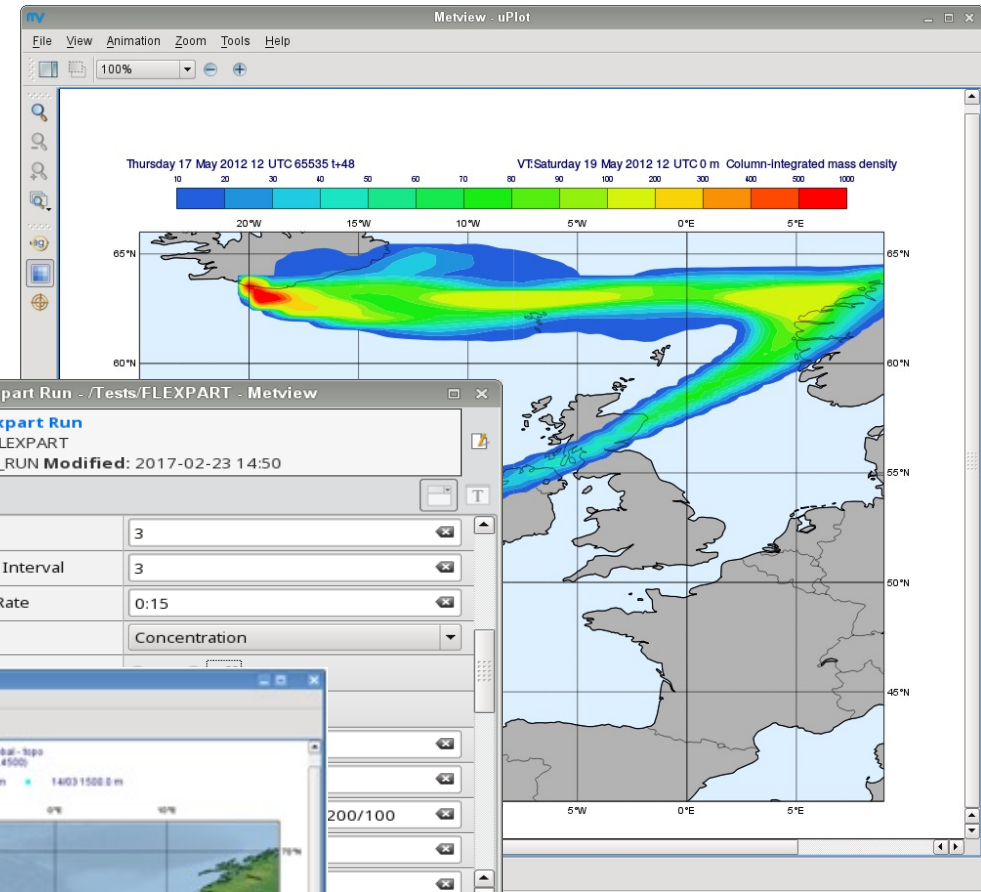
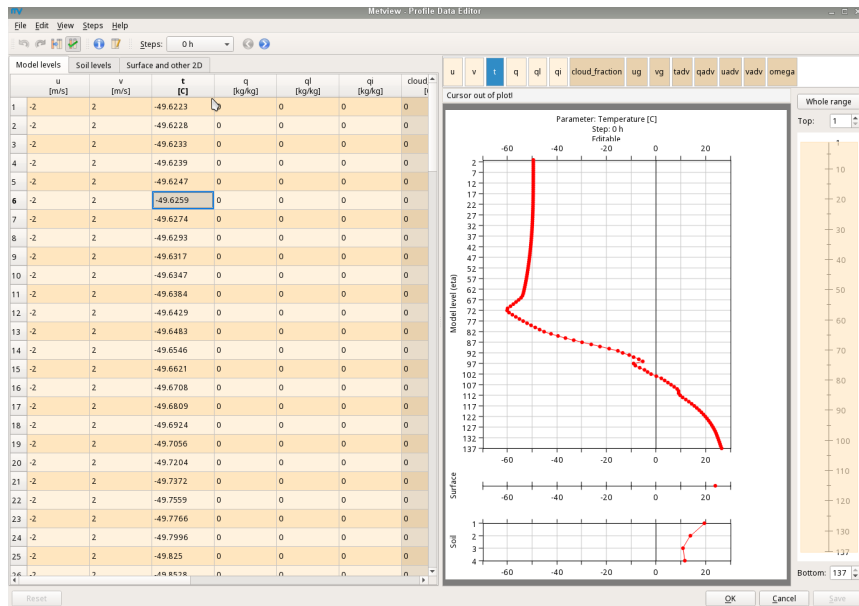
Visualisation - Layout

Layout editor allows any number of different views to be combined



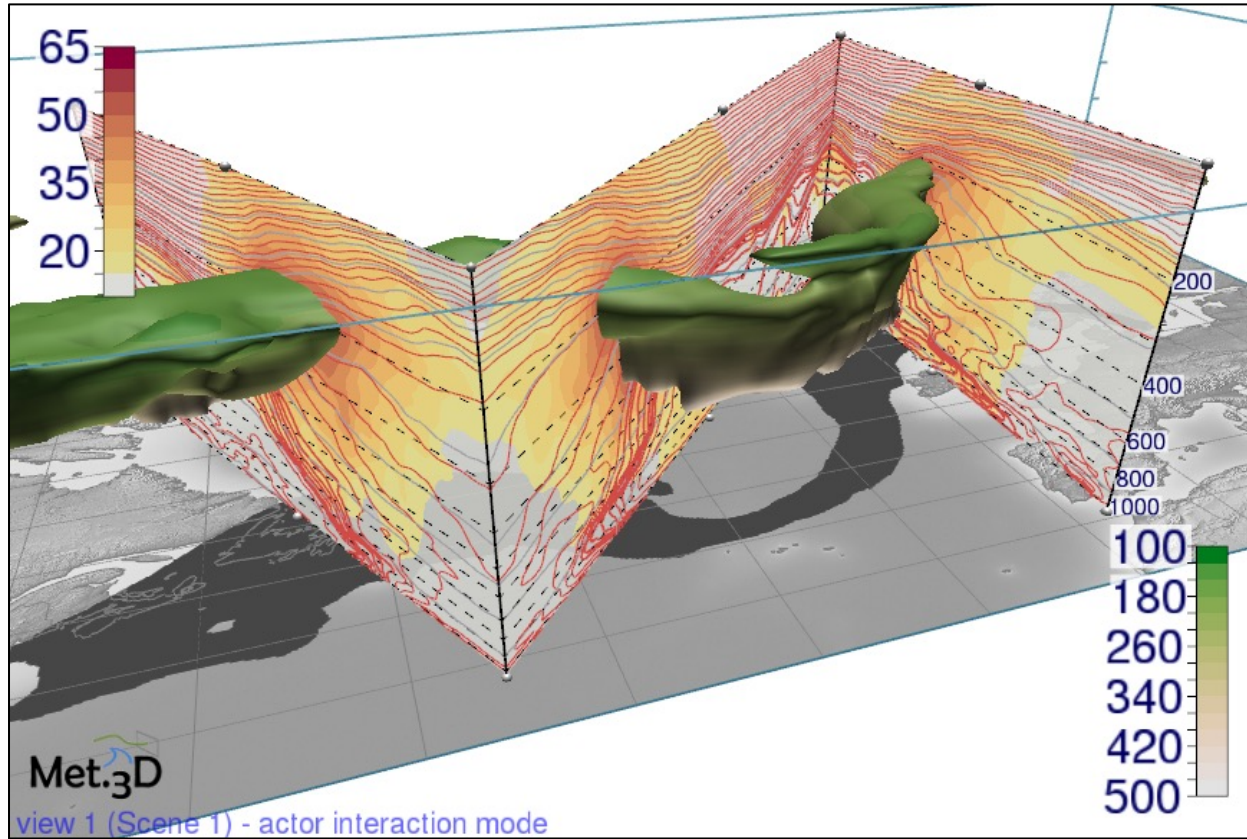
Running models

- Metview is able to prepare data for, run, and plot output from:
- FLEXTRA (trajectory)
- FLEXPART (particle dispersion)
- SCM (Single Column Model)

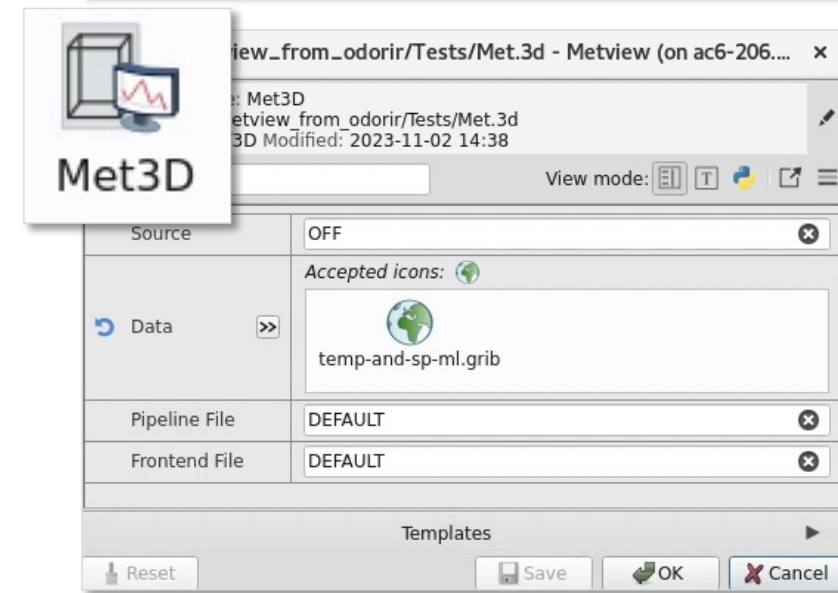
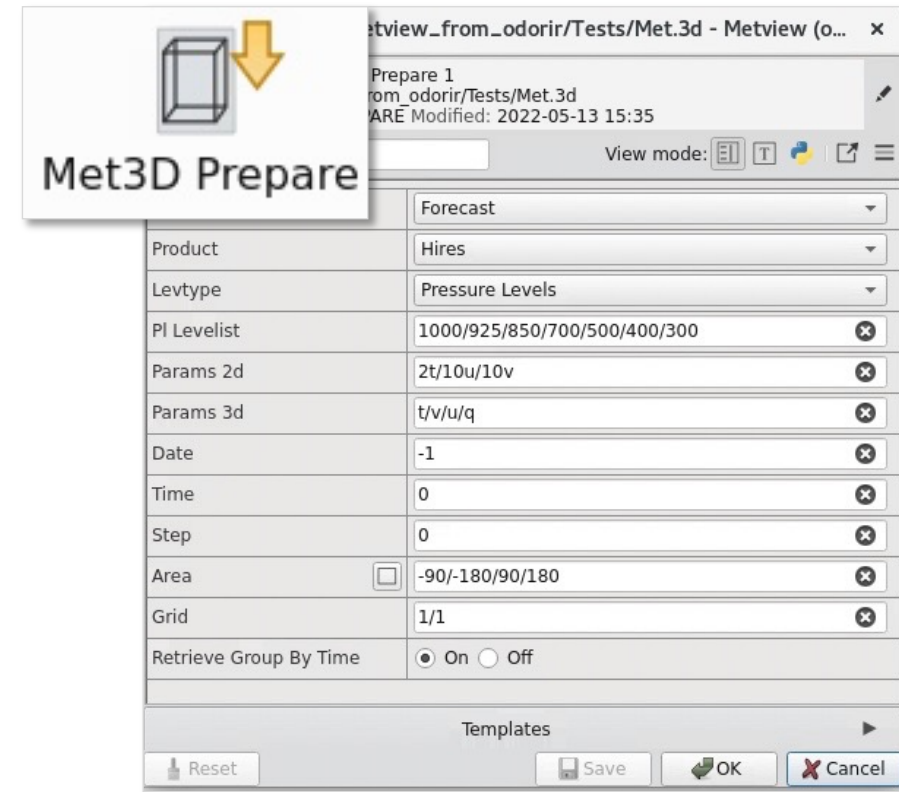


3D

- Metview can prepare data for, and launch Met.3D



Imagery produced by Met.3D (met3d.wavestoweather.de)



Python API



- Powerful, high-level scripting library
- Native handling of major data types
- Some nice features, e.g. model-obs differences (gridded minus scattered data)
- Can extract numpy, pandas and xarrays from Metview's data types

```
# filter out a timestep
wg = mv.read(data=g, step=78)

# create mask (with 0s and 1s) for windgust > 22 m/s
wg_mask = wg > 22

# compute probability
prob = mv.mean(wg_mask) * 100

# define contour shading
cont = mv.mcont(
    legend="on",
    contour_line_colour="charcoal",
    contour_highlight="off",
```

<code>w_from_omega()</code>	Computes the vertical velocity in m/s from pressure velocity
Thermodynamics	
<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

lifted_condensation_level

`lifted_condensation_level(t, td, p)`

Computes the Lifted Condensation Level (LCL) for a parcel ascending from a given temperature, dewpoint and pressure.

- Parameters**
- `t` (number, ndarray or `Fieldset`) - initial temperature (K)
 - `td` (number, ndarray or `Fieldset`) - initial dew point temperature (K)
 - `p` (number, ndarray or `Fieldset`) - initial pressure (Pa)

Return type dict or None

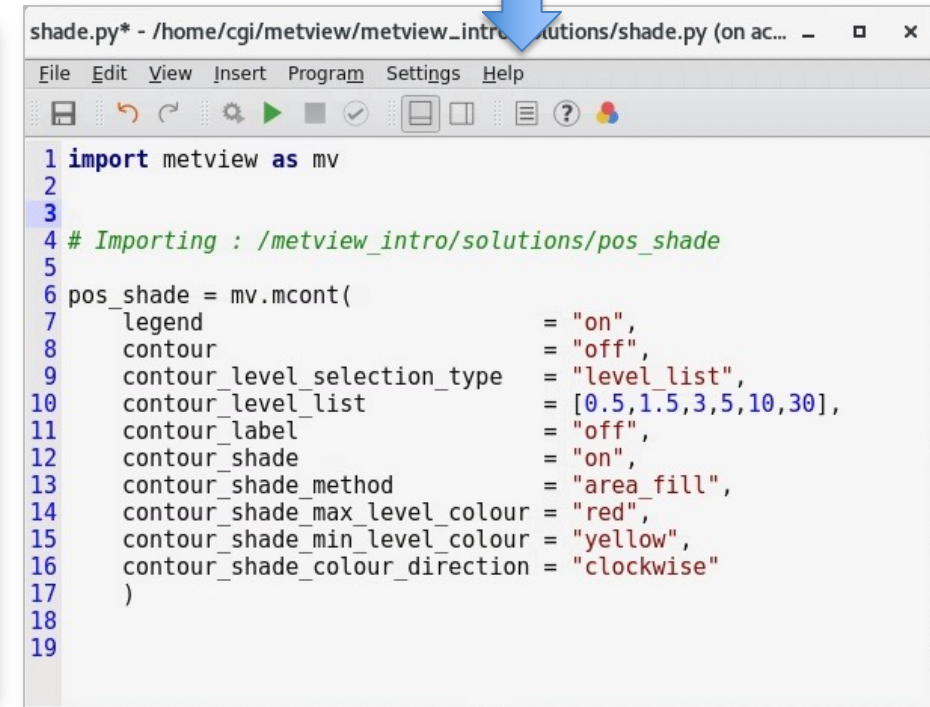
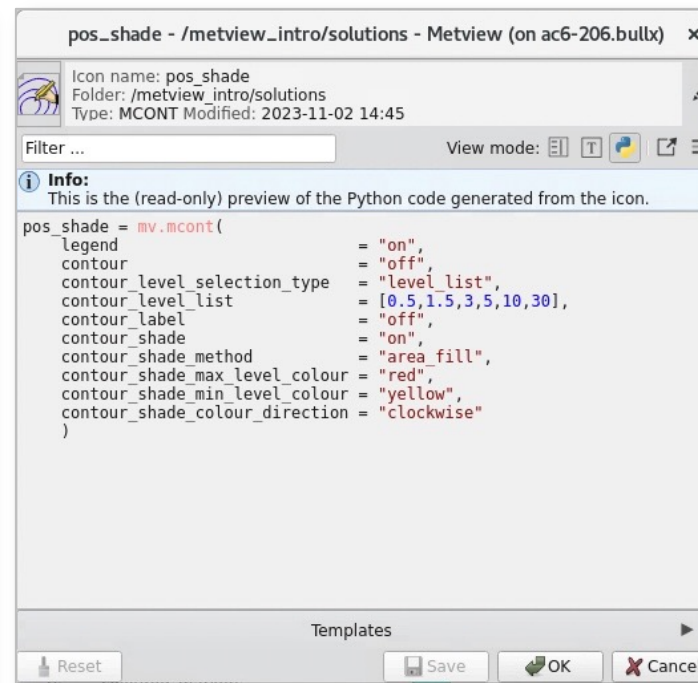
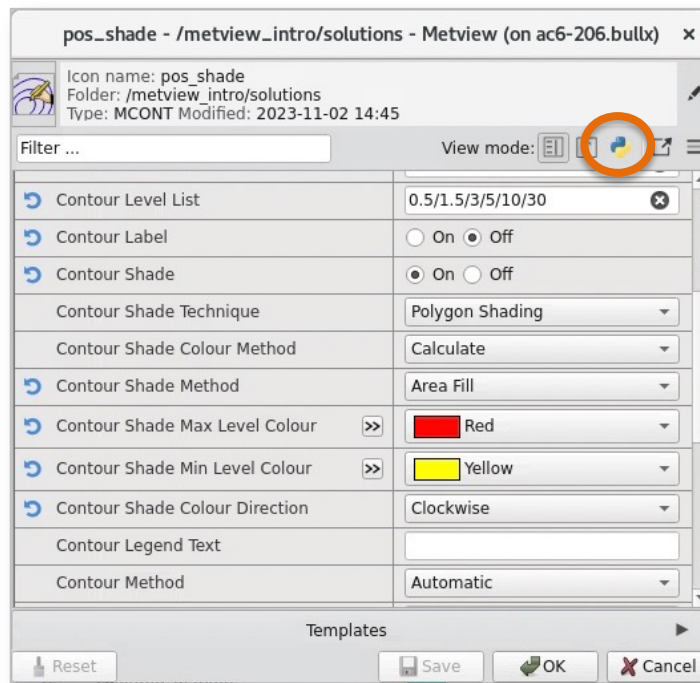
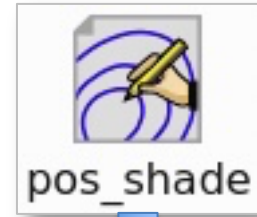
The LCL is the level where the parcel becomes saturated during an adiabatic ascend. First, the LCL temperature is computed with the formula from [Bolton1980](#):

$$t_{LCL} = 56.0 + \frac{1}{\frac{1}{td-56} + \frac{\log(\frac{t}{td})}{800}}$$

Python API



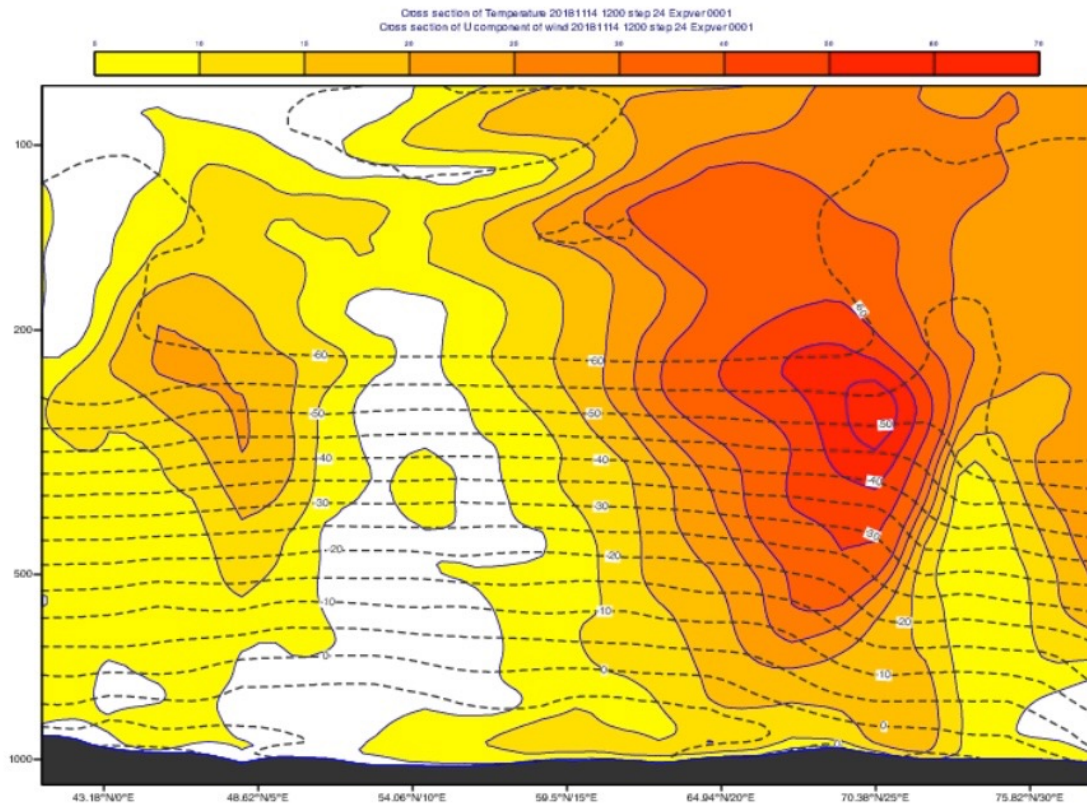
- Every icon has a corresponding Python function call
- Two ways to generate Python code from icons:
 - 1) click on the Python icon inside an icon editor, then copy & paste the code from there
 - 2) drop an icon into Metview's Code Editor



Examples

- See the Gallery for Python examples

Cross Section for Model Level Data with Orography



GRIB - Polygon Masking

GRIB - Shapefile Masking

GRIB - De-accumulate Precipitation Forecast

GRIB - ERA5 SST El Nino Maps

The gallery displays various meteorological maps and plots. It includes four small maps showing different regions, a larger map of Europe with a color scale, and several maps showing precipitation forecasts and SST anomalies. The ERA5 SST El Nino Maps show a global view of sea surface temperature anomalies.

```
# compute divergent wind from spectral divergence. The result is
# still spectral (T255)
d = mv.read(data=g, param="d")
dwind_sh = mv.divwind(data=d, truncation=255)

# transform spectral divergent wind into gridpoint space (reduced)
dwind = mv.read(data=dwind_sh, grid=218)

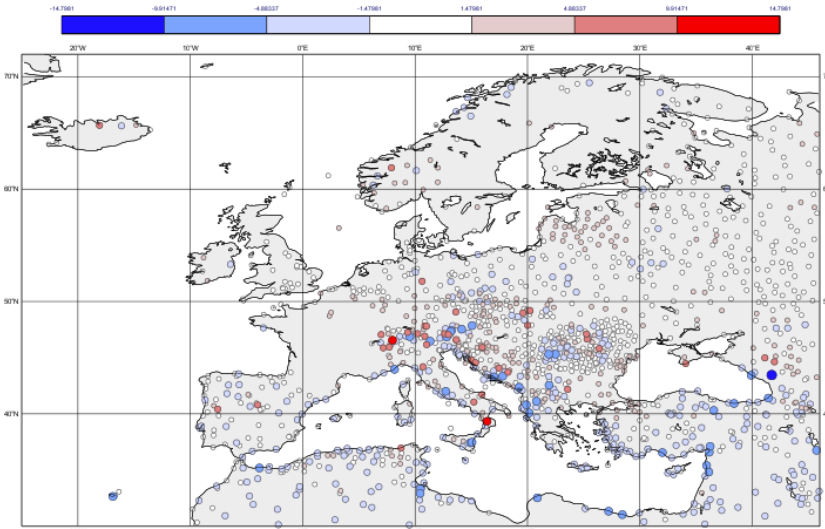
# compute rotational wind from spectral vorticity. The result is
# still spectral (T255)
vo = mv.read(data=g, param="vo")
rwind_sh = mv.divrot(data=vo, truncation=255)

# transform spectral rotational wind into gridpoint space (reduced)
rwind = mv.read(data=rwind_sh, grid=218)

# define coastlines
coast = mv.mcoast(
    map_coastline_colour="cream",
    map_coastline_land_shade="on",
    map_coastline_land_shade_colour="RGB(0.2,0.2,0.2)",
    map_coastline_sea_shade="on",
    map_coastline_sea_shade_colour="RGB(0.3,0.3,0.3)",
    map_grid_colour="RGB(0.9333,0.9333,0.9333)",
)
```

Examples

- See also the Jupyter Notebooks



Out [14]:
We can easily convert this to a pandas dataframe for further analysis.

In [15]:

```
df = diff.to_dataframe()
```

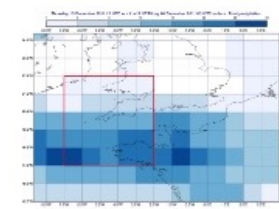
Print a summary of the whole data set:

In [16]:

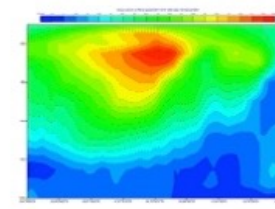
```
df.describe()
```

Out [16]:

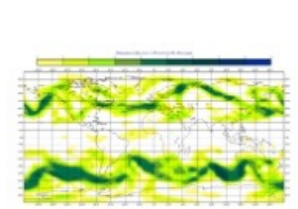
	latitude	longitude	value	level
count	1471.000000	1471.000000	1471.000000	1471.0
mean	46.557104	21.160707	-0.201723	0.0
std	8.350950	14.272239	2.417394	0.0
min	30.110000	-22.590000	-10.236664	0.0



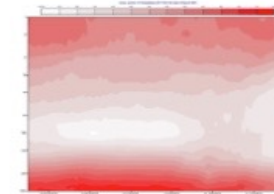
Advanced regridding of data



Analysing data (GRIB)



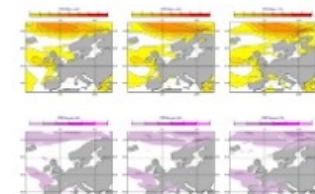
Computing and plotting wind speed (GRIB)



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

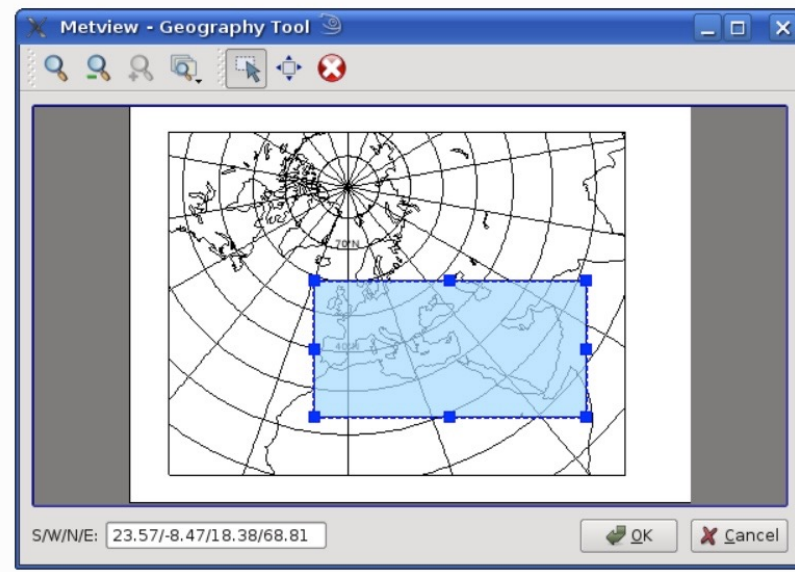
Tutorials

- Plenty of material online including tutorials

Now we want to set the area used in the view. Although we can interactively zoom into smaller areas in the **Display Window**, we now want to store a particular area so that we can use exactly the same one again and again. Set the **Map Area Definition** to **Corners** and click on the **Geography Tool** button next to the **Area** parameter (shown in the picture below).



This tool helps you define a region.

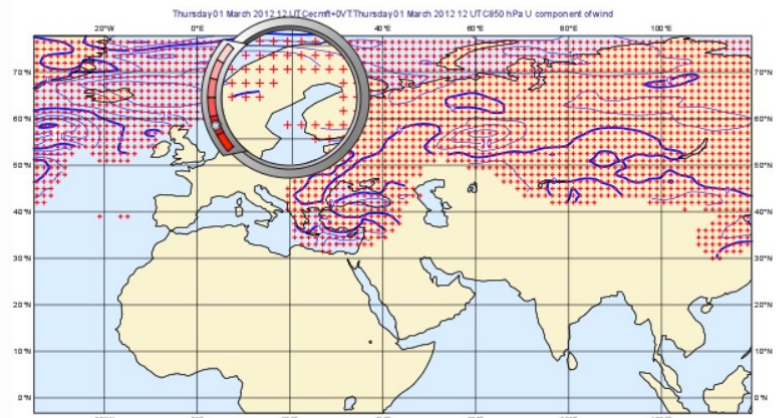


Tutorials on specific topics

- Webinar (2022): [Slicing and dicing GRIB data](#) and associated notebook [Slicing and dicing GRIB data](#)
- Webinar (2021): [Stop, think, interpolate!](#) and associated notebook [Advanced regridding of data](#)
- Workshop presentation from UEF 2022: [Interactive data visualisation and pre-processing with ECMWF's Metview software](#)

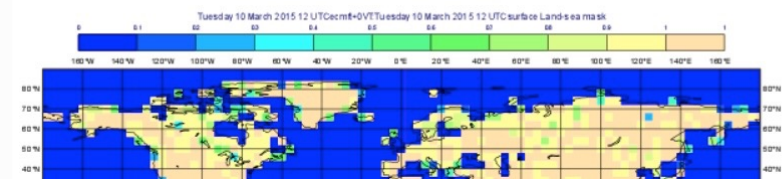
- [BUFR tutorial](#)
- [ODB Tutorial 1](#)
- [ODB Tutorial 2](#)
- [FLEXPART tutorial](#)
- [FLEXTRA tutorial](#)
- [VAPOR Tutorial](#)
- [Metview WMS Tutorial](#)
- [The SCM Interface in M](#)

Overview



Fields and observations can often contain missing values - it can be important to understand the implications of these, and also how to use them to remove unwanted data points. Using a mask of missing values can enable Metview to perform computations on a specific subset of points.

Computing the mean surface temperature over land

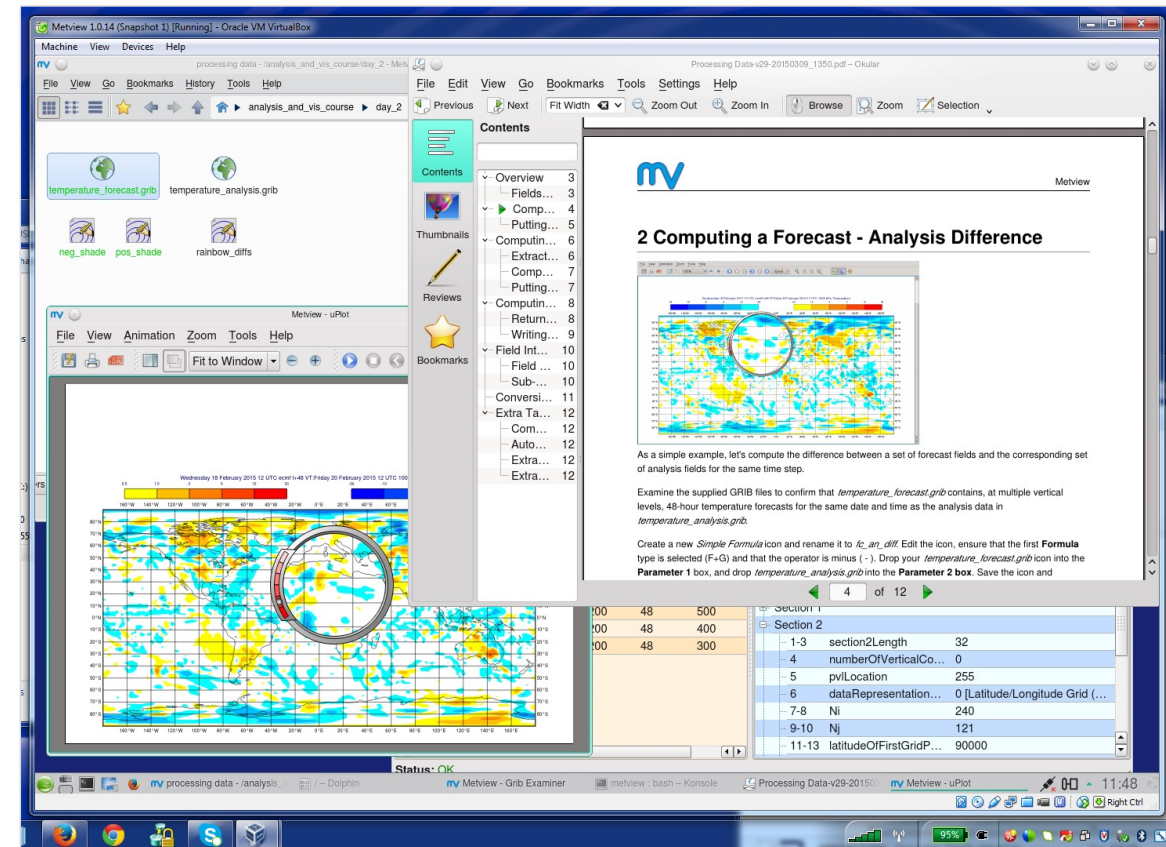


Course modules

- [A Simple Visualisation](#)
- [Customising Your Plot](#)
- [Case Study: Plotting Hurricane Sandy on a Map](#)
- [Data Part 1](#)
- [Processing Data](#)
- [Analysis Views](#)
- [Layout in Metview](#)
- [Case Study: Cross Section of Sandy](#)
- [Data Part 2](#)
- [Handling Time in Metview](#)
- [Graph Plotting in Metview](#)
- [Case study: Plotting the Track of Hurricane Sandy](#)
- [Working with graphical output](#)
- [Organising Macros](#)
- [Missing Values and Masks](#)
- [Optimising Your Workflow](#)
- [Customising Your Plot Title](#)
- [Case study: Ensemble Forecast](#)
- [Running Metview in Batch Mode](#)
- [Working with Folders and Icons](#)
- [Exploring Metview](#)

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`



Exercise (in GatherTown)

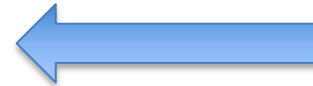
- Find the Metview documentation on ReadTheDocs:
- <https://metview.readthedocs.io/en/latest/index.html>



- Navigate: Tutorials -> [ECMWF New Users Metview Tutorial](#)

Introduction to Metview

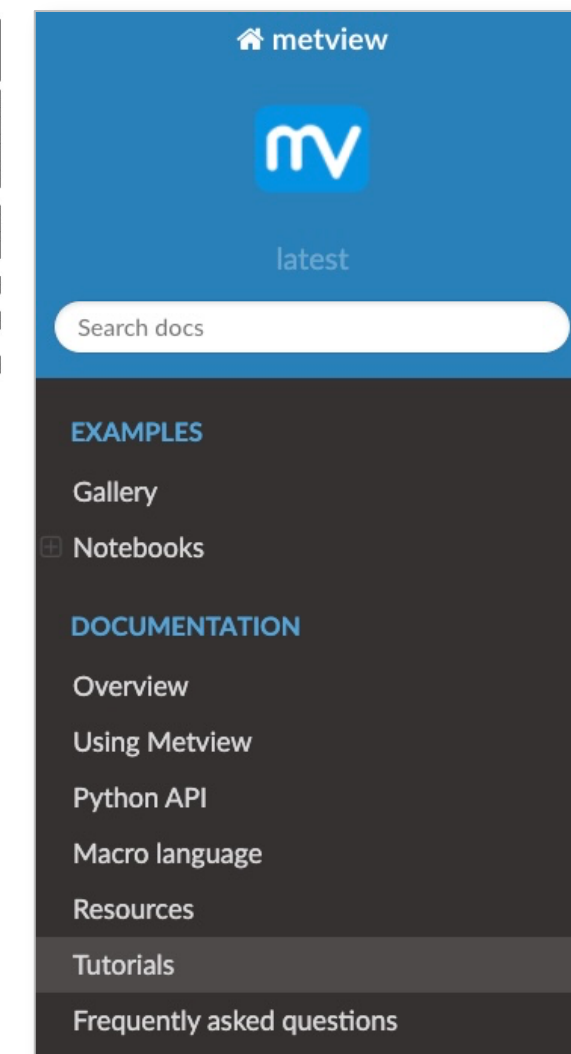
- [A Quick Tour of Metview](#)
- [Metview 90 minute introduction](#)
- [ECMWF New Users Metview Tutorial](#)



ECMWF New Users Metview Tutorial

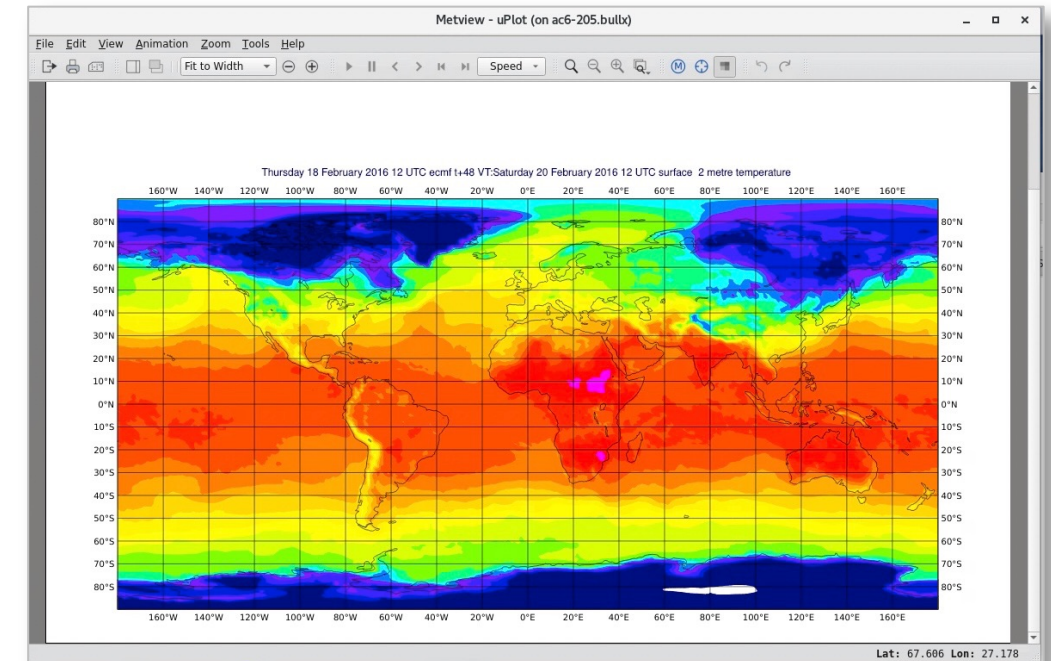
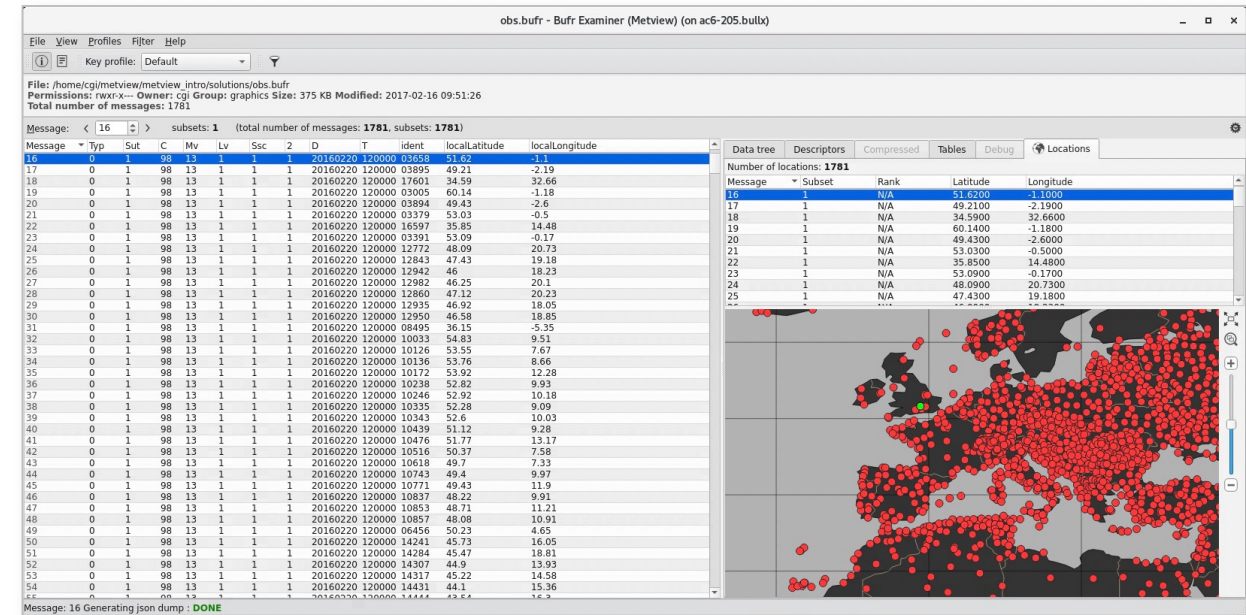
Note

This tutorial was written for ECMWF's Introduction for New Users course (COM_MARS and COM_INTRO) and shows how to retrieve data from MARS using Metview, perform some basic manipulations and plot the result.



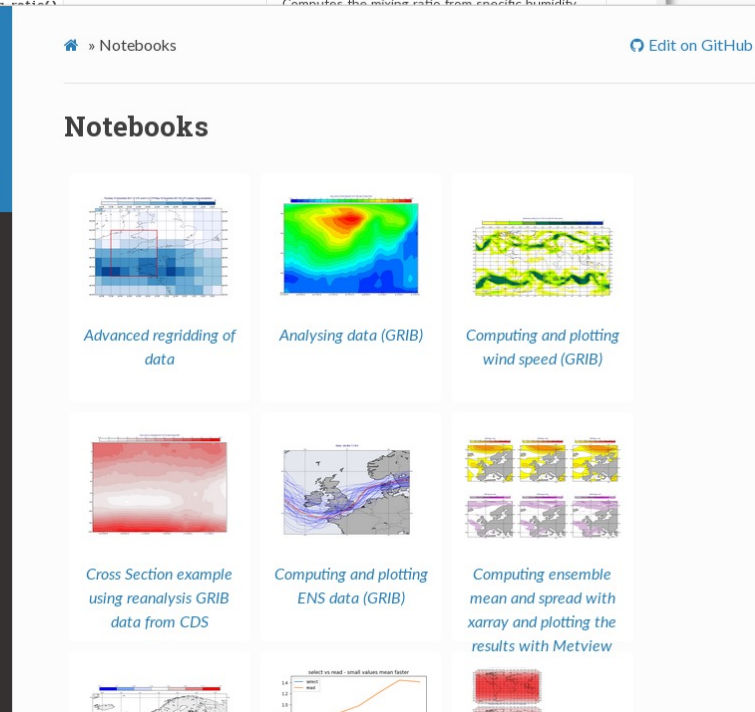
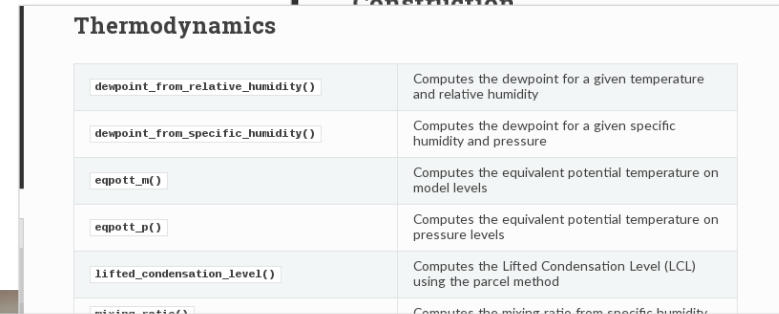
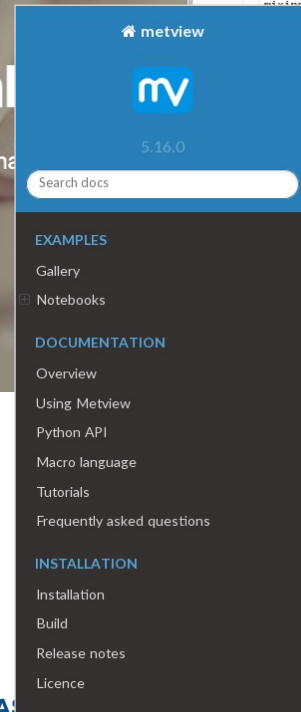
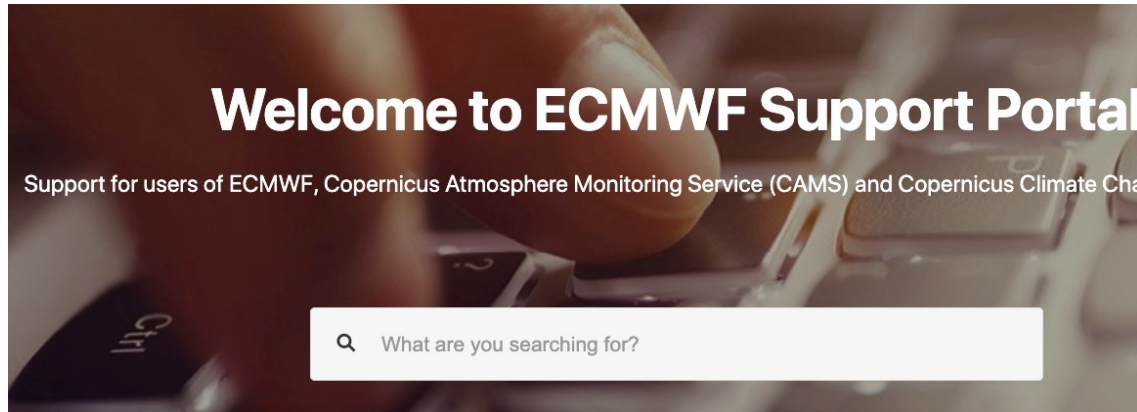
Pro Tips (command-line tools)

- `metview -e <GRIB/BUFR/... file>`
 - Starts up the examiner
- `metview -p <file>`
 - Starts up the plot window with the given data – tries to find the corresponding ecCharts style if available
- `metview -slog`
 - Gives more debug output to stdout
- `metview -u /path/to/new/metview/home`
 - Use another home dir (`$HOME/metview` is default)



For more information...

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





ECMWF's next-generation data-processing toolkit family



earthkit-data

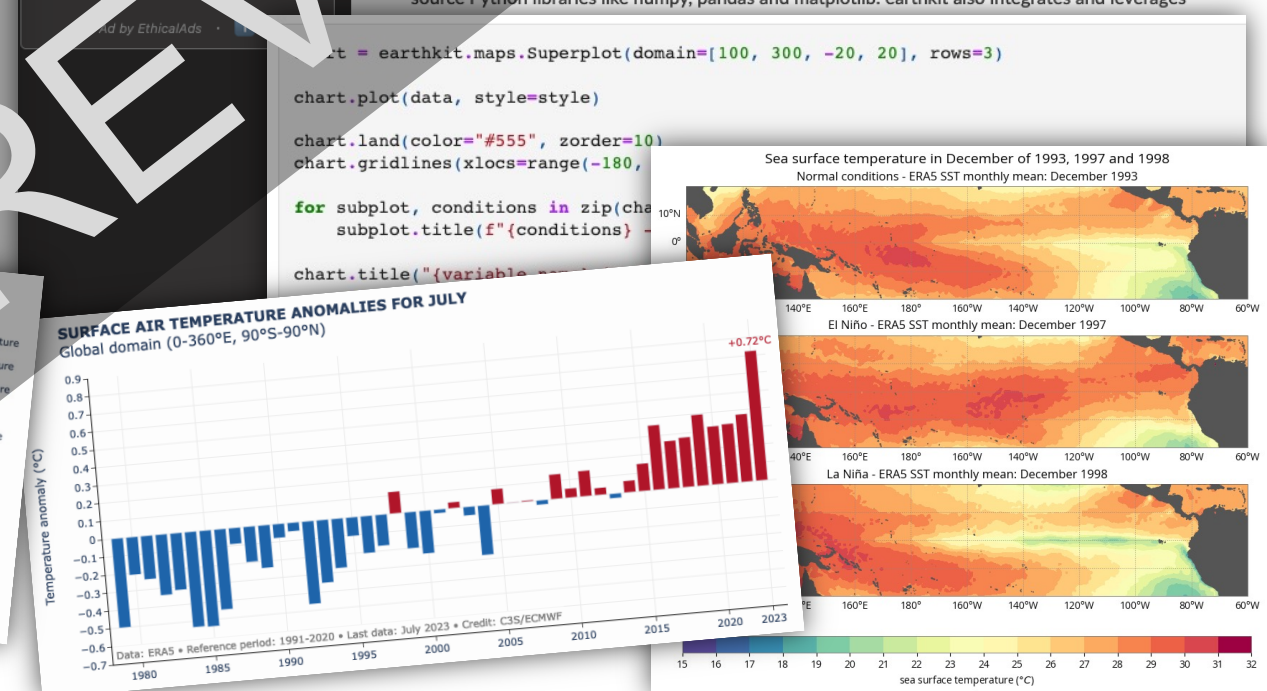
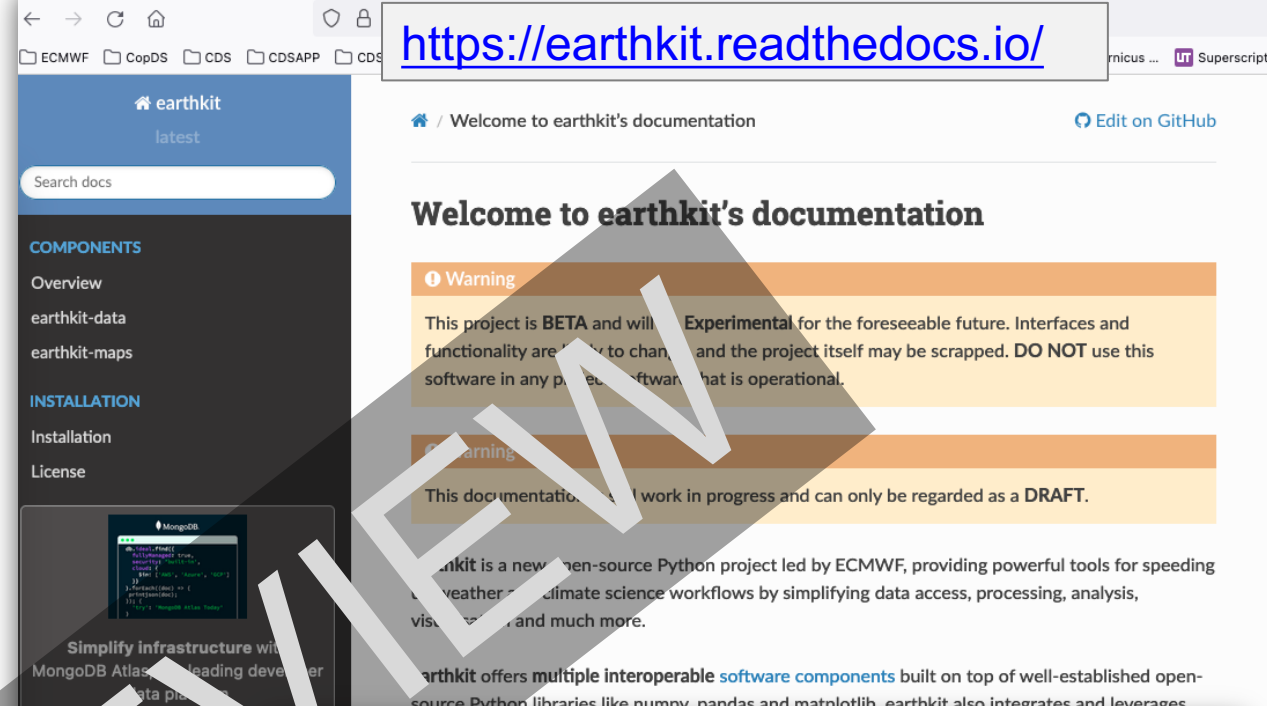
earthkit-plots

earthkit-meteo

earthkit-climate

earthkit-maps

earthkit-regrid



PREVIEW

Finding a range of extreme values

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

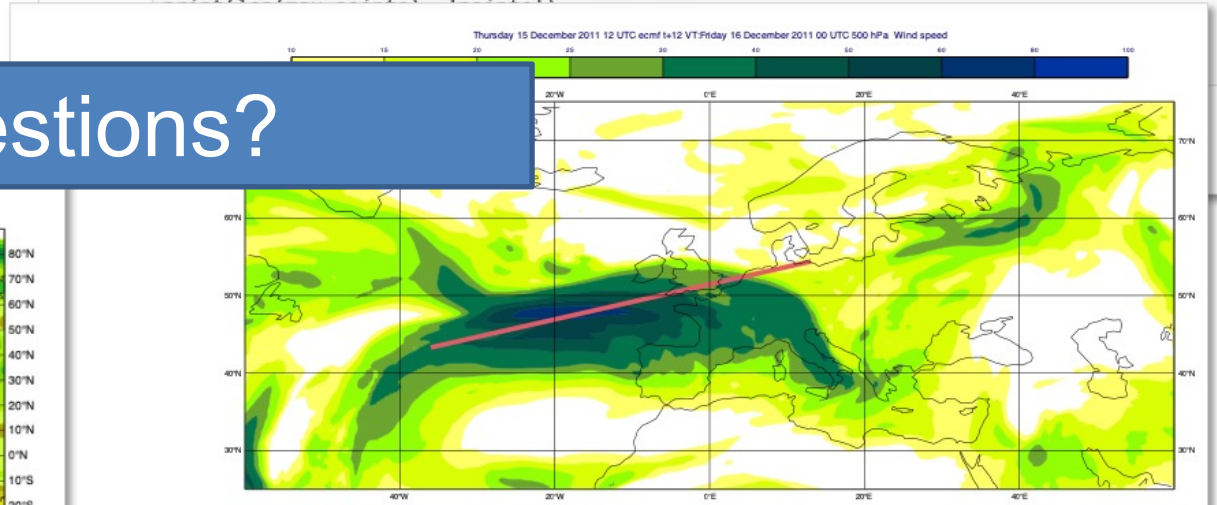
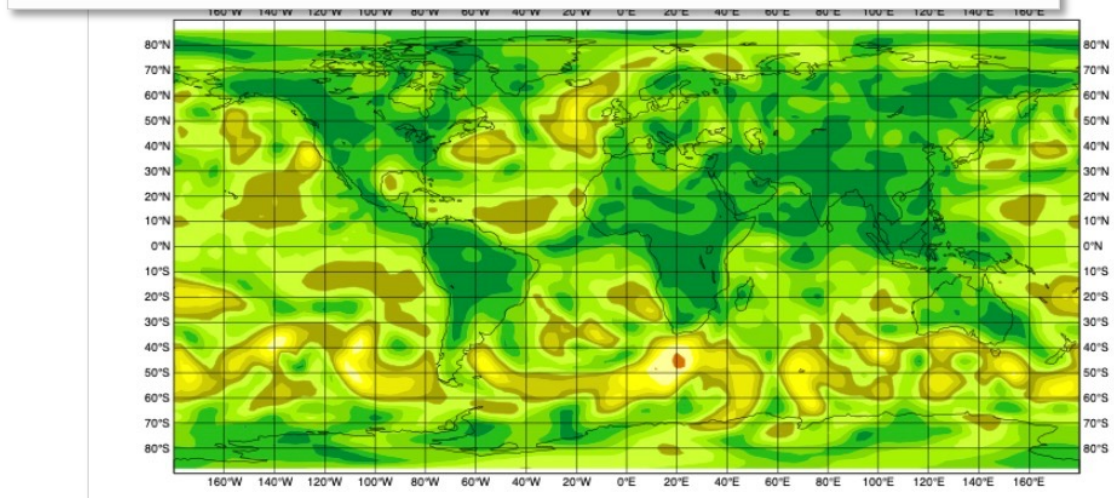
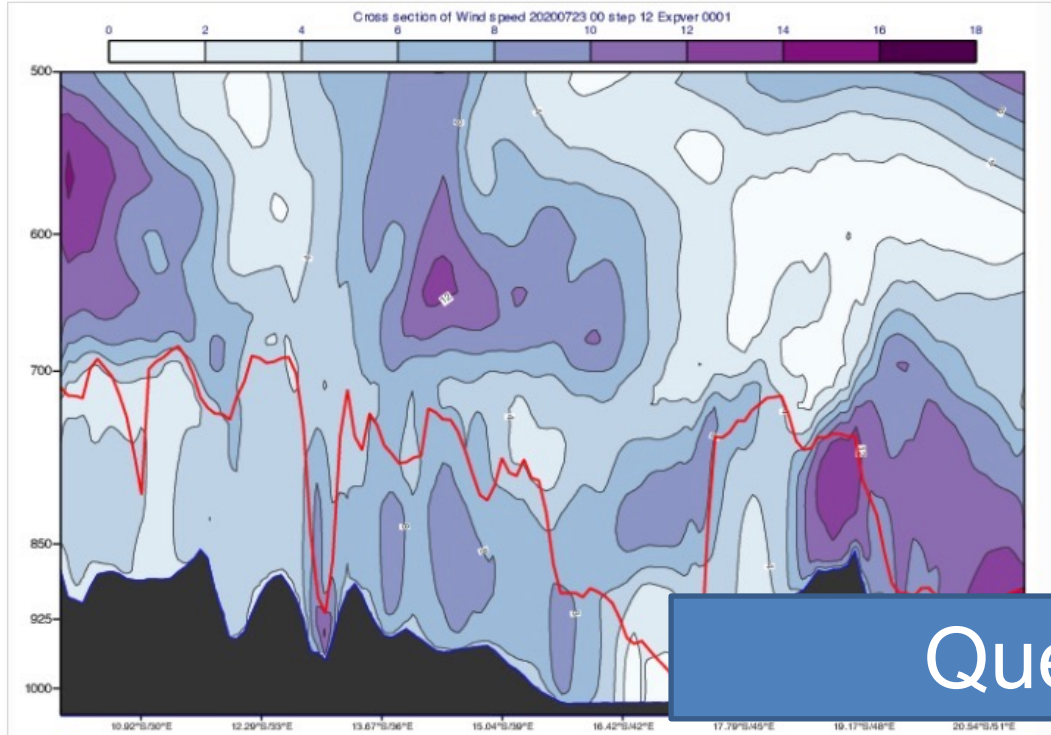
```
[12]: mv.find(wg0, [max0*0.95, max0])
```

```
[12]: [[48.5, -5.0], [48.5, -4.5], [47.0, -7.0], [47.0, -3.5], [47.0, -3.0]]
```

If we want to work with these points in Metview, the easiest way is to use the `gfind()` function to return a `Geopoints` variable:

```
[13]: max_points = mv.gfind(wg0, max0, max0*0.05)
```

Questions?



Define a cross section view along an area of interest

```
n [36]: xs_view = mv.mxsectview(  
    bottom_level = 1000.0,  
    top_level    = 100,  
    line         = line  
)
```

Create a colour scale to plot the data with

```
n [37]: xs_shade = mv.mcont(  
    legend = "on",
```

