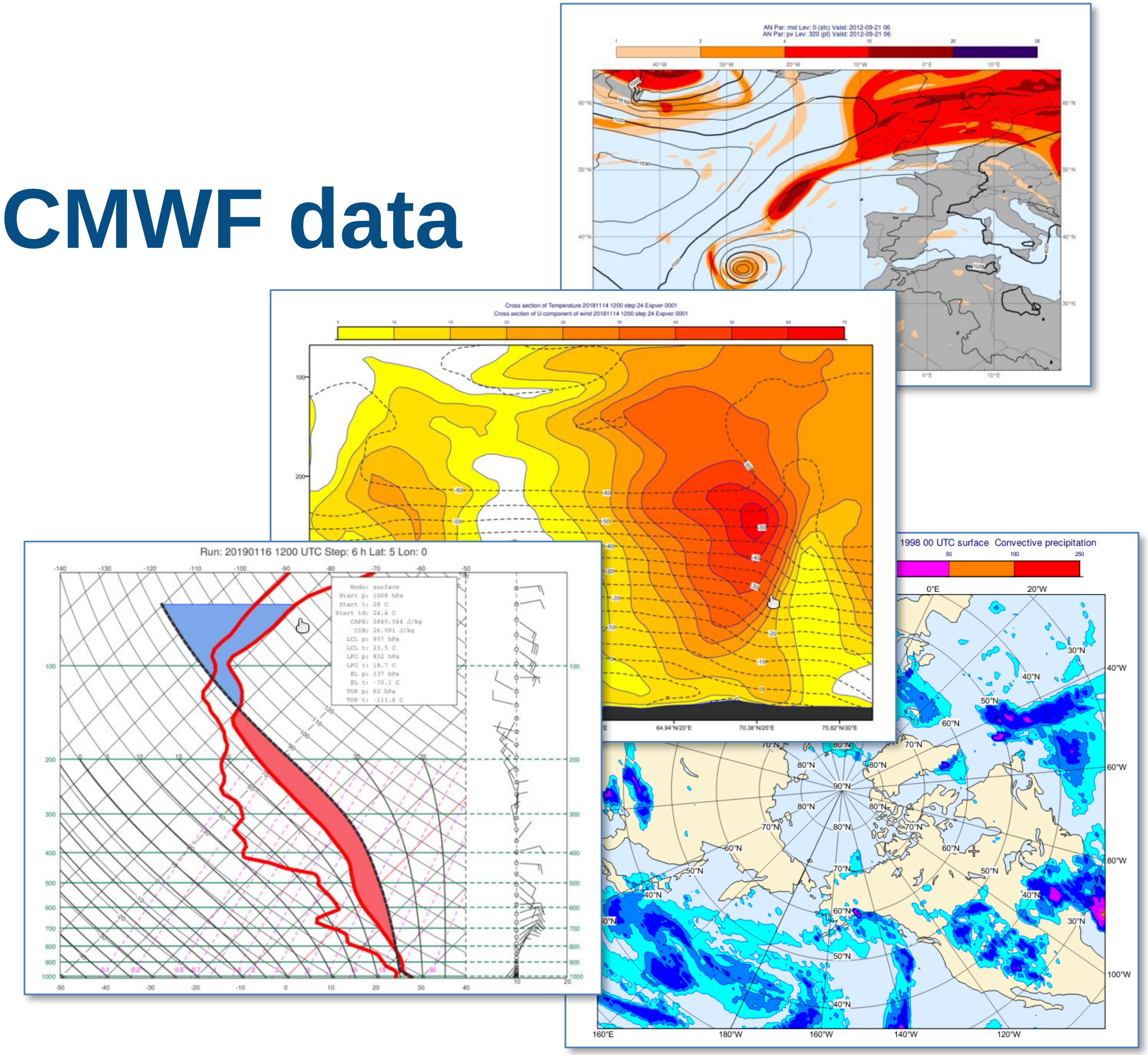
Visualisation of ECMWF data

ECMWF

Iain Russell,
Sándor Kertész

Development Section, ECMWF



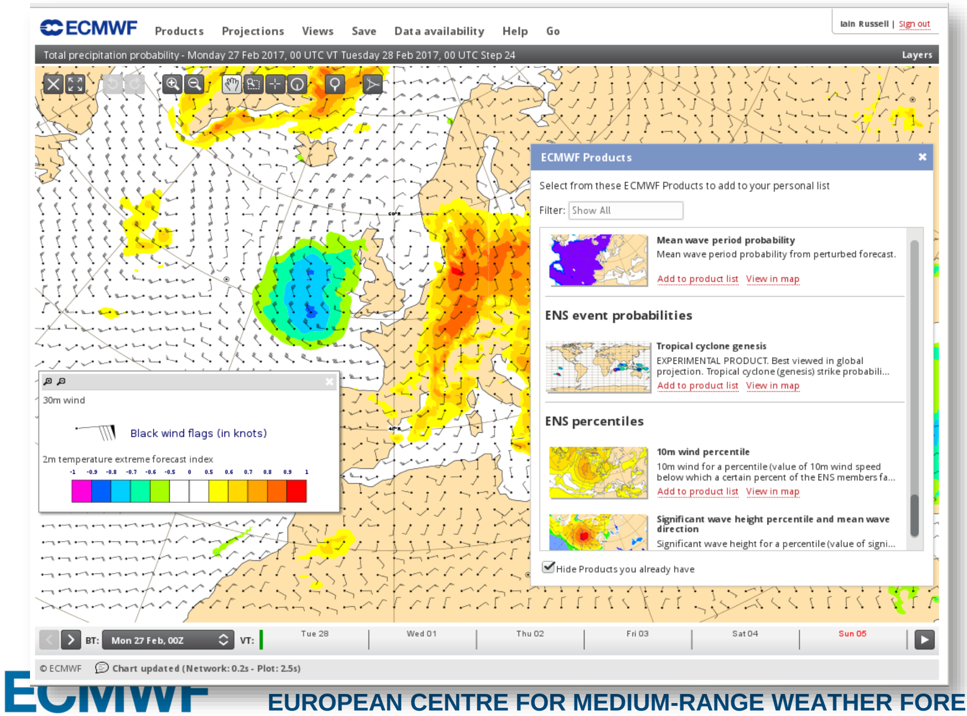
Outline

- Overview of ECMWF visualisation systems
  - ecCharts & Metview
- ecCharts summary
- Getting started with Metview
- GRIB data in Metview
  - Getting data into Metview
  - Styling
  - Comparison of plotting algorithms
  - Using ecCharts styles and layers
- BUFR data in Metview
- Scattered data in Metview
  - Geopoints, CSV, lists of values
- NetCDF and ODB data in Metview
- Map projections, analysis views (e.g. cross section), layout
- Scripting, and using other Python tools
- Where to find out more

ECMWF Visualisation Packages

- ecCharts
  - Runs in a web browser
  - Pre-defined graphical products from ECMWF's recent forecast data
  - Uses Magics for graphics
  - Restricted service

- Metview
  - Desktop application running on Linux or Mac OS X
  - Post-processing and visualisation using any ECMWF data (recent, past or experimental)
  - Uses Magics for graphics
  - Open Source



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

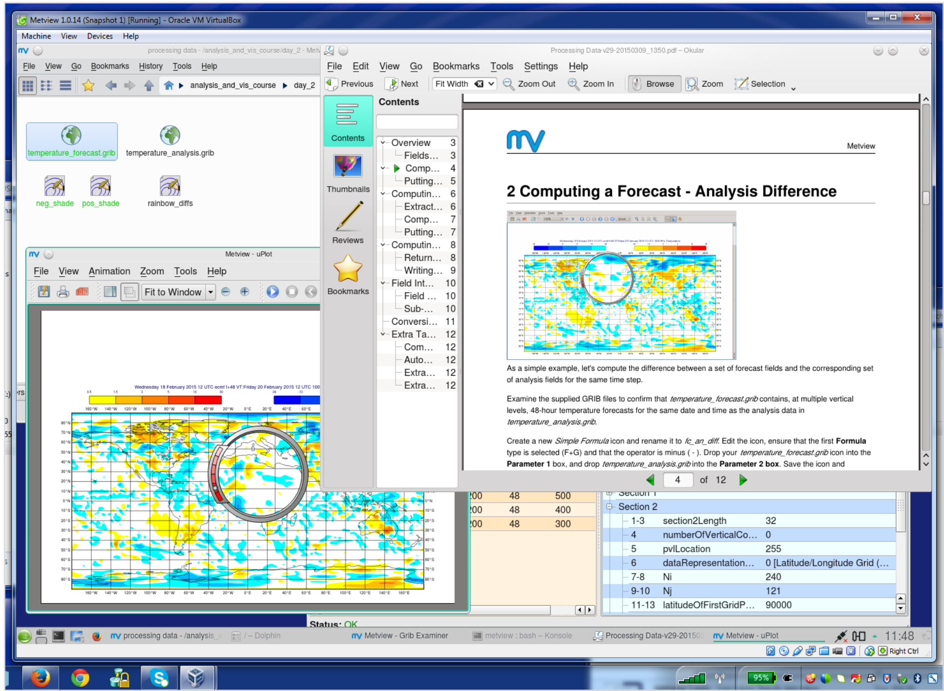
ENS percentiles

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)



2 Computing a Forecast - Analysis Difference

As a simple example, let's compute the difference between a set of forecast fields and the corresponding set of analysis fields for the same time step.

Examine the supplied GRIB files to confirm that *temperature_forecast.grib* contains, at multiple vertical levels, 48-hour temperature forecasts for the same date and time as the analysis data in *temperature_analysis.grib*.

Create a new *Simple Formula* icon and rename it to *fc_an_diff*. Edit the icon, ensure that the first **Formula** type is selected (F+G) and that the operator is minus (-). Drop your *temperature_forecast.grib* icon into the **Parameter 1** box, and drop *temperature_analysis.grib* into the **Parameter 2 box**. Save the icon and

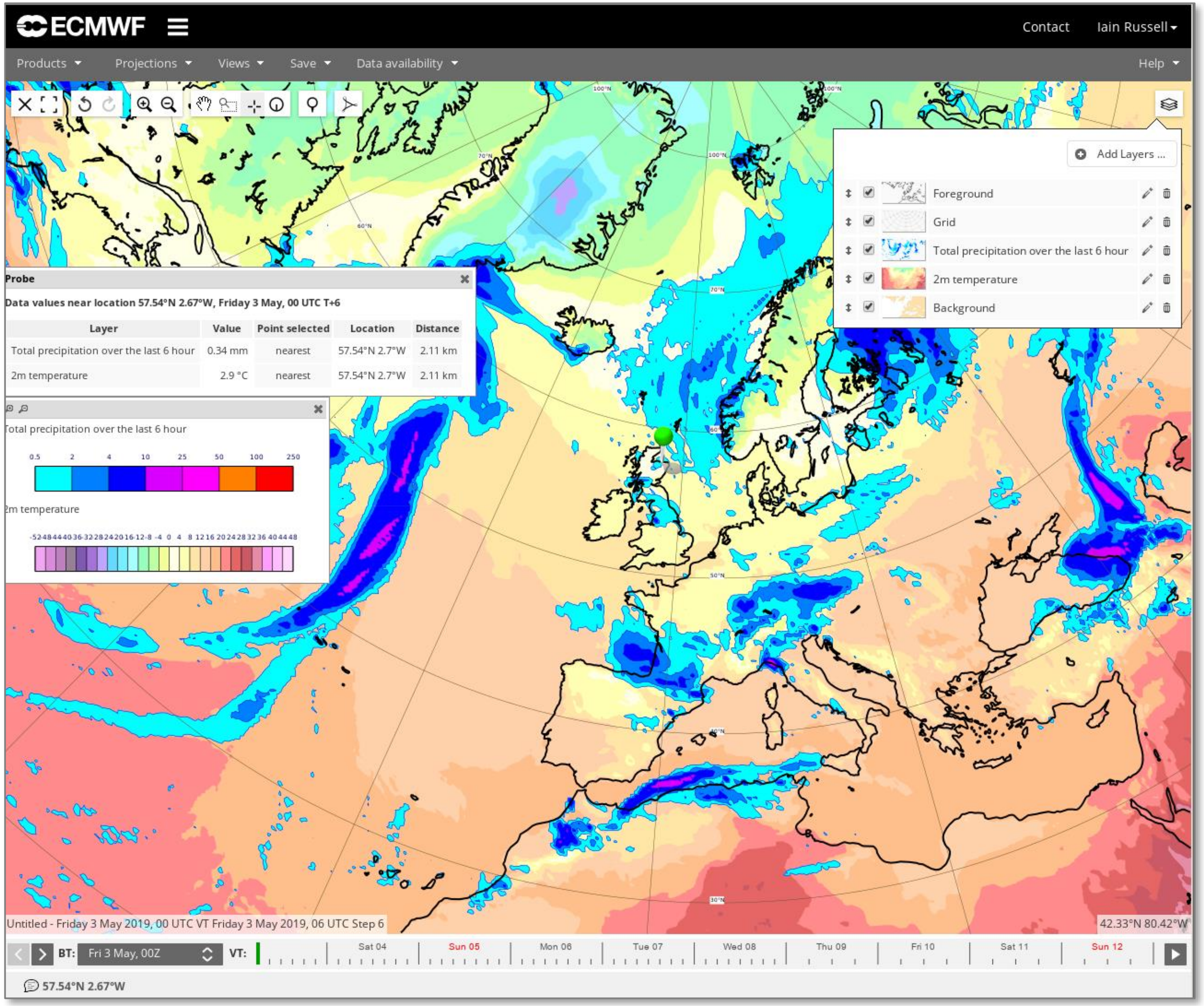
200	48	500
200	48	400
200	48	300

Section 1
Section 2

1-3	section2Length	32
4	numberOfVerticalCo...	0
5	pvlLocation	255
6	dataRepresentation...	0 [Latitude/Longitude Grid (...
7-8	Ni	240
9-10	Nj	121
11-13	latitudeOfFirstGridP...	90000

ecCharts Overview

- Interactive web service for forecasters to view ECMWF forecast products
  - Real-time data (around 250 layers) from medium range (HRES, ENS, wave) and extended range, and Copernicus CAMS service that are updated as soon as they are available from our dissemination system
- Also provides a WMS service so that plots can be embedded in other applications
- *ecCharts is a restricted service that is only available to Member and Cooperating State forecasters and licensed subscribers of ECMWF Web Products*



Probe

Data values near location 57.54°N 2.67°W, Friday 3 May, 00 UTC T+6

Layer	Value	Point selected	Location	Distance
Total precipitation over the last 6 hour	0.34 mm	nearest	57.54°N 2.7°W	2.11 km
2m temperature	2.9 °C	nearest	57.54°N 2.7°W	2.11 km

Untitled - Friday 3 May 2019, 00 UTC VT Friday 3 May 2019, 06 UTC Step 6

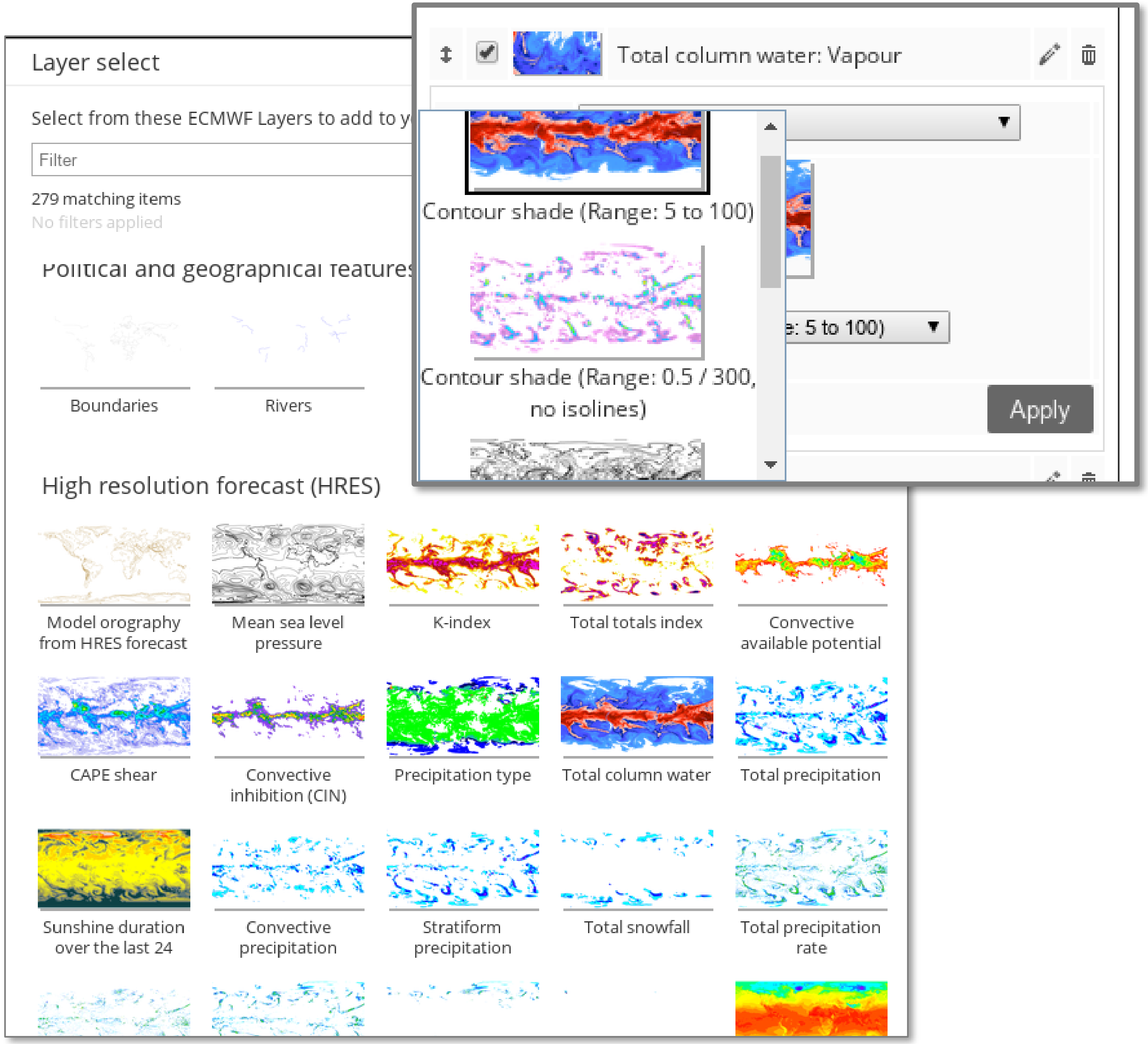
ecCharts Layers and Styles

- ecCharts has some key concepts:

- **Layer** = data + visual styling
  - Most layers have several styles available

- **Product** = set of combined layers

- Users cannot fine-tune most aspects of the styling, but many styles available, based on years of work and experience



Layer select

Select from these ECMWF Layers to add to y

Filter

279 matching items

No filters applied

Political and geographical features

High resolution forecast (HRES)

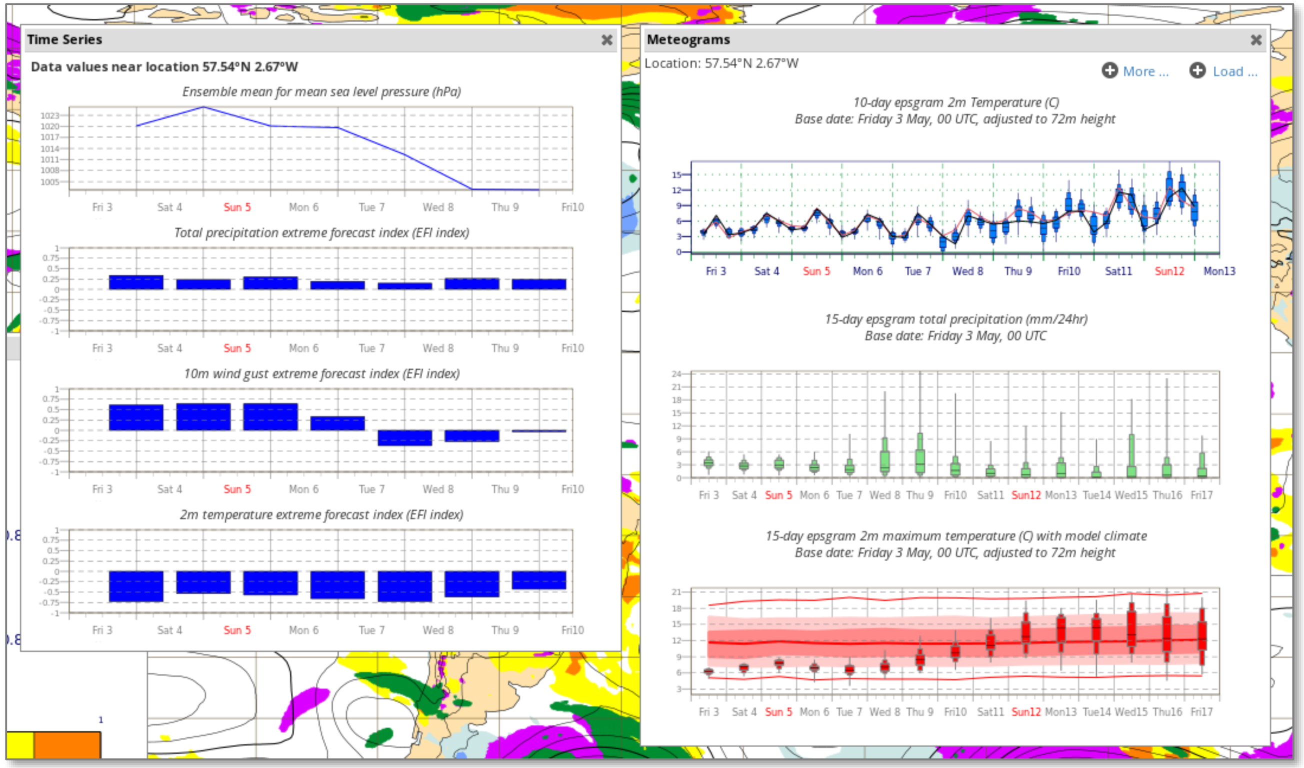
Total column water: Vapour

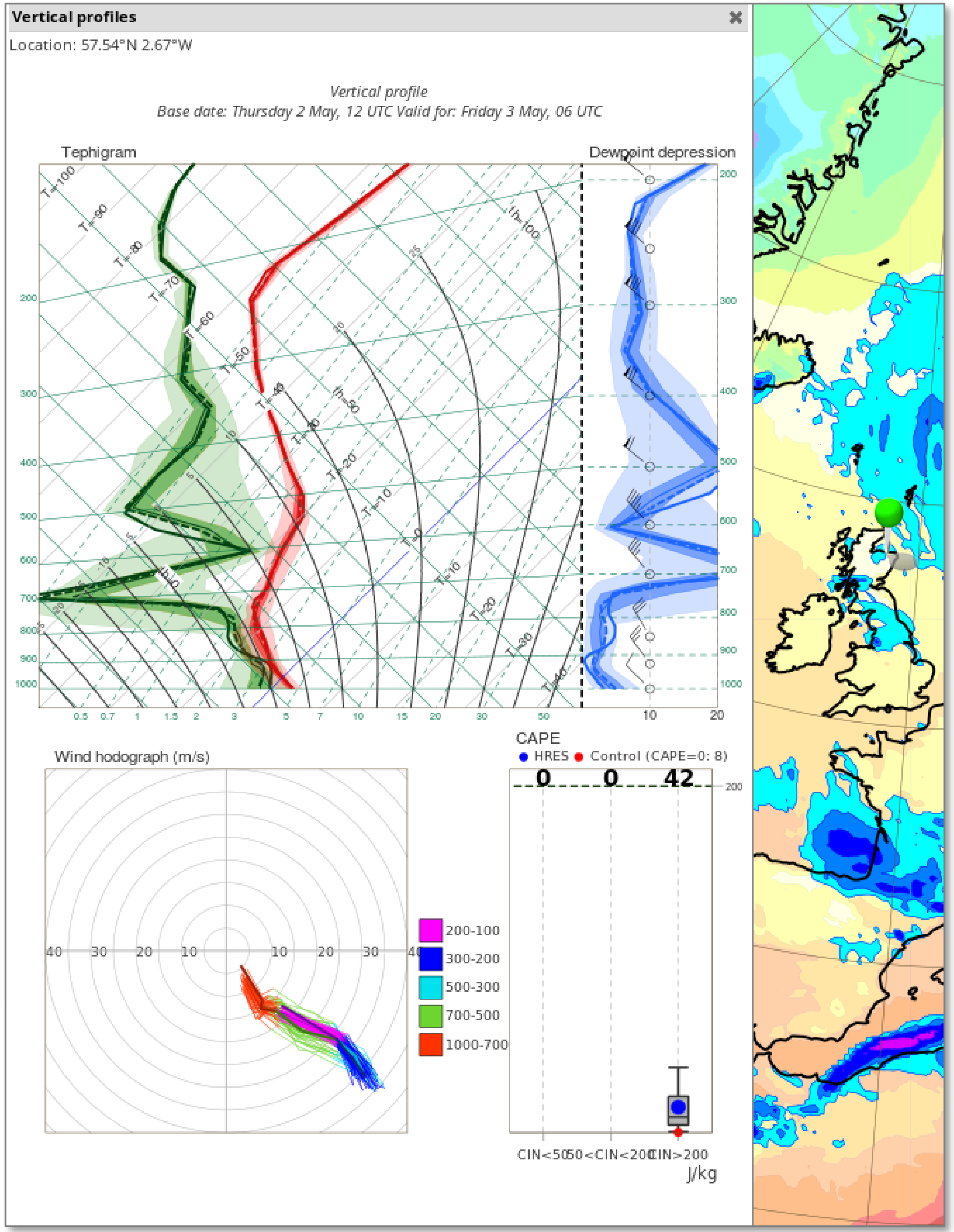
Contour shade (Range: 5 to 100)

Contour shade (Range: 0.5 / 300, no isolines)

ecCharts Point Data Tools

- Various tools to display a wealth of data at a given point on the map


Time Series
Data values near location 57.54°N 2.67°W
Ensemble mean for mean sea level pressure (hPa)
Total precipitation extreme forecast index (EFI index)
10m wind gust extreme forecast index (EFI index)
2m temperature extreme forecast index (EFI index)
Fri 3 Sat 4 Sun 5 Mon 6 Tue 7 Wed 8 Thu 9 Fri10
Meteograms
Location: 57.54°N 2.67°W
More ...
Load ...
10-day epsgram 2m Temperature (C)
Base date: Friday 3 May, 00 UTC, adjusted to 72m height
15-day epsgram total precipitation (mm/24hr)
Base date: Friday 3 May, 00 UTC
15-day epsgram 2m maximum temperature (C) with model climate
Base date: Friday 3 May, 00 UTC, adjusted to 72m height
Fri 3 Sat 4 Sun 5 Mon 6 Tue 7 Wed 8 Thu 9 Fri10 Sat11 Sun12 Mon13 Tue14 Wed15 Thu16 Fri17

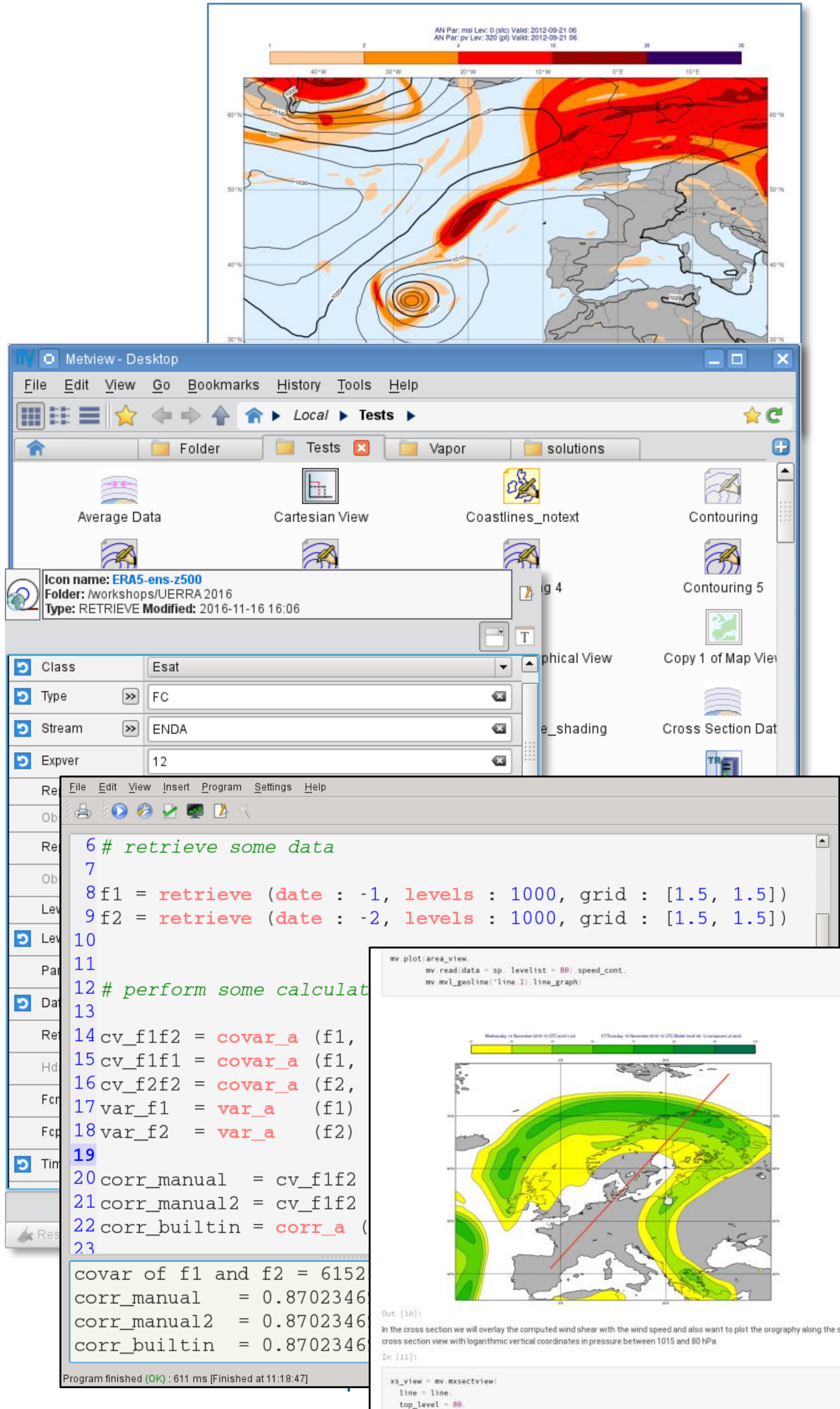


What is Metview?

- Workstation software for researchers and operational analysts
  - Runs on UNIX, from laptops to supercomputers (Linux and Mac OS X)

- Retrieve/manipulate/visualise/examine meteorological data
- Interactive usage or scripting (Macro or Python)
- Can access MARS, either locally or through the Web API

- Serving users of ECMWF data since 1993
- Open Source under Apache Licence 2.0
- Metview is a co-operation project with INPE (Brazil)



Icon name: ERA5-ens-z500
Folder: /workshops/UERRA 2016
Type: RETRIEVE **Modified:** 2016-11-16 16:06

Class	Esat
Type	FC
Stream	ENDA
Expver	12

```
# 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 calculat

cv_f1f2 = covar_a (f1,
cv_f1f1 = covar_a (f1,
cv_f2f2 = covar_a (f2,
var_f1  = var_a   (f1)
var_f2  = var_a   (f2)

corr_manual  = cv_f1f2
corr_manual2 = cv_f1f2
corr_builtin = corr_a (
```

```
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]

```
mv.plot(area_view,
        mv.read(data = sp, levelist = 80),speed_cont,
        mv.mvl_geoline(*line,1),line_graph)
```

Out [10]:

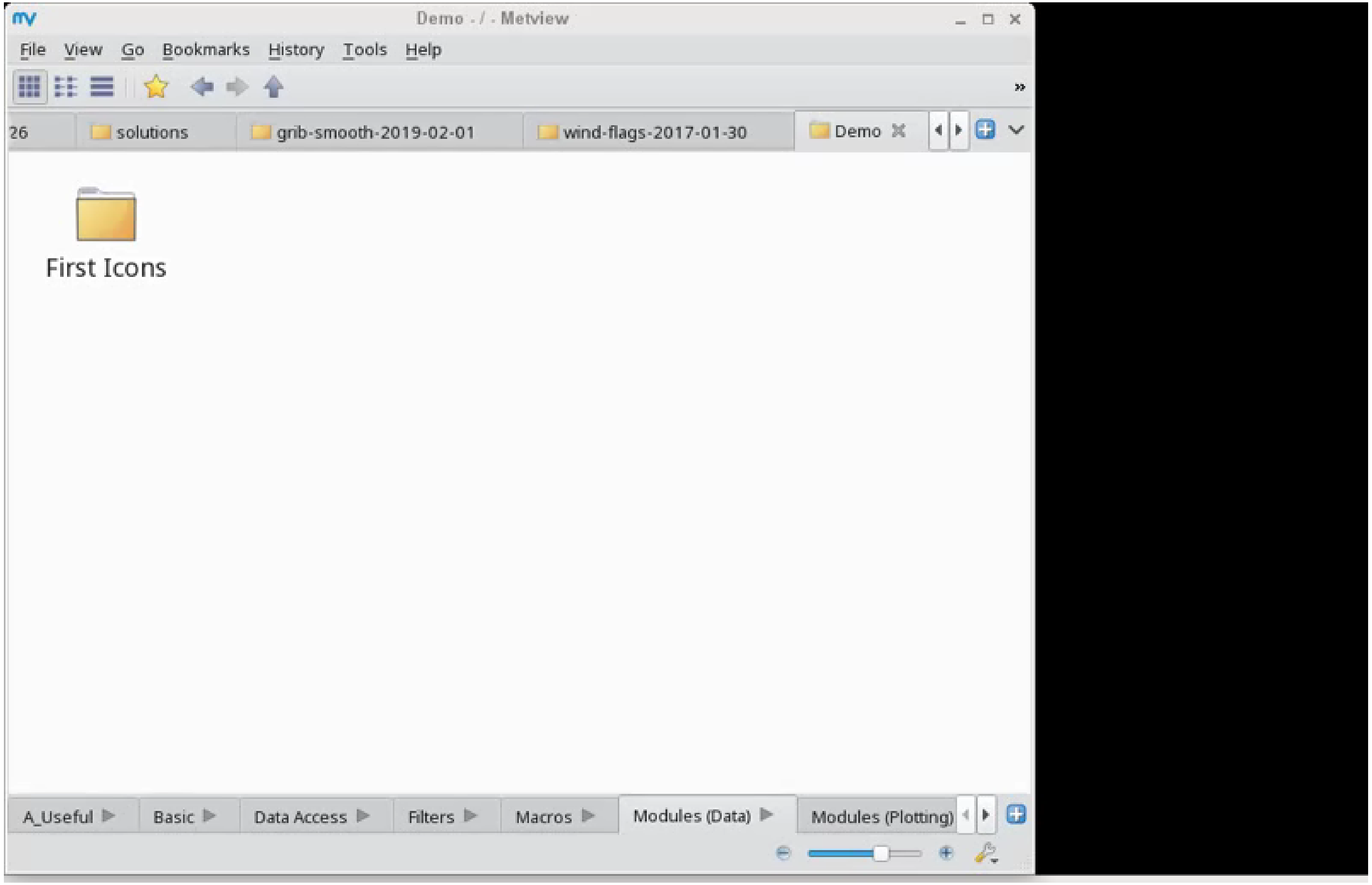
In the cross section we will overlay the computed wind shear with the wind speed and also want to plot the orography along the s cross section view with logarithmic vertical coordinates in pressure between 1015 and 80 hPa.

In [11]:

```
xs_view = mv.mxsectview(
  line = line,
  top_level = 80,
```

Getting data files into Metview

- Copy files into `$HOME/metview/…`
- Create links
- Macro/Python can read data files from anywhere

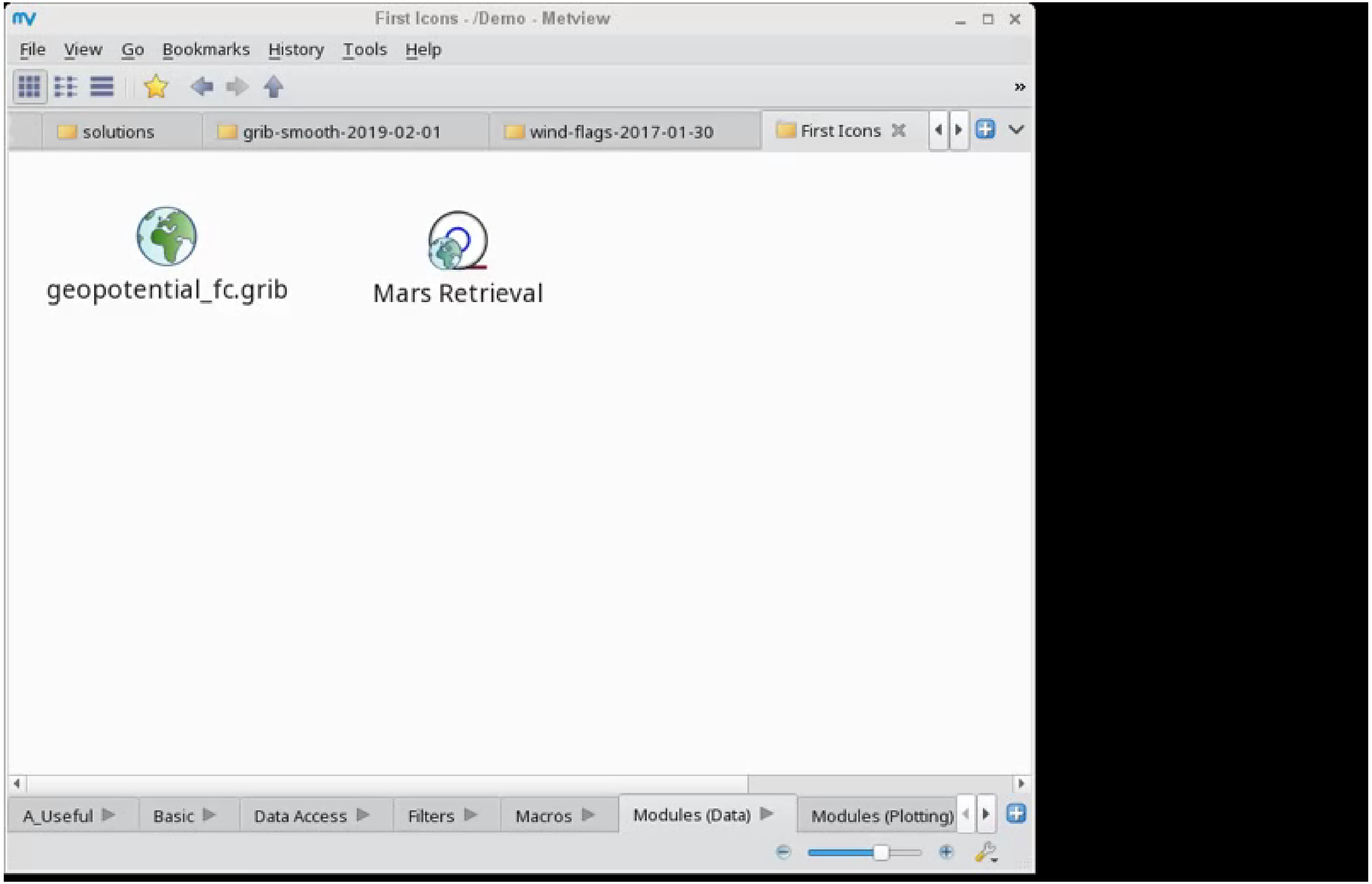


ECMWF

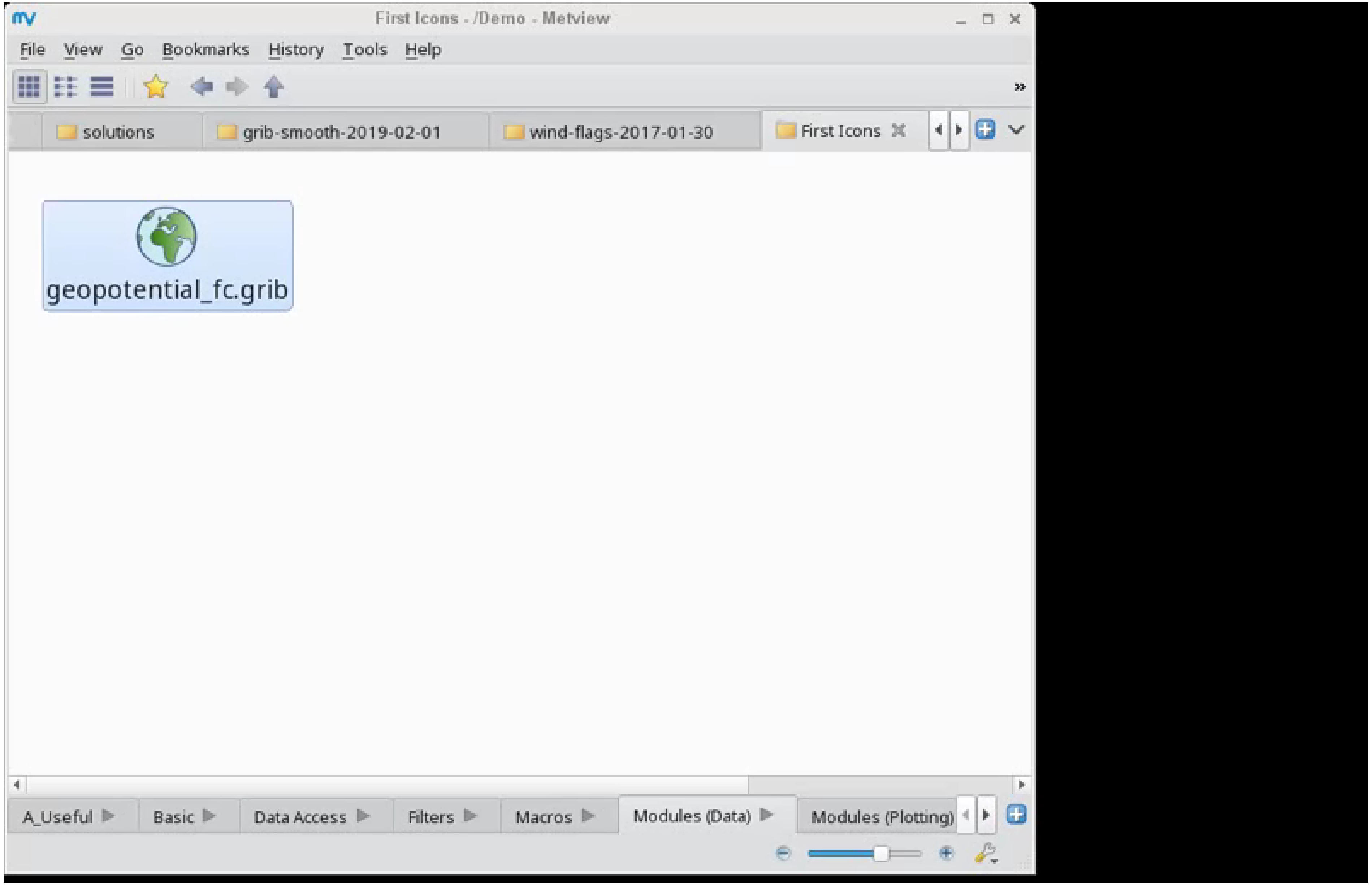
© ECMWF - slides at https://confluence.ecmwf.int/metview/Webinars

Retrieving MARS data with Metview

- Use **MARS Retrieval** icon

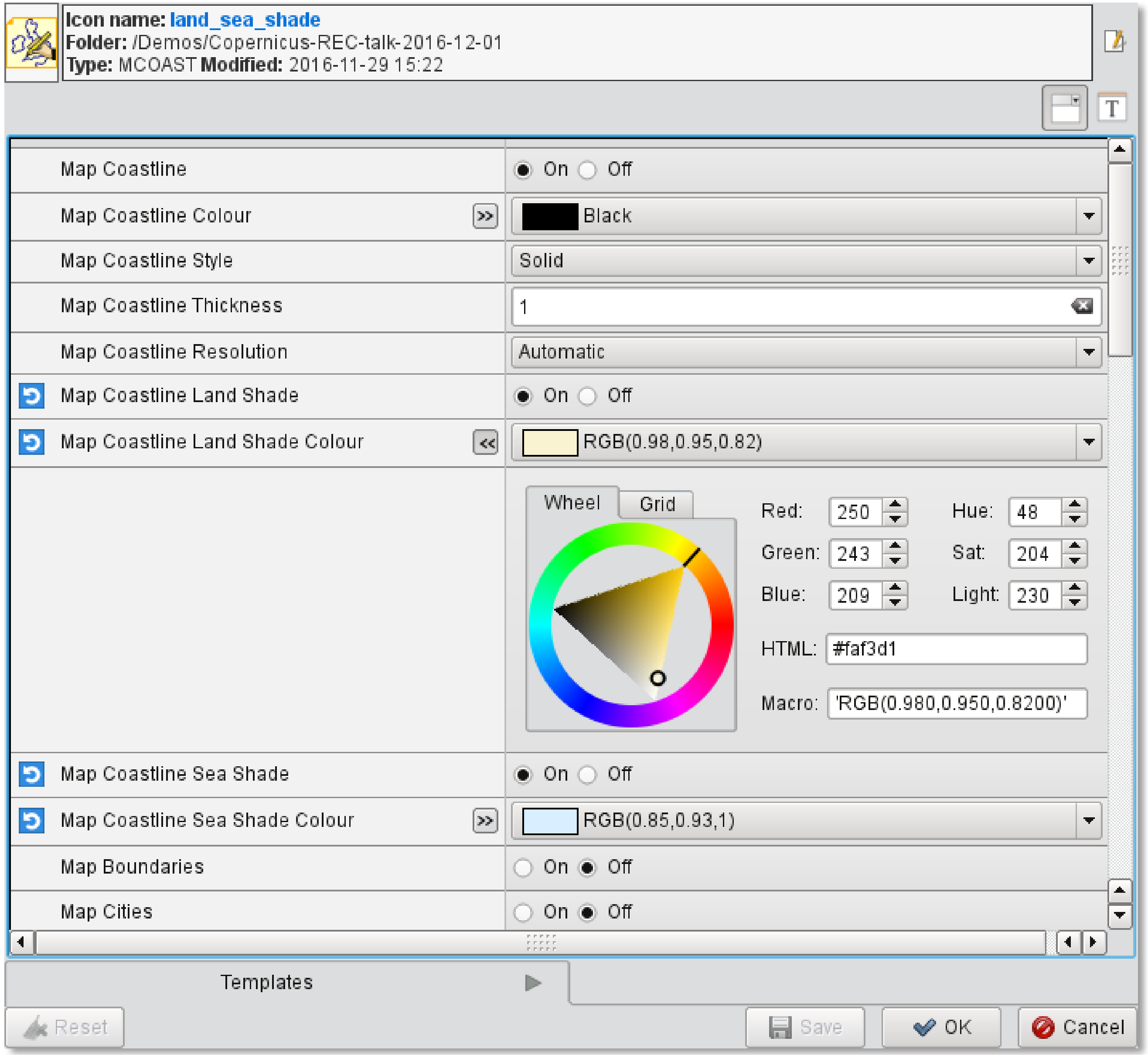


Interactive visualisation



Visual Definitions

- Metview has a class of icons called **Visual Definition Icons**
- These control various aspects of the plotting
- Such as… coastlines, legend, text, contouring, symbol plotting



Icon name: land_sea_shade
Folder: /Demos/Copernicus-REC-talk-2016-12-01
Type: MCOAST **Modified:** 2016-11-29 15:22

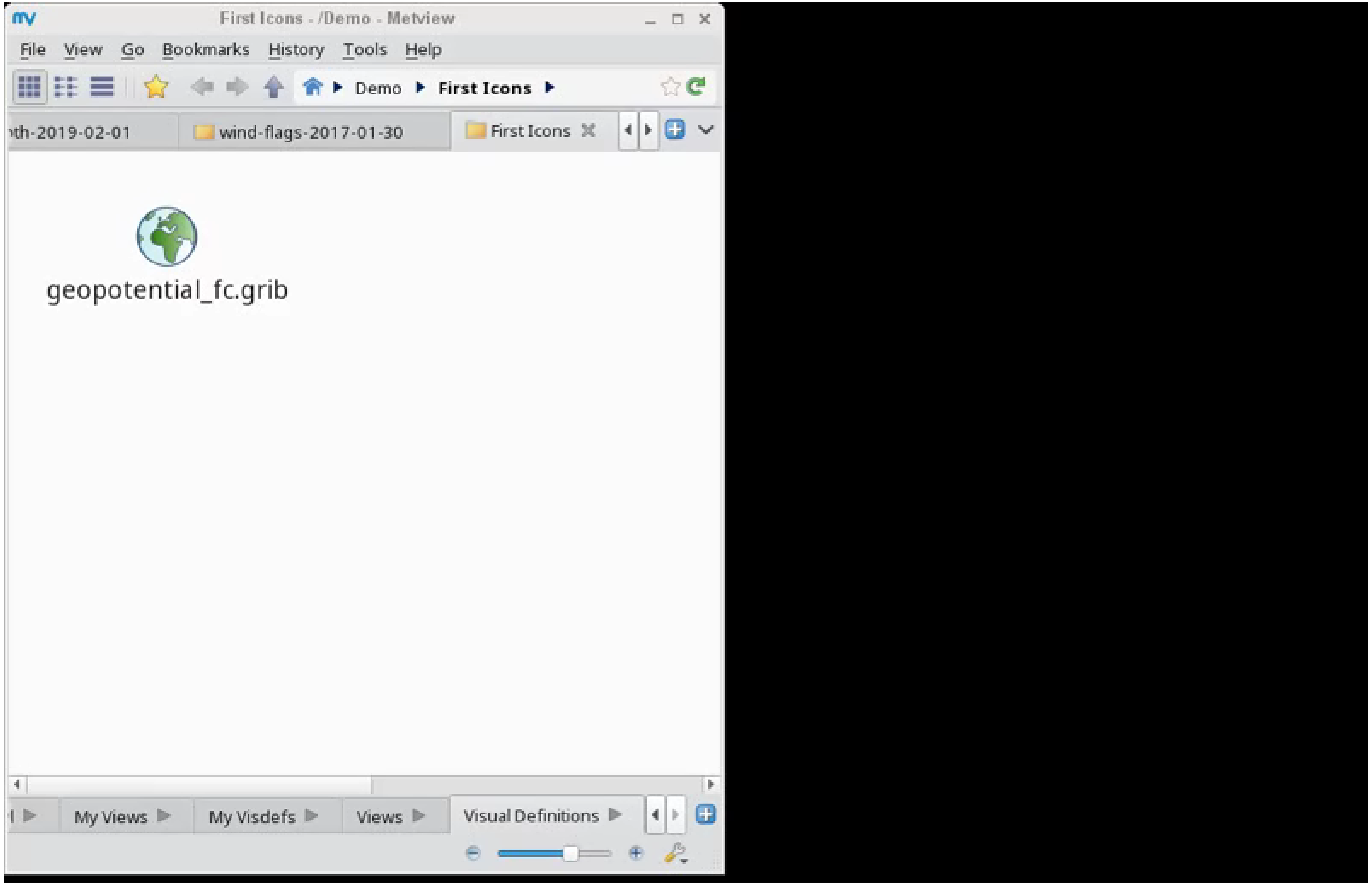
Map Coastline	● On ○ Off
Map Coastline Colour	Black
Map Coastline Style	Solid
Map Coastline Thickness	1
Map Coastline Resolution	Automatic
Map Coastline Land Shade	● On ○ Off
Map Coastline Land Shade Colour	RGB(0.98,0.95,0.82)
Map Coastline Sea Shade	● On ○ Off
Map Coastline Sea Shade Colour	RGB(0.85,0.93,1)
Map Boundaries	○ On ● Off
Map Cities	○ On ● Off

Wheel | Grid

Red: 250 Hue: 48
Green: 243 Sat: 204
Blue: 209 Light: 230
HTML: #faf3d1
Macro: 'RGB(0.980,0.950,0.8200)'

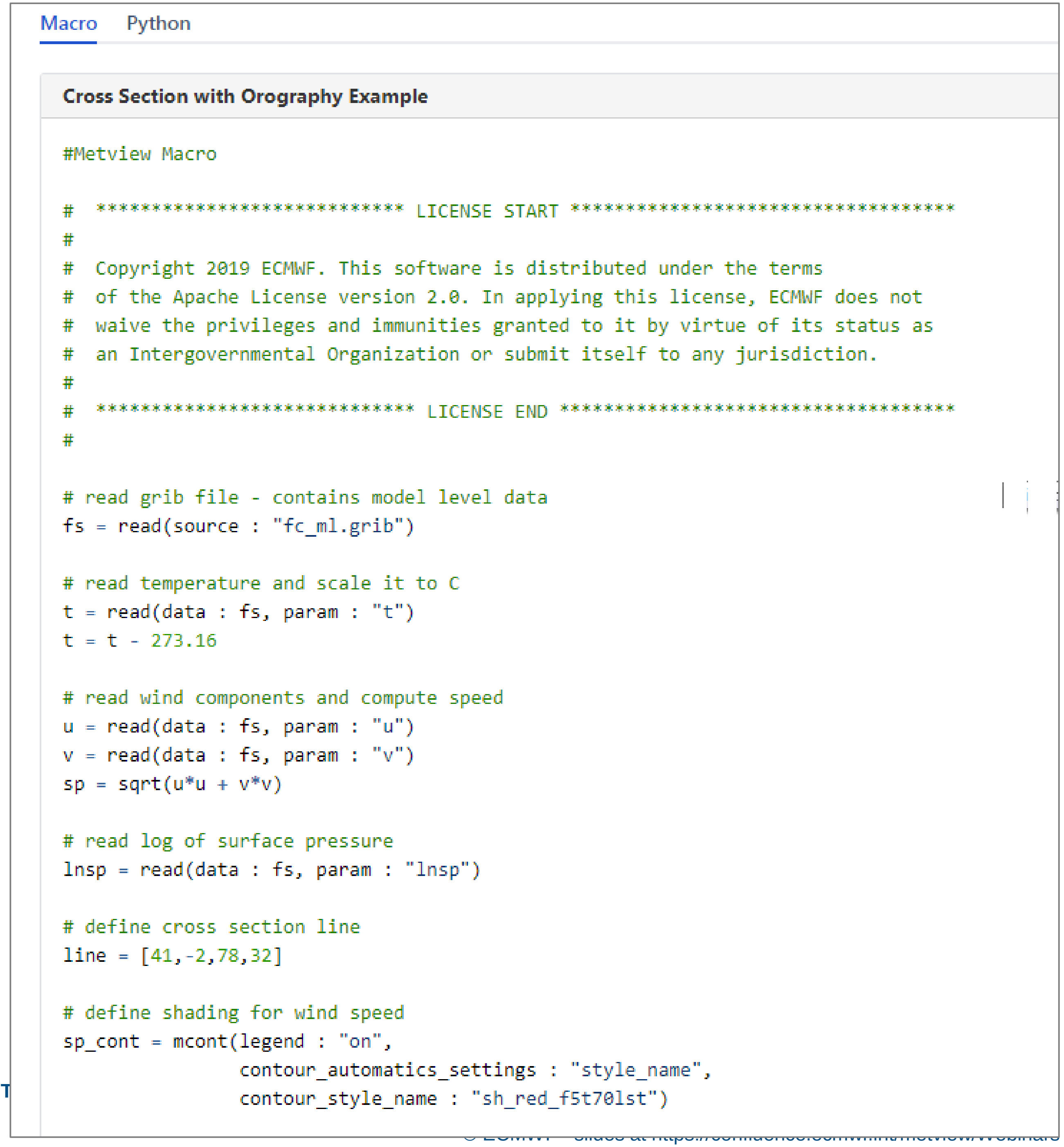
Adding land shading to the plot

- Use the **Coastlines** icon



Generating scripts

- Everything that can be done interactively with icons can be done via scripting
- Either via **Python** or Metview’s own scripting language, **Macro**
- They both offer the same functionality, but Python can interface with other Python libraries
- Scripting offers extra functionality and more flexibility
- Scripts can be generated from the plot window or by dropping icons into Metview’s code editor

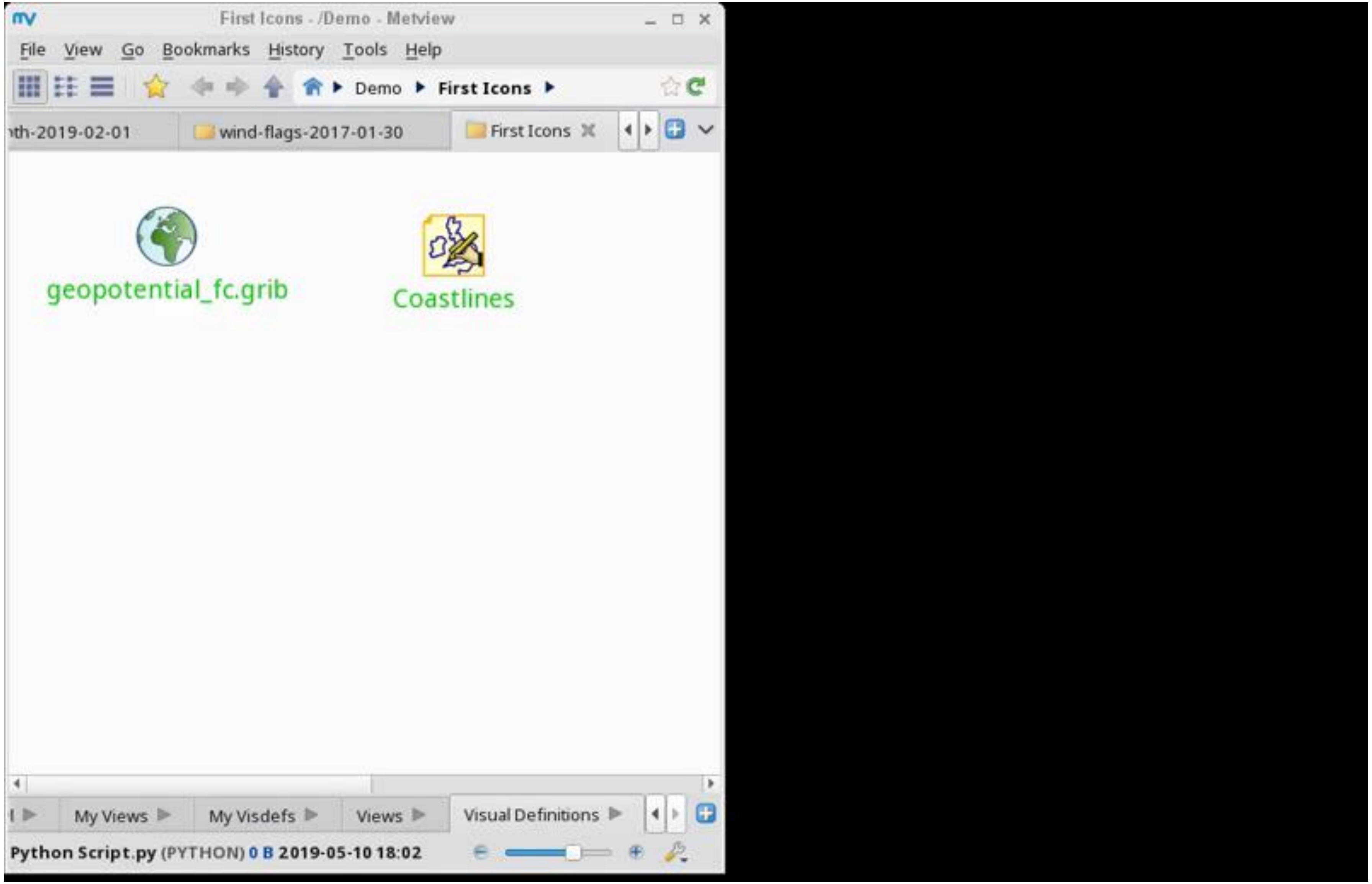


Cross Section with Orography Example

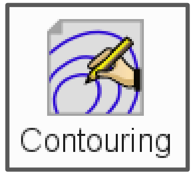
```
#Metview Macro

#  *************************** LICENSE START ***********************************
#
#  Copyright 2019 ECMWF. This software is distributed under the terms
#  of the Apache License version 2.0. In applying this license, ECMWF does not
#  waive the privileges and immunities granted to it by virtue of its status as
#  an Intergovernmental Organization or submit itself to any jurisdiction.
#
#  ***************************** LICENSE END ***********************************
#

# read grib file - contains model level data
fs = read(source : "fc_ml.grib")

# read temperature and scale it to C
t = read(data : fs, param : "t")
t = t - 273.16

# read wind components and compute speed
u = read(data : fs, param : "u")
v = read(data : fs, param : "v")
sp = sqrt(u*u + v*v)

# read log of surface pressure
lnsp = read(data : fs, param : "lnsp")

# define cross section line
line = [41,-2,78,32]

# define shading for wind speed
sp_cont = mcont(legend : "on",
                contour_automatics_settings : "style_name",
                contour_style_name : "sh_red_f5t70lst")
```

Generating scripts using drag & drop

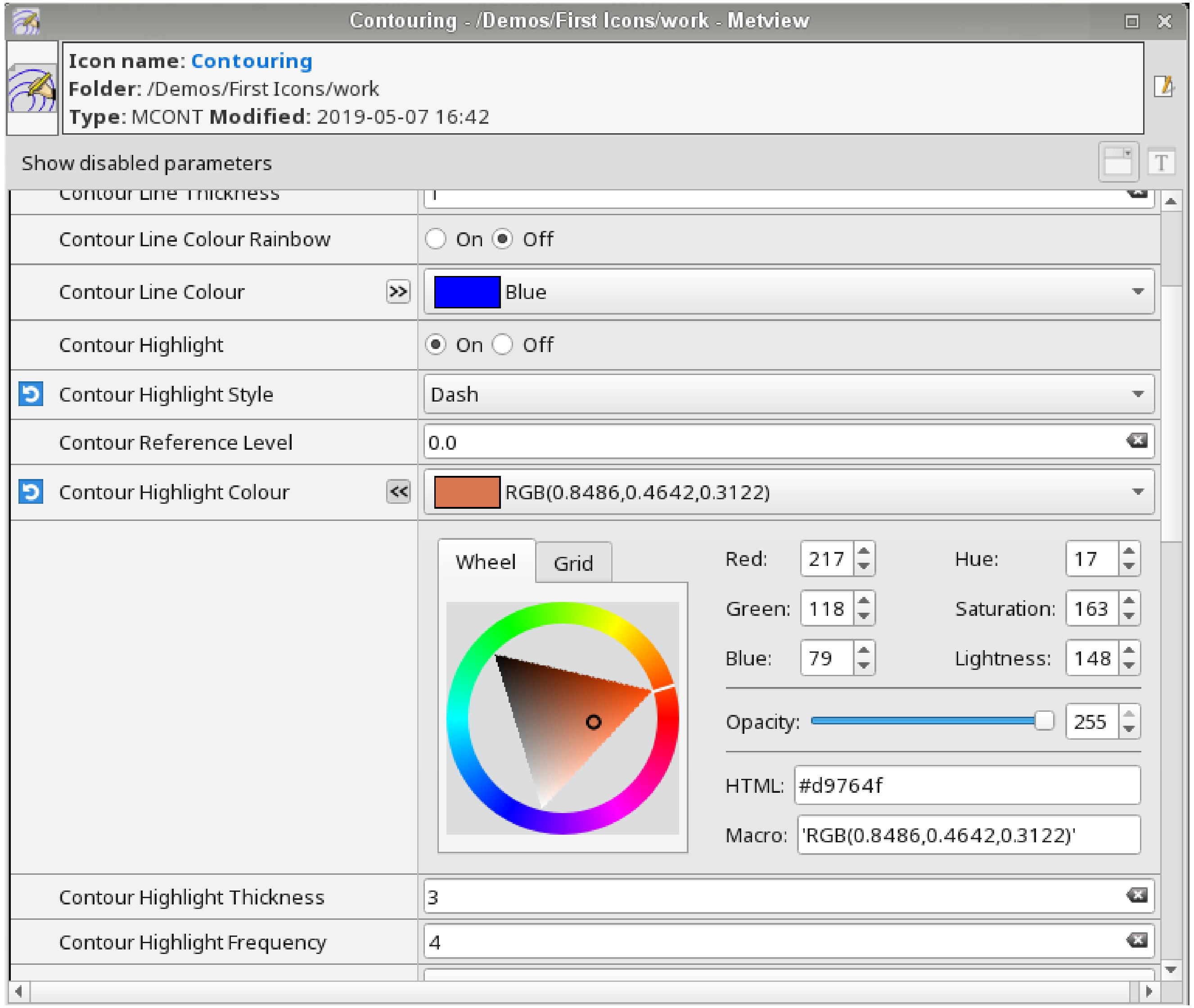
- All icons can be dropped into Metview's code editor to generate code
- More can be added by hand!



Contouring Icon



- For GRIB data, the visual definition icon we are most concerned with is the **Contouring** icon
- From here we can access all the ecCharts styles or create our own
- Drop the icon into the plot window to apply it



Contouring - /Demos/First Icons/work - Metview

Icon name: Contouring
Folder: /Demos/First Icons/work
Type: MCONT **Modified:** 2019-05-07 16:42

Show disabled parameters

Contour Line Thickness: 1

Contour Line Colour Rainbow: ○ On ● Off

Contour Line Colour: Blue

Contour Highlight: ● On ○ Off

Contour Highlight Style: Dash

Contour Reference Level: 0.0

Contour Highlight Colour: RGB(0.8486,0.4642,0.3122)

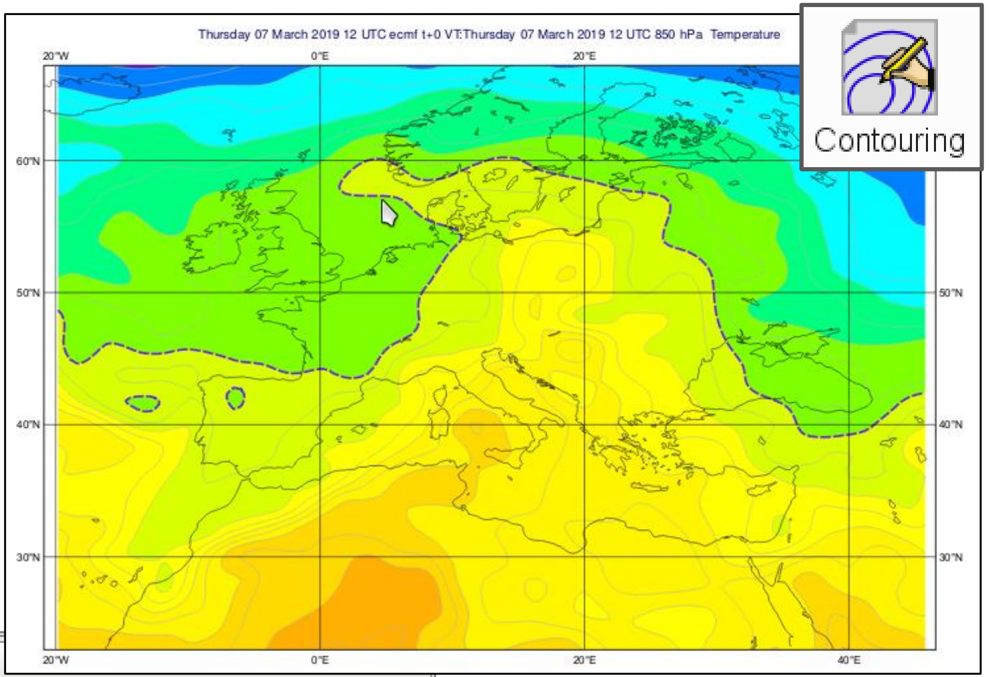
Red: 217
Green: 118
Blue: 79
Hue: 17
Saturation: 163
Lightness: 148
Opacity: 255
HTML: #d9764f
Macro: 'RGB(0.8486,0.4642,0.3122)'

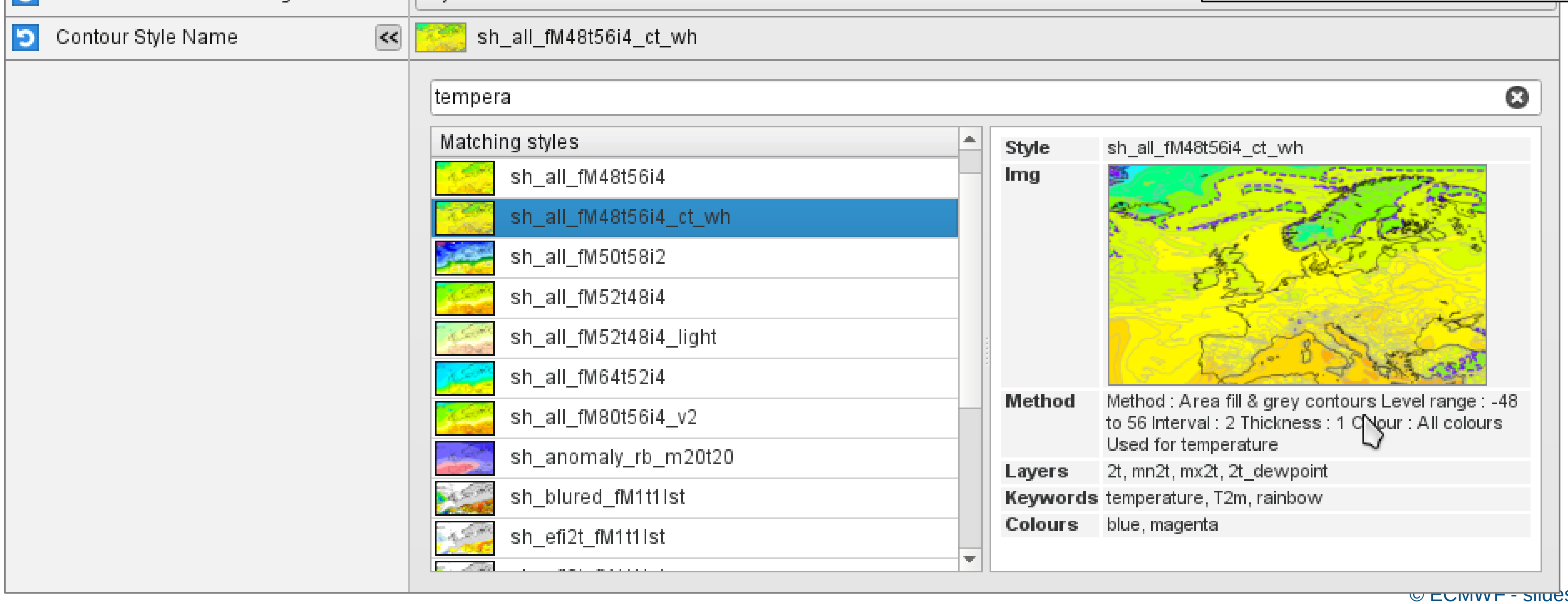
Contour Highlight Thickness: 3

Contour Highlight Frequency: 4

Using ecCharts styles in Metview

- The Contouring icon editor provides a list of styles from ecCharts
  - “Contour Automatic Setting = ECMWF”
    - style will be chosen based on meta-data
  - “Contour Automatic Setting = Style Name”
    - Choose from selection


Thursday 07 March 2019 12 UTC ecmf t+0 VT:Thursday 07 March 2019 12 UTC 850 hPa Temperature
20°W
0°E
20°E
40°E
60°N
50°N
40°N
30°N
Contouring



Contour Automatic Setting: Style Name

Contour Style Name: sh_all_fM48t56i4_ct_wh

tempera

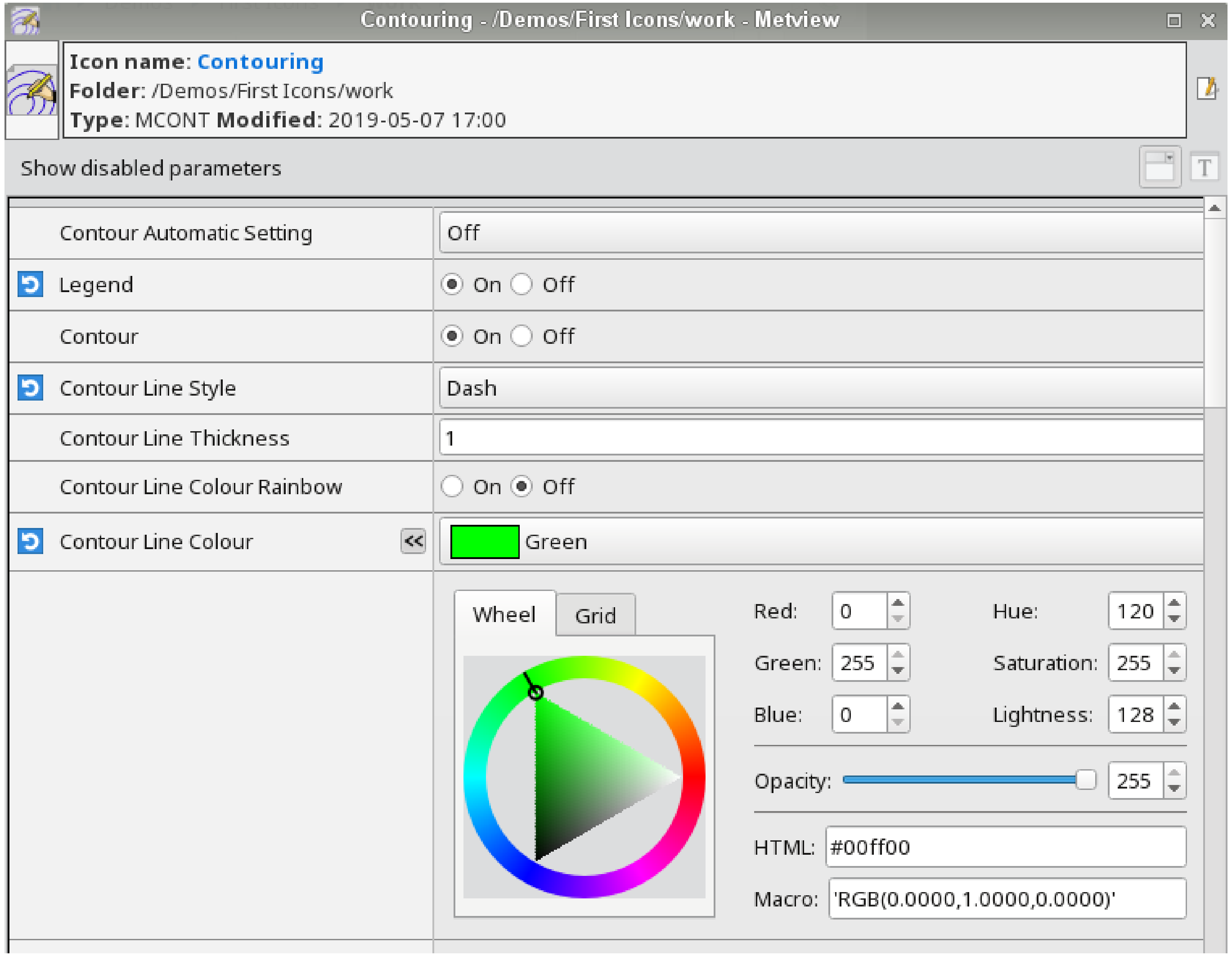
Matching styles
- sh_all_fM48t56i4
- sh_all_fM48t56i4_ct_wh
- sh_all_fM50t58i2
- sh_all_fM52t48i4
- sh_all_fM52t48i4_light
- sh_all_fM64t52i4
- sh_all_fM80t56i4_v2
- sh_anomaly_rb_m20t20
- sh_blured_fM1t1lst
- sh_efi2t_fM1t1lst

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

Creating a style from scratch

Contouring

- Deactivate Contour Automatic Setting
- This enables all the other options
- E.g.
- Isoline style, colour, labelling
- Shading, colour schemes, grid points, …



Contouring - /Demos/First Icons/work - Metview

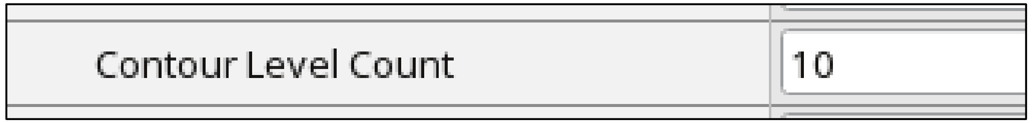
Icon name: Contouring
Folder: /Demos/First Icons/work
Type: MCONT **Modified:** 2019-05-07 17:00

Show disabled parameters

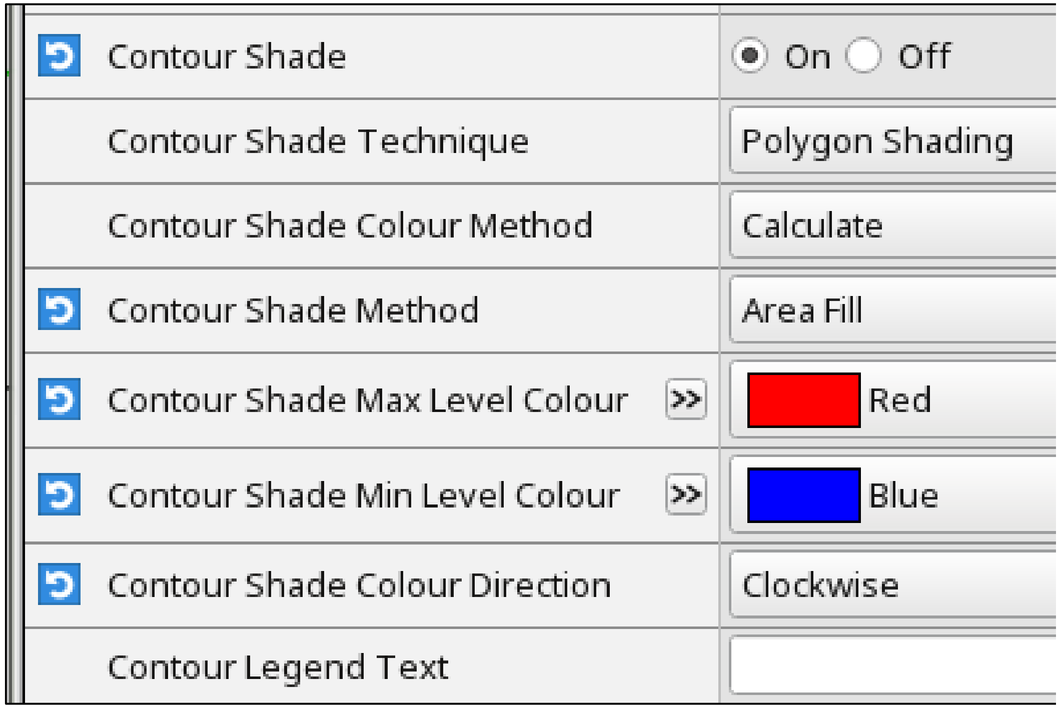
Contour Automatic Setting	Off
Legend	● On ○ Off
Contour	● On ○ Off
Contour Line Style	Dash
Contour Line Thickness	1
Contour Line Colour Rainbow	○ On ● Off
Contour Line Colour	🟩 Green

Red: 0 Hue: 120
Green: 255 Saturation: 255
Blue: 0 Lightness: 128
Opacity: 255
HTML: #00ff00
Macro: 'RGB(0.0000,1.0000,0.0000)'

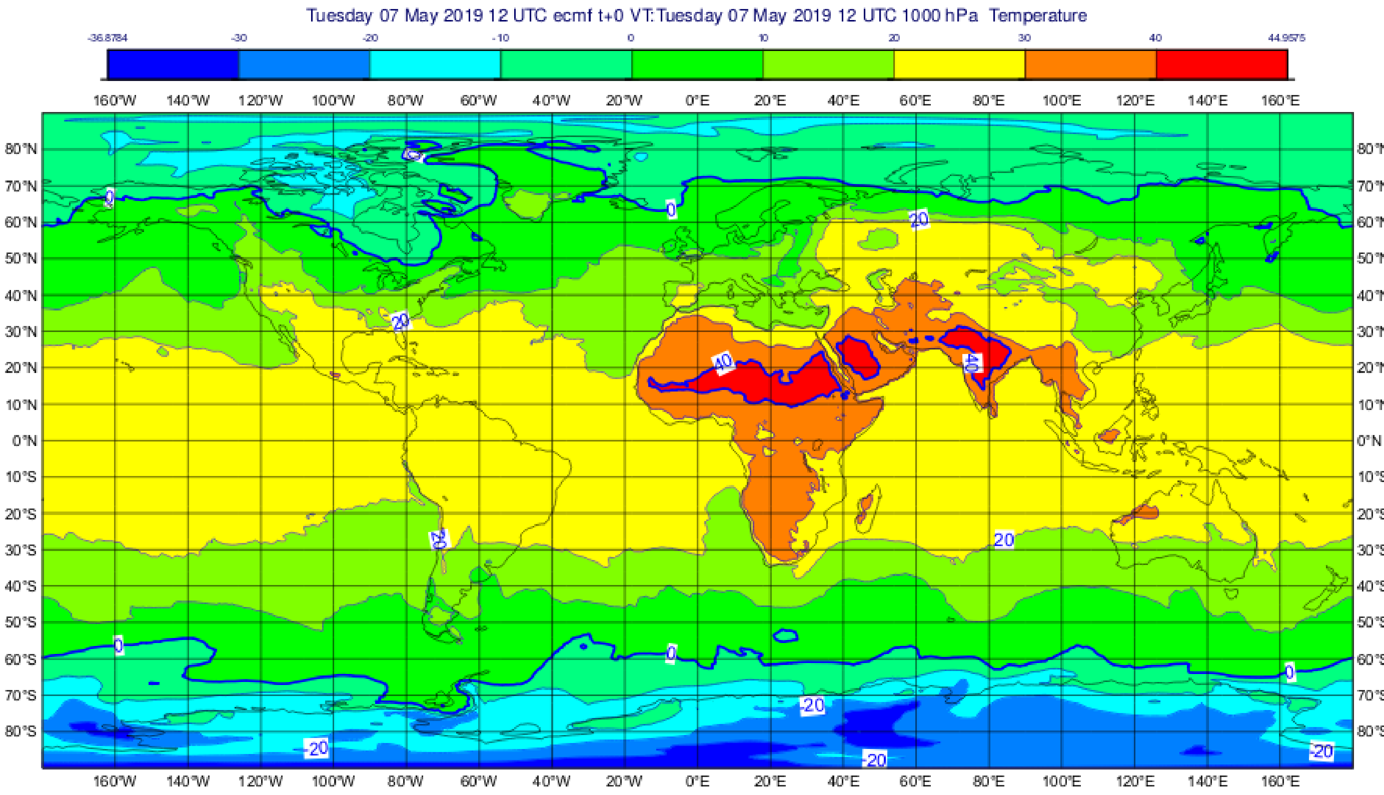
Polygon shading: Calculate 10 colours from blue to red



Contour Level Count	10

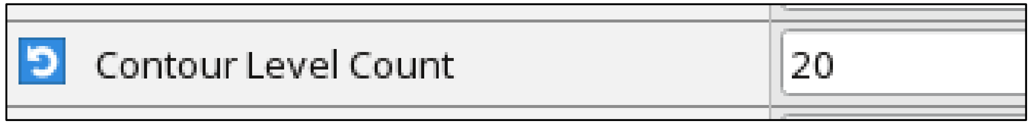


Contour Shade	● On ○ Off
Contour Shade Technique	Polygon Shading
Contour Shade Colour Method	Calculate
Contour Shade Method	Area Fill
Contour Shade Max Level Colour	🟥 Red
Contour Shade Min Level Colour	🟦 Blue
Contour Shade Colour Direction	Clockwise
Contour Legend Text	

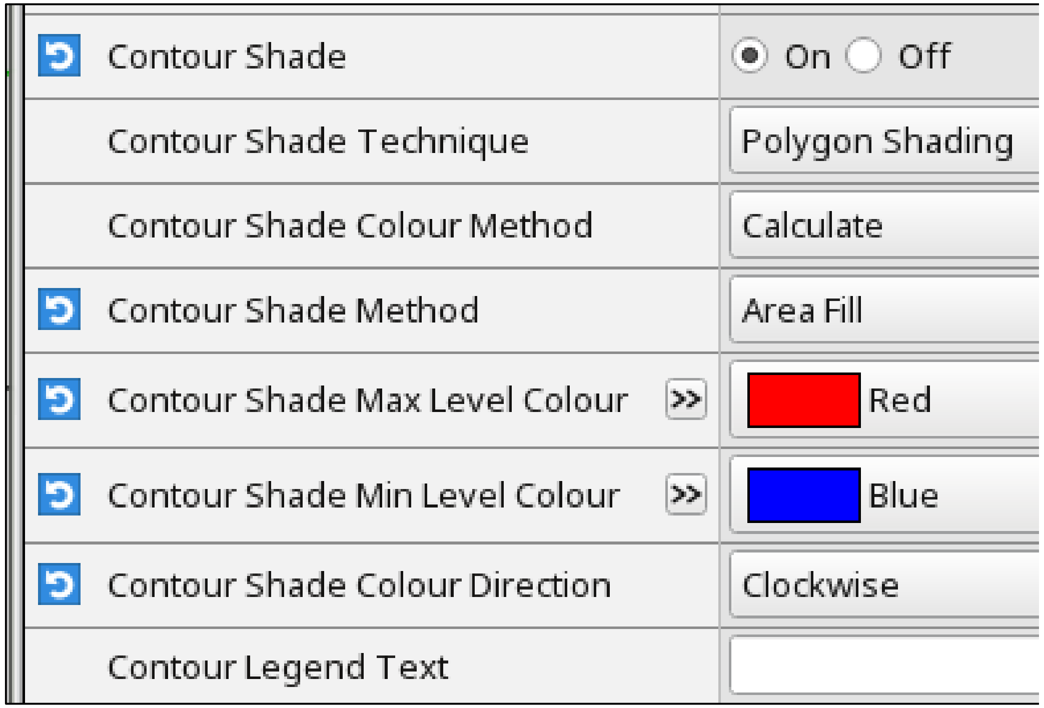


Tuesday 07 May 2019 12 UTC ecmf t+0 VT:Tuesday 07 May 2019 12 UTC 1000 hPa Temperature
-36.8784
-30
-20
-10
0
10
20
30
40
44.9575
160°W 140°W 120°W 100°W 80°W 60°W 40°W 20°W 0°E 20°E 40°E 60°E 80°E 100°E 120°E 140°E 160°E
80°N 70°N 60°N 50°N 40°N 30°N 20°N 10°N 0°N 10°S 20°S 30°S 40°S 50°S 60°S 70°S 80°S
0
20
40
-20

Polygon shading: Calculate 20 colours from blue to red



Contour Level Count: 20



Contour Shade: ● On ○ Off

Contour Shade Technique: Polygon Shading

Contour Shade Colour Method: Calculate

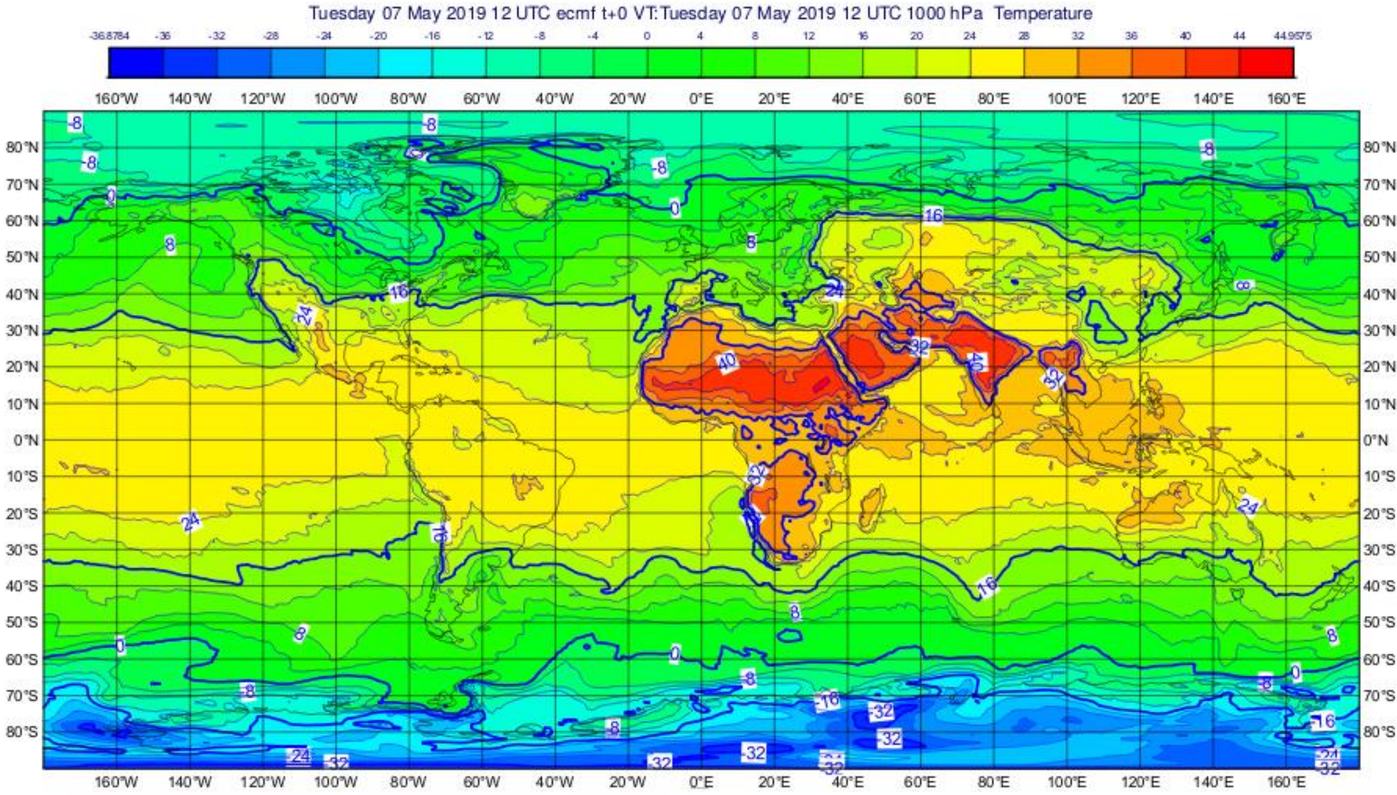
Contour Shade Method: Area Fill

Contour Shade Max Level Colour: 🟥 Red

Contour Shade Min Level Colour: 🟦 Blue

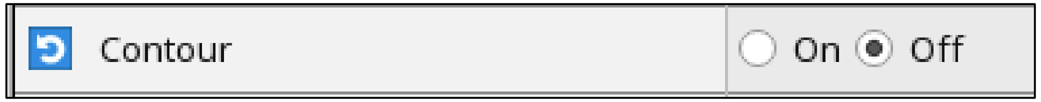
Contour Shade Colour Direction: Clockwise

Contour Legend Text:

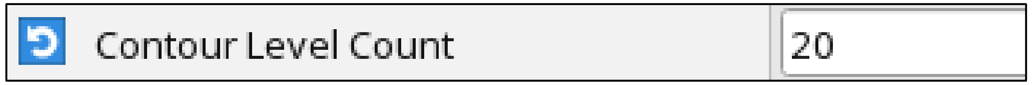


Tuesday 07 May 2019 12 UTC ecmwf t+0 VT:Tuesday 07 May 2019 12 UTC 1000 hPa Temperature

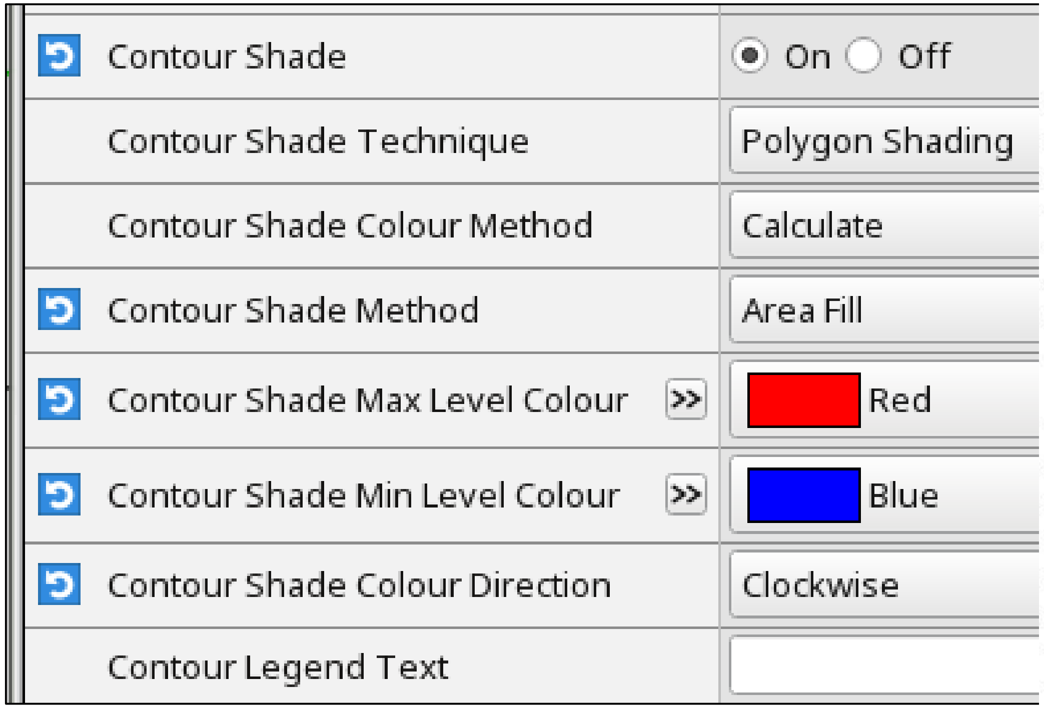
Polygon shading: remove isolines



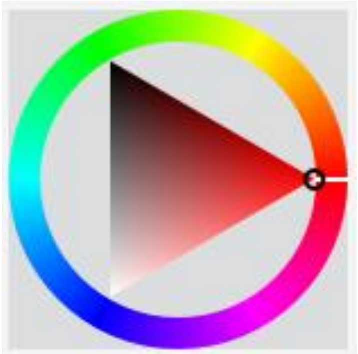
Contour	○ On ● Off

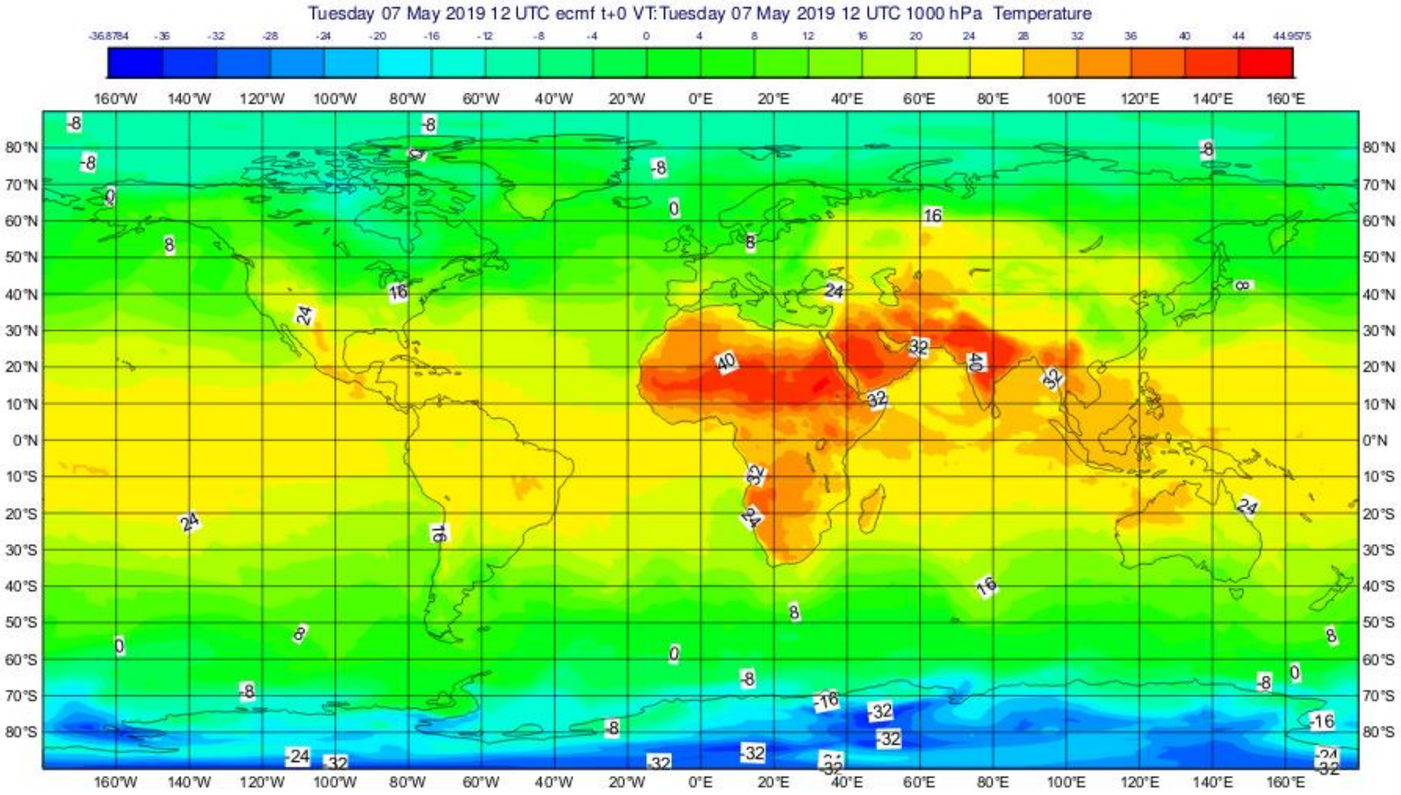


Contour Level Count	20



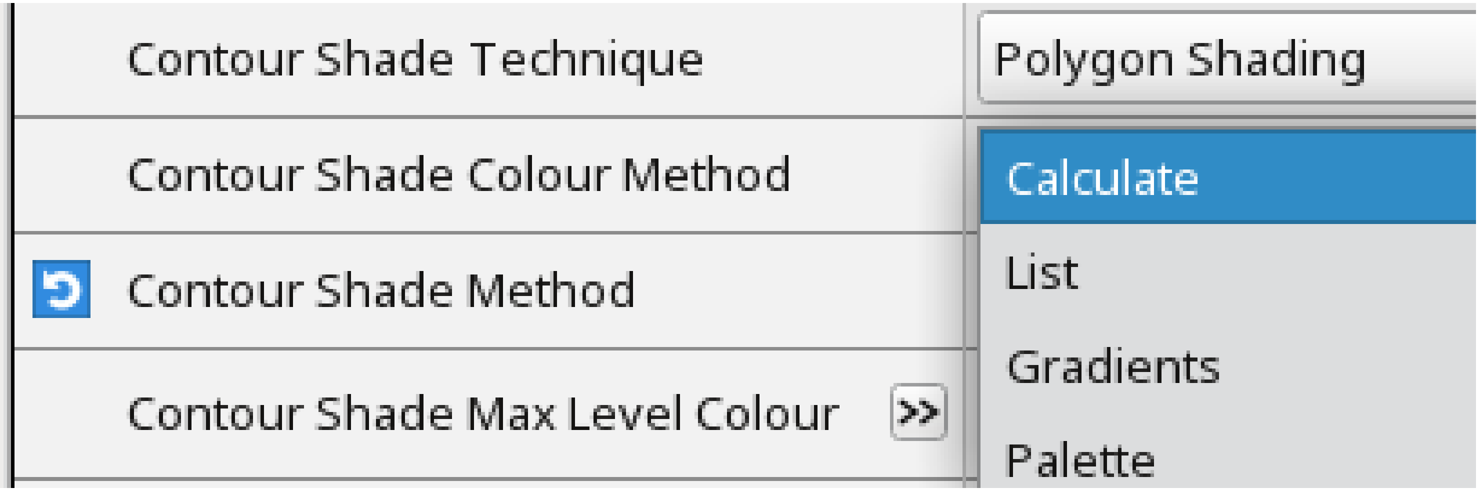
Contour Shade	● On ○ Off
Contour Shade Technique	Polygon Shading
Contour Shade Colour Method	Calculate
Contour Shade Method	Area Fill
Contour Shade Max Level Colour	🟥 Red
Contour Shade Min Level Colour	🟦 Blue
Contour Shade Colour Direction	Clockwise
Contour Legend Text	




Tuesday 07 May 2019 12 UTC ecmwf t+0 VT:Tuesday 07 May 2019 12 UTC 1000 hPa Temperature
-36.8784 -36 -32 -28 -24 -20 -16 -12 -8 -4 0 4 8 12 16 20 24 28 32 36 40 44 44.9575
160°W 140°W 120°W 100°W 80°W 60°W 40°W 20°W 0°E 20°E 40°E 60°E 80°E 100°E 120°E 140°E 160°E
80°N 70°N 60°N 50°N 40°N 30°N 20°N 10°N 0°N 10°S 20°S 30°S 40°S 50°S 60°S 70°S 80°S

Multiple ways to specify the colours used in shading

- Set **Contour Shade Colour Method**
- The previous examples used **Calculate**



Contour Shade Technique: Polygon Shading

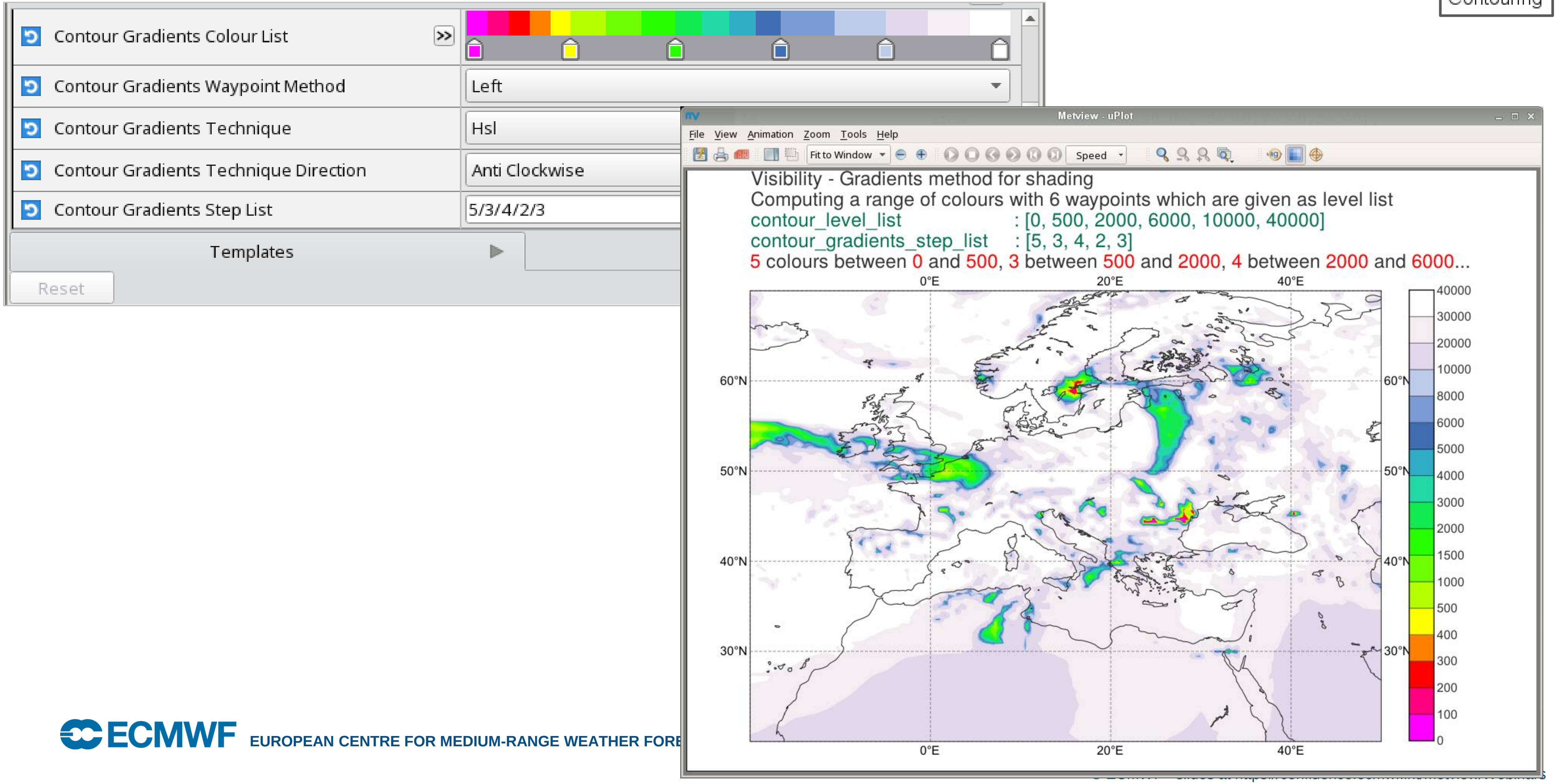
Contour Shade Colour Method: Calculate

Contour Shade Method

Contour Shade Max Level Colour

Shading with multiple colour gradients (“Gradients”)

Contouring



- Contour Gradients Colour List
- Contour Gradients Waypoint Method: Left
- Contour Gradients Technique: Hsl
- Contour Gradients Technique Direction: Anti Clockwise
- Contour Gradients Step List: 5/3/4/2/3

Visibility - Gradients method for shading

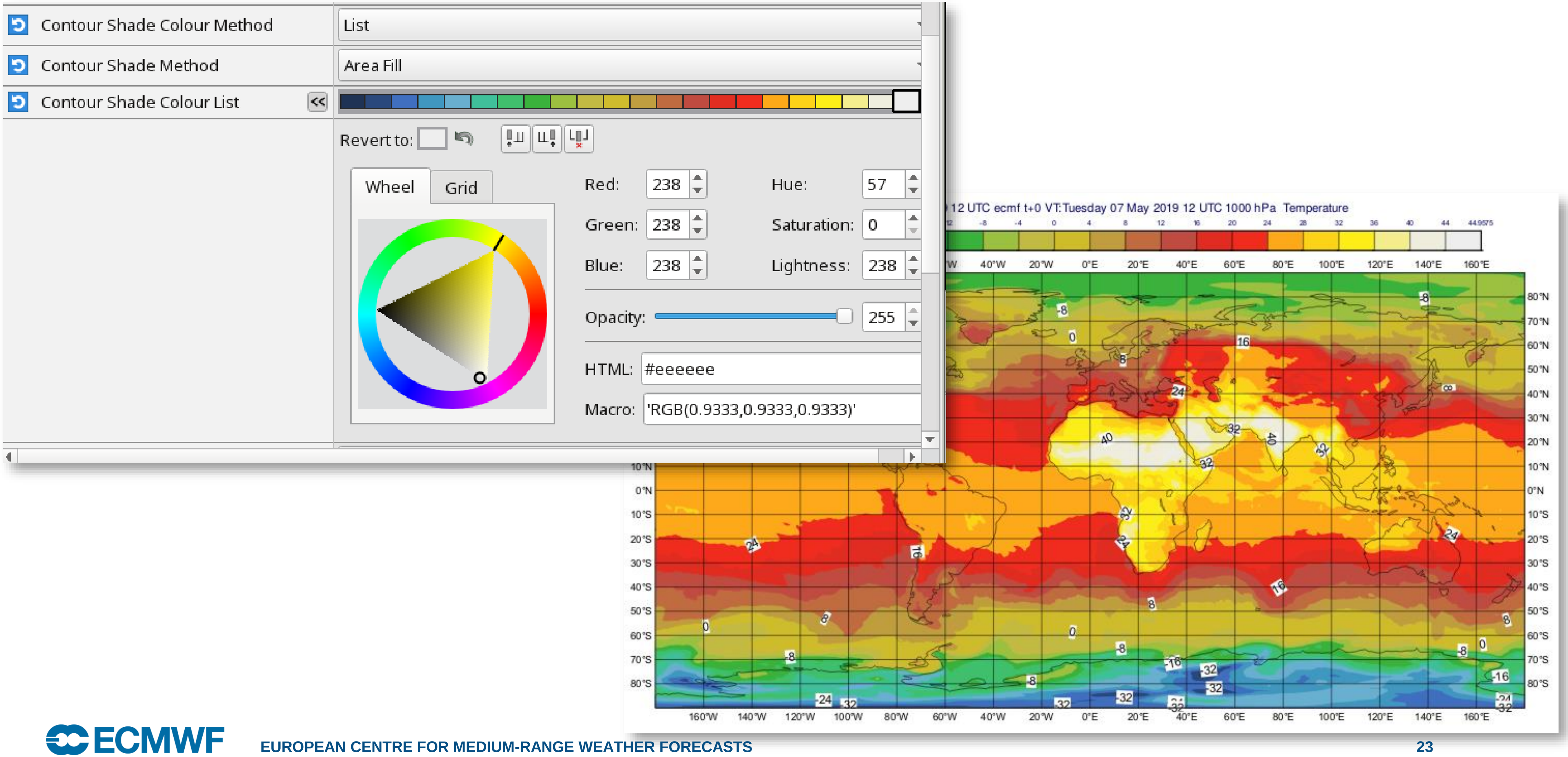
Computing a range of colours with 6 waypoints which are given as level list

contour_level_list : [0, 500, 2000, 6000, 10000, 40000]

contour_gradients_step_list : [5, 3, 4, 2, 3]

5 colours between 0 and 500, 3 between 500 and 2000, 4 between 2000 and 6000...

Shading with a user-defined list of colours (“List”)



Contour Shade Colour Method: List

Contour Shade Method: Area Fill

Contour Shade Colour List

Revert to:

Red: 238

Green: 238

Blue: 238

Hue: 57

Saturation: 0

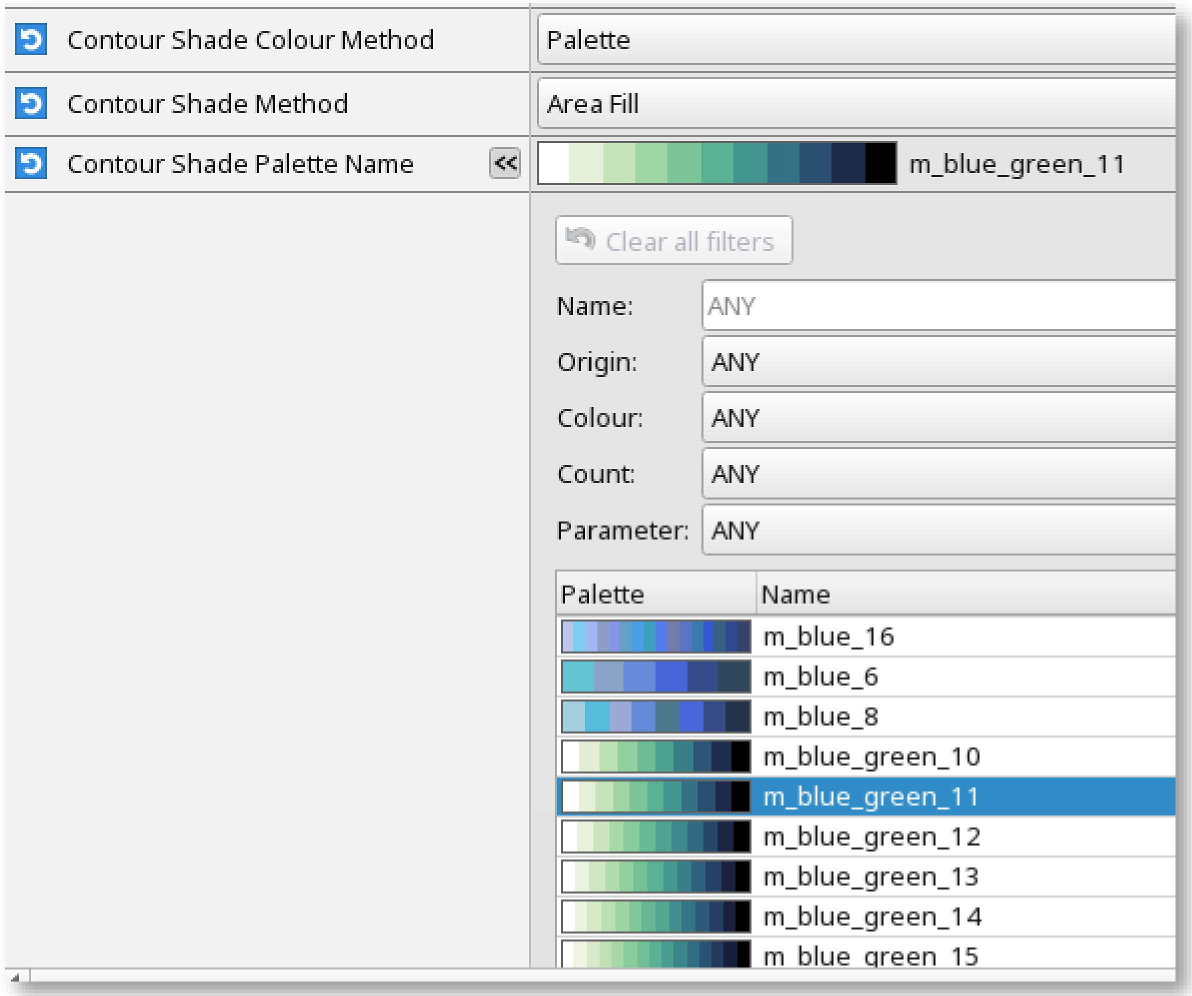
Lightness: 238

Opacity: 255

HTML: #eeeeee

Macro: 'RGB(0.9333,0.9333,0.9333)'

Shading with a pre-defined colour palette ("Palette")



Contour Shade Colour Method: Palette

Contour Shade Method: Area Fill

Contour Shade Palette Name: m_blue_green_11

Name: ANY

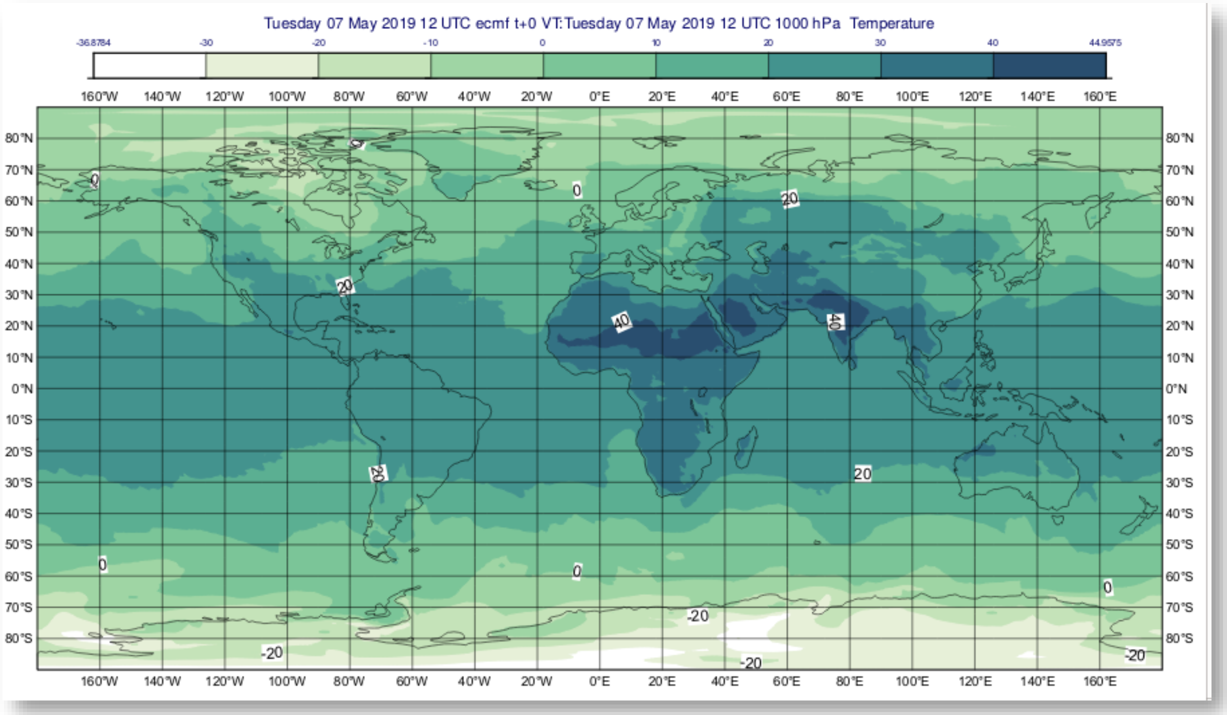
Origin: ANY

Colour: ANY

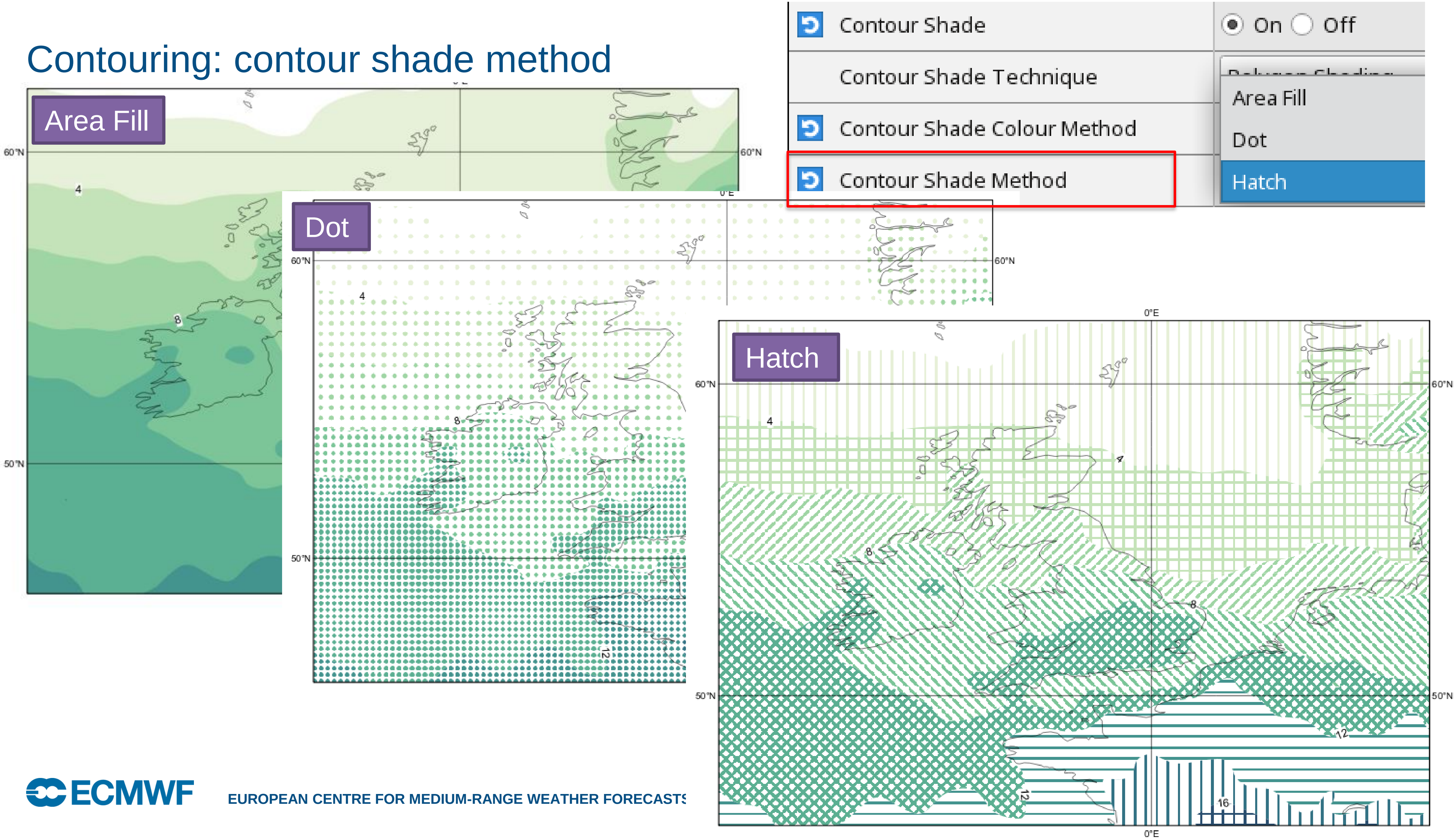
Count: ANY

Parameter: ANY

Palette	Name
	m_blue_16
	m_blue_6
	m_blue_8
	m_blue_green_10
	m_blue_green_11
	m_blue_green_12
	m_blue_green_13
	m_blue_green_14
	m_blue_green_15



Contouring: contour shade method



Contour Shade ● On ○ Off

Contour Shade Technique Polygon Shading

Contour Shade Colour Method

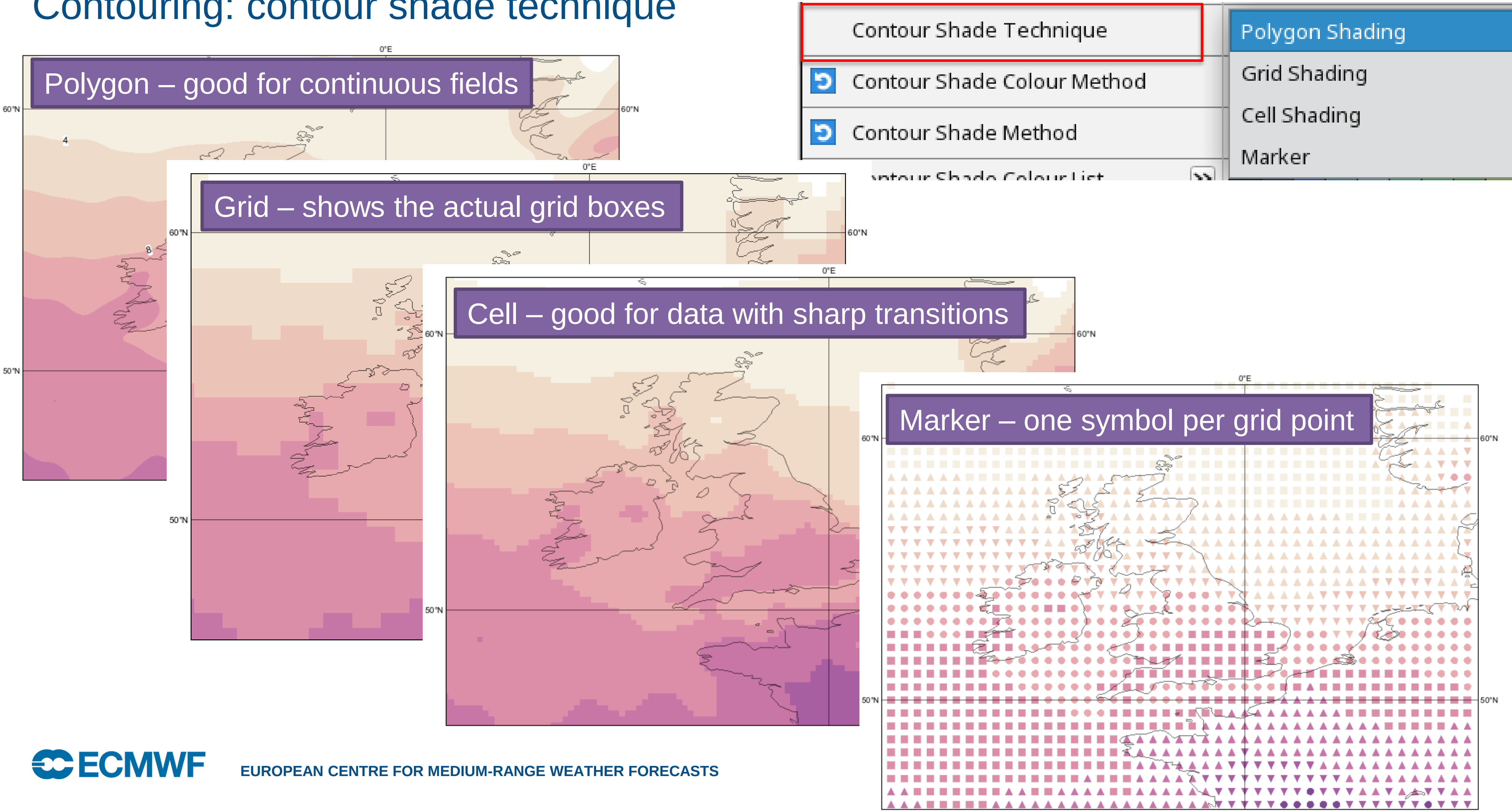
Contour Shade Method

Area Fill

Dot

Hatch

Contouring: contour shade technique



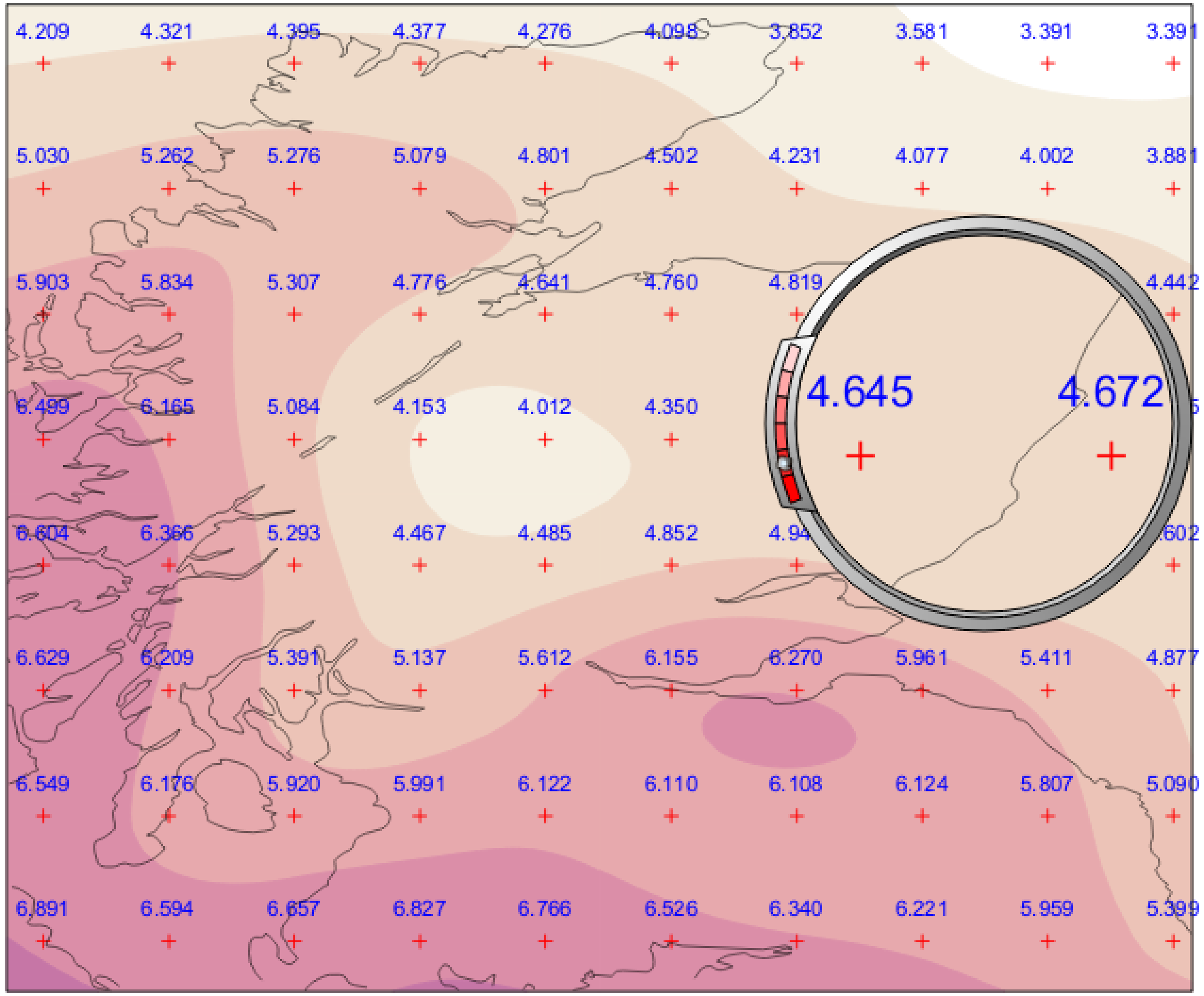
Polygon – good for continuous fields

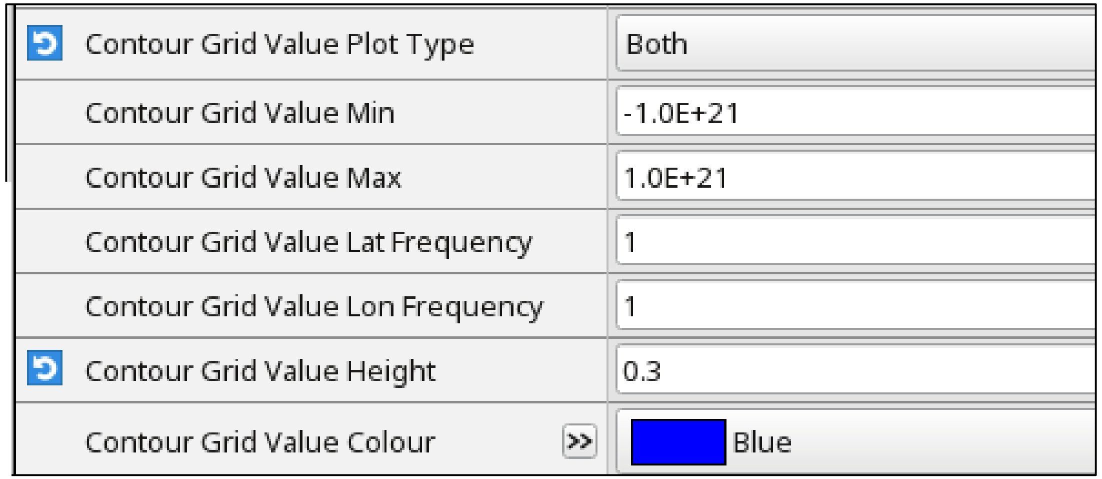
Grid – shows the actual grid boxes

Cell – good for data with sharp transitions

Marker – one symbol per grid point

Plotting grid points and values



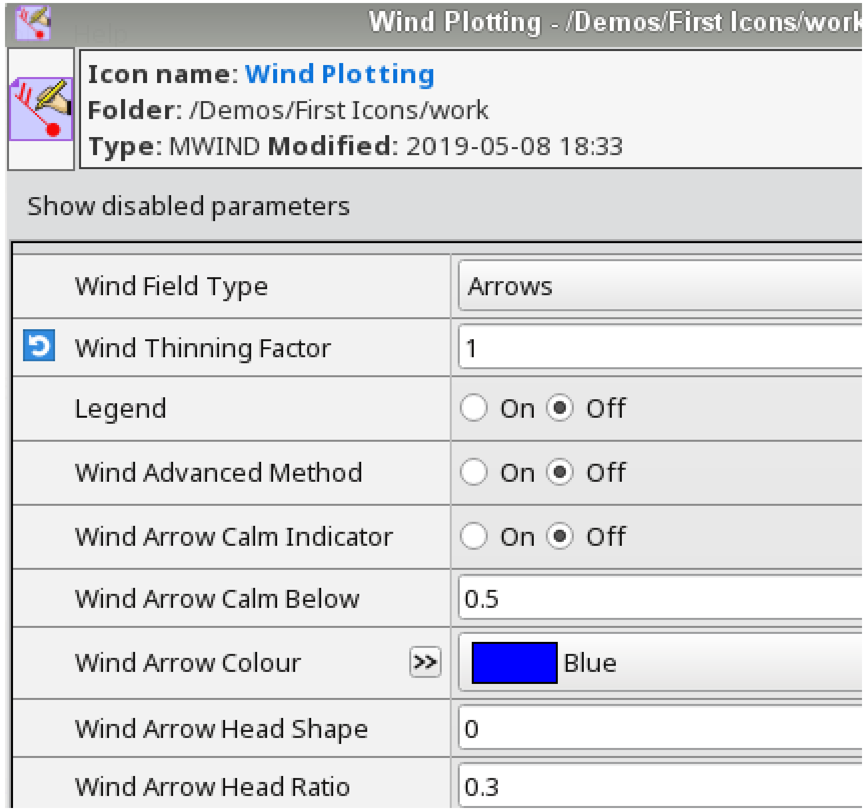


Contour Grid Value Plot Type	Both
Contour Grid Value Min	-1.0E+21
Contour Grid Value Max	1.0E+21
Contour Grid Value Lat Frequency	1
Contour Grid Value Lon Frequency	1
Contour Grid Value Height	0.3
Contour Grid Value Colour	🟦 Blue

Wind plotting

Wind Plotting

- Metview recognises fields that are vector pairs, e.g. 10U/10V
- The **Wind Plotting** icon provides parameters for customising the plotting of wind fields



Wind Plotting - /Demos/First Icons/work

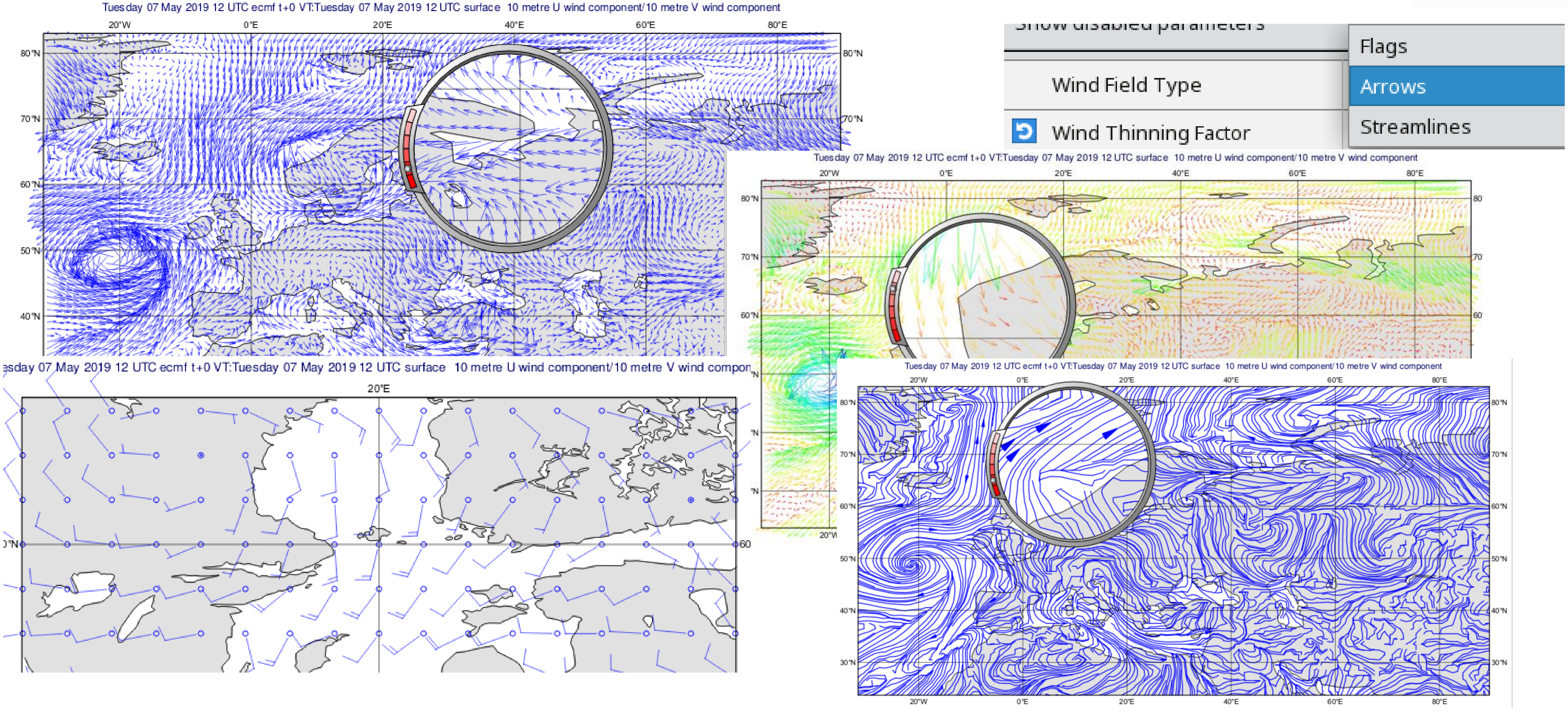
Icon name: Wind Plotting
Folder: /Demos/First Icons/work
Type: MWIND **Modified:** 2019-05-08 18:33

Show disabled parameters

Wind Field Type	Arrows
Wind Thinning Factor	1
Legend	○ On ● Off
Wind Advanced Method	○ On ● Off
Wind Arrow Calm Indicator	○ On ● Off
Wind Arrow Calm Below	0.5
Wind Arrow Colour	🟦 Blue
Wind Arrow Head Shape	0
Wind Arrow Head Ratio	0.3

Wind plotting

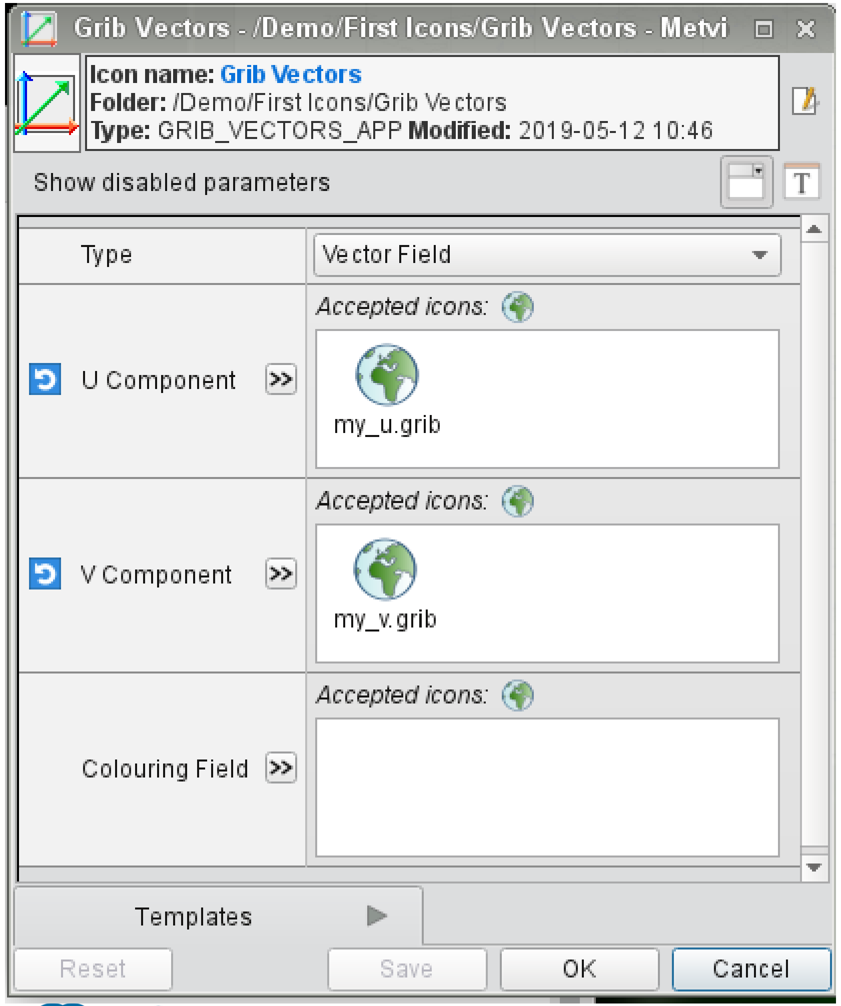
Wind Plotting



Specifying own vector components

Grib Vectors

- The **Grib Vectors** icon allows you to combine your own fields into a vector pair for plotting



Grib Vectors - /Demo/First Icons/Grib Vectors - Metvi

Icon name: Grib Vectors
Folder: /Demo/First Icons/Grib Vectors
Type: GRIB_VECTORS_APP **Modified:** 2019-05-12 10:46

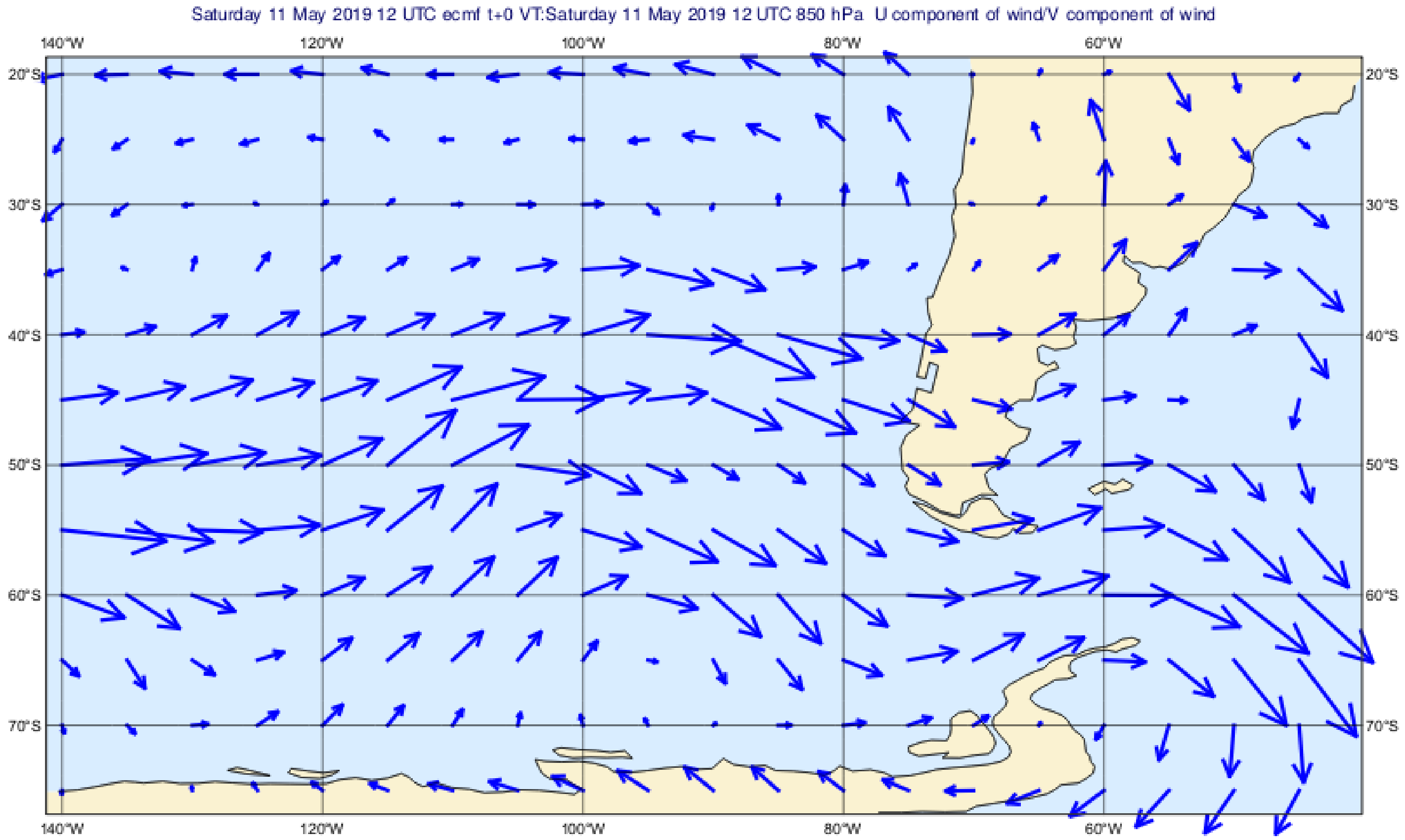
Show disabled parameters

Type: Vector Field

U Component: *Accepted icons:* my_u.grib

V Component: *Accepted icons:* my_v.grib

Colouring Field: *Accepted icons:*

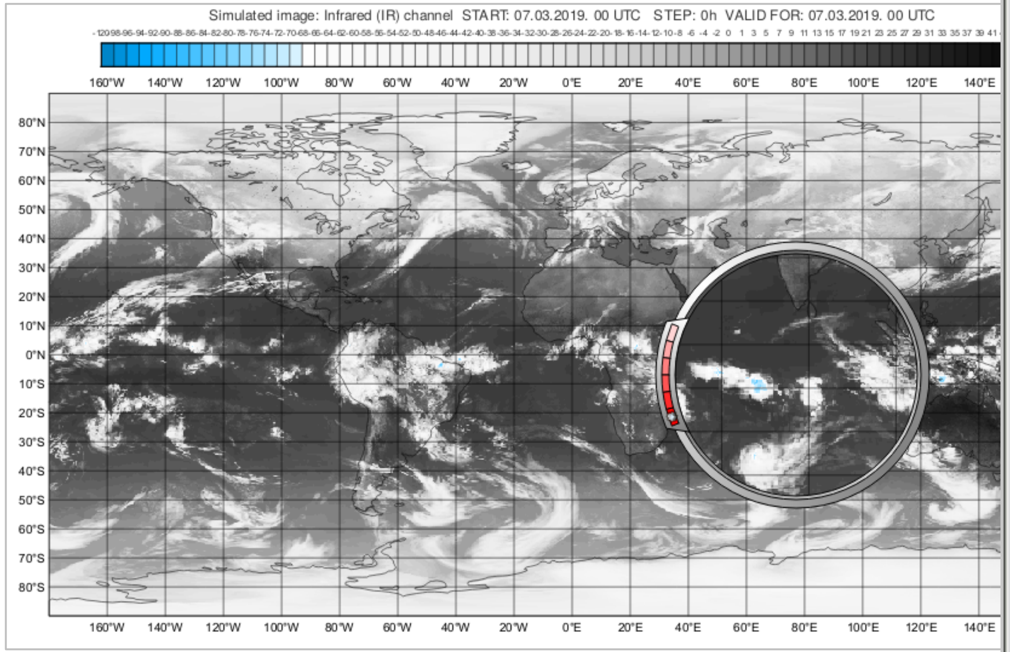


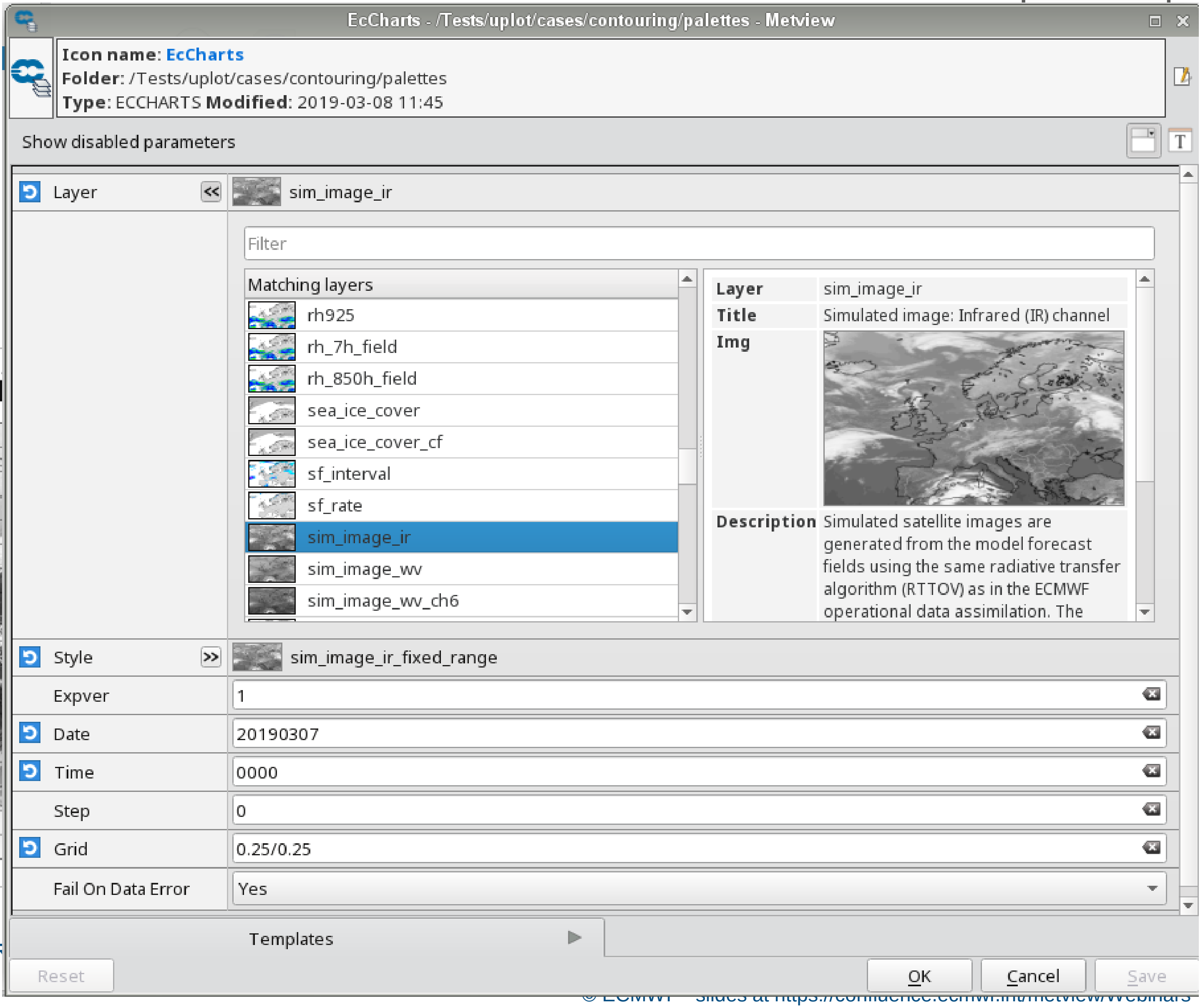
Saturday 11 May 2019 12 UTC ecmf t+0 VT:Saturday 11 May 2019 12 UTC 850 hPa U component of wind/V component of wind

Complete ecCharts Layers

- The ecCharts icon combines data and styling – retrieves pre-defined data from MARS and styles it as per ecCharts

EcCharts





EcCharts - /Tests/uplot/cases/contouring/palettes - Metview

Icon name: EcCharts
Folder: /Tests/uplot/cases/contouring/palettes
Type: ECCHARTS **Modified:** 2019-03-08 11:45

Show disabled parameters

Layer: sim_image_ir

Filter

Matching layers:
rh925
rh_7h_field
rh_850h_field
sea_ice_cover
sea_ice_cover_cf
sf_interval
sf_rate
sim_image_ir
sim_image_wv
sim_image_wv_ch6

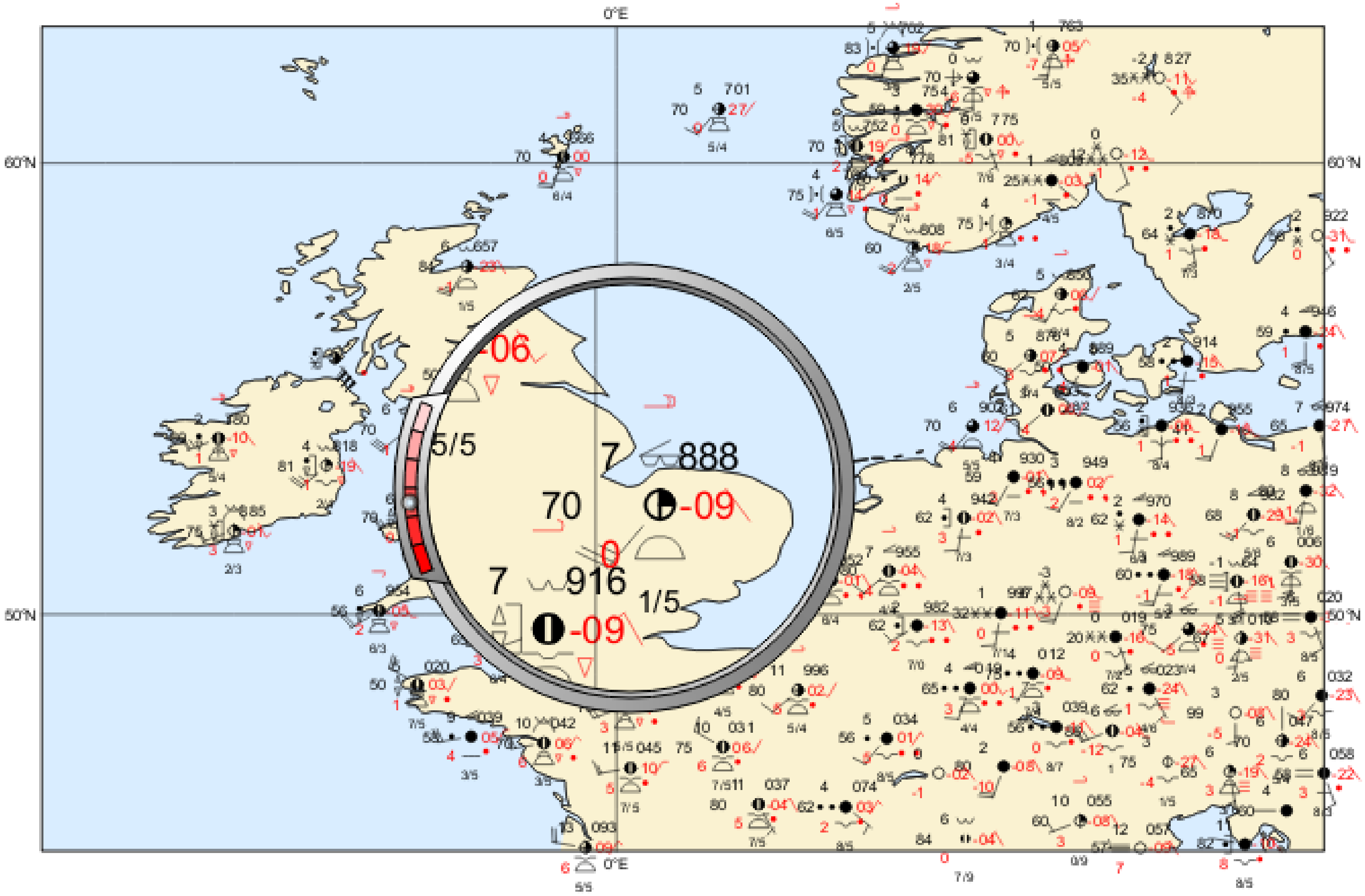
Layer	sim_image_ir
Title	Simulated image: Infrared (IR) channel
Img	
Description	Simulated satellite images are generated from the model forecast fields using the same radiative transfer algorithm (RTTOV) as in the ECMWF operational data assimilation. The

Style	sim_image_ir_fixed_range
Expver	1
Date	20190307
Time	0000
Step	0
Grid	0.25/0.25
Fail On Data Error	Yes

BUFR Plotting

BUFR

- BUFR is a very flexible format!
- Designed to store conventional observations
- But if the files follow some standard templates, we can plot them directly

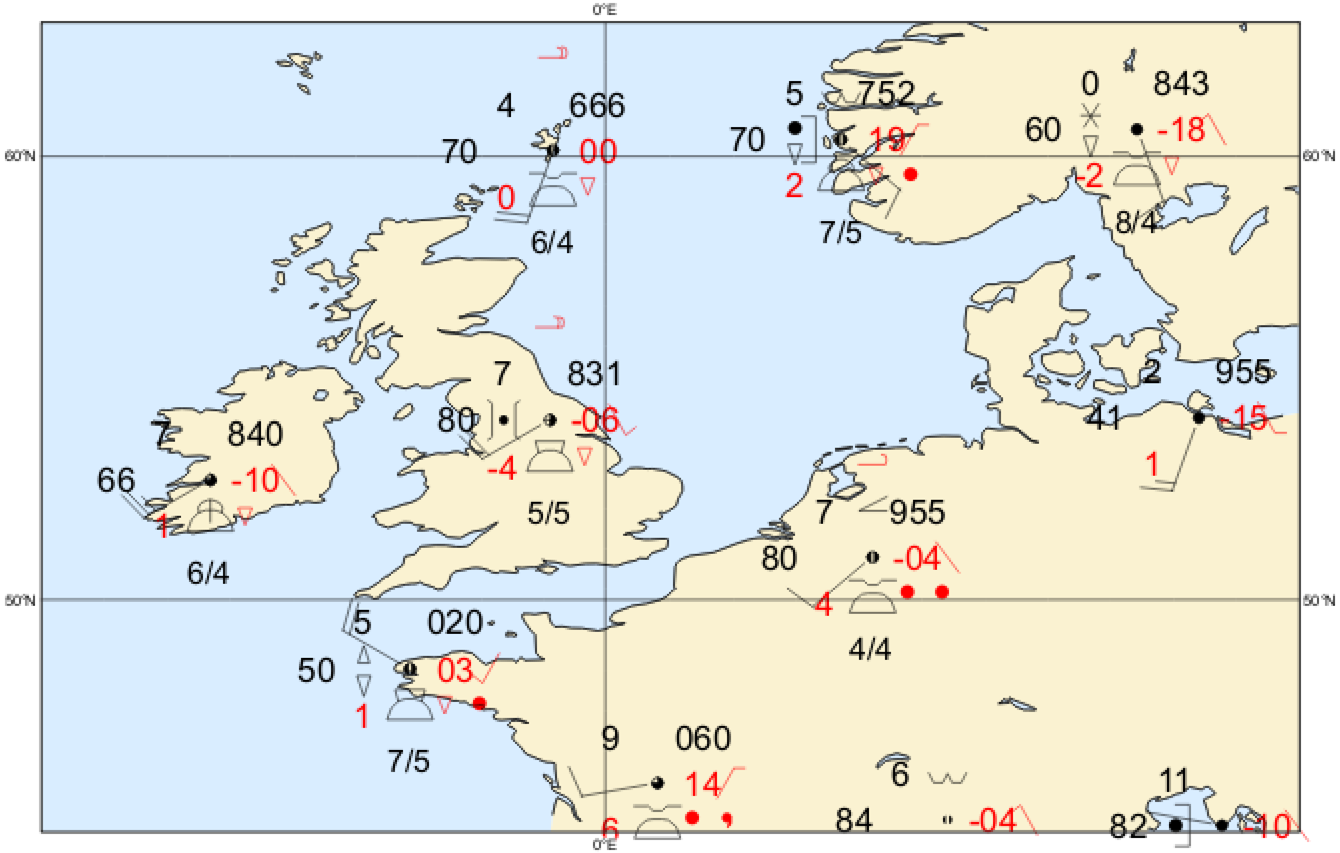


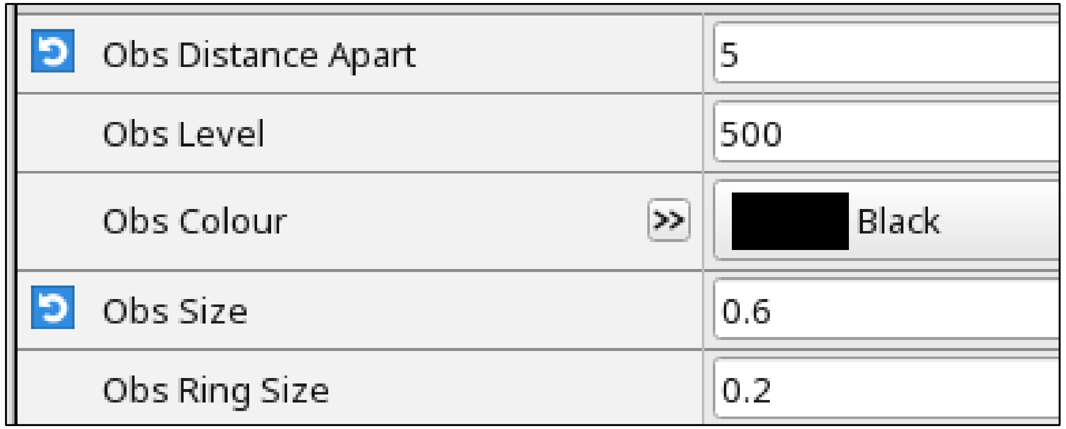
BUFR Plotting

- Customisation is via the **Observation Plotting** icon, e.g. thinning and size

Observation Plotting

BUFR

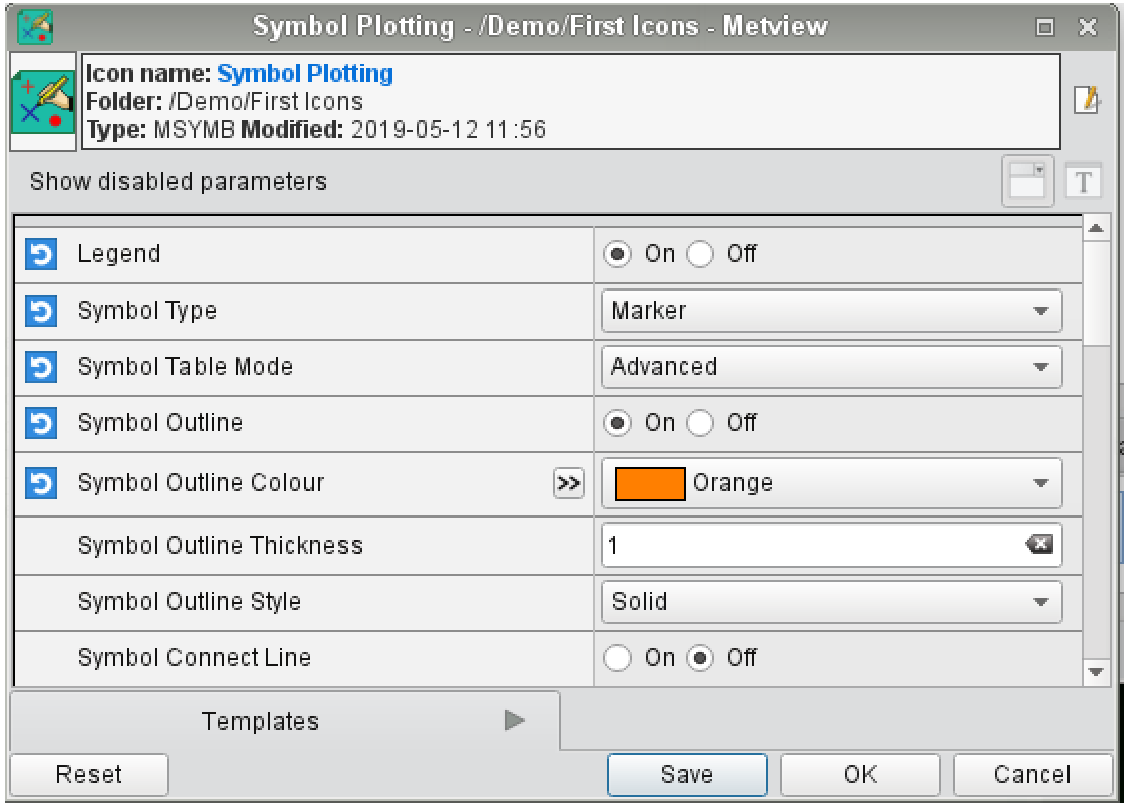




Obs Distance Apart	5
Obs Level	500
Obs Colour	Black
Obs Size	0.6
Obs Ring Size	0.2

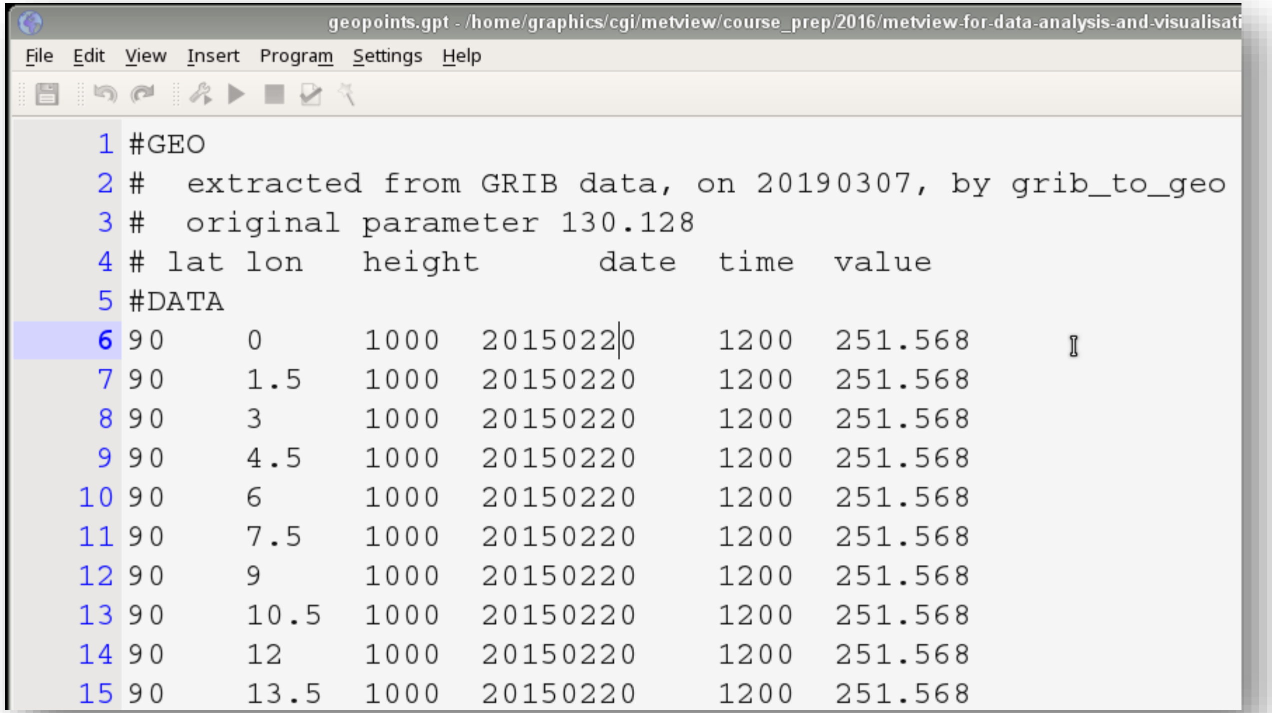
Scattered data

- Geopoints, CSV, lists of values, ODB, NetCDF
- Use **Symbol Plotting** icon to apply styling



Icon name: Symbol Plotting
Folder: /Demo/First Icons
Type: MSYMB Modified: 2019-05-12 11:56

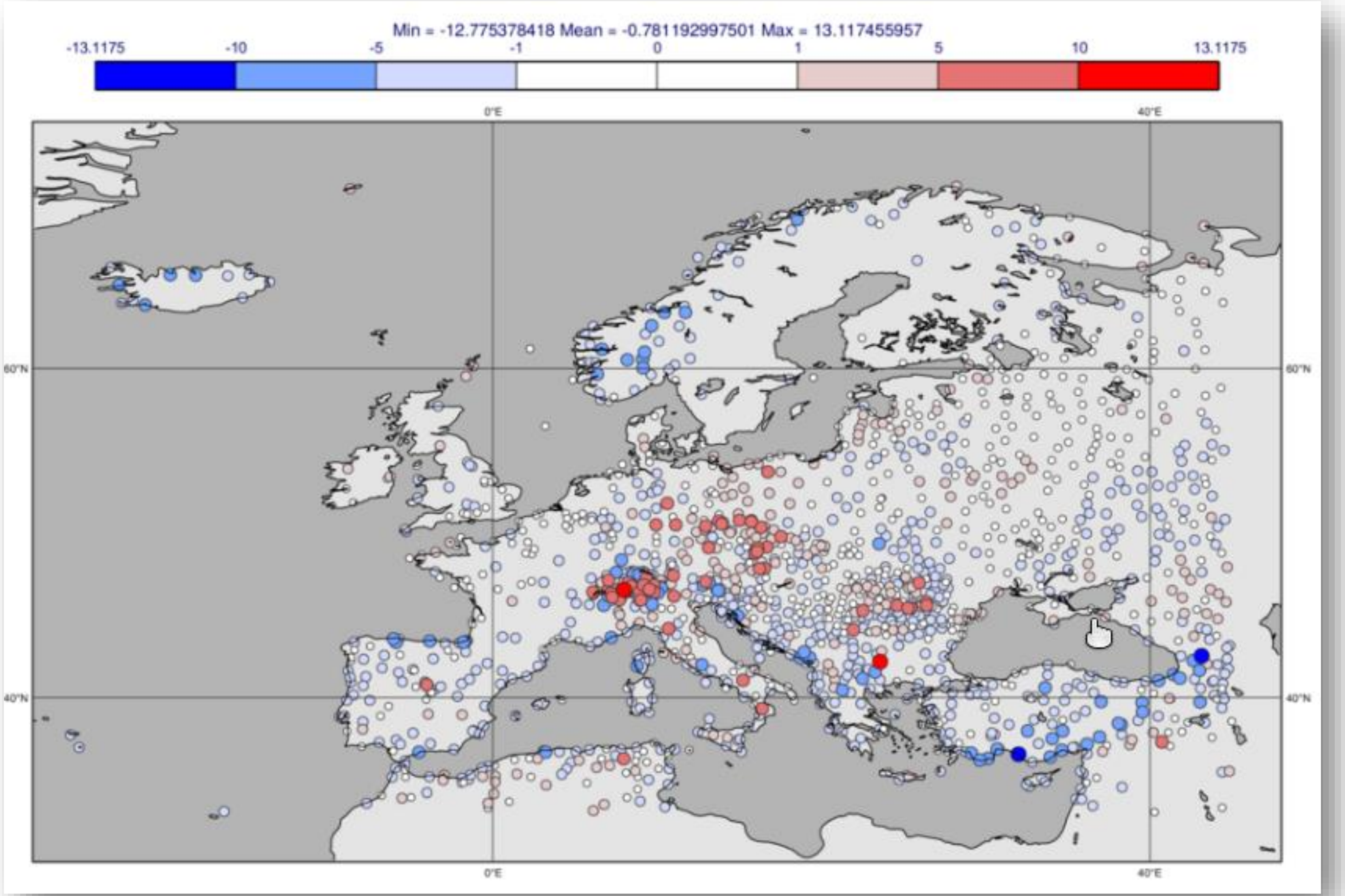
Legend ● On ○ Off
Symbol Type: Marker
Symbol Table Mode: Advanced
Symbol Outline ● On ○ Off
Symbol Outline Colour: Orange
Symbol Outline Thickness: 1
Symbol Outline Style: Solid
Symbol Connect Line ○ On ● Off


geopoints.gpt - /home/graphics/cgi/metview/course_prep/2016/metview-for-data-analysis-and-visualisati
File
Edit
View
Insert
Program
Settings
Help

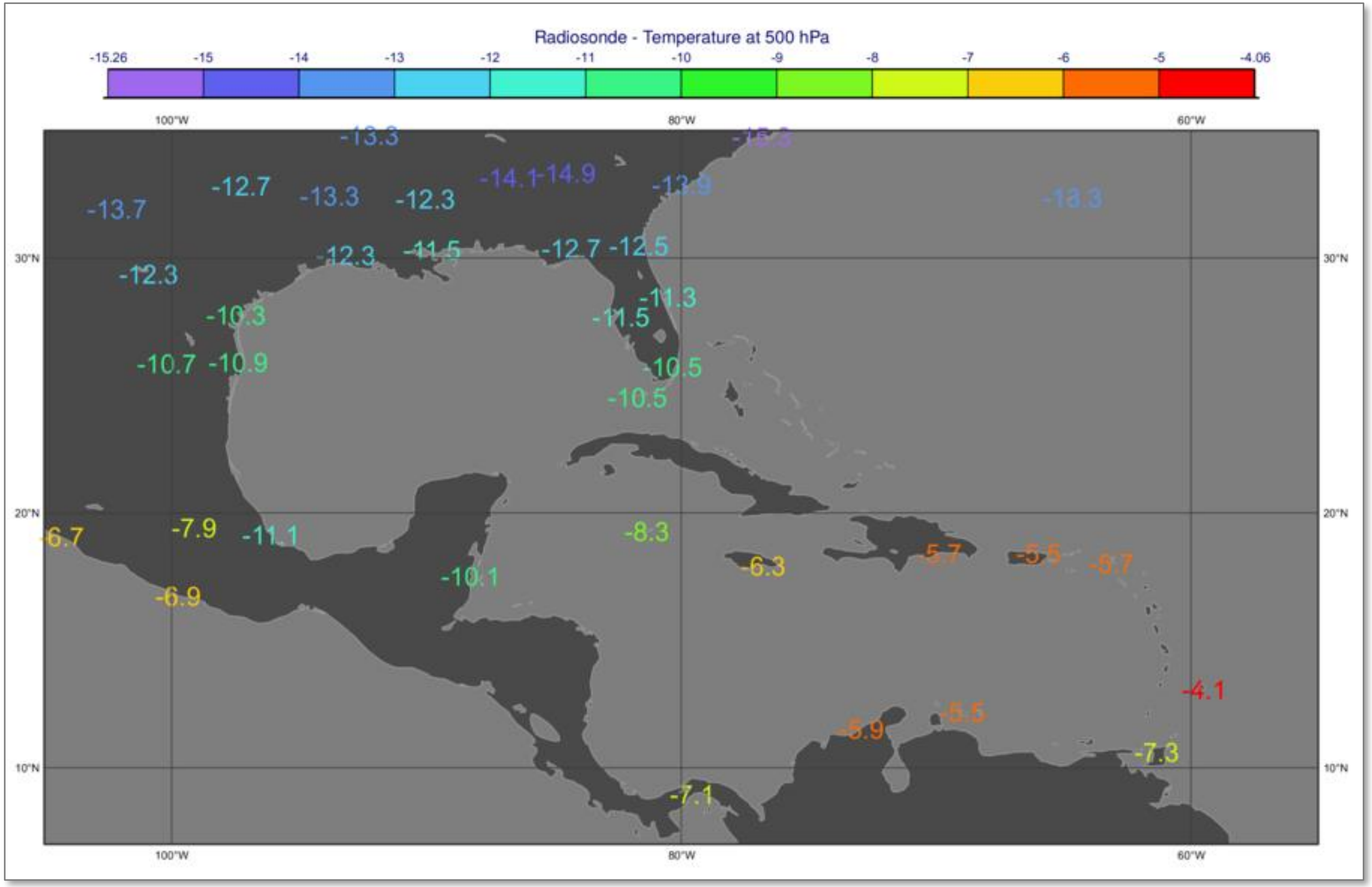
```
#GEO
#  extracted from GRIB data, on 20190307, by grib_to_geo
#  original parameter 130.128
# lat lon   height      date   time   value
#DATA
90     0     1000   20150220     1200   251.568
90     1.5   1000   20150220     1200   251.568
90     3     1000   20150220     1200   251.568
90     4.5   1000   20150220     1200   251.568
90     6     1000   20150220     1200   251.568
90     7.5   1000   20150220     1200   251.568
90     9     1000   20150220     1200   251.568
90     10.5  1000   20150220     1200   251.568
90     12    1000   20150220     1200   251.568
90     13.5  1000   20150220     1200   251.568
```

Geopoints format – columns in ASCII text

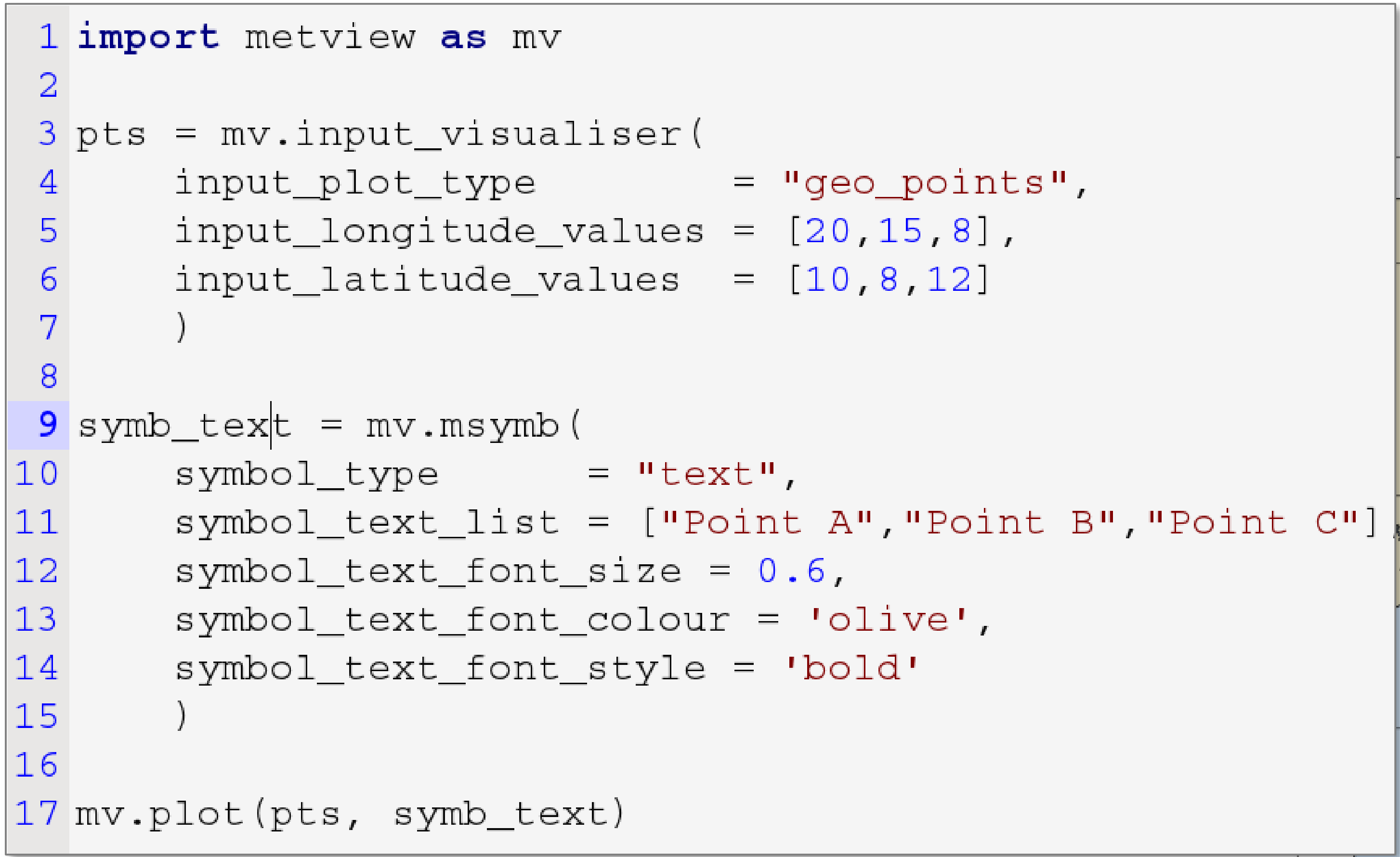
Scattered data plotting – Symbol Type = Marker


Min = -12.775378418 Mean = -0.781192997501 Max = 13.117455957
-13.1175
-10
-5
-1
0
1
5
10
13.1175
0°E
40°E
60°N
40°N
0°E
40°E

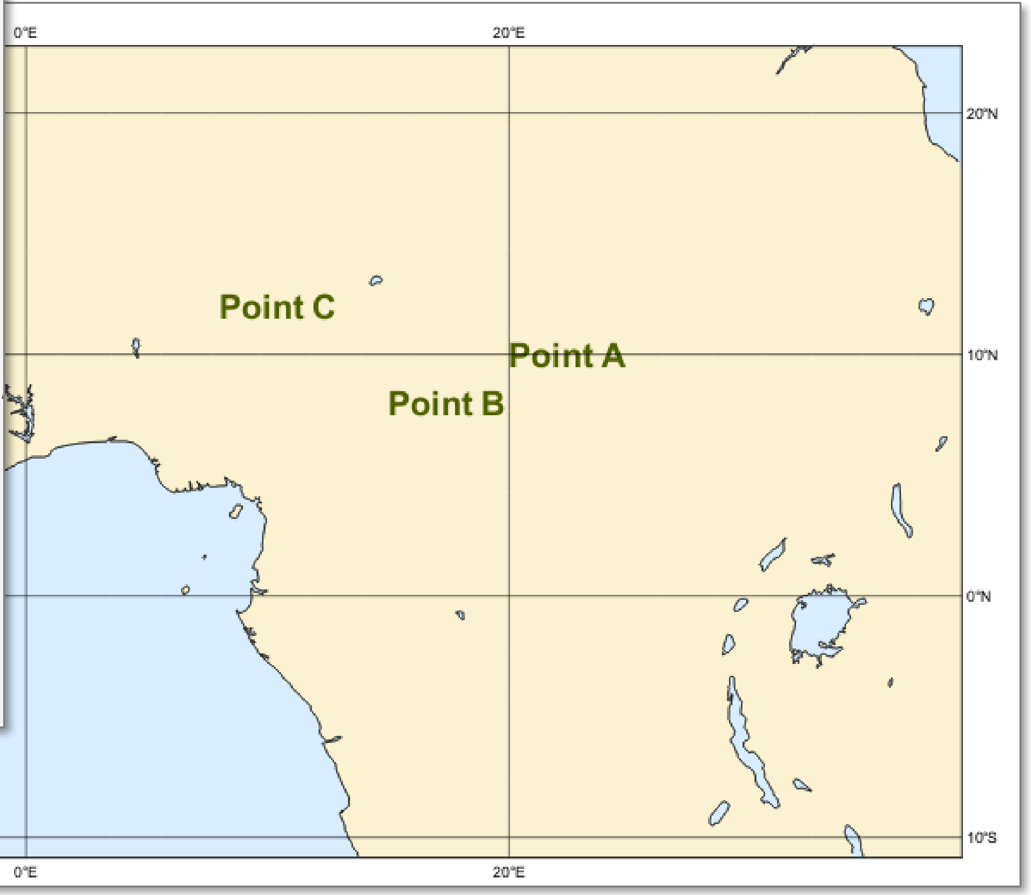
Scattered data plotting – Symbol Type = Number



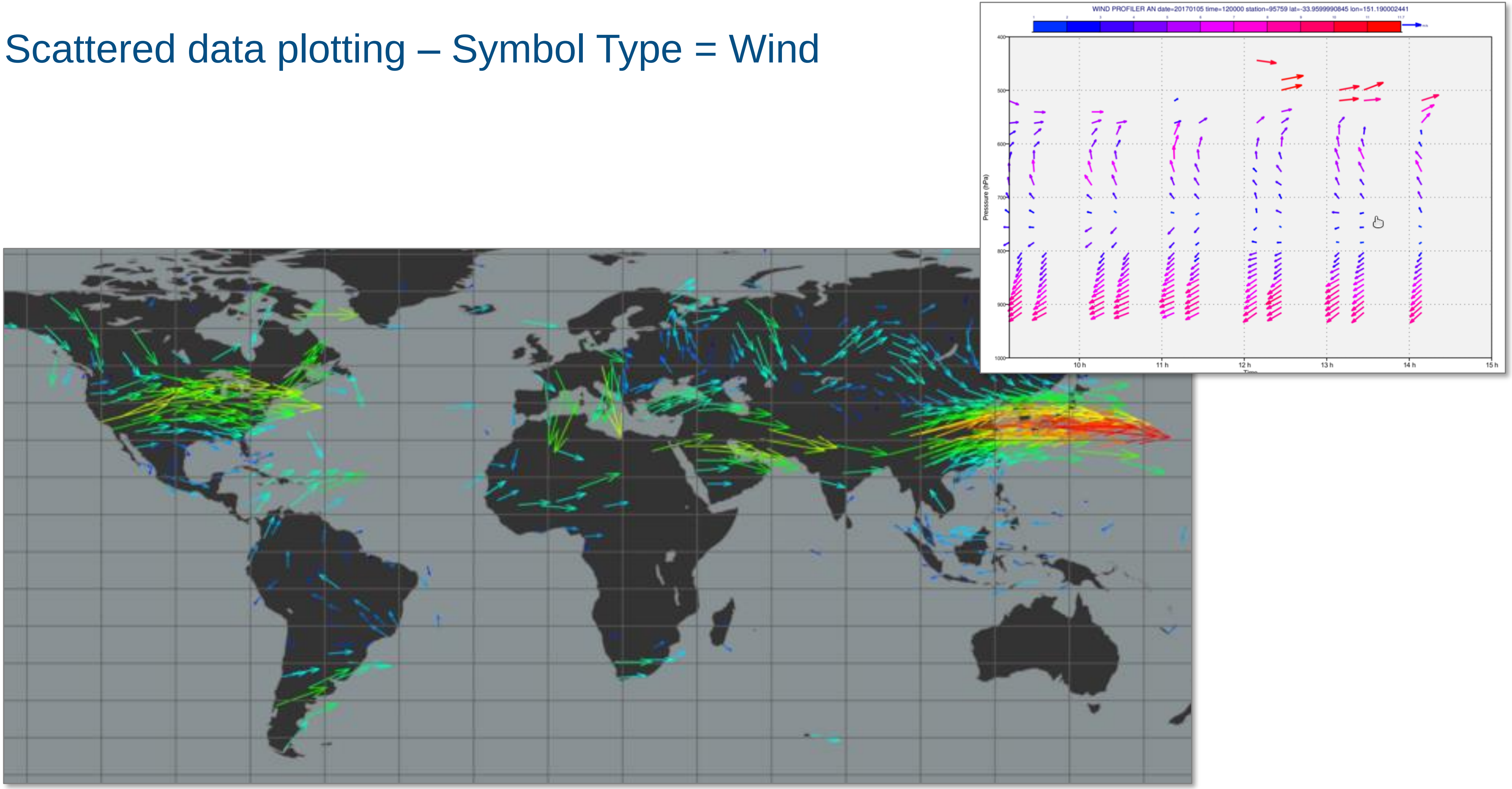
Scattered data plotting – Symbol Type = Text



```python
import metview as mv

pts = mv.input_visualiser(
    input_plot_type          = "geo_points",
    input_longitude_values = [20,15,8],
    input_latitude_values  = [10,8,12]
    )

symb_text = mv.msymb(
    symbol_type       = "text",
    symbol_text_list = ["Point A","Point B","Point C"]
    symbol_text_font_size = 0.6,
    symbol_text_font_colour = 'olive',
    symbol_text_font_style = 'bold'
    )

mv.plot(pts, symb_text)
```

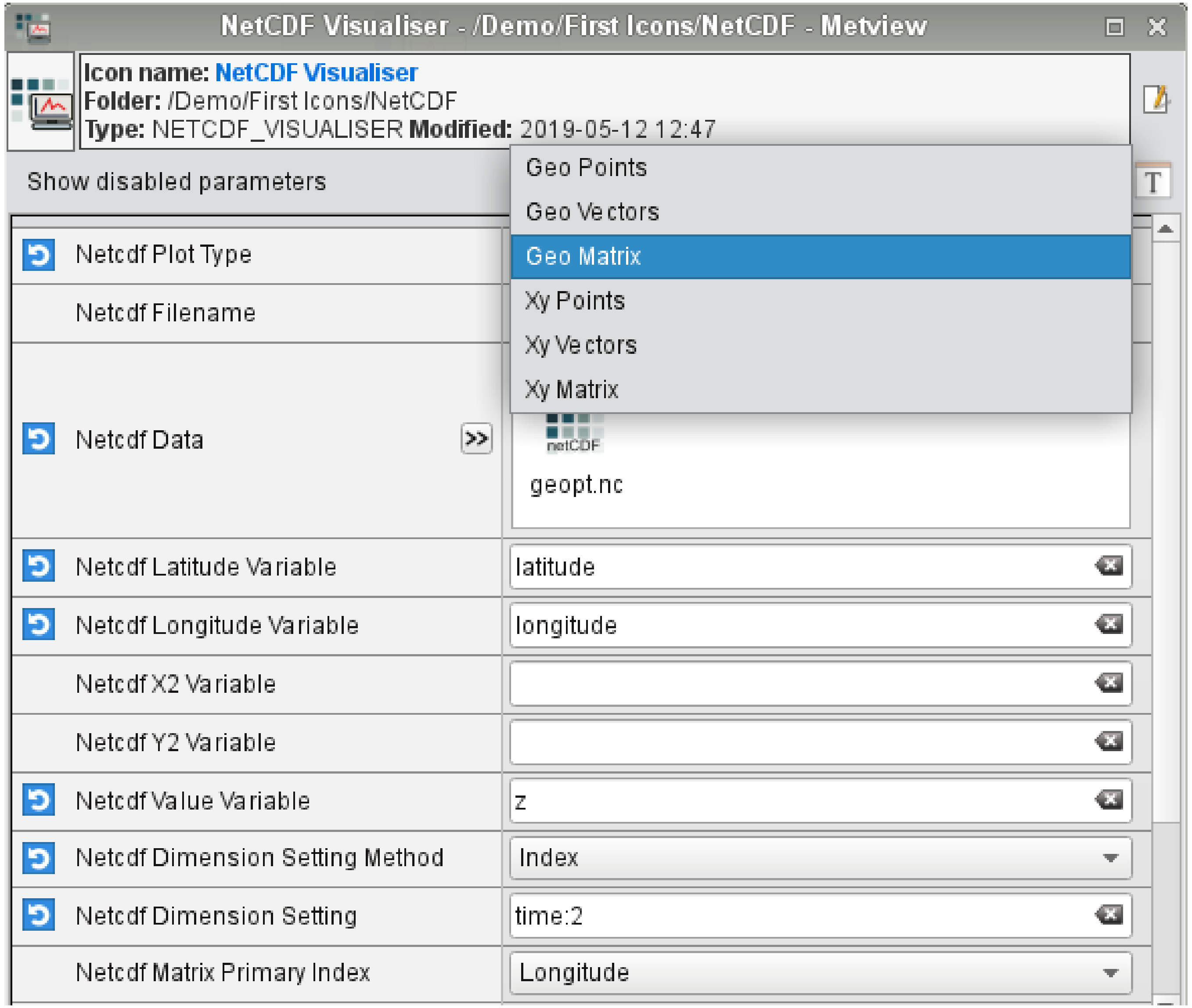


Scattered data plotting – Symbol Type = Wind


WIND PROFILER AN date=20170105 time=120000 station=95759 lat=-33.9599990845 lon=151.190002441
1
2
3
4
5
6
7
8
9
10
11
11.7
m/s
Presssure (hPa)
400
500
600
700
800
900
1000
10 h
11 h
12 h
13 h
14 h
15 h
Time

NetCDF Specifics

- NetCDF is a very flexible format
- Use the **NetCDF Visualiser** icon to tell Metview which variable / dimensions you wish to plot



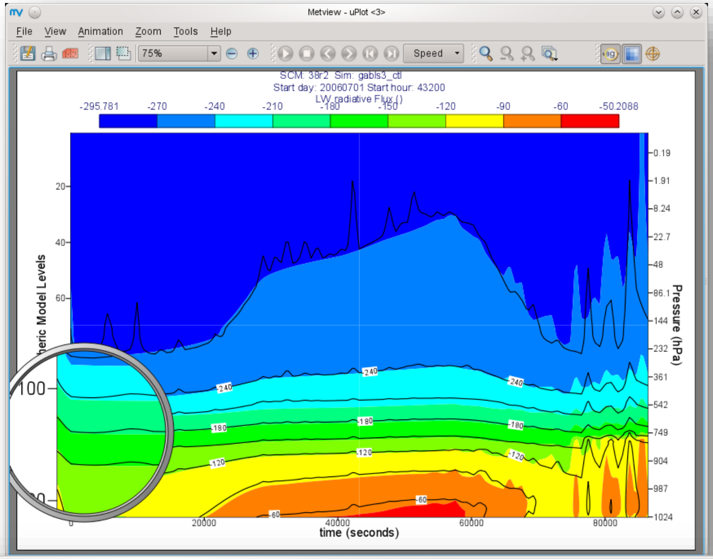
NetCDF Visualiser - /Demo/First Icons/NetCDF - Metview

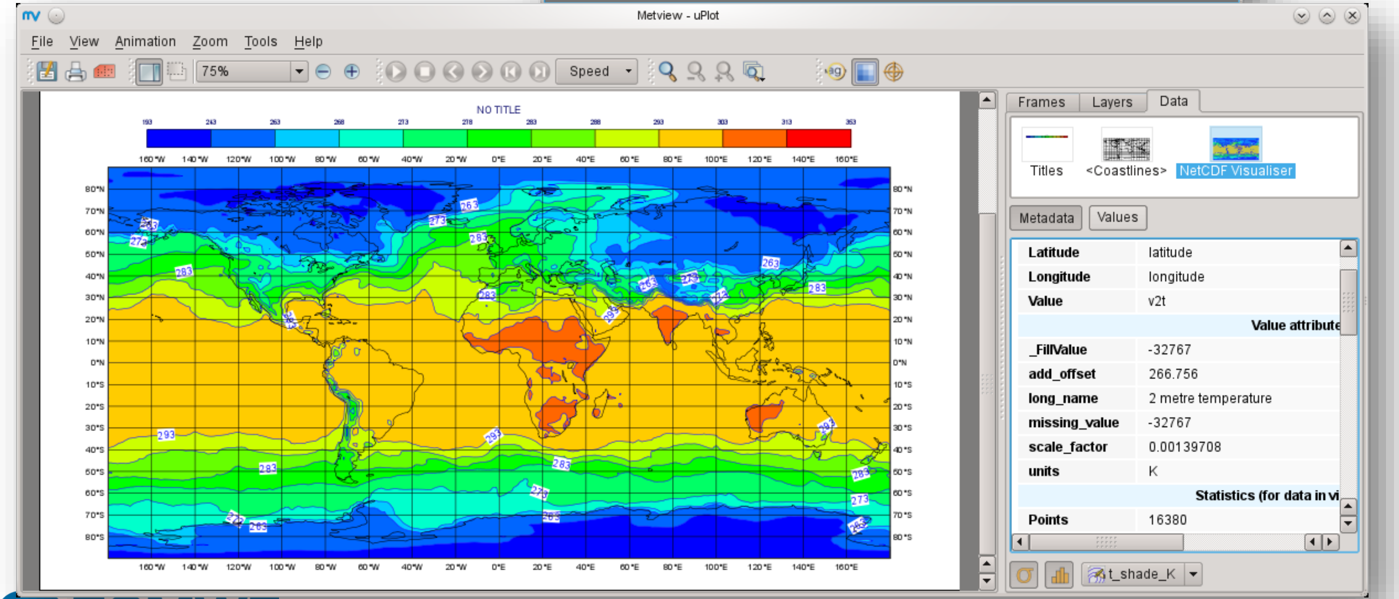
Icon name: NetCDF Visualiser
Folder: /Demo/First Icons/NetCDF
Type: NETCDF_VISUALISER **Modified:** 2019-05-12 12:47

Netcdf Plot Type	Geo Matrix
Netcdf Filename	
Netcdf Data	geopt.nc
Netcdf Latitude Variable	latitude
Netcdf Longitude Variable	longitude
Netcdf X2 Variable	
Netcdf Y2 Variable	
Netcdf Value Variable	z
Netcdf Dimension Setting Method	Index
Netcdf Dimension Setting	time:2
Netcdf Matrix Primary Index	Longitude

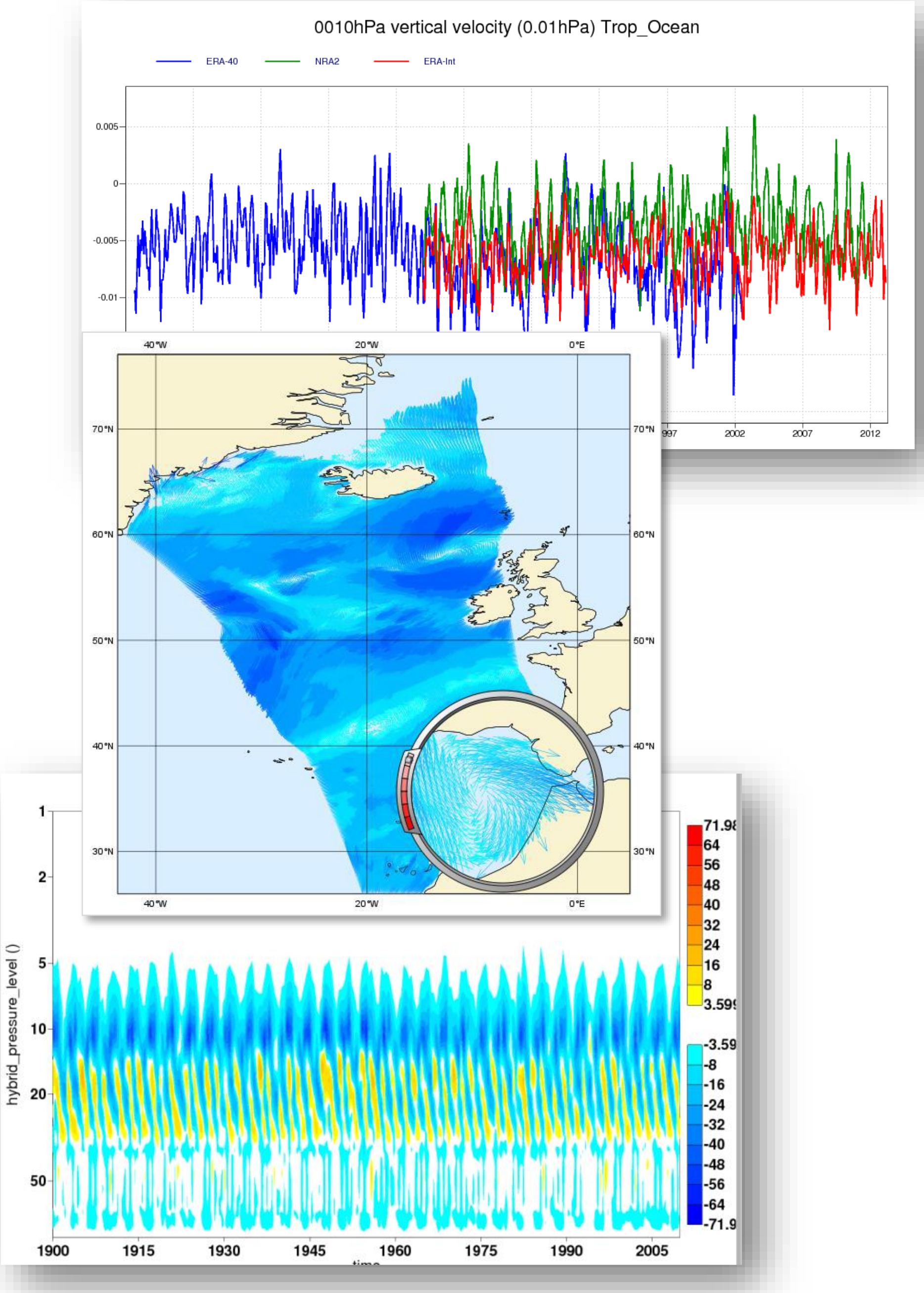
Example NetCDF plots

- Matrix data uses **Contouring**
- Point data uses **Symbol Plotting**
- Line data uses **Graph Plotting**





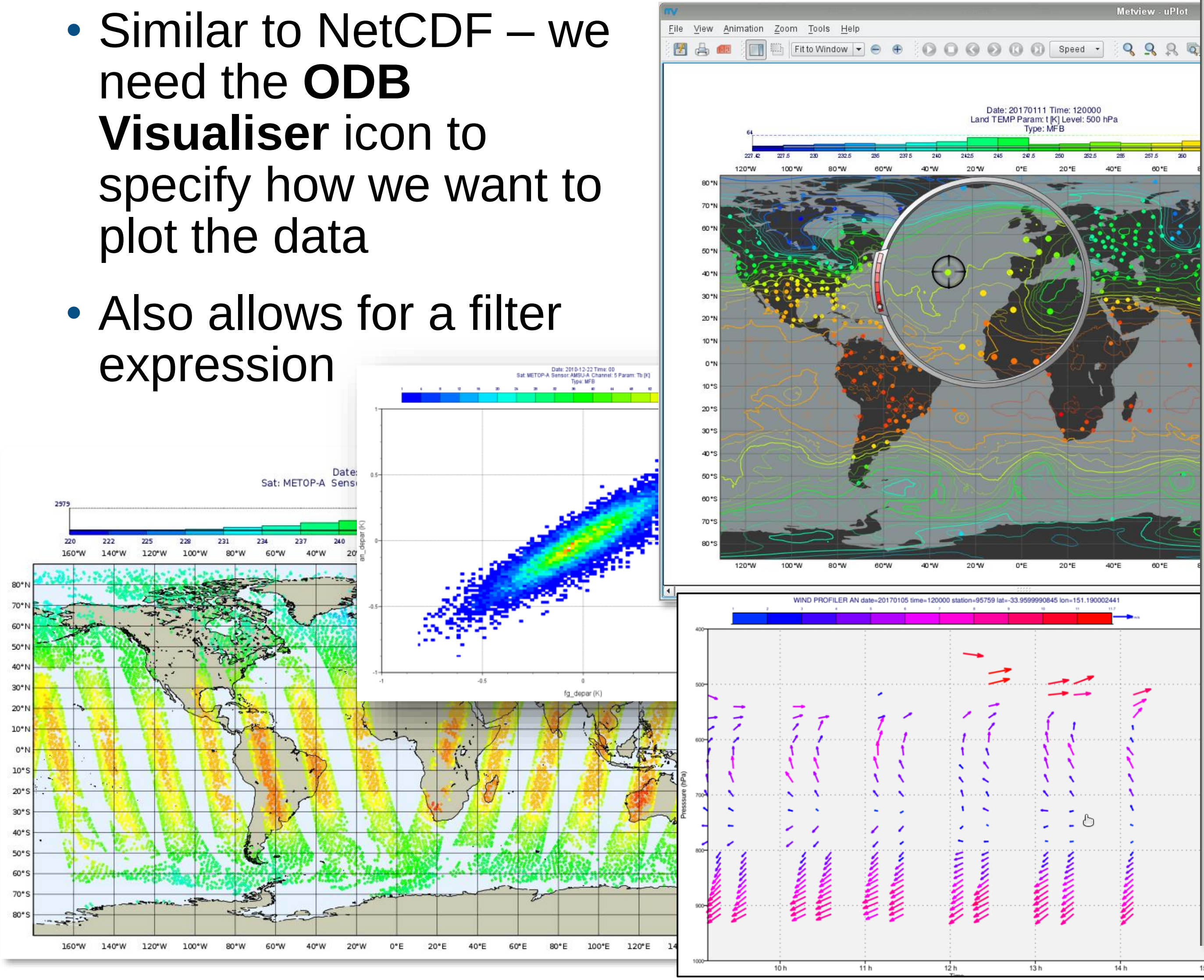
Latitude	latitude
Longitude	longitude
Value	v2t
	Value attribute
_FillValue	-32767
add_offset	266.756
long_name	2 metre temperature
missing_value	-32767
scale_factor	0.00139708
units	K
	Statistics (for data in vi
Points	16380

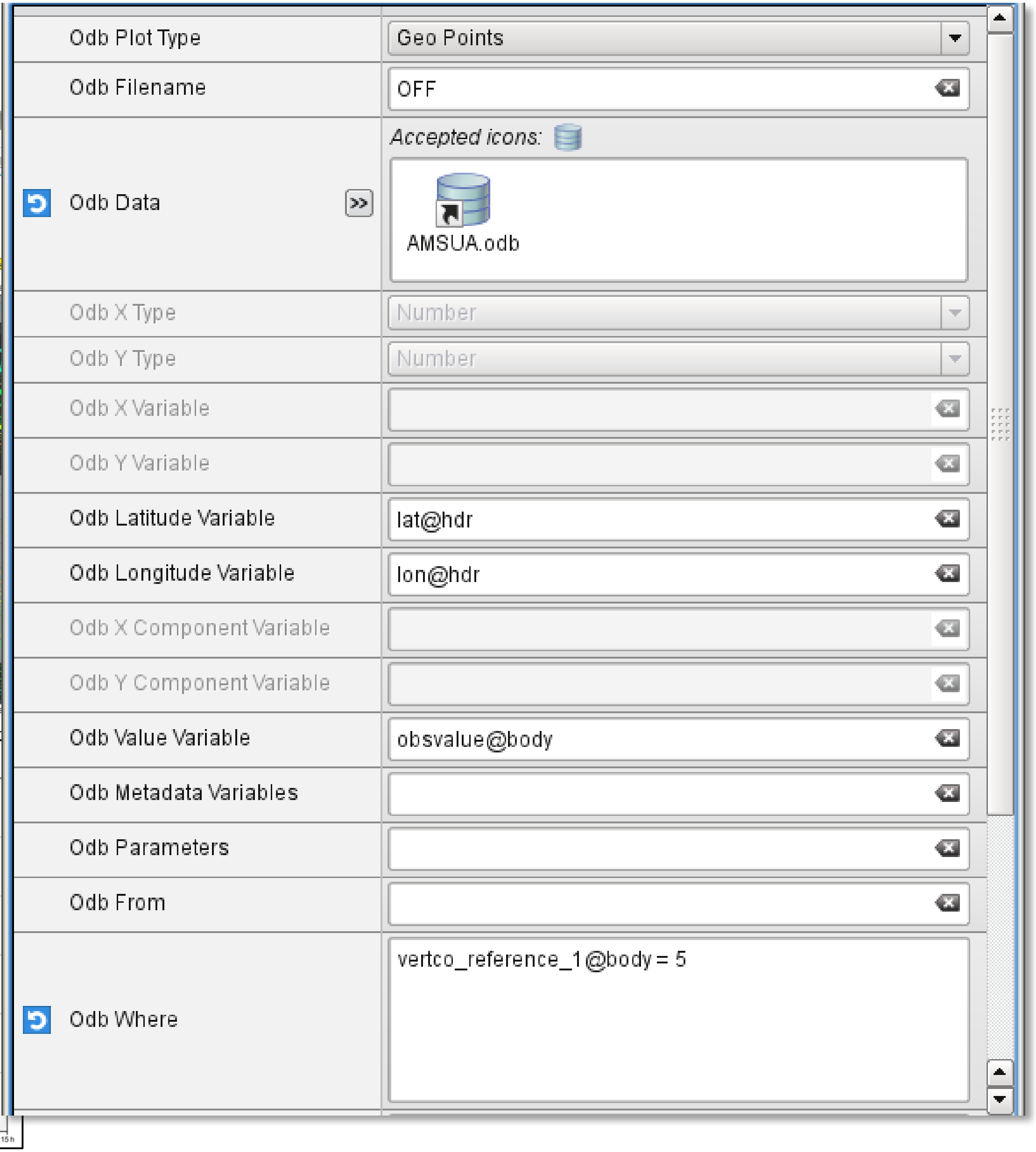


ODB Specifics

Developed at ECMWF to handle observations in data assimilation

- Similar to NetCDF – we need the **ODB Visualiser** icon to specify how we want to plot the data
- Also allows for a filter expression

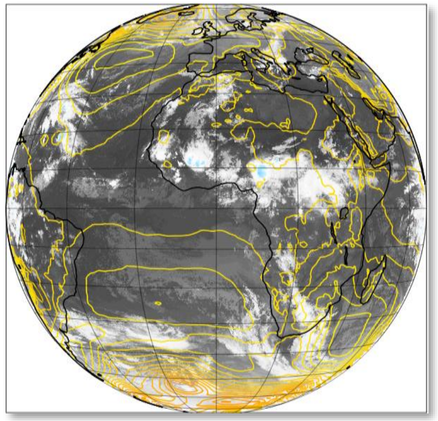




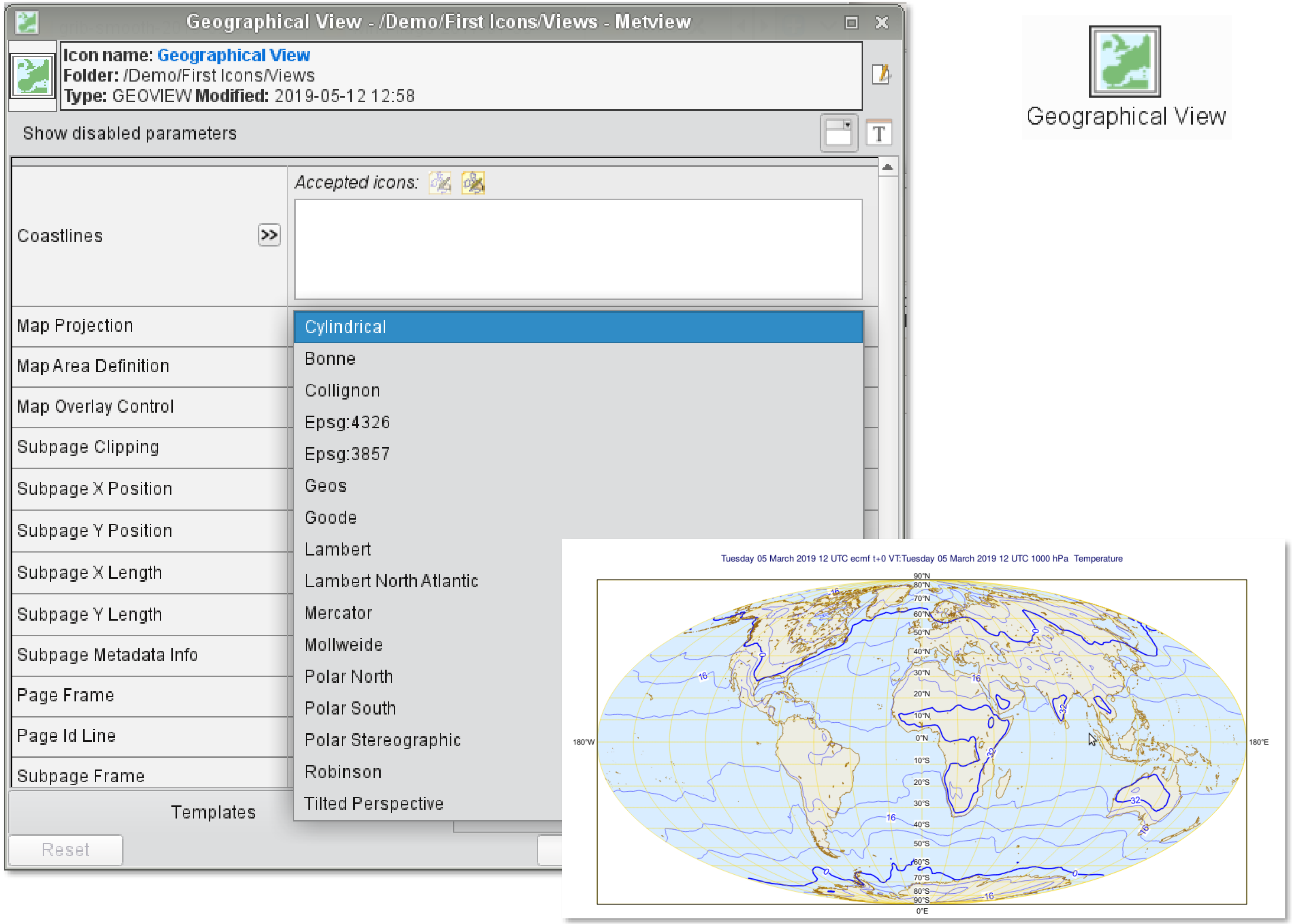
Odb Plot Type	Geo Points
Odb Filename	OFF
Odb Data	Accepted icons: AMSUA.odb
Odb X Type	Number
Odb Y Type	Number
Odb X Variable	
Odb Y Variable	
Odb Latitude Variable	lat@hdr
Odb Longitude Variable	lon@hdr
Odb X Component Variable	
Odb Y Component Variable	
Odb Value Variable	obsvalue@body
Odb Metadata Variables	
Odb Parameters	
Odb From	
Odb Where	vertco_reference_1@body = 5

Geographic Views

- Use the **Geographical View** icon to choose map projections and to store sub-areas



Geographical View



Geographical View - /Demo/First Icons/Views - Metview

Icon name: Geographical View
Folder: /Demo/First Icons/Views
Type: GEOVIEW **Modified:** 2019-05-12 12:58

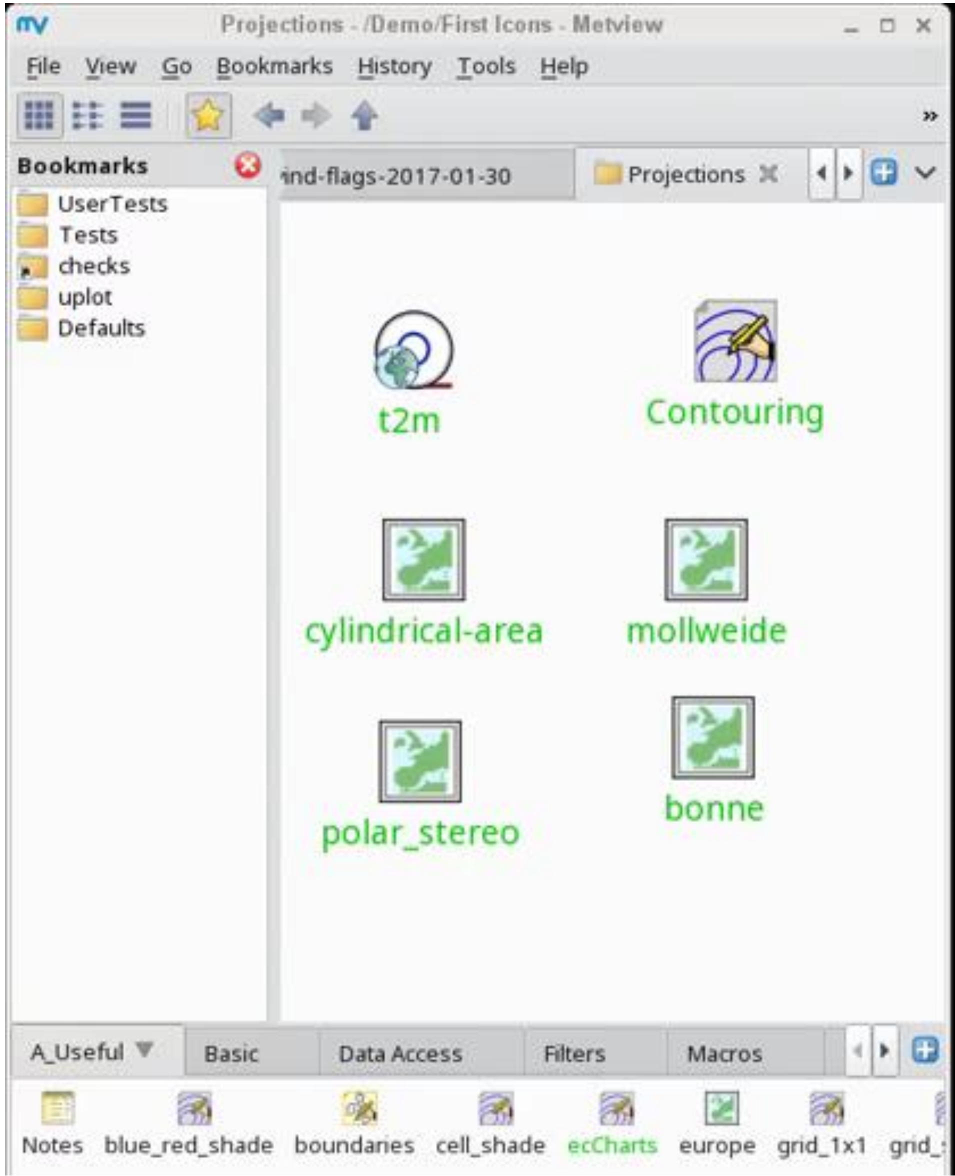
Show disabled parameters

Coastlines — *Accepted icons:*

Map Projection — Cylindrical

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

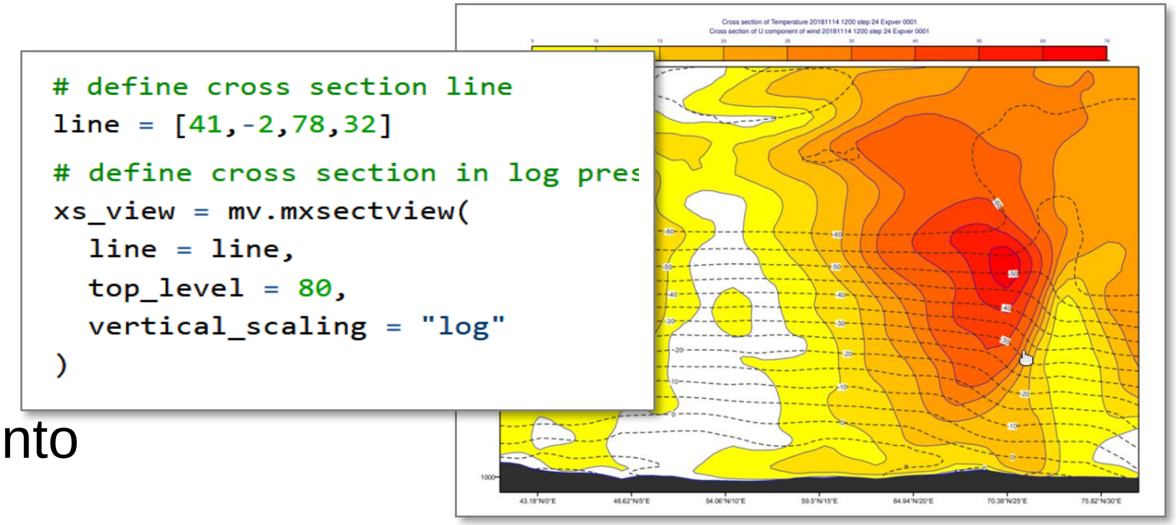
Geographic Views



Analysis Views

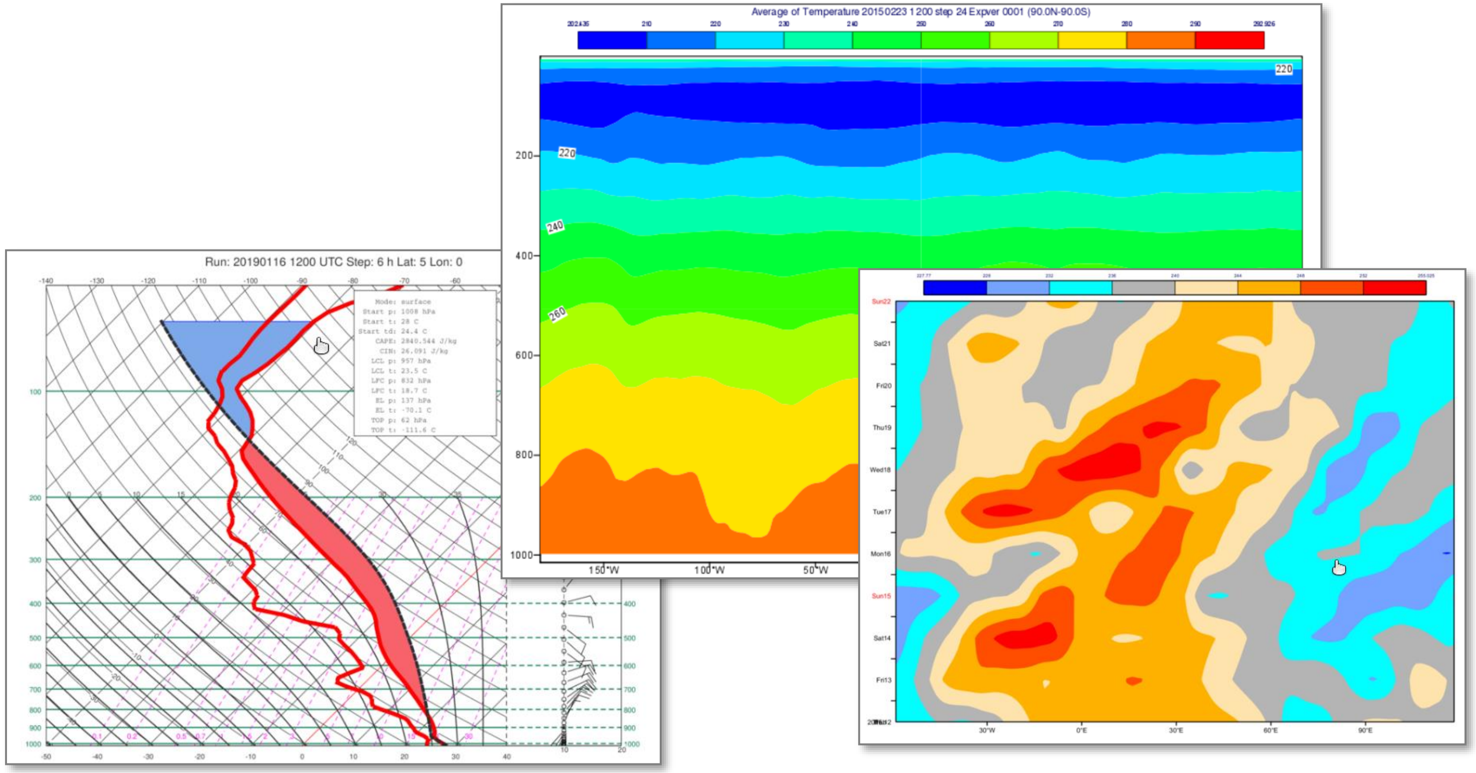
- Some views perform some processing and present the data in a different way
- Set up the particular **View** icon
- Visualise it, then drop data + visual definitions into the plot window

- Cross Section
- Vertical Profile
- Thermodynamic diagrams
- Average (zonal and meridional)
- Hovmoeller

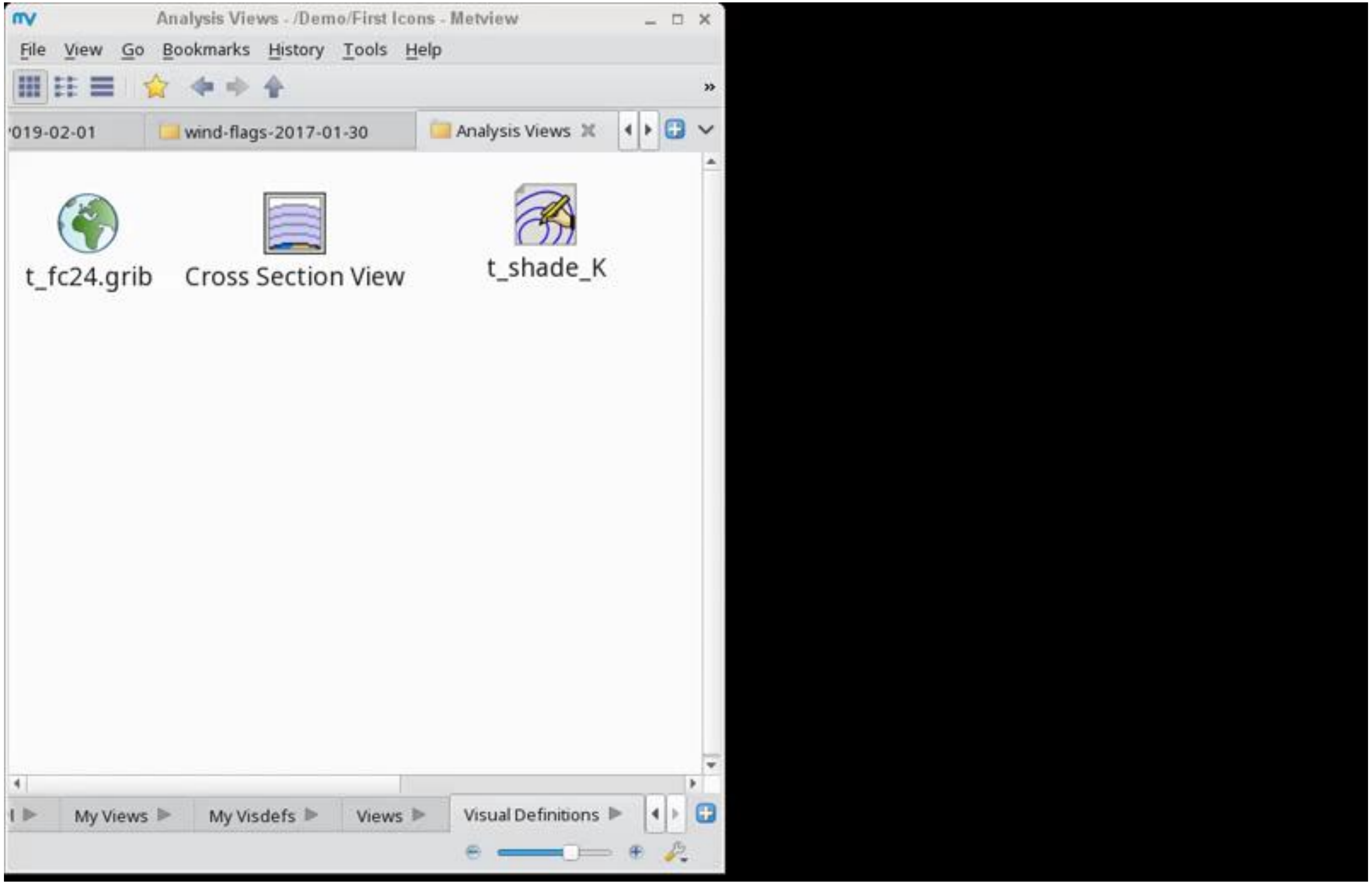


```
# define cross section line
line = [41,-2,78,32]
# define cross section in log pres
xs_view = mv.mxsectview(
  line = line,
  top_level = 80,
  vertical_scaling = "log"
)
```

Cross section of Temperature 20181114 1200 step 24 Expver 0001
Cross section of U component of wind 20181114 1200 step 24 Expver 0001



Cross Section Example

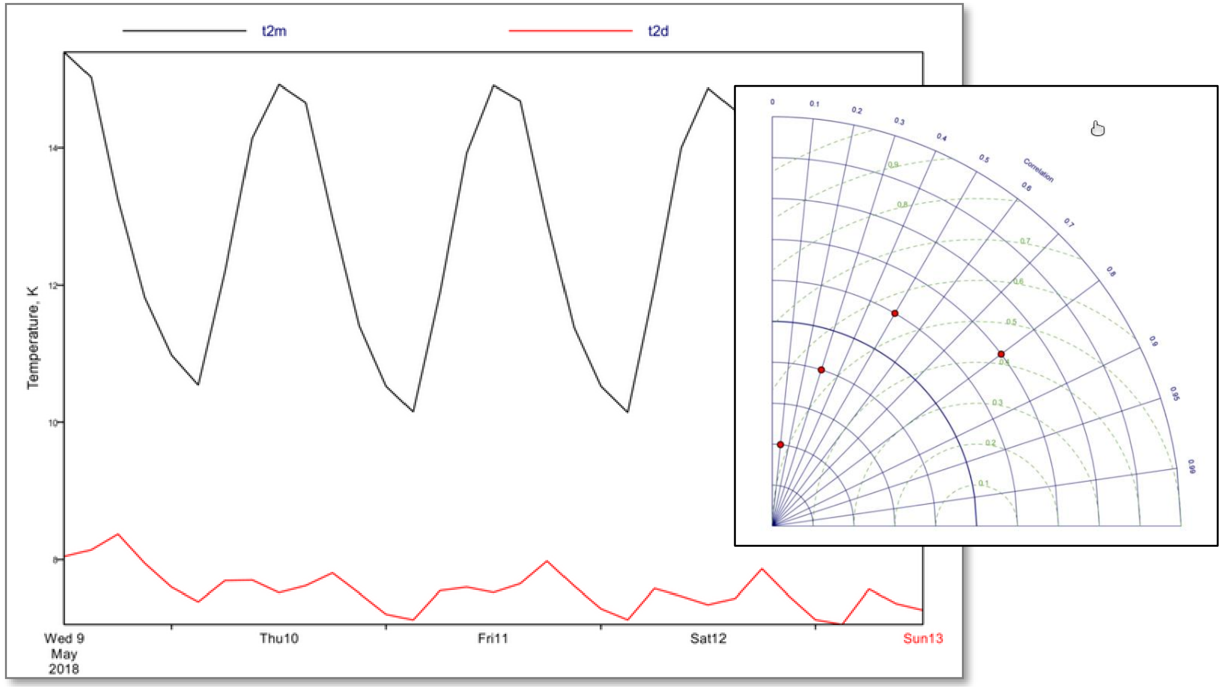


Cartesian View

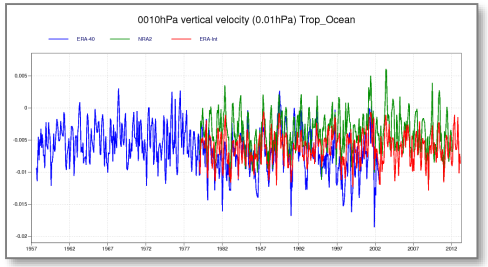
Cartesian View

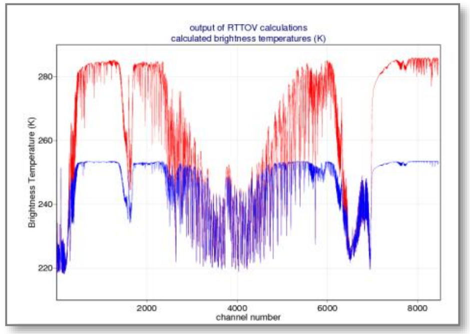
- Simple X/Y view for plotting ‘generic’ data, e.g. scatterplots, time series, … also Taylor diagrams

```
vaxis = mv.maxis(axis_title_text = 'Temperature, K',
                 axis_title_height = 0.5)

ts_view = mv.cartesianview(
    x_automatic = "on",
    x_axis_type = "date",
    y_automatic = "on",
    horizontal_axis = haxis,
    vertical_axis   = vaxis)
```



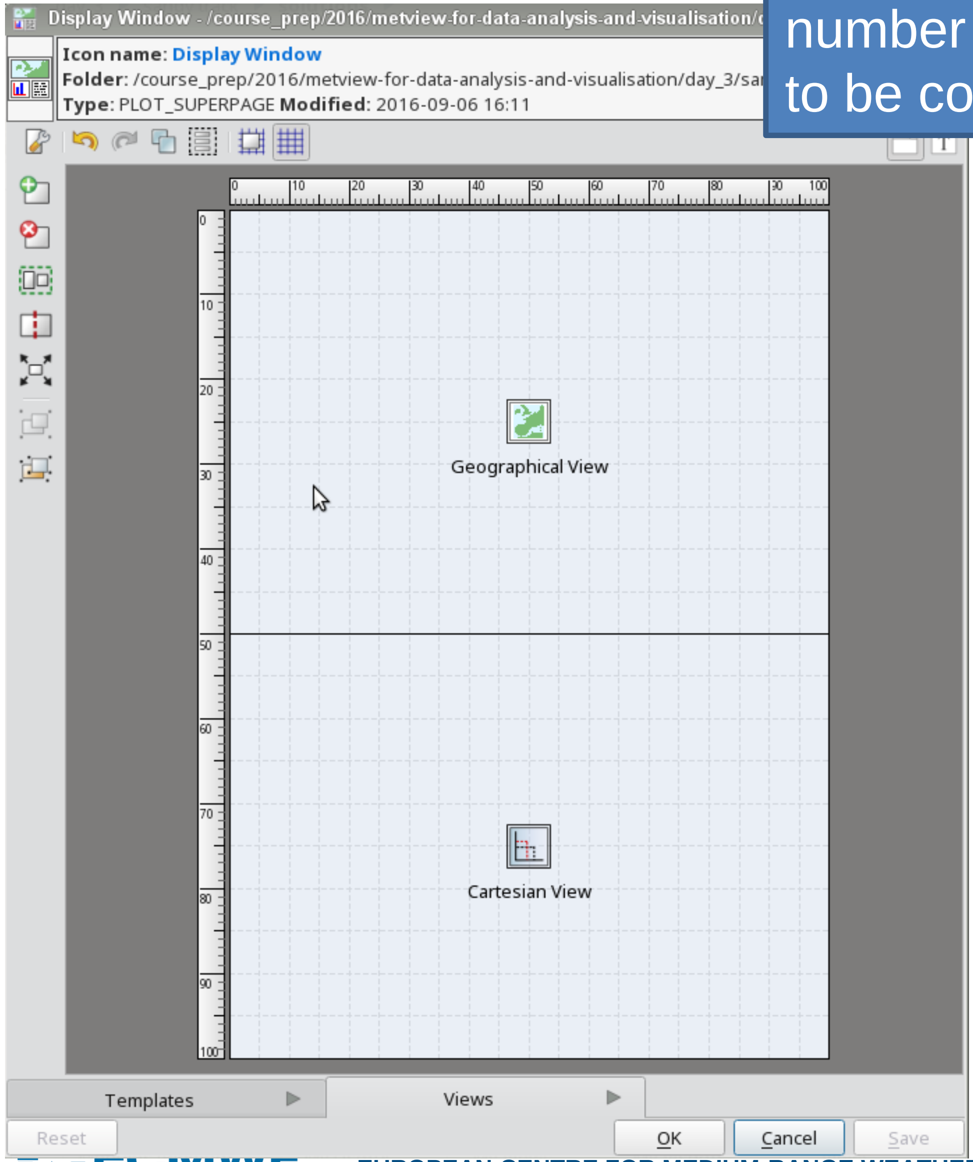
```
# plot everything into the Cartesian view
mv.plot(ts_view, curve_2t, graph_2t, curve_2d, graph_2d, legend)
```





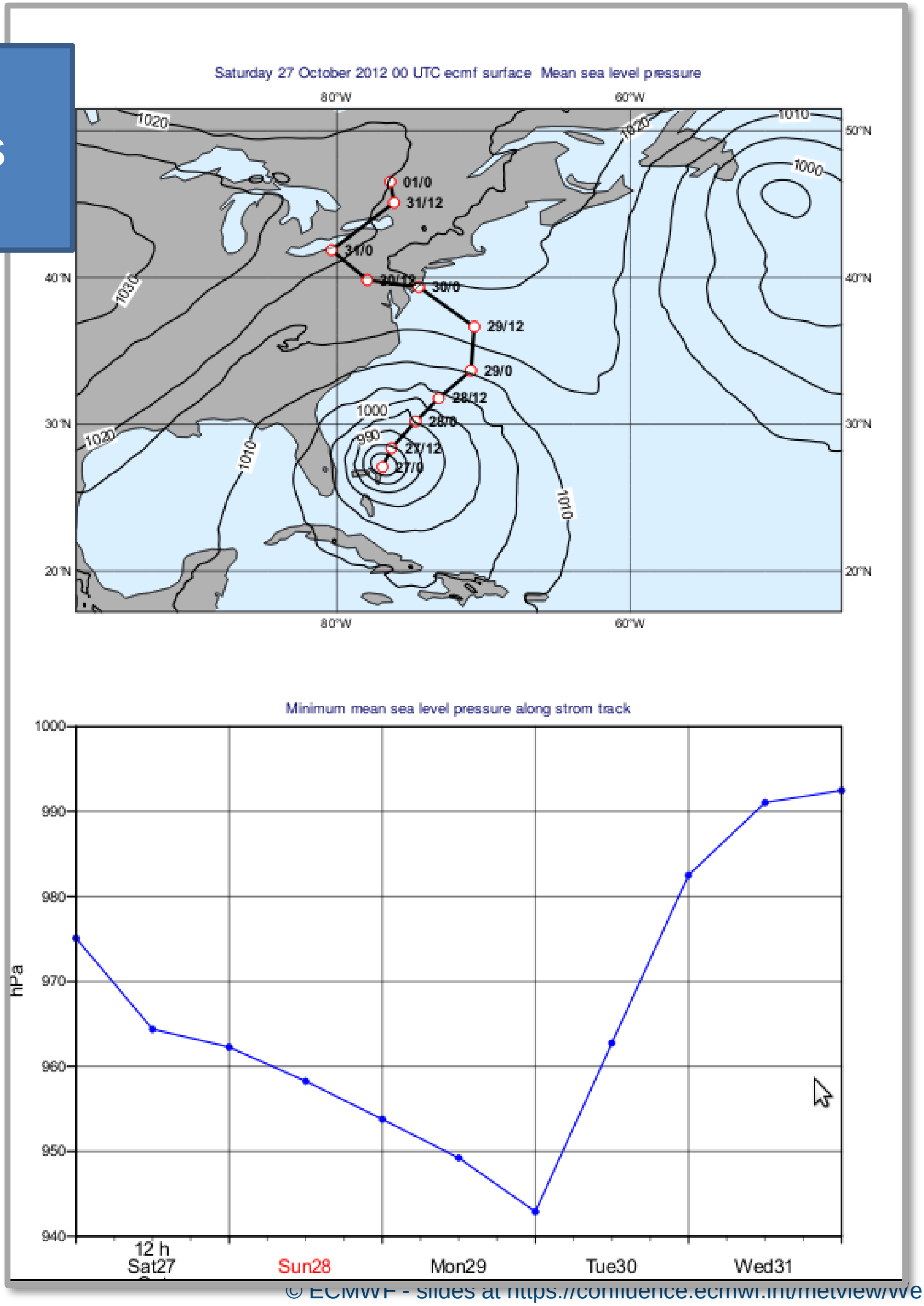
Layout (Display Window)

Layout editor allows any number of different views to be combined



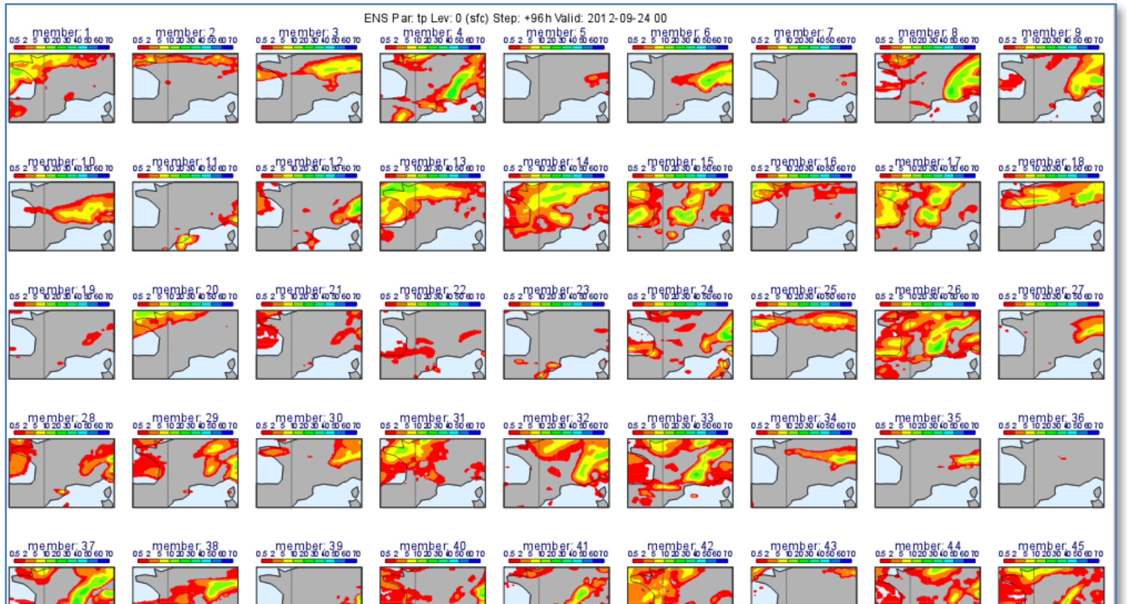
Display Window - /course_prep/2016/metview-for-data-analysis-and-visualisation/

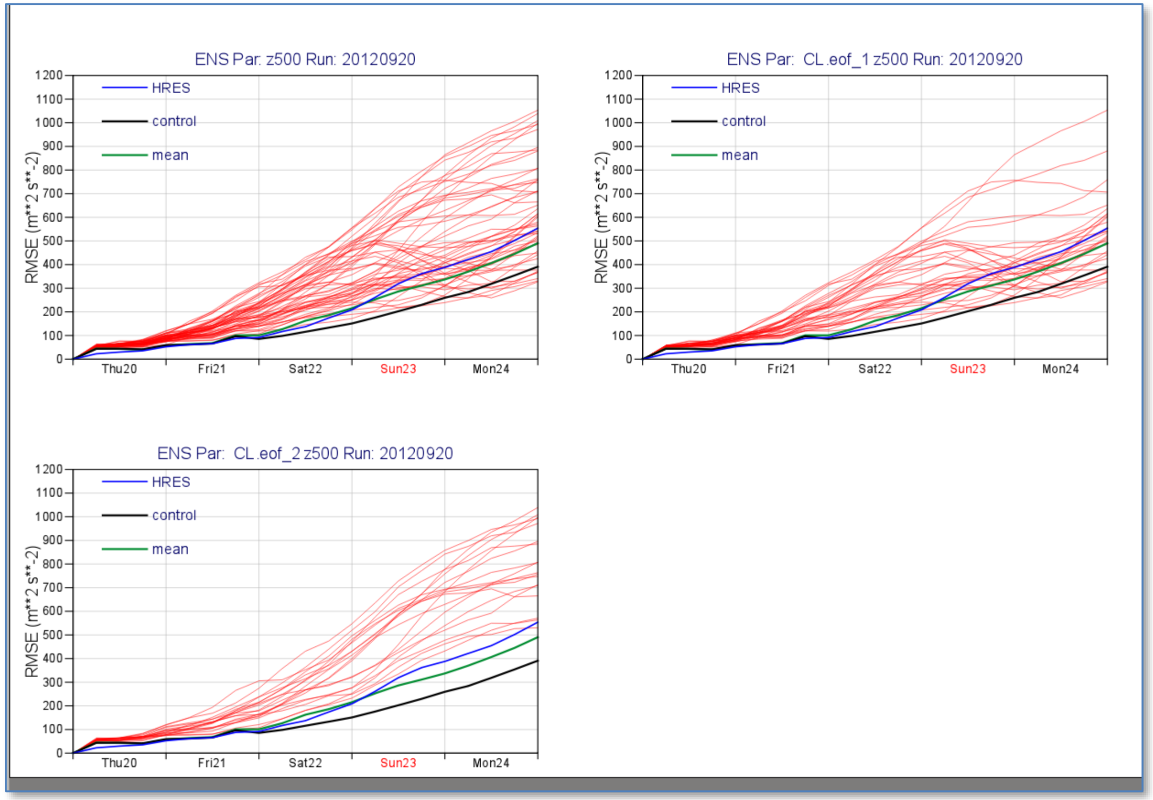
Icon name: Display Window
Folder: /course_prep/2016/metview-for-data-analysis-and-visualisation/day_3/sa
Type: PLOT_SUPERPAGE Modified: 2016-09-06 16:11

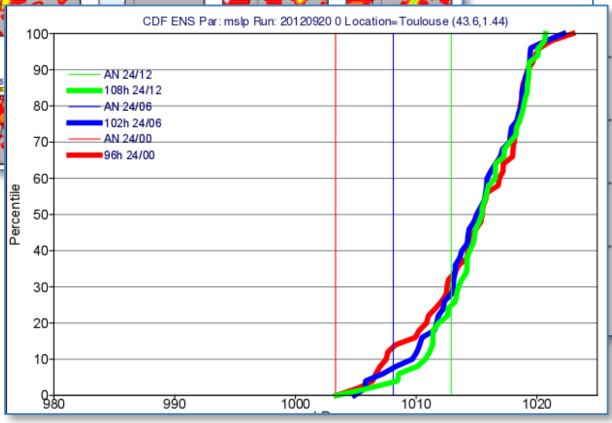


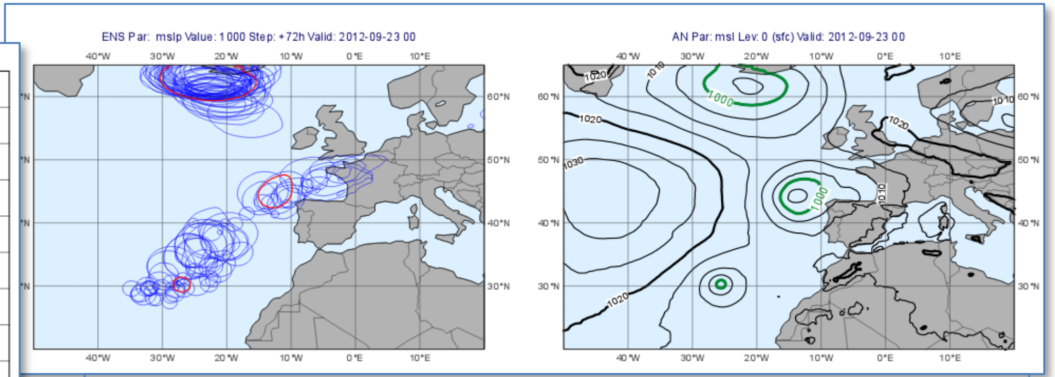
Metview in OpenIFS 2016

- A large emphasis on studying ensemble data







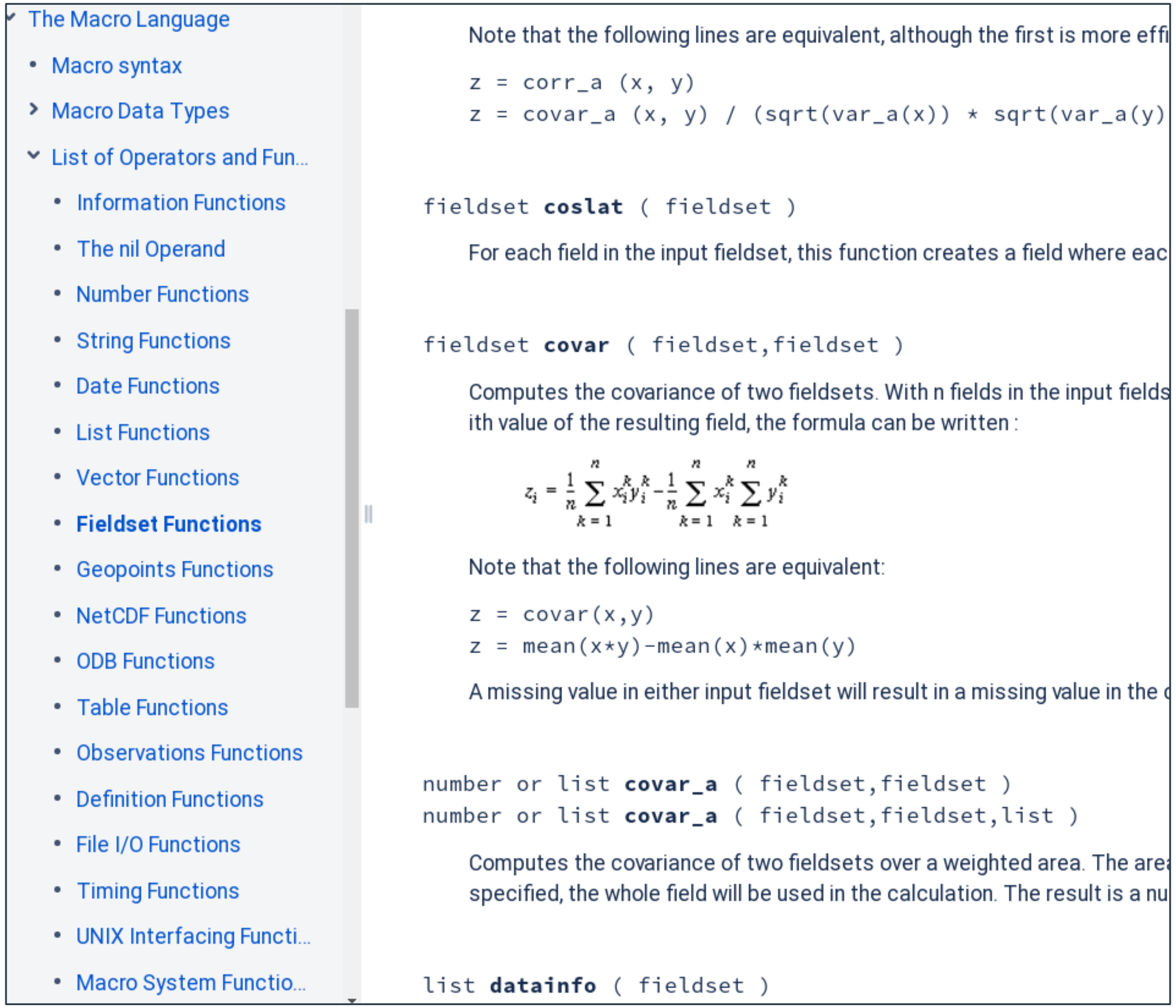

ENS Par: mslp Value: 1000 Step: +72h Valid: 2012-09-23 00
AN Par: msl Lev: 0 (sfc) Valid: 2012-09-23 00
40°W
30°W
20°W
10°W
0°E
10°E
60°N
50°N
40°N
30°N
1000
1010
1020
1030

Stamp, RMSE plumes, CDF and spaghetti plots

The “Data analysis and visualisation using Metview” course covers ENS data

Scripting

- Macro language
  - Built-in scripting language
  - Gives access to all interactive functionality
  - Plus many more functions
- Python language
  - Gives Python access to all Macro functionality
  - Plus all Python functionality



Note that the following lines are equivalent, although the first is more effi

```
z = corr_a (x, y)
z = covar_a (x, y) / (sqrt(var_a(x)) * sqrt(var_a(y)
```

```
fieldset coslat ( fieldset )
```

For each field in the input fieldset, this function creates a field where eac

```
fieldset covar ( fieldset,fieldset )
```

Computes the covariance of two fieldsets. With n fields in the input fields ith value of the resulting field, the formula can be written :

$$z_i = \frac{1}{n}\sum_{k=1}^{n} x_i^k y_i^k - \frac{1}{n}\sum_{k=1}^{n} x_i^k \sum_{k=1}^{n} y_i^k$$

Note that the following lines are equivalent:

```
z = covar(x,y)
z = mean(x*y)-mean(x)*mean(y)
```

A missing value in either input fieldset will result in a missing value in the c

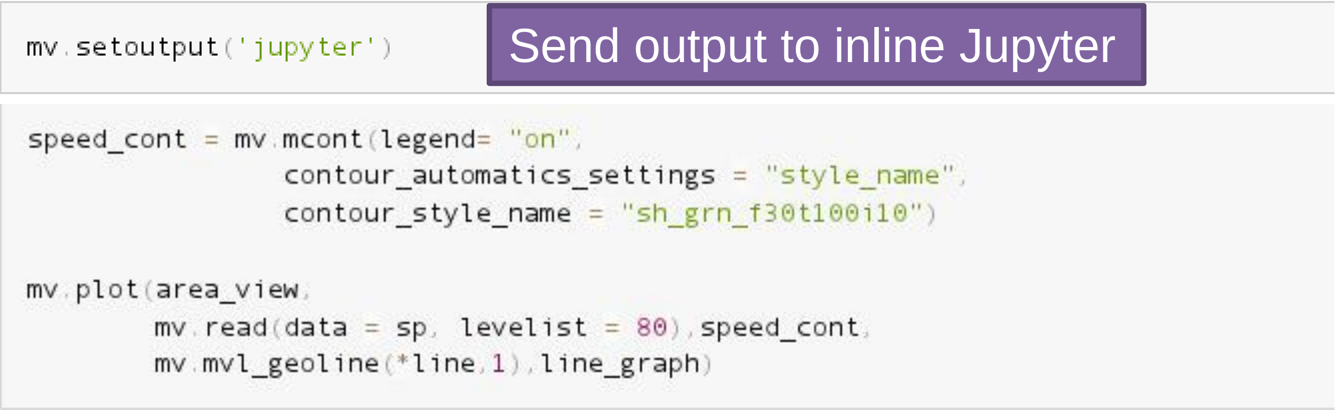
```
number or list covar_a ( fieldset,fieldset )
number or list covar_a ( fieldset,fieldset,list )
```

Computes the covariance of two fieldsets over a weighted area. The area specified, the whole field will be used in the calculation. The result is a nu

```
list datainfo ( fieldset )
```

Scripting and output formats

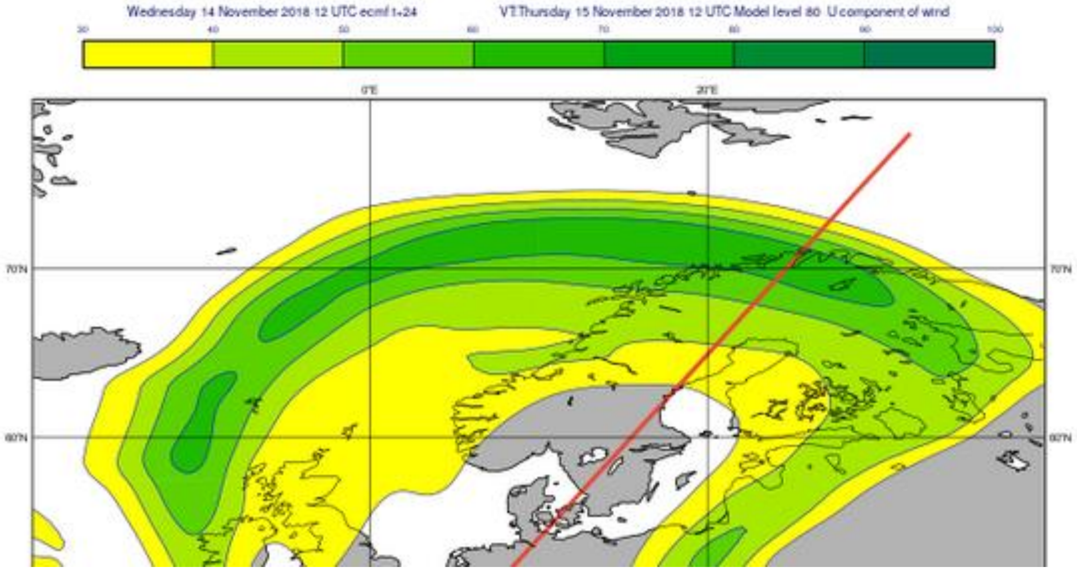
- setoutput() – sets the output format (ps, pdf, png, svg, kml) or sends to Jupyter



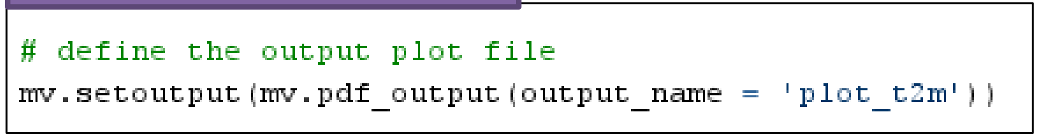
```
mv.setoutput('jupyter')
```

Send output to inline Jupyter

```
speed_cont = mv.mcont(legend= "on",
                      contour_automatics_settings = "style_name",
                      contour_style_name = "sh_grn_f30t100i10")

mv.plot(area_view,
        mv.read(data = sp, levelist = 80),speed_cont,
        mv.mvl_geoline(*line,1),line_graph)
```


Wednesday 14 November 2018 12 UTC ecmf t+24
VT:Thursday 15 November 2018 12 UTC Model level 80 U component of wind
30
40
50
60
70
80
90
100
0°E
20°E
70°N
60°N

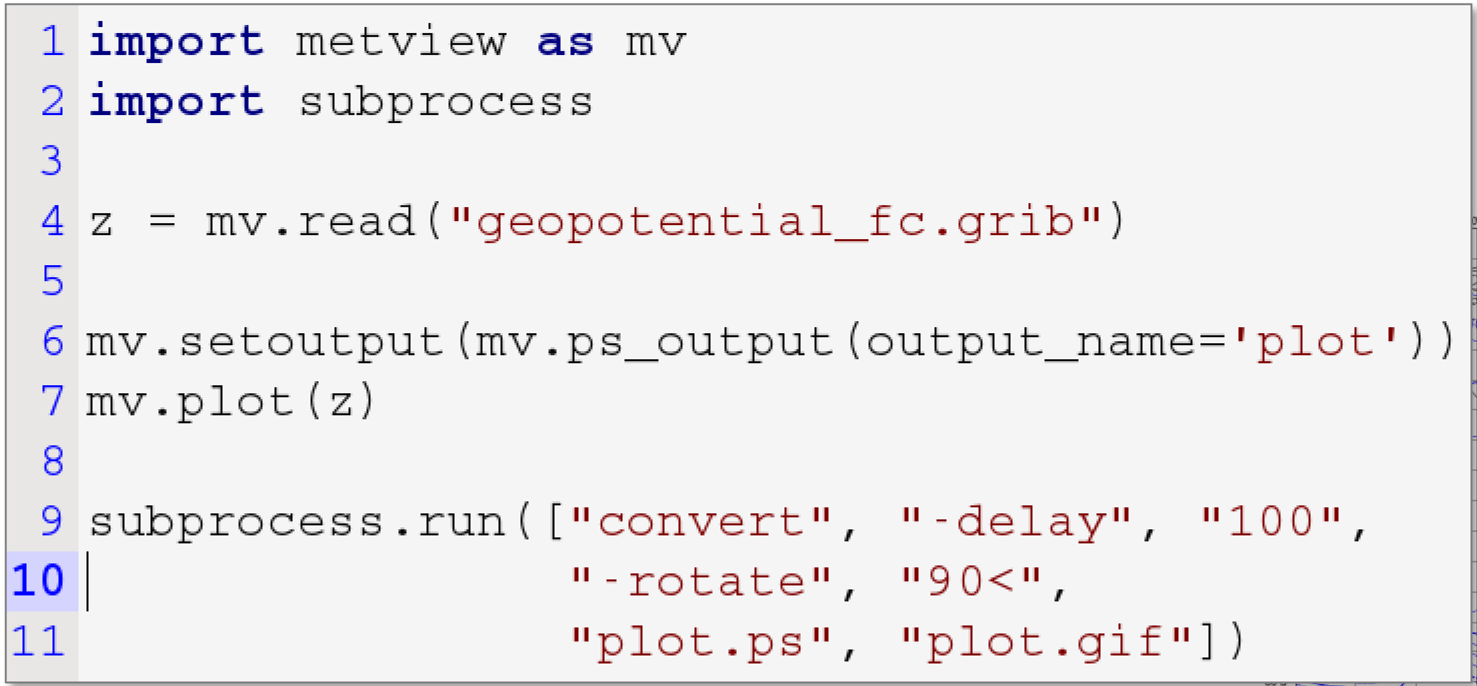
Send output to pdf file



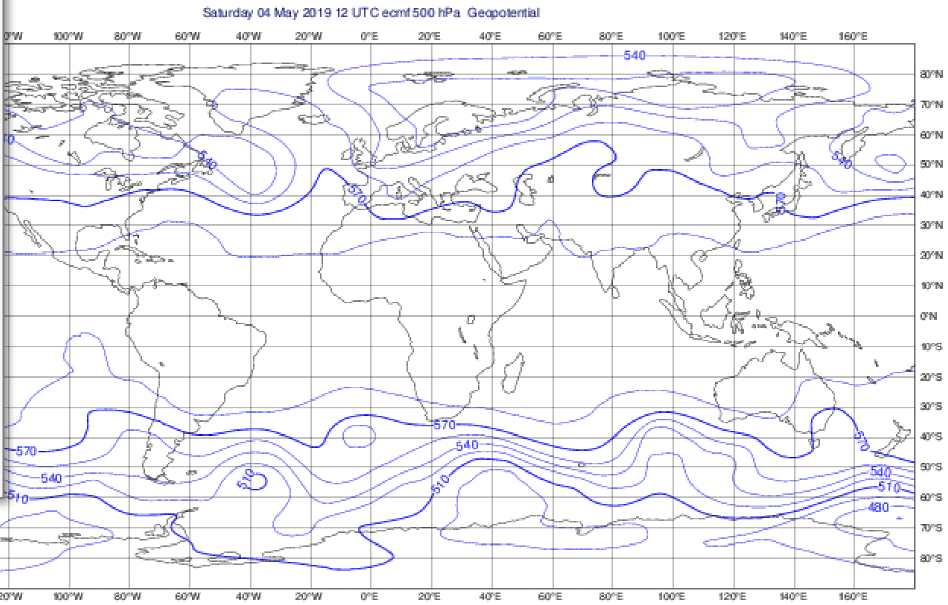
```
# define the output plot file
mv.setoutput(mv.pdf_output(output_name = 'plot_t2m'))
```

Animations

- Metview can produce multi-plot files (PostScript, PDF) and multi-file plots (PNG, SVG, KML)
- An easy way to get an animation out of these is to use the ImageMagick **convert** command

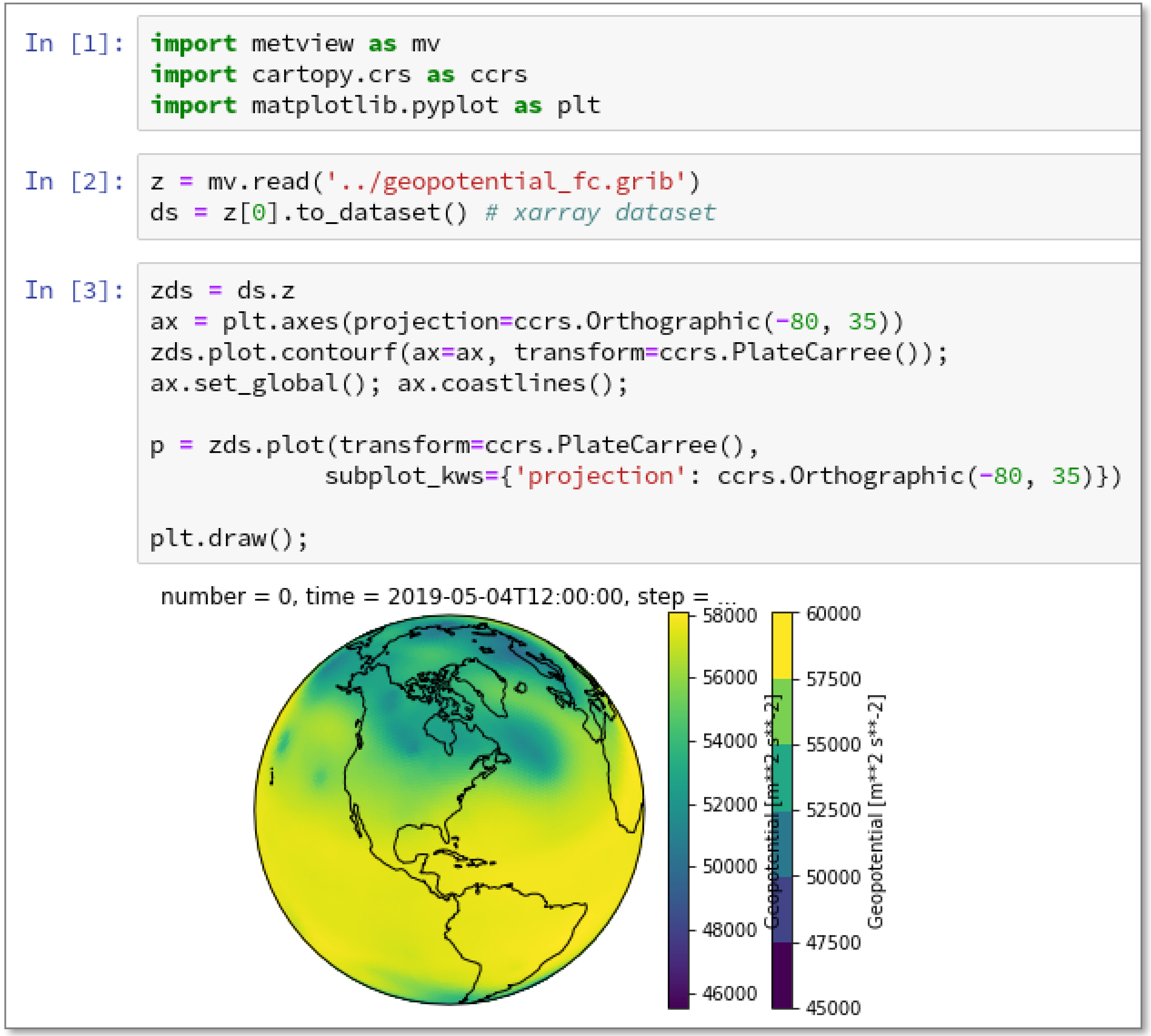


```python
import metview as mv
import subprocess

z = mv.read("geopotential_fc.grib")

mv.setoutput(mv.ps_output(output_name='plot'))
mv.plot(z)

subprocess.run(["convert", "-delay", "100",
                "-rotate", "90<",
                "plot.ps", "plot.gif"])
```



Other Python tools – xarray / cartopy / matplotlib

- e.g. use Metview or cfgrib* to get GRIB data into an xarray, then plot using cartopy and matplotlib



```
In [1]: import metview as mv
        import cartopy.crs as ccrs
        import matplotlib.pyplot as plt

In [2]: z = mv.read('../geopotential_fc.grib')
        ds = z[0].to_dataset() # xarray dataset

In [3]: zds = ds.z
        ax = plt.axes(projection=ccrs.Orthographic(-80, 35))
        zds.plot.contourf(ax=ax, transform=ccrs.PlateCarree());
        ax.set_global(); ax.coastlines();

        p = zds.plot(transform=ccrs.PlateCarree(),
                     subplot_kws={'projection': ccrs.Orthographic(-80, 35)})

        plt.draw();
```

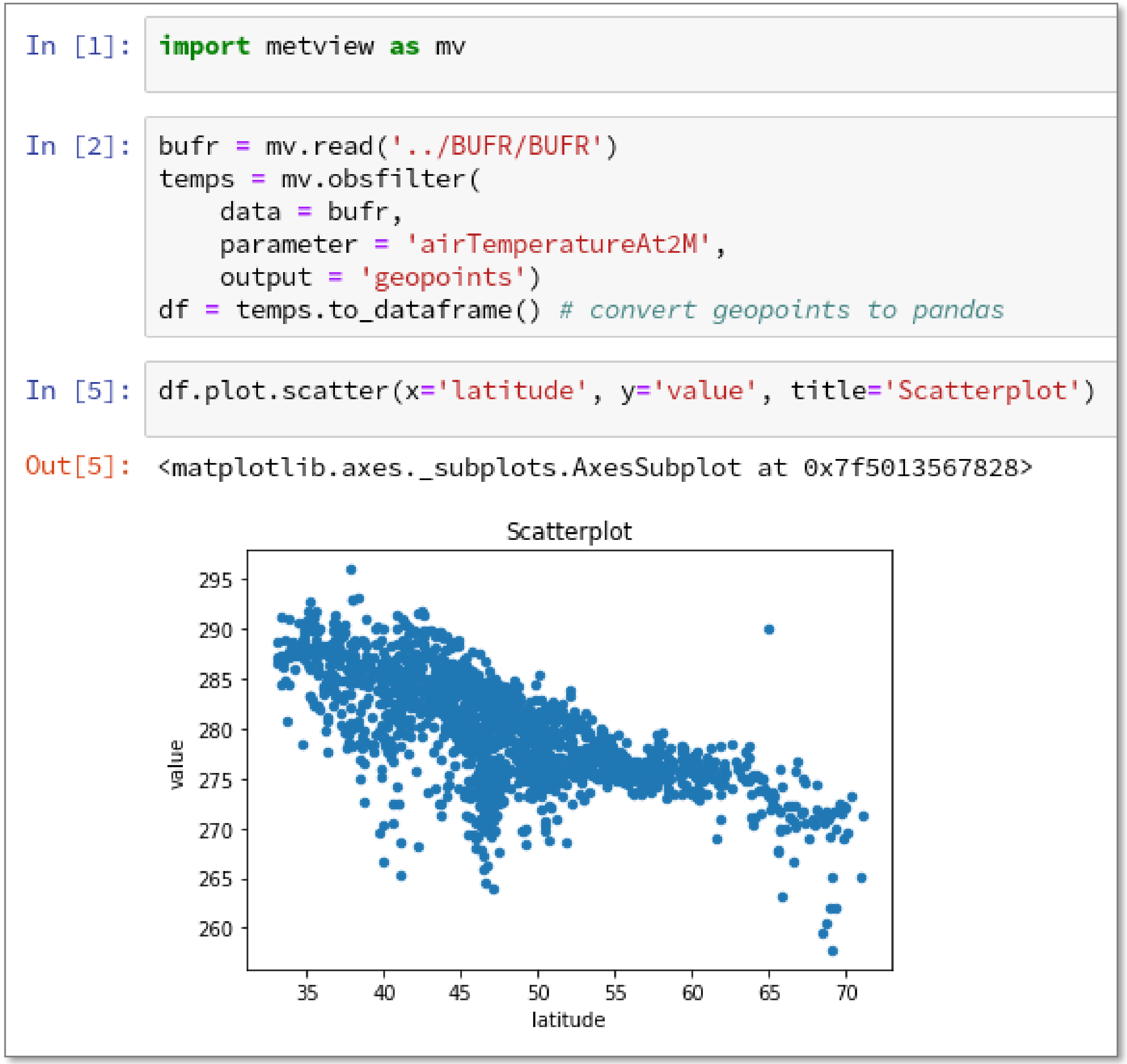
number = 0, time = 2019-05-04T12:00:00, step = ...
58000
56000
54000
52000
50000
48000
46000
Geopotential [m**2 s**-2]
60000
57500
55000
52500
50000
47500
45000
Geopotential [m**2 s**-2]

xarray

- *cfgrib is an ECMWF/B-Open development for loading GRIB data into xarray

Other Python tools – pandas

- e.g. use Metview to convert BUFR, geopoints or ODB into a pandas dataframe, then plot
- All Metview data objects can also export an numpy array of values via the `values()` method



```
In [1]: import metview as mv
```

```
In [2]: bufr = mv.read('../BUFR/BUFR')
        temps = mv.obsfilter(
            data = bufr,
            parameter = 'airTemperatureAt2M',
            output = 'geopoints')
        df = temps.to_dataframe() # convert geopoints to pandas
```

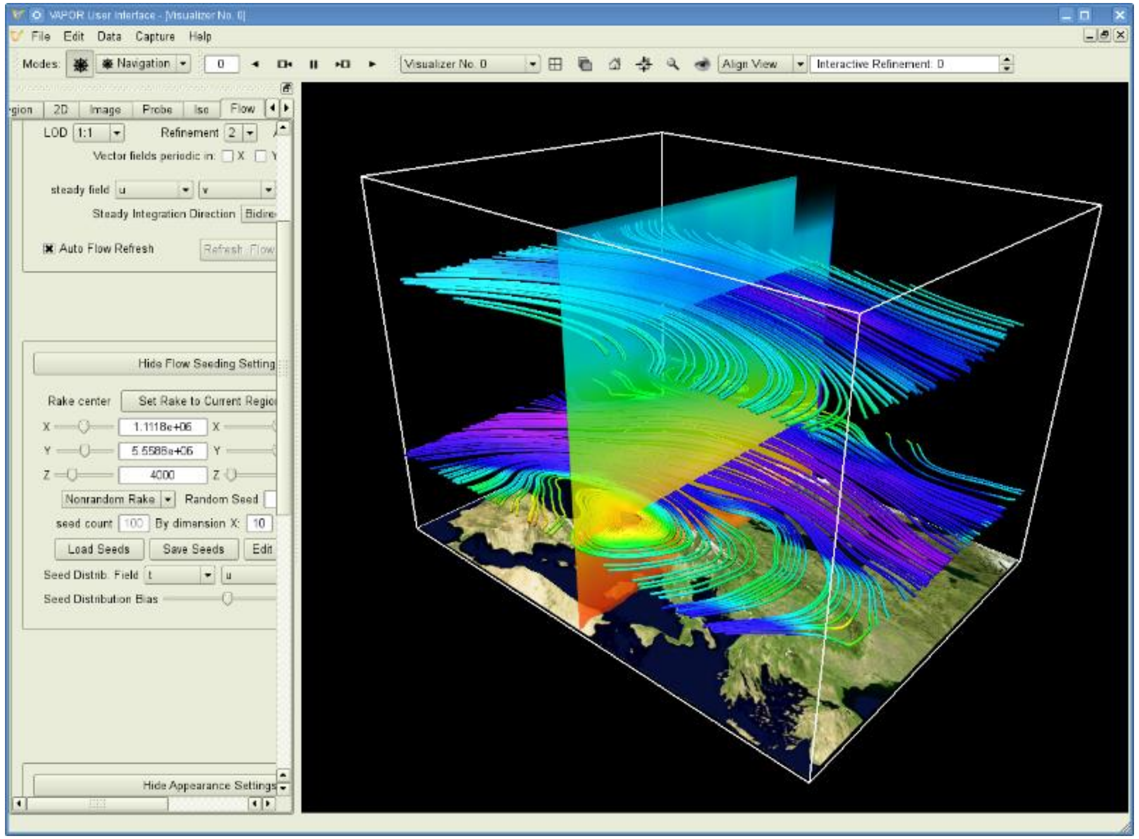
```
In [5]: df.plot.scatter(x='latitude', y='value', title='Scatterplot')

Out[5]: <matplotlib.axes._subplots.AxesSubplot at 0x7f5013567828>
```

pandas

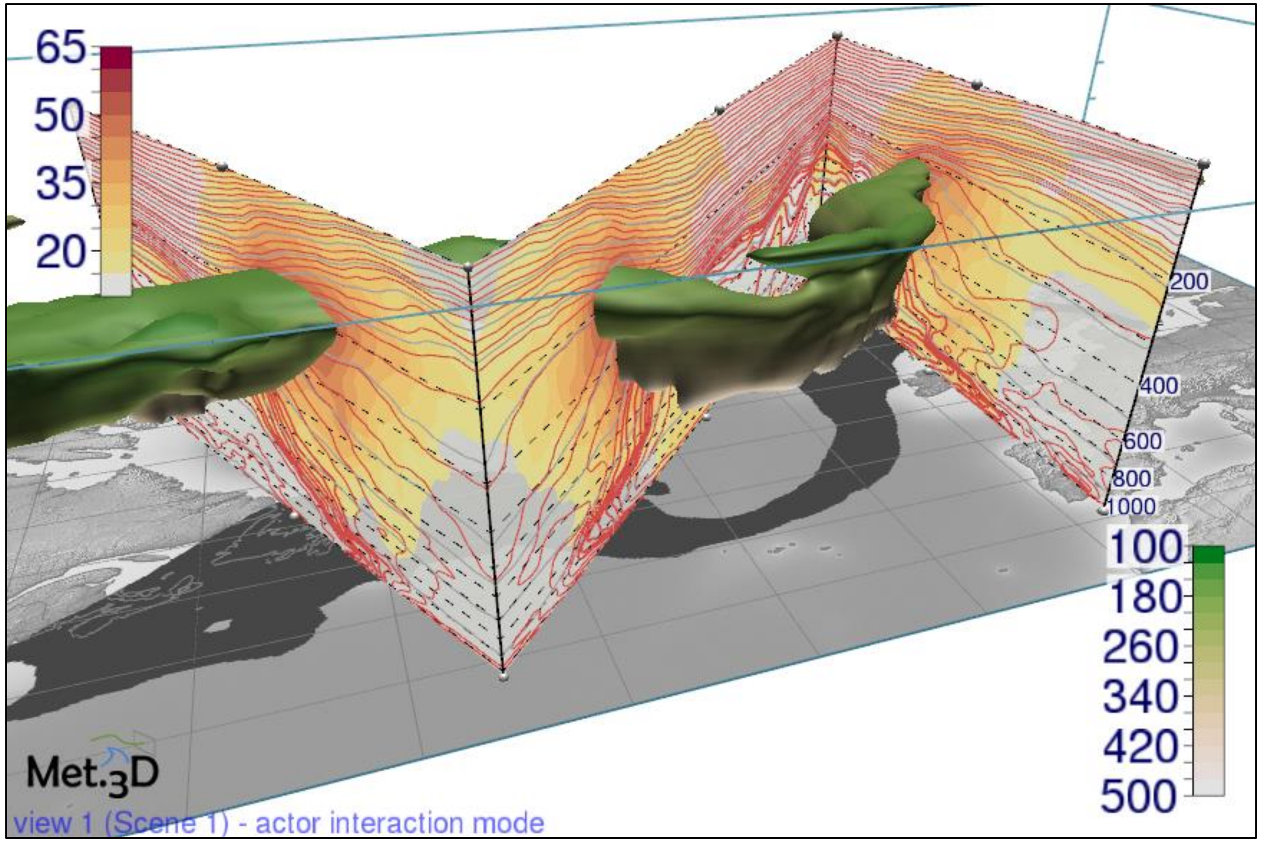
$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$

3D

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



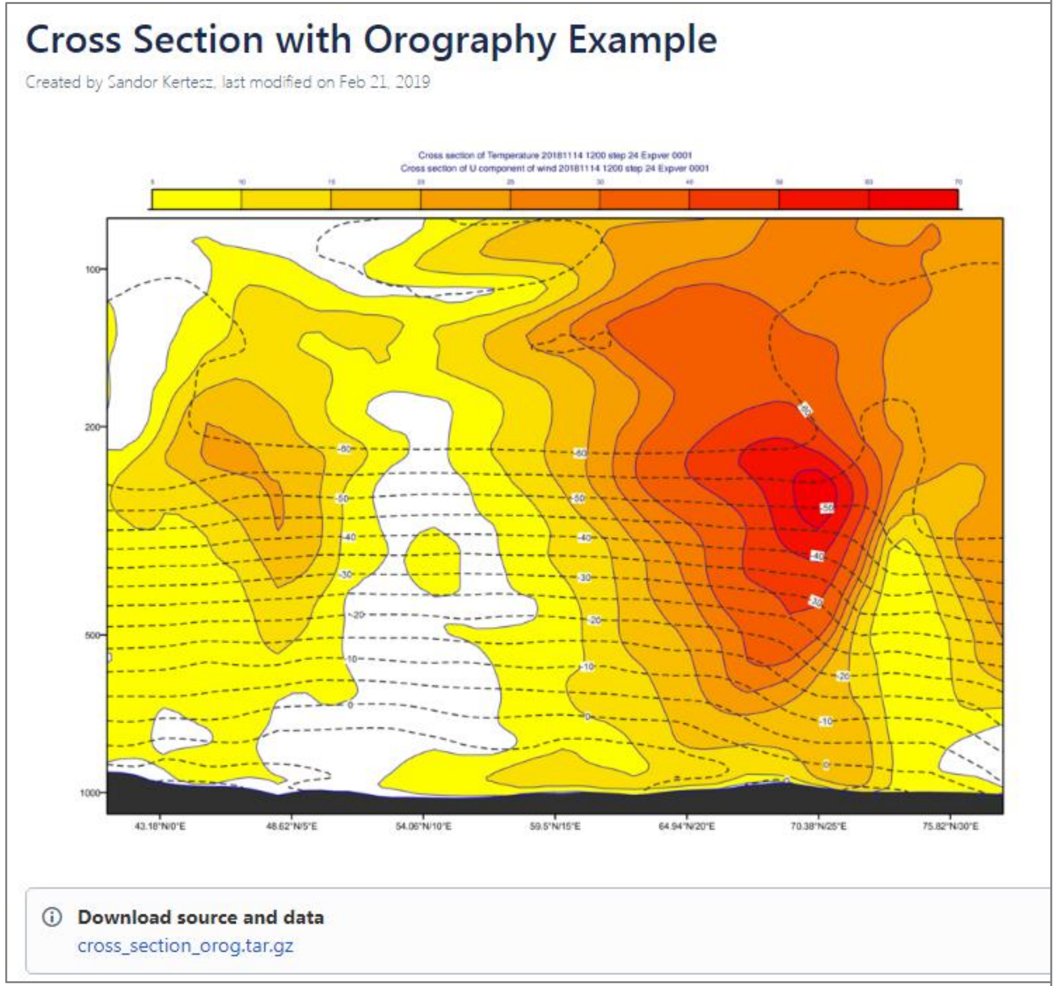
Imagery produced by VAPOR



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

Where to find out more

- See the Gallery for Macro and Python examples



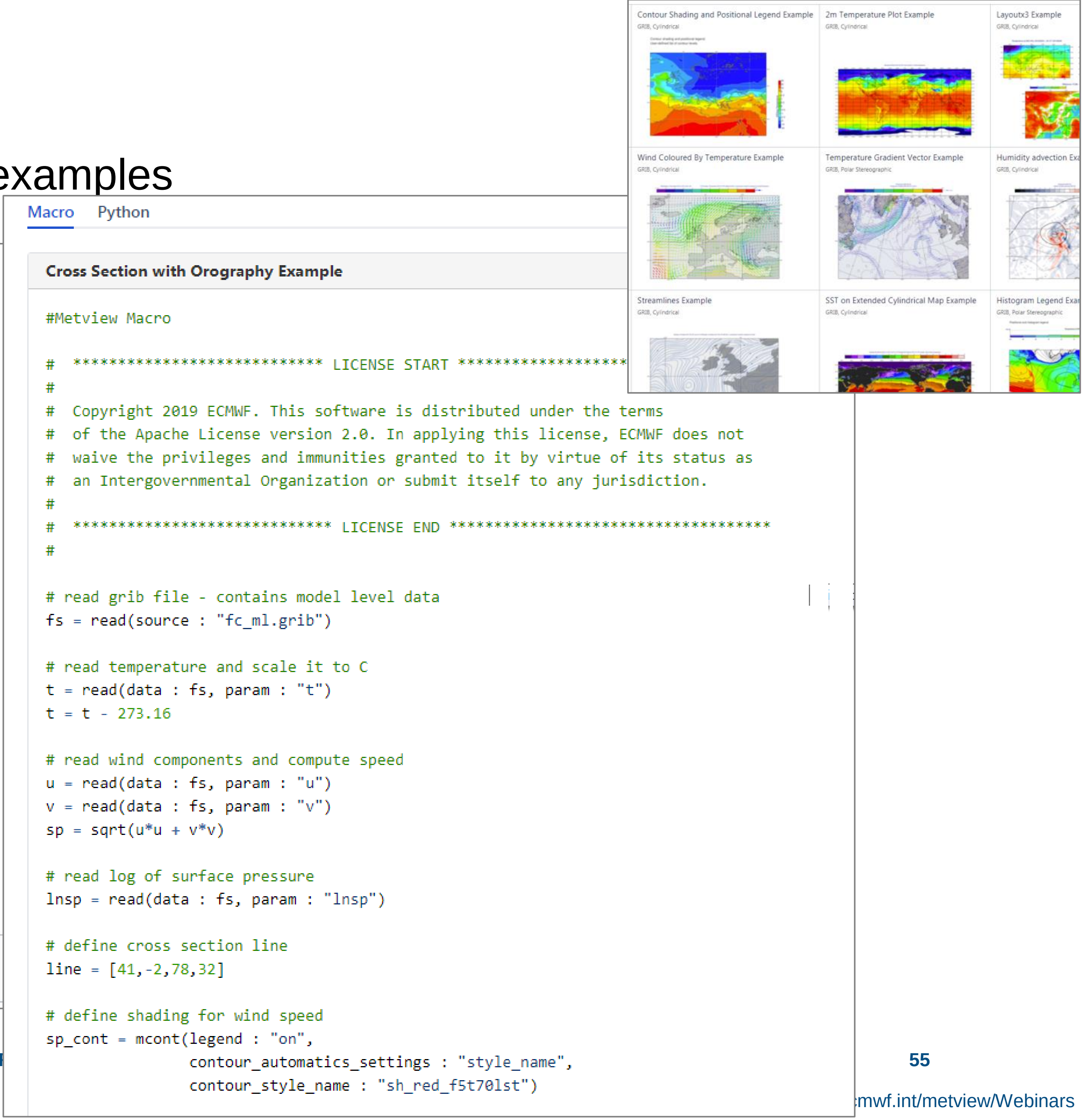
Cross Section with Orography Example

Created by Sandor Kertesz, last modified on Feb 21, 2019

Cross section of Temperature 20181114 1200 step 24 Expver 0001
Cross section of U component of wind 20181114 1200 step 24 Expver 0001
5
10
15
20
25
30
40
50
60
70
100
200
500
1000
43.16°N/0°E
48.62°N/5°E
54.06°N/10°E
59.5°N/15°E
64.94°N/20°E
70.38°N/25°E
75.82°N/30°E

ⓘ **Download source and data**

cross_section_orog.tar.gz



Macro Python

Cross Section with Orography Example

```
#Metview Macro

#  **************************** LICENSE START ******************
#
#  Copyright 2019 ECMWF. This software is distributed under the terms
#  of the Apache License version 2.0. In applying this license, ECMWF does not
#  waive the privileges and immunities granted to it by virtue of its status as
#  an Intergovernmental Organization or submit itself to any jurisdiction.
#
#  ***************************** LICENSE END ************************************
#

# read grib file - contains model level data
fs = read(source : "fc_ml.grib")

# read temperature and scale it to C
t = read(data : fs, param : "t")
t = t - 273.16

# read wind components and compute speed
u = read(data : fs, param : "u")
v = read(data : fs, param : "v")
sp = sqrt(u*u + v*v)

# read log of surface pressure
lnsp = read(data : fs, param : "lnsp")

# define cross section line
line = [41,-2,78,32]

# define shading for wind speed
sp_cont = mcont(legend : "on",
                contour_automatics_settings : "style_name",
                contour_style_name : "sh_red_f5t70lst")
```

Contour Shading and Positional Legend Example
GRIB, Cylindrical

2m Temperature Plot Example
GRIB, Cylindrical

Layoutx3 Example
GRIB, Cylindrical

Wind Coloured By Temperature Example
GRIB, Cylindrical

Temperature Gradient Vector Example
GRIB, Polar Stereographic

Humidity advection Exa
GRIB, Cylindrical

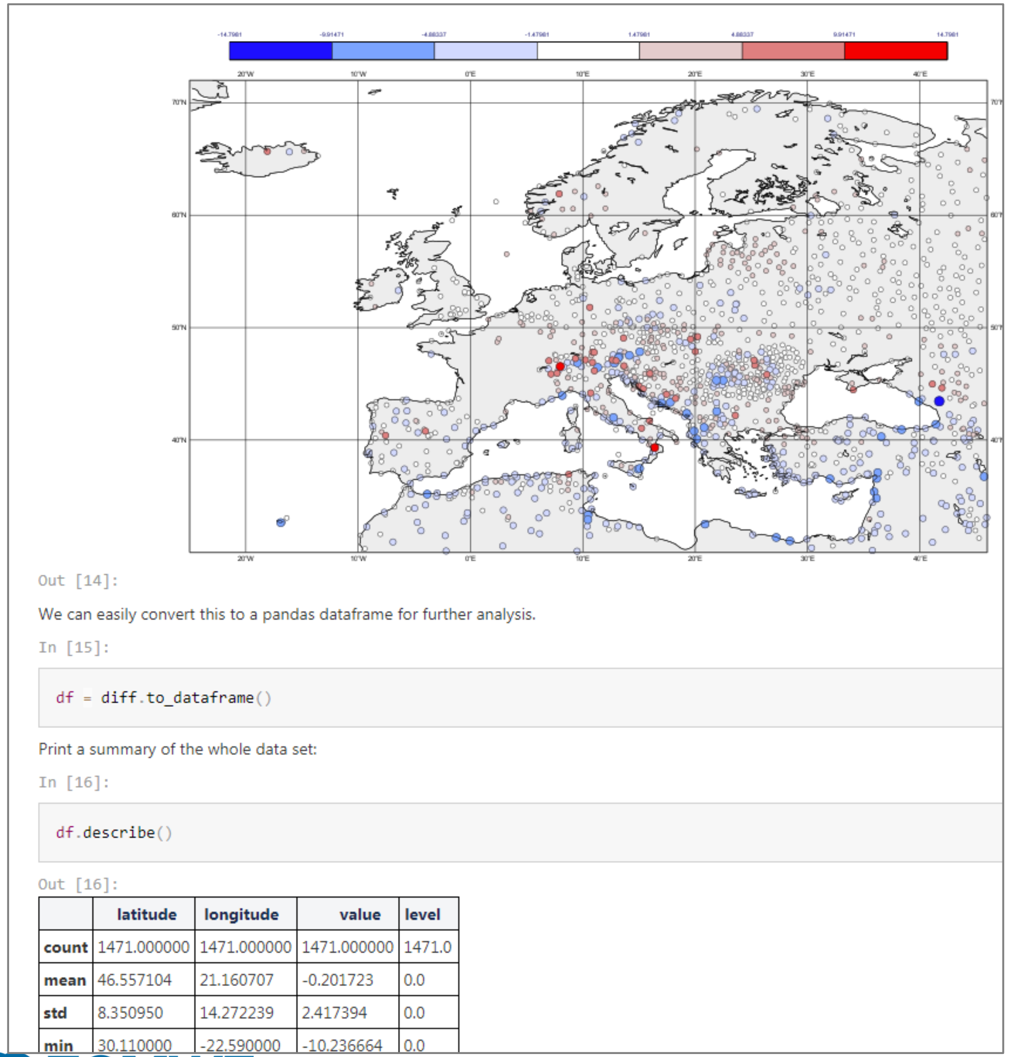
Streamlines Example
GRIB, Cylindrical

SST on Extended Cylindrical Map Example
GRIB, Cylindrical

Histogram Legend Exa
GRIB, Polar Stereographic

Where to find out more

- See the Jupyter Notebooks for more Python



Out [14]:

We can easily convert this to a pandas dataframe for further analysis.

In [15]:

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

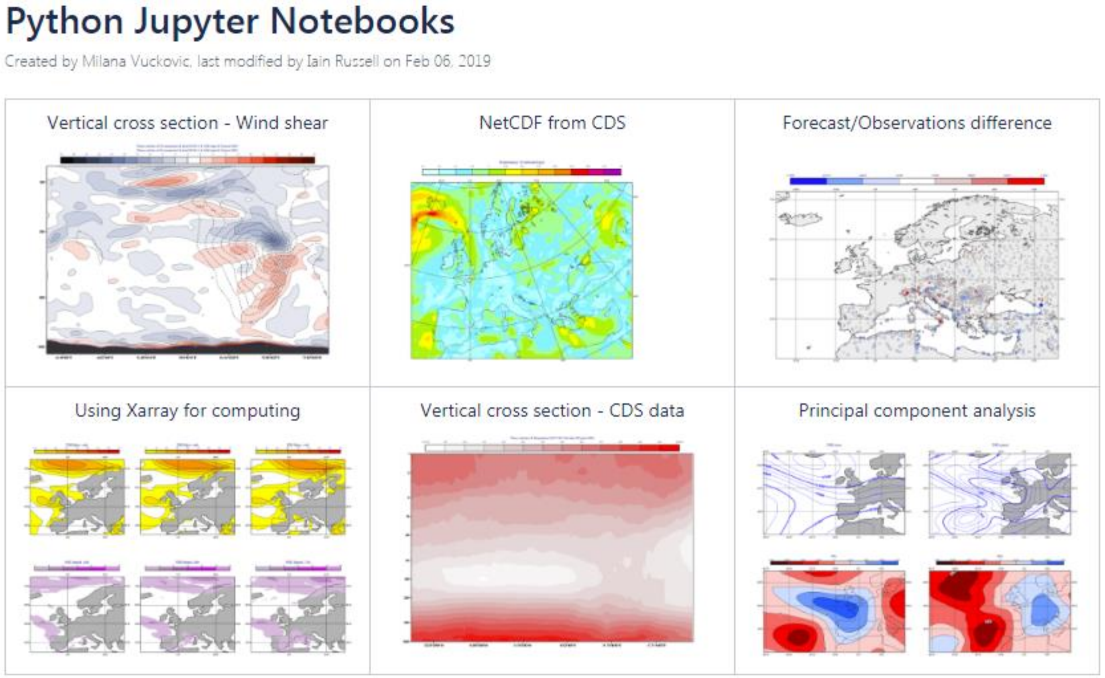
Print a summary of the whole data set:

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



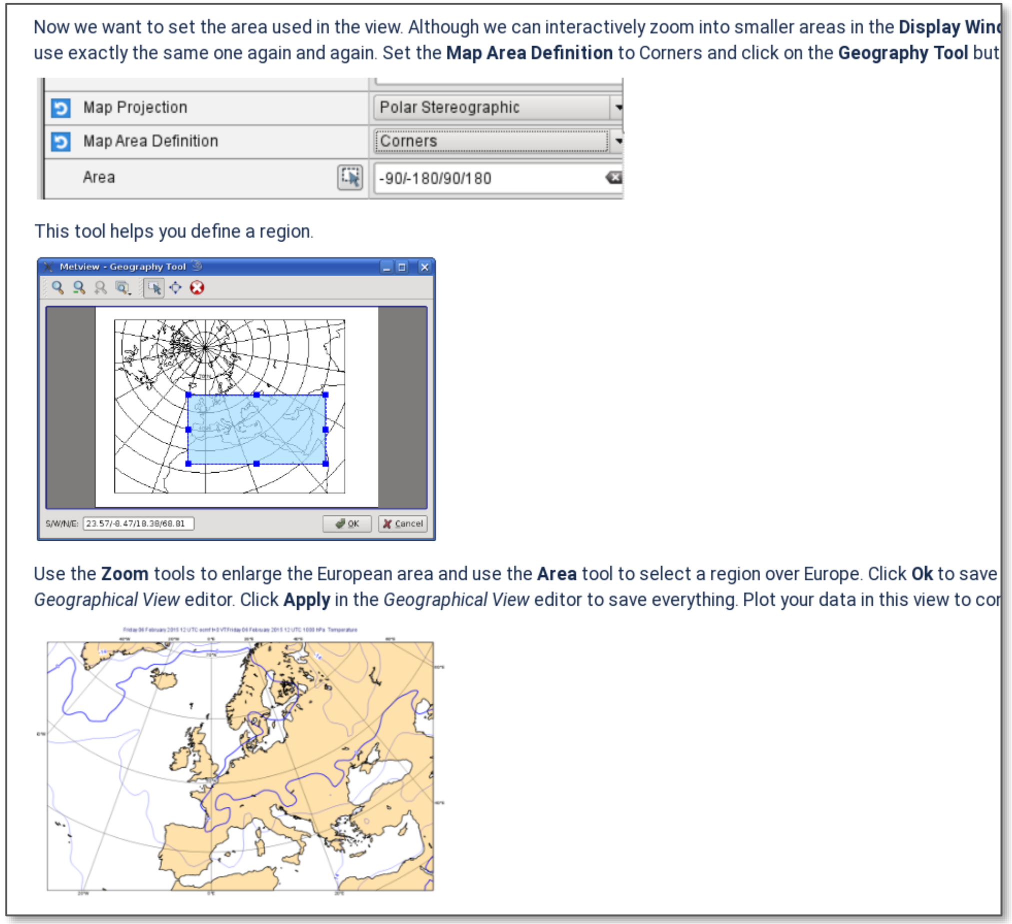
Python Jupyter Notebooks

Created by Milana Vuckovic, last modified by Iain Russell on Feb 06, 2019

Vertical cross section - Wind shear
NetCDF from CDS
Forecast/Observations difference
Using Xarray for computing
Vertical cross section - CDS data
Principal component analysis

Tutorials

- Lots of material online including tutorials

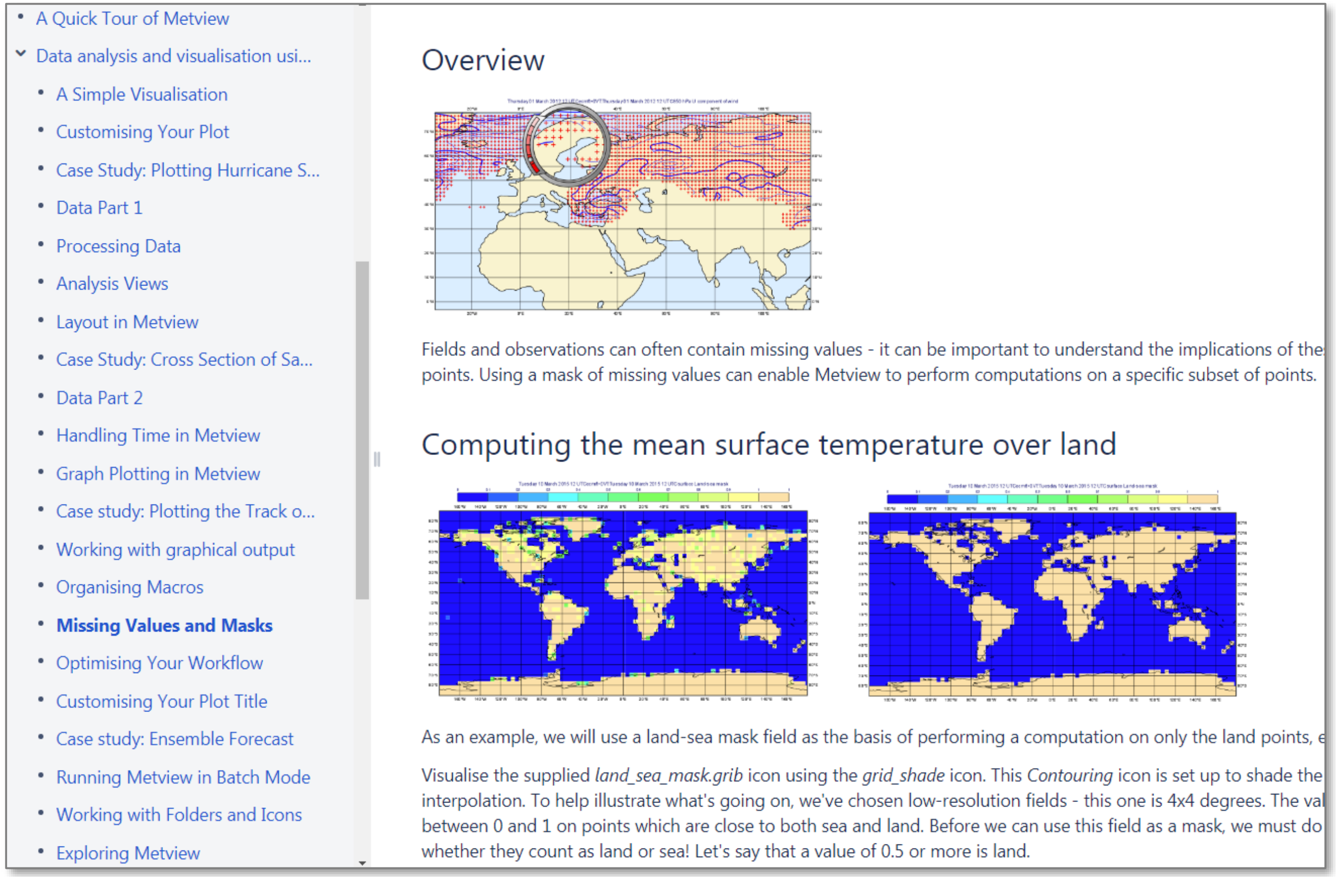


Now we want to set the area used in the view. Although we can interactively zoom into smaller areas in the **Display Wind** use exactly the same one again and again. Set the **Map Area Definition** to Corners and click on the **Geography Tool** but

Map Projection	Polar Stereographic
Map Area Definition	Corners
Area	-90/-180/90/180

This tool helps you define a region.

Use the **Zoom** tools to enlarge the European area and use the **Area** tool to select a region over Europe. Click **Ok** to save *Geographical View* editor. Click **Apply** in the *Geographical View* editor to save everything. Plot your data in this view to co



- A Quick Tour of Metview
- Data analysis and visualisation usi...
  - A Simple Visualisation
  - Customising Your Plot
  - Case Study: Plotting Hurricane S...
  - Data Part 1
  - Processing Data
  - Analysis Views
  - Layout in Metview
  - Case Study: Cross Section of Sa...
  - Data Part 2
  - Handling Time in Metview
  - Graph Plotting in Metview
  - Case study: Plotting the Track o...
  - 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

Overview

Fields and observations can often contain missing values - it can be important to understand the implications of the 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

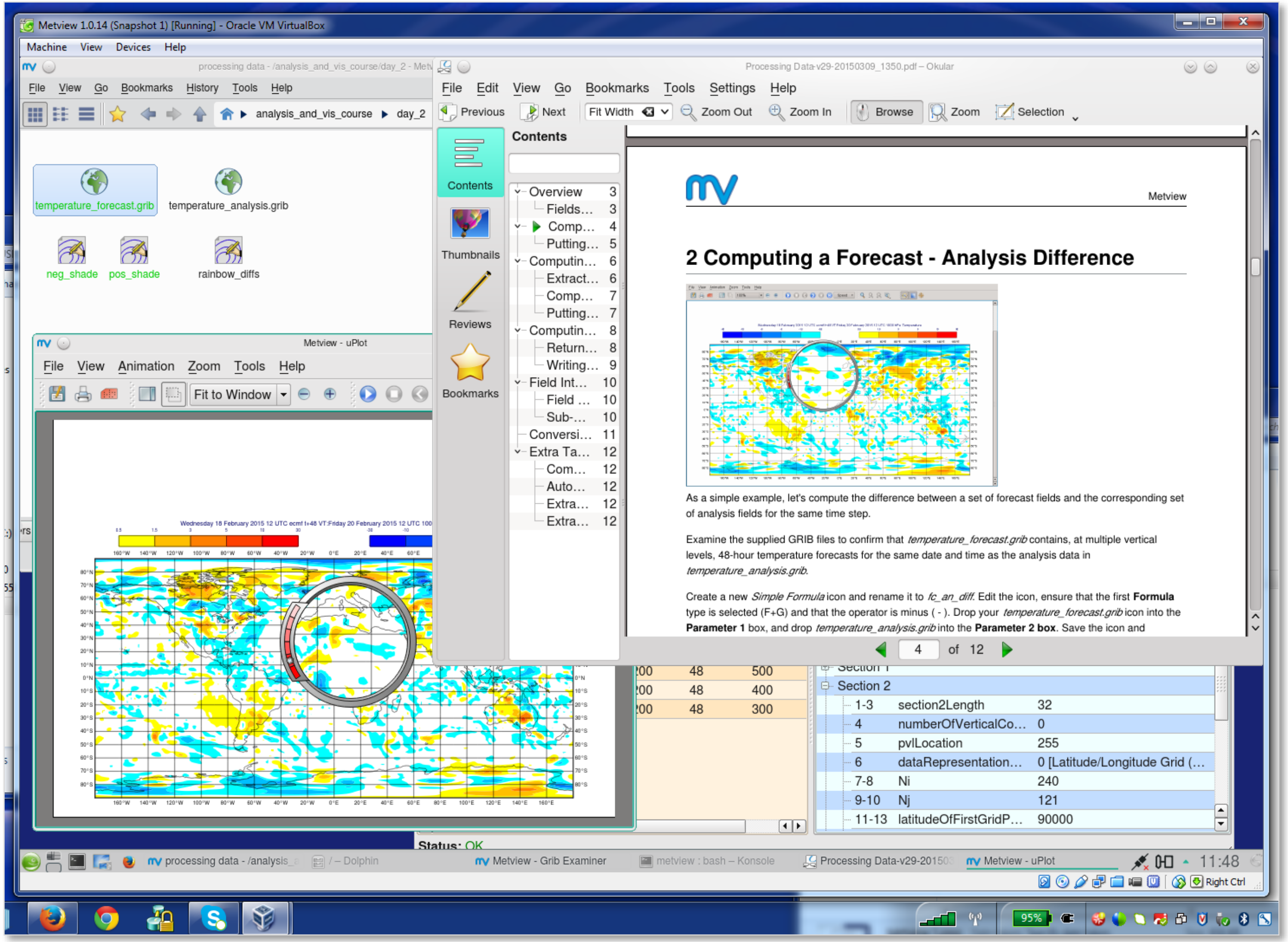
As an example, we will use a land-sea mask field as the basis of performing a computation on only the land points, e

Visualise the supplied *land_sea_mask.grib* icon using the *grid_shade* icon. This *Contouring* icon is set up to shade the interpolation. To help illustrate what's going on, we've chosen low-resolution fields - this one is 4x4 degrees. The val between 0 and 1 on points which are close to both sea and land. Before we can use this field as a mask, we must do whether they count as land or sea! Let's say that a value of 0.5 or more is land.

Metview Availability

- **Available on ECMWF systems:**
  - Versioned using the ‘module’ system
  - `[module swap metview/new]`
  - `metview`

- **On other systems**:
  - Install from binaries (rpm, deb)
  - Conda (via conda-forge)
  - Build from source
  - Build from bundle
  - pip install metview (Python)


Metview 1.0.14 (Snapshot 1) [Running] - Oracle VM VirtualBox
Machine View Devices Help
processing data - /analysis_and_vis_course/day_2 - Metview
File View Go Bookmarks History Tools Help
analysis_and_vis_course ▸ day_2
temperature_forecast.grib
temperature_analysis.grib
neg_shade
pos_shade
rainbow_diffs
Metview - uPlot
File View Animation Zoom Tools Help
Fit to Window
Processing Data-v29-20150309_1350.pdf – Okular
File Edit View Go Bookmarks Tools Settings Help
Previous Next Fit Width Zoom Out Zoom In Browse Zoom Selection
Contents
Thumbnails
Reviews
Bookmarks
Overview 3
Fields… 3
Comp… 4
Putting… 5
Computin… 6
Extract… 6
Comp… 7
Putting… 7
Computin… 8
Return… 8
Writing… 9
Field Int… 10
Field … 10
Sub-… 10
Conversi… 11
Extra Ta… 12
Com… 12
Auto… 12
Extra… 12
Extra… 12
Metview
4 of 12
Status: OK
processing data - /analysis
/ – Dolphin
Metview - Grib Examiner
metview : bash – Konsole
Processing Data-v29-20150
Metview - uPlot
11:48
Right Ctrl
95%

2 Computing a Forecast - Analysis Difference

As a simple example, let's compute the difference between a set of forecast fields and the corresponding set of analysis fields for the same time step.

Examine the supplied GRIB files to confirm that *temperature_forecast.grib* contains, at multiple vertical levels, 48-hour temperature forecasts for the same date and time as the analysis data in *temperature_analysis.grib*.

Create a new *Simple Formula* icon and rename it to *fc_an_diff*. Edit the icon, ensure that the first **Formula** type is selected (F+G) and that the operator is minus (-). Drop your *temperature_forecast.grib* icon into the **Parameter 1** box, and drop *temperature_analysis.grib* into the **Parameter 2 box**. Save the icon and

200	48	500
200	48	400
200	48	300

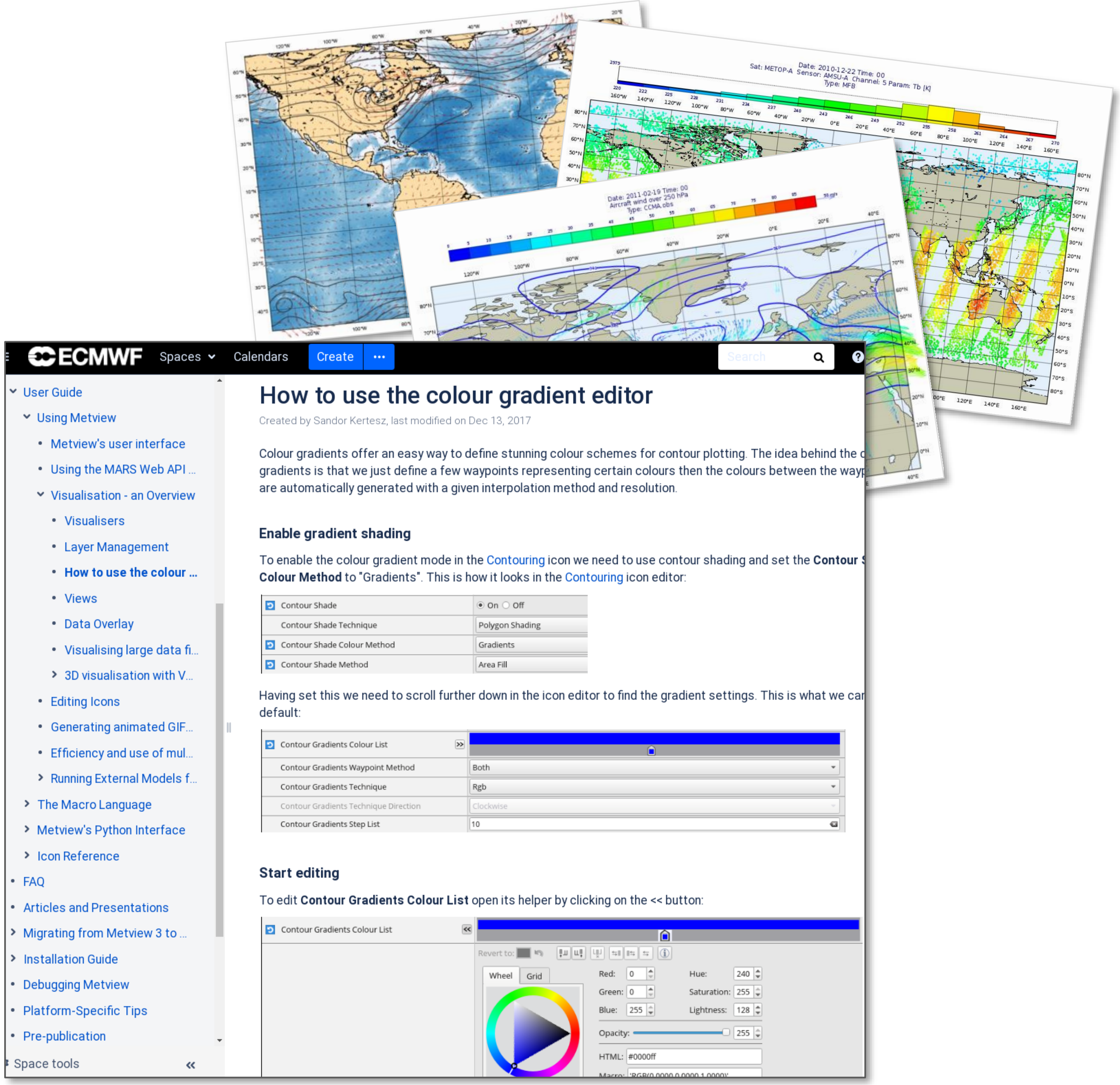
Section 1

Section 2

1-3	section2Length	32
4	numberOfVerticalCo…	0
5	pvlLocation	255
6	dataRepresentation…	0 [Latitude/Longitude Grid (…
7-8	Ni	240
9-10	Nj	121
11-13	latitudeOfFirstGridP…	90000

For more information…

- Ask for help:
  - Software.Support@ecmwf.int
- Visit our web pages:
  - http://confluence.ecmwf.int/metview


Date: 2010-12-22 Time: 00
Sat: METOP-A Sensor: AMSU-A Channel: 5 Param: Tb [K]
Type: MFB
Date: 2011-02-19 Time: 00
Aircraft wind over 250 hPa
Type: CCMA obs
ECMWF
Spaces
Calendars
Create
Search
User Guide
Using Metview
Metview's user interface
Using the MARS Web API ...
Visualisation - an Overview
Visualisers
Layer Management
How to use the colour ...
Views
Data Overlay
Visualising large data fi...
3D visualisation with V...
Editing Icons
Generating animated GIF...
Efficiency and use of mul...
Running External Models f...
The Macro Language
Metview's Python Interface
Icon Reference
FAQ
Articles and Presentations
Migrating from Metview 3 to ...
Installation Guide
Debugging Metview
Platform-Specific Tips
Pre-publication
Space tools

How to use the colour gradient editor

Created by Sandor Kertesz, last modified on Dec 13, 2017

Colour gradients offer an easy way to define stunning colour schemes for contour plotting. The idea behind the c
gradients is that we just define a few waypoints representing certain colours then the colours between the wayp
are automatically generated with a given interpolation method and resolution.

Enable gradient shading

To enable the colour gradient mode in the Contouring icon we need to use contour shading and set the **Contour S**
Colour Method to "Gradients". This is how it looks in the Contouring icon editor:

Contour Shade	● On ○ Off
Contour Shade Technique	Polygon Shading
Contour Shade Colour Method	Gradients
Contour Shade Method	Area Fill

Having set this we need to scroll further down in the icon editor to find the gradient settings. This is what we car
default:

Contour Gradients Colour List	
Contour Gradients Waypoint Method	Both
Contour Gradients Technique	Rgb
Contour Gradients Technique Direction	Clockwise
Contour Gradients Step List	10

Start editing

To edit **Contour Gradients Colour List** open its helper by clicking on the << button:

Contour Gradients Colour List

Revert to:

Wheel Grid

Red: 0
Green: 0
Blue: 255
Hue: 240
Saturation: 255
Lightness: 128
Opacity: 255
HTML: #0000ff
Macro: 'RGB(0.0000,0.0000,1.0000)'

Questions?