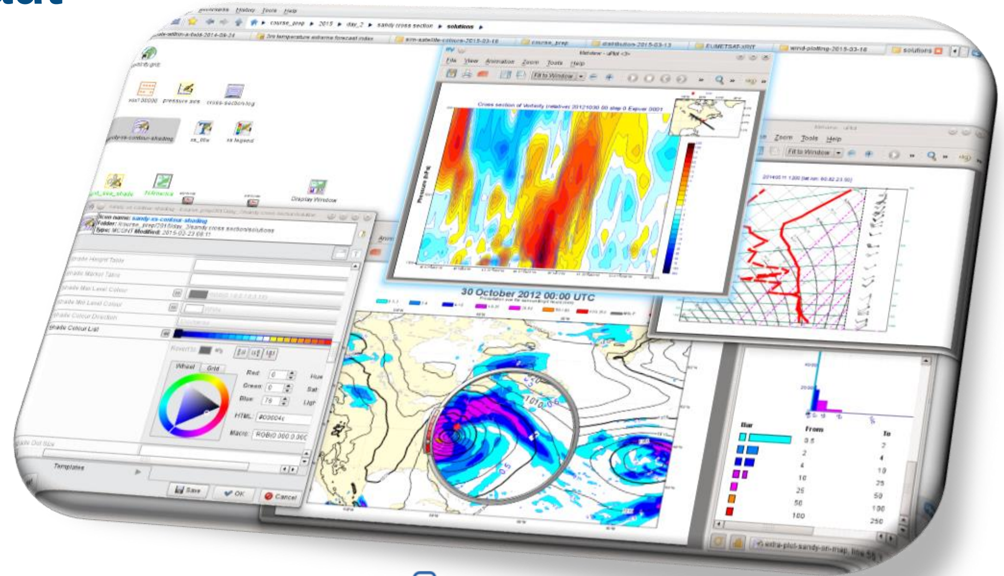
Data Analysis and Visualisation Using Metview

Computer User Training Course 2016

Iain Russell, Fernando Ii, Sándor Kertész, Stephan Siemen, Sylvie Lamy-Thépaut

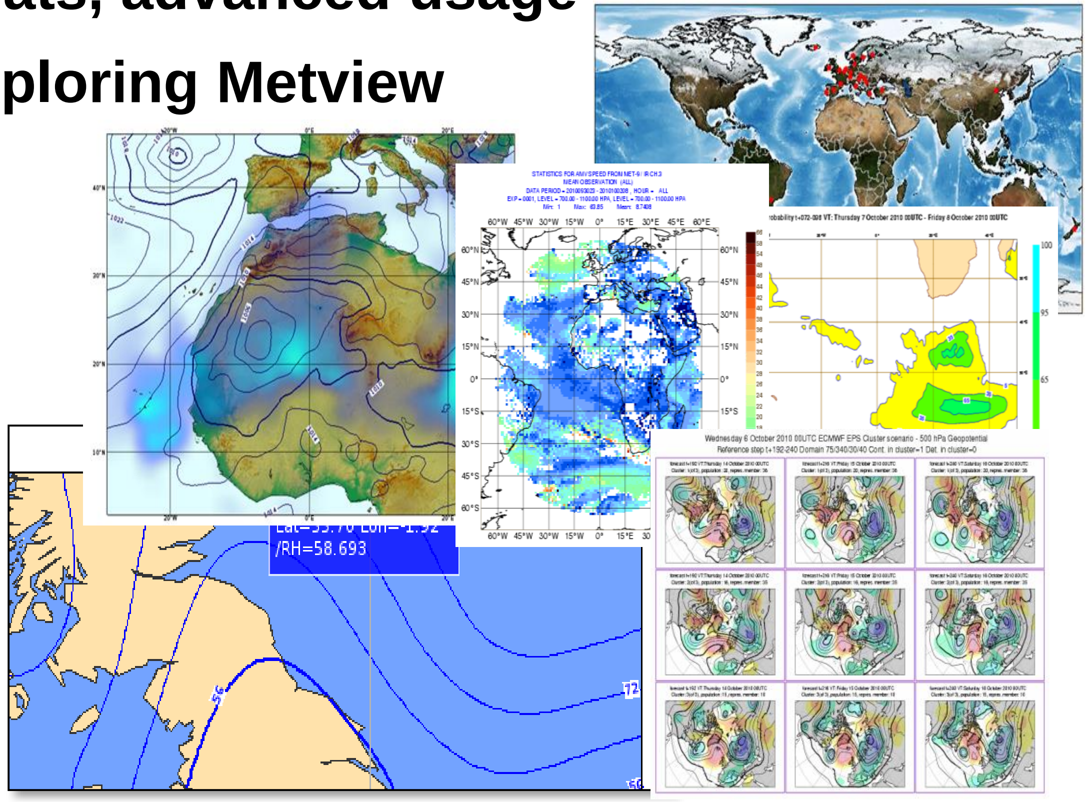
Development Section

metview@ecmwf.int



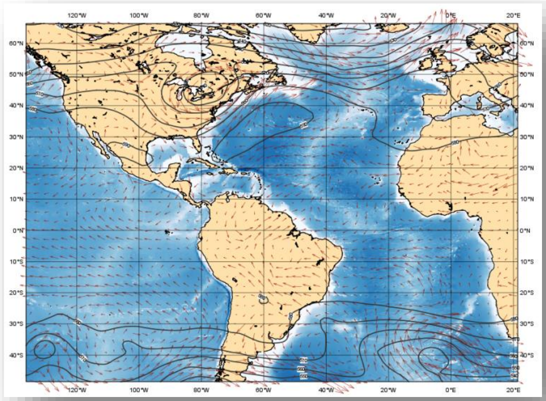
Outline

- Day 1: Introduction, main features
- Day 2: Data (1) and processing
- Day 3: Data (2), time and graphs
- Day 4: Graphics formats, advanced usage
- Day 5: Batch jobs, exploring Metview



Metview: meteorological workstation

- **Retrieve/manipulate/visualise meteorological data**
- **Working environment for operational and research meteorologists**
- **Allows analysts and researchers to easily build products interactively and run them in batch mode**



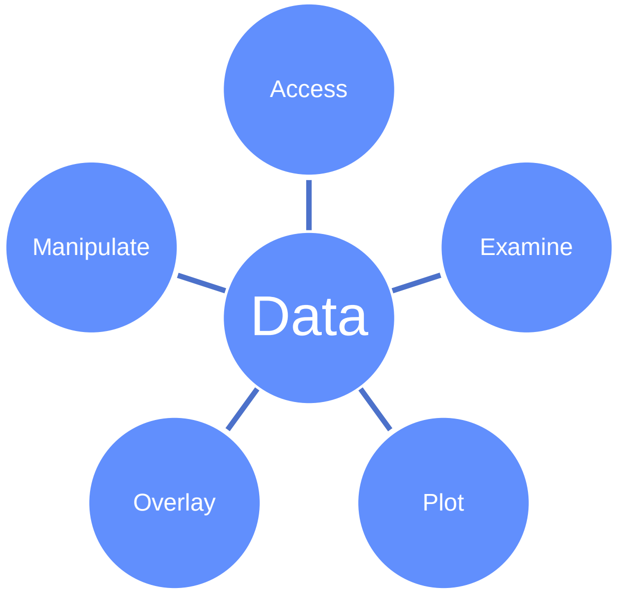
Built on core ECMWF technologies:

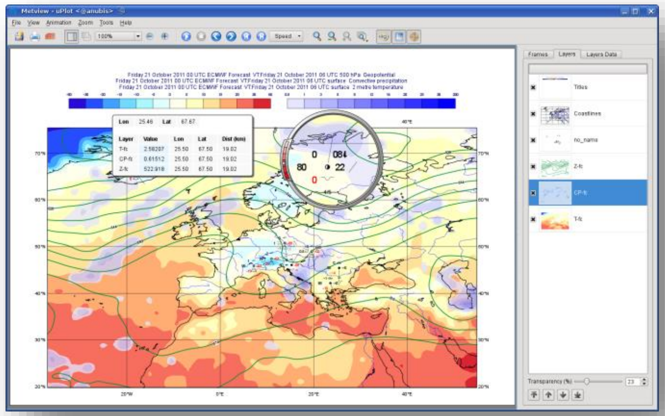
MARS, GRIB_API, Magics, ODB, Emoslib

(ecCodes, MIR)

- **Open Source under Apache Licence 2.0**
  - ***Increased interest from research community***
- **Metview is a co-operation project with INPE (Brazil)**

What is Metview?

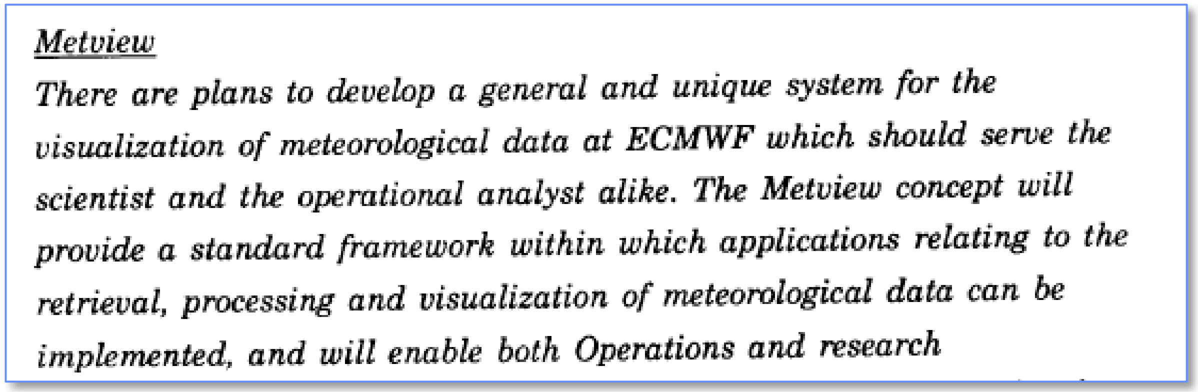




- **Can be run interactively or in batch**
- **Can be easily installed and runs self-contained standalone**
  - **From laptops to supercomputers**
  - **No special data servers required (but can be easily connected to MARS or local databases)**

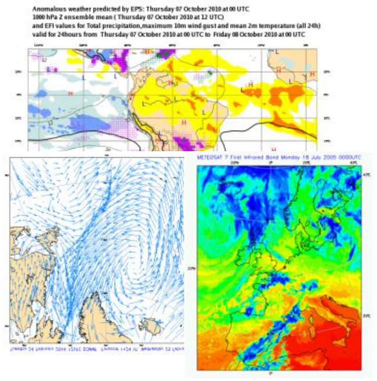
Metview history

- **Announced at first EGOWS in June 1990 (Oslo)**



Metview

There are plans to develop a general and unique system for the visualization of meteorological data at ECMWF which should serve the scientist and the operational analyst alike. The Metview concept will provide a standard framework within which applications relating to the retrieval, processing and visualization of meteorological data can be implemented, and will enable both Operations and research



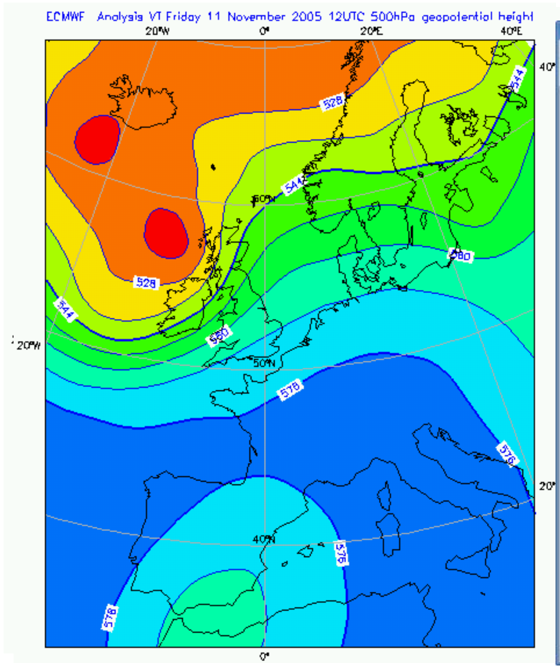
● **First prototype in 1991**	***INPE***
● **First operational version in 1993**	***Metview 1.0***
● **OpenGL graphics introduced in 1998**	***Metview 2.0***
● **New user interface in 2000**	***Metview 3.0***
● **Magics++ and Qt introduced in 2010**	***Metview 4.0***
● **New Qt Desktop introduced in 2014**	***Metview 4.5***
● **Remove all Motif code in 2016**	***Metview 5.0***

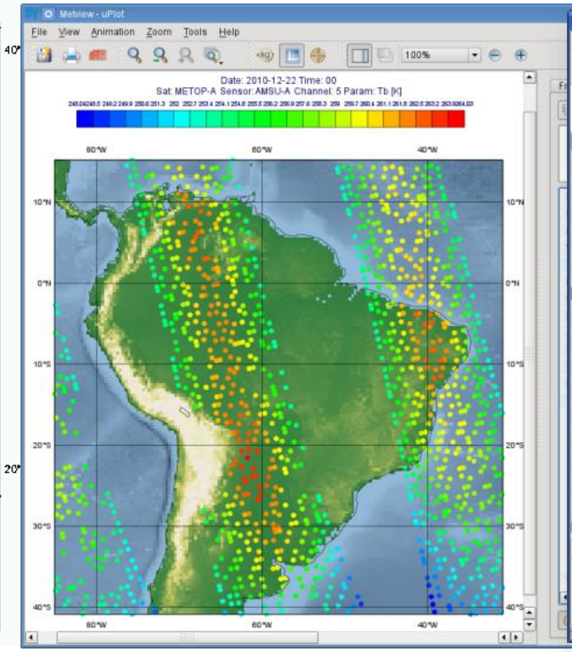
Main features

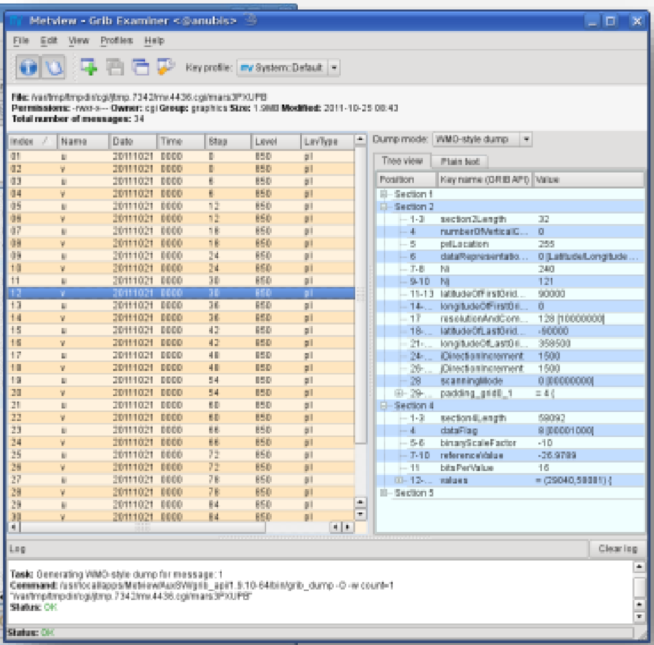
1) Data handling

- Supports a variety of data types (meteorological and non-meteorological)
- Rich set of modules and functions for data manipulation

GRIB
BUFR
NetCDF
ODB
Geopoints
ASCII


ECMWF Analysis VT Friday 11 November 2005 12UTC 500hPa geopotential height
20°W
0°
20°E
40°E
40°
20°W
20°
528
544
560
576
578
580
60°N
50°N
40°N
0°





File: /var/tmp/tmpdir/cgi/tmp.7342/mv.4436.cgi/mars3PXUPB
Permissions: -rwx-x--- **Owner:** cgi **Group:** graphics **Size:** 1.9MB **Modified:** 2011-10-25 08:43
Total number of messages: 34

Index	Name	Date	Time	Step	Level	LevType
01	u	20111021	0000	0	850	pl
02	v	20111021	0000	0	850	pl
03	u	20111021	0000	6	850	pl
04	v	20111021	0000	6	850	pl
05	u	20111021	0000	12	850	pl
06	v	20111021	0000	12	850	pl
07	u	20111021	0000	18	850	pl
08	v	20111021	0000	18	850	pl
09	u	20111021	0000	24	850	pl
10	v	20111021	0000	24	850	pl
11	u	20111021	0000	30	850	pl
12	v	20111021	0000	30	850	pl
13	u	20111021	0000	36	850	pl
14	v	20111021	0000	36	850	pl
15	u	20111021	0000	42	850	pl
16	v	20111021	0000	42	850	pl
17	u	20111021	0000	48	850	pl
18	v	20111021	0000	48	850	pl
19	u	20111021	0000	54	850	pl
20	v	20111021	0000	54	850	pl
21	u	20111021	0000	60	850	pl
22	v	20111021	0000	60	850	pl
23	u	20111021	0000	66	850	pl
24	v	20111021	0000	66	850	pl
25	u	20111021	0000	72	850	pl
26	v	20111021	0000	72	850	pl
27	u	20111021	0000	78	850	pl
28	v	20111021	0000	78	850	pl
29	u	20111021	0000	84	850	pl
30	v	20111021	0000	84	850	pl

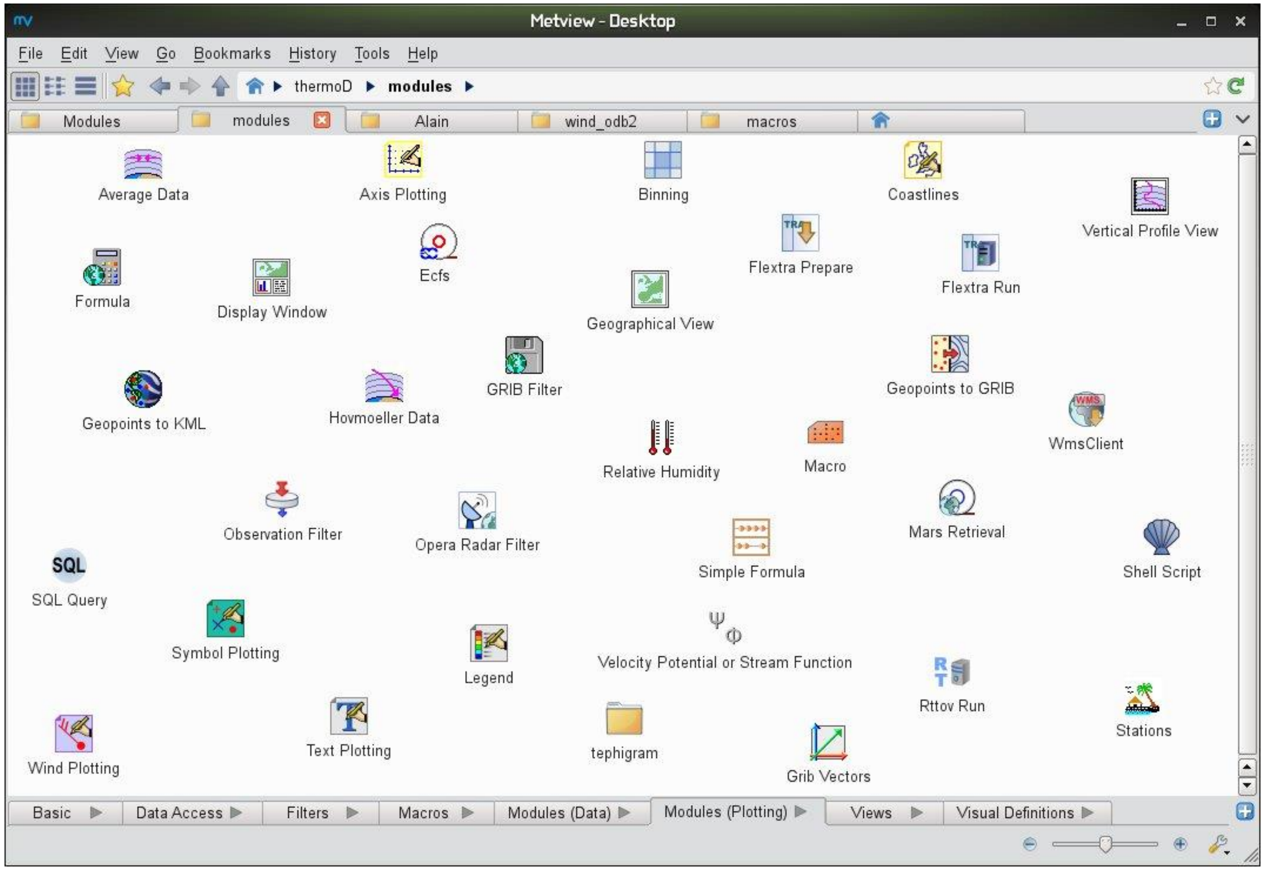
Position	Key name (GRIB API)	Value
Section 1		
Section 2		
1-3	section2Length	32
4	numberOfVerticalC...	0
5	pvlLocation	255
6	dataRepresentatio...	0 [Latitude/Longitude...
7-8	Ni	240
9-10	Nj	121
11-13	latitudeOfFirstGrid...	90000
14-...	longitudeOfFirstGri...	0
17	resolutionAndCom...	128 [10000000]
18-...	latitudeOfLastGrid...	-90000
21-...	longitudeOfLastGri...	358500
24-...	iDirectionIncrement	1500
26-...	jDirectionIncrement	1500
28	scanningMode	0 [00000000]
29-...	padding_grid0_1	= 4 {
Section 4		
1-3	section4Length	58092
4	dataFlag	8 [00001000]
5-6	binaryScaleFactor	-10
7-10	referenceValue	-26.9789
11	bitsPerValue	16
12-...	values	= (29040,58081) {
Section 5		

```
Task: Generating WMO-style dump for message: 1
Command: /usr/local/apps/Metview/AuxSW/grib_api/1.9.10-64bit/grib_dump -O -w count=1
"/var/tmp/tmpdir/cgi/tmp.7342/mv.4436.cgi/mars3PXUPB"
Status: OK
```

Status: OK

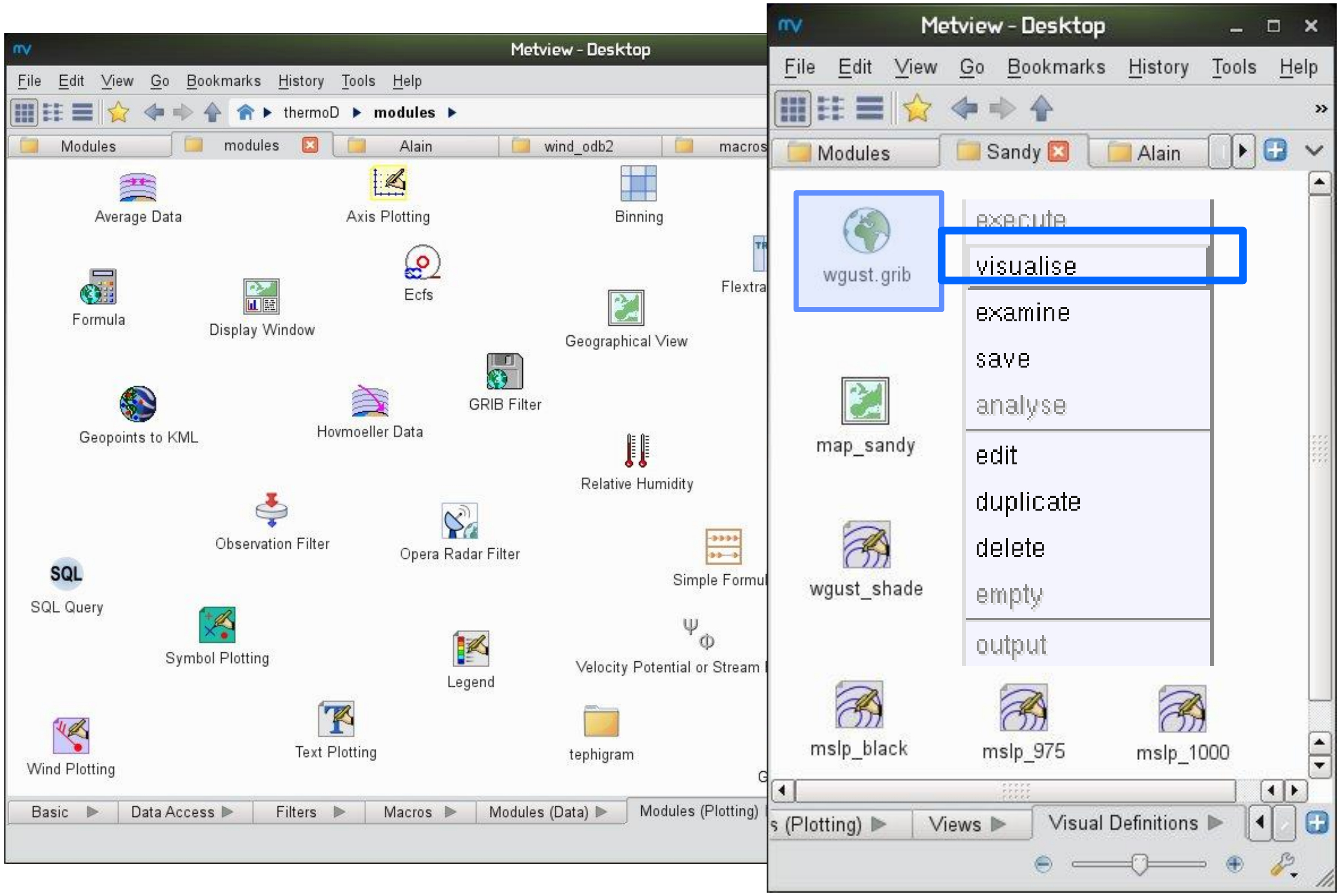
Main features

2) Icon-based interface



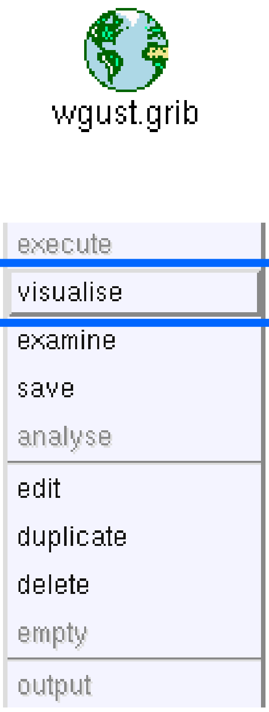
Main features

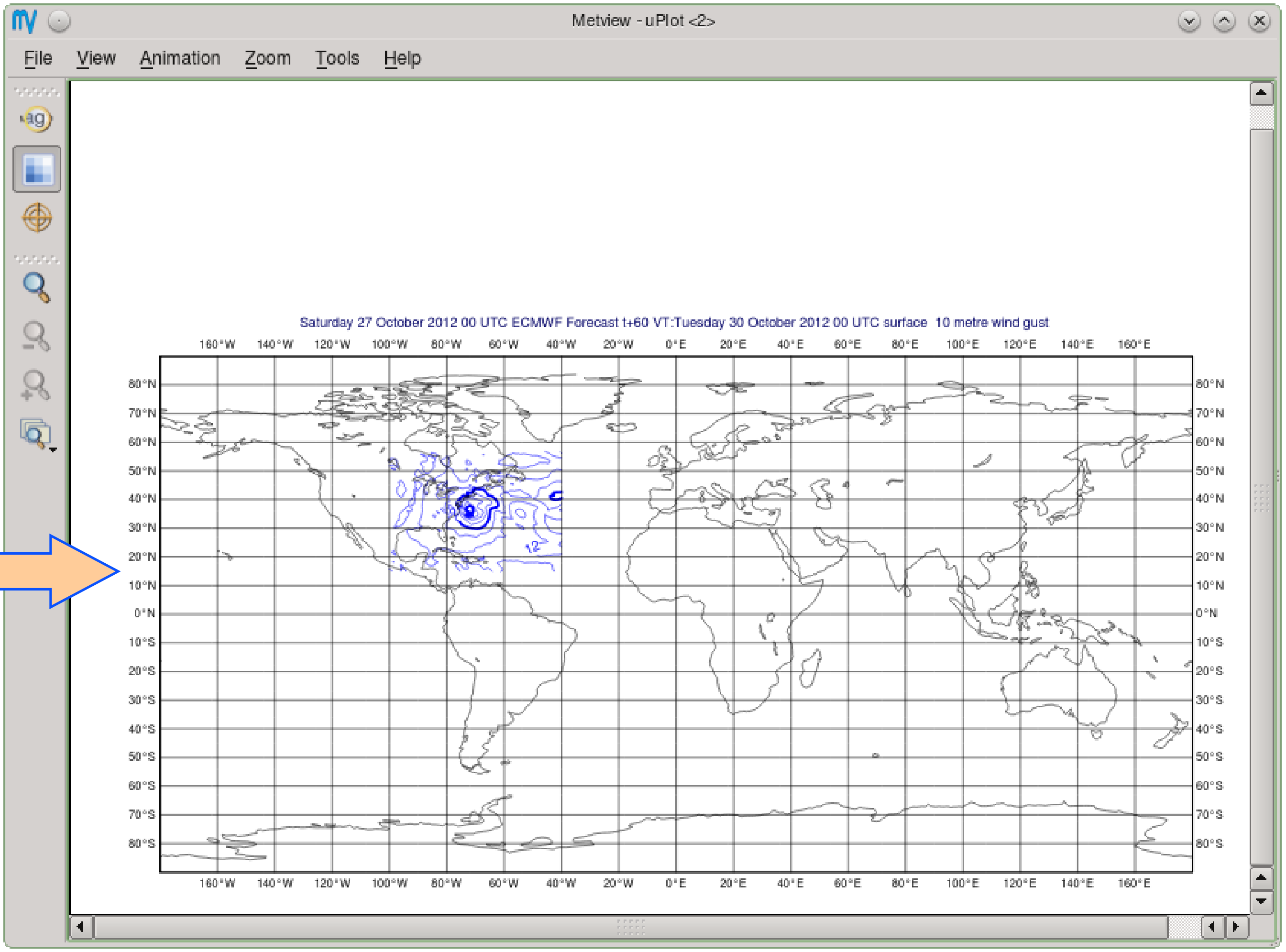
3) Drag and Drop support



Visualisation

GRIB file

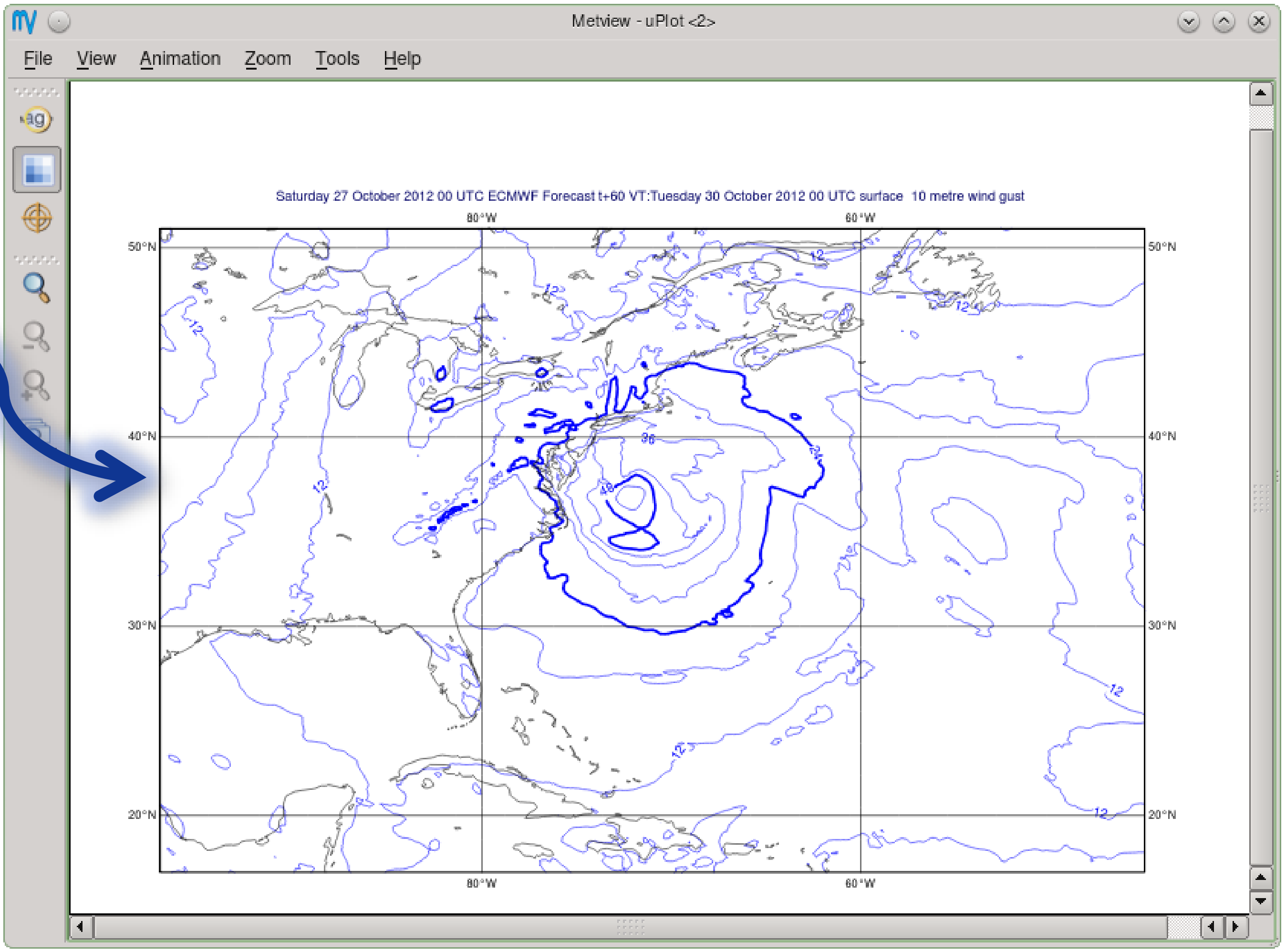



Metview - uPlot <2>
File
View
Animation
Zoom
Tools
Help
Saturday 27 October 2012 00 UTC ECMWF Forecast t+60 VT:Tuesday 30 October 2012 00 UTC surface 10 metre wind gust

Drag and Drop

Map view

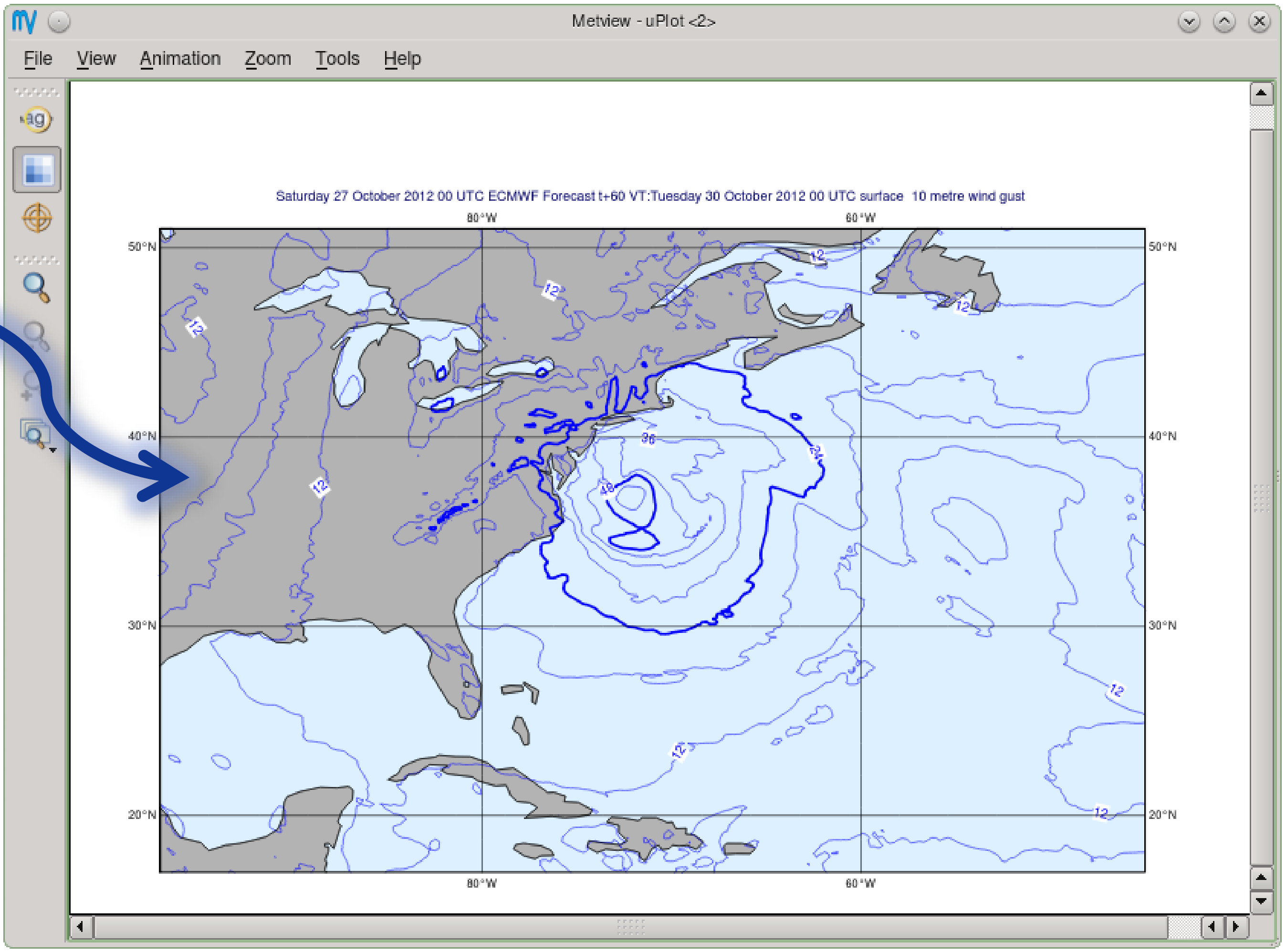
map_sandy


Metview - uPlot <2>
File
View
Animation
Zoom
Tools
Help
Saturday 27 October 2012 00 UTC ECMWF Forecast t+60 VT:Tuesday 30 October 2012 00 UTC surface 10 metre wind gust
80°W
60°W
50°N
40°N
30°N
20°N

Drag and Drop

Coastlines

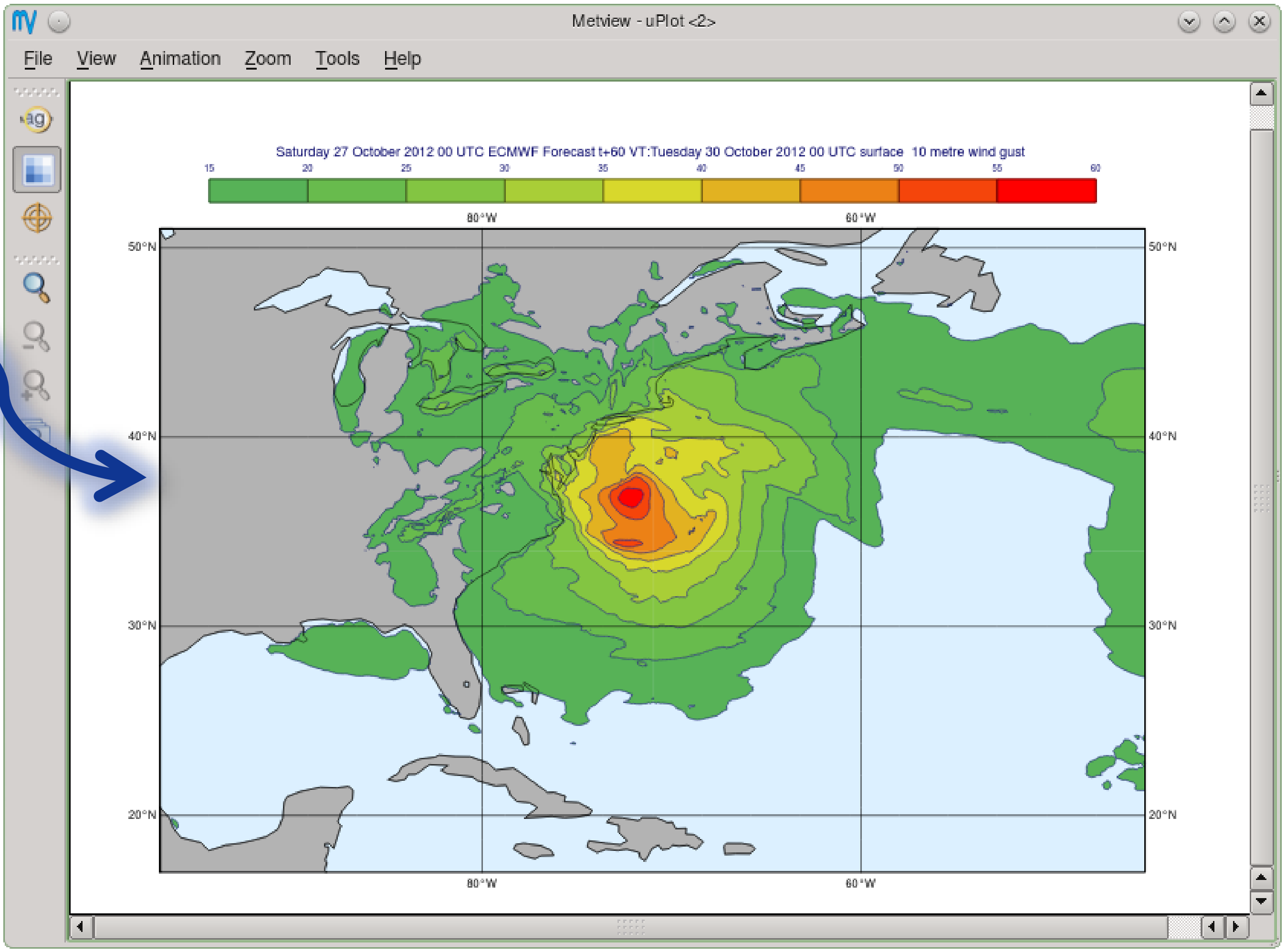
coast_grey_light


Metview - uPlot <2>
File View Animation Zoom Tools Help
Saturday 27 October 2012 00 UTC ECMWF Forecast t+60 VT:Tuesday 30 October 2012 00 UTC surface 10 metre wind gust
80°W
60°W
50°N
40°N
30°N
20°N

Drag and Drop

Contour shading

wgust_shade


Metview - uPlot <2>
File
View
Animation
Zoom
Tools
Help
Saturday 27 October 2012 00 UTC ECMWF Forecast t+60 VT:Tuesday 30 October 2012 00 UTC surface 10 metre wind gust
15
20
25
30
35
40
45
50
55
60
80°W
60°W
50°N
40°N
30°N
20°N

Drag and Drop - Overlay

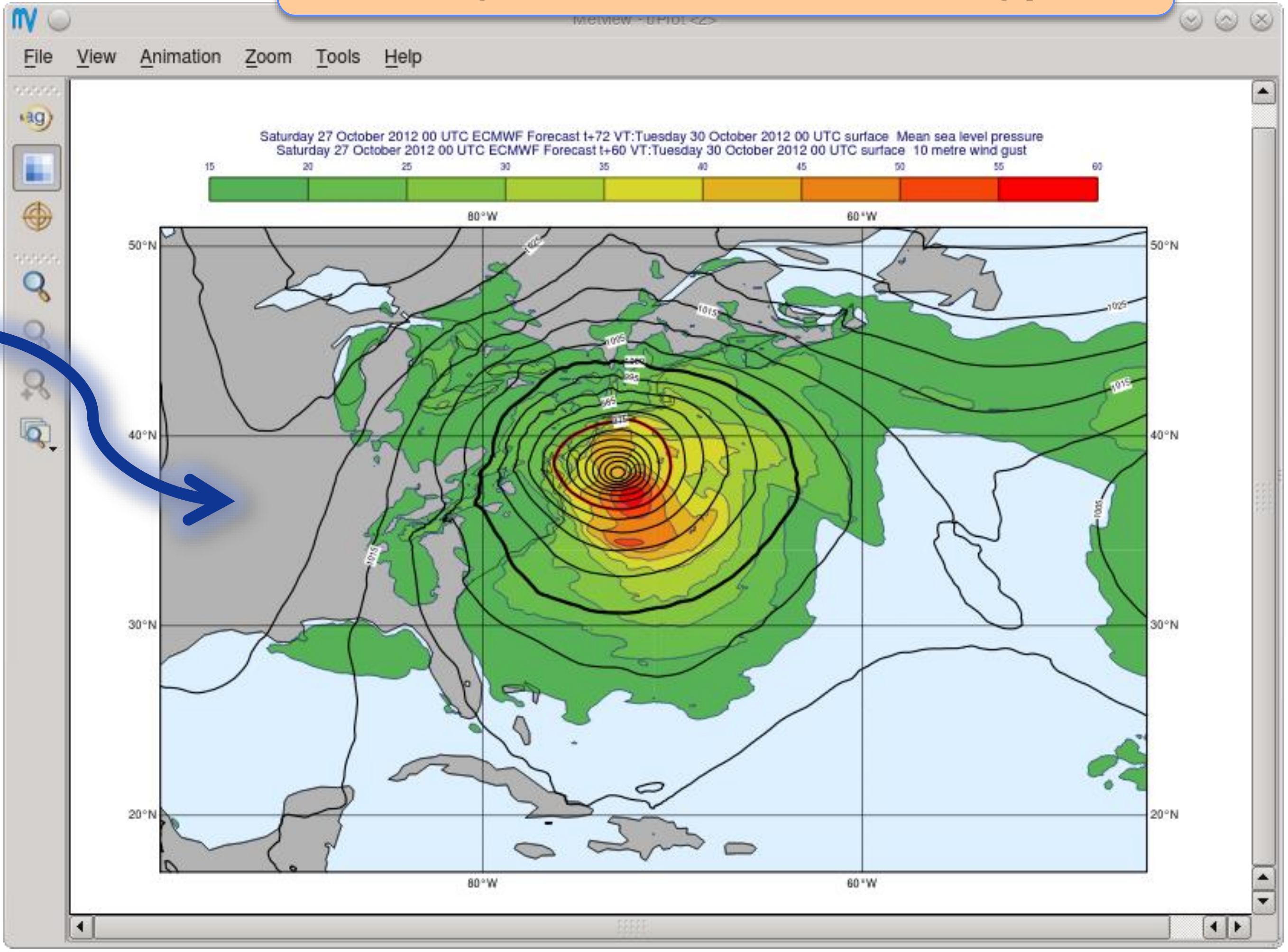
Overlay works for all the data types!

MSLP (GRIB)

mslp.grib

mslp_black mslp_975 mslp_1000

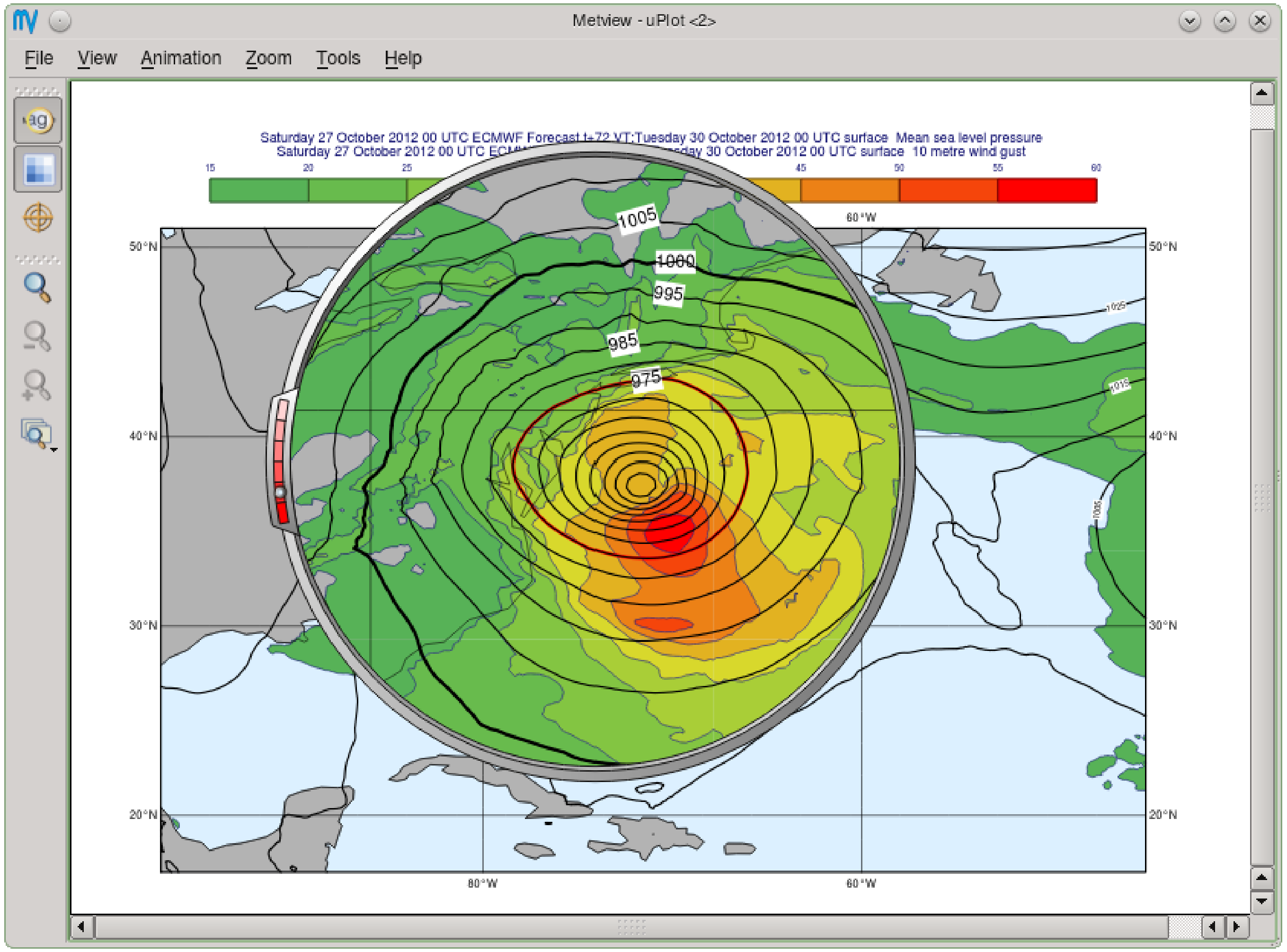
Contouring



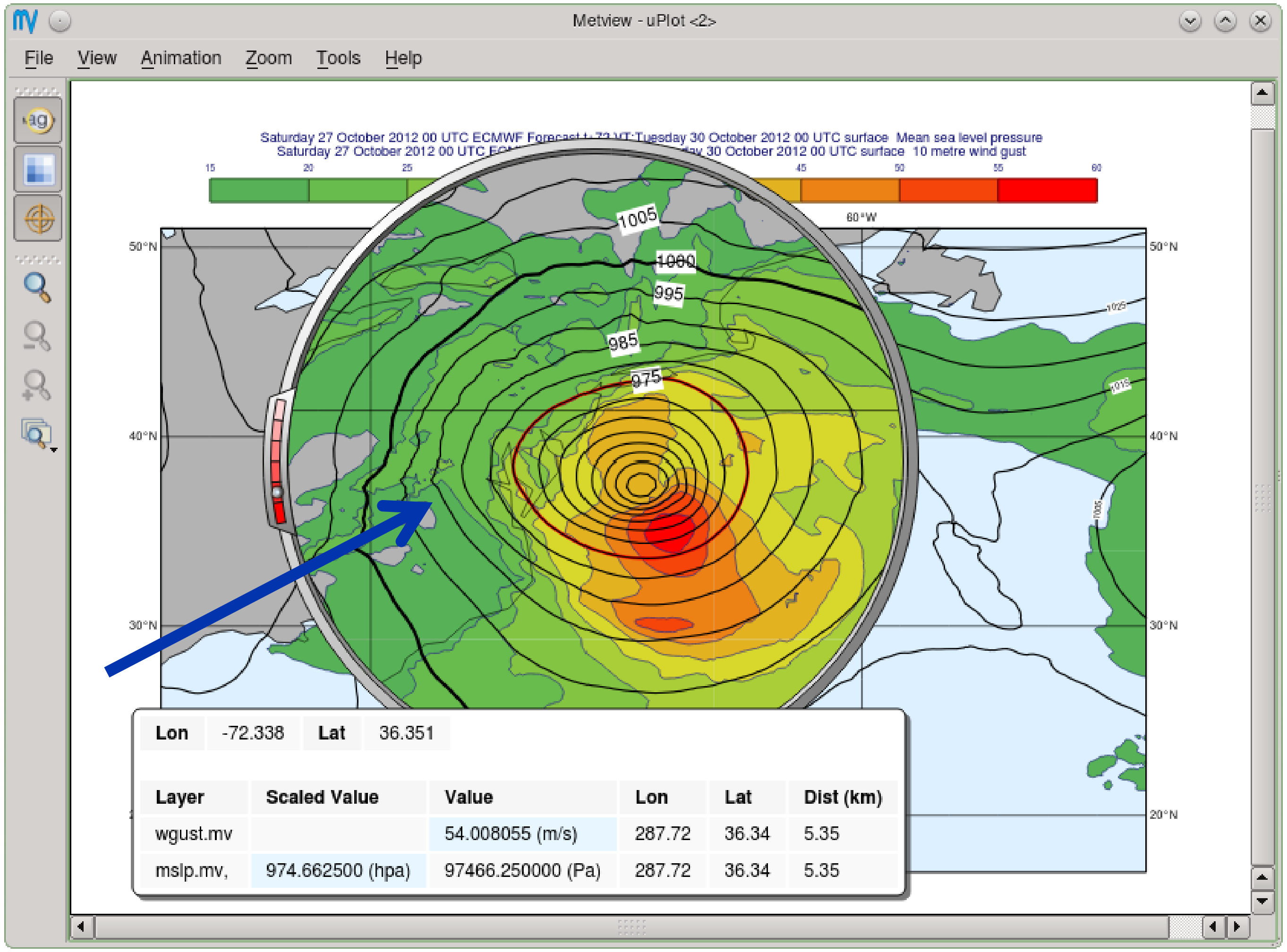
Saturday 27 October 2012 00 UTC ECMWF Forecast t+72 VT:Tuesday 30 October 2012 00 UTC surface Mean sea level pressure
Saturday 27 October 2012 00 UTC ECMWF Forecast t+60 VT:Tuesday 30 October 2012 00 UTC surface 10 metre wind gust

15	20	25	30	35	40	45	50	55	60

Display Window - Magnifier


Metview - uPlot <2>
File
View
Animation
Zoom
Tools
Help
Saturday 27 October 2012 00 UTC ECMWF Forecast t+72 VT:Tuesday 30 October 2012 00 UTC surface Mean sea level pressure
Saturday 27 October 2012 00 UTC surface 10 metre wind gust
1005
1000
995
985
975
50°N
40°N
30°N
20°N
80°W
60°W

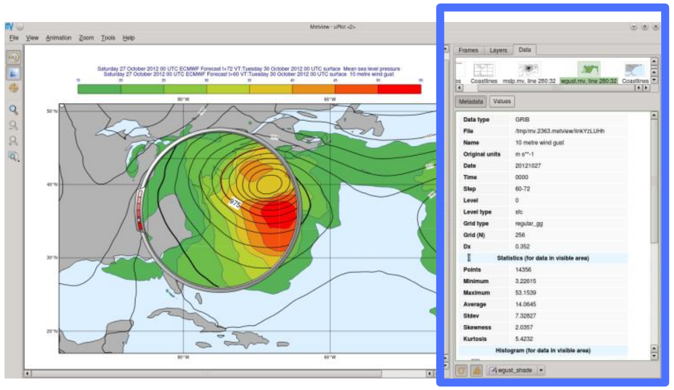
Display Window - Cursor Data


Metview - uPlot <2>
File View Animation Zoom Tools Help
Saturday 27 October 2012 00 UTC ECMWF Forecast t+72 VT:Tuesday 30 October 2012 00 UTC surface Mean sea level pressure
Saturday 27 October 2012 00 UTC ECMWF Forecast t+72 VT:Tuesday 30 October 2012 00 UTC surface 10 metre wind gust
15
20
25
45
50
55
60
60°W
50°N
40°N
30°N
20°N
1005
1000
995
985
975

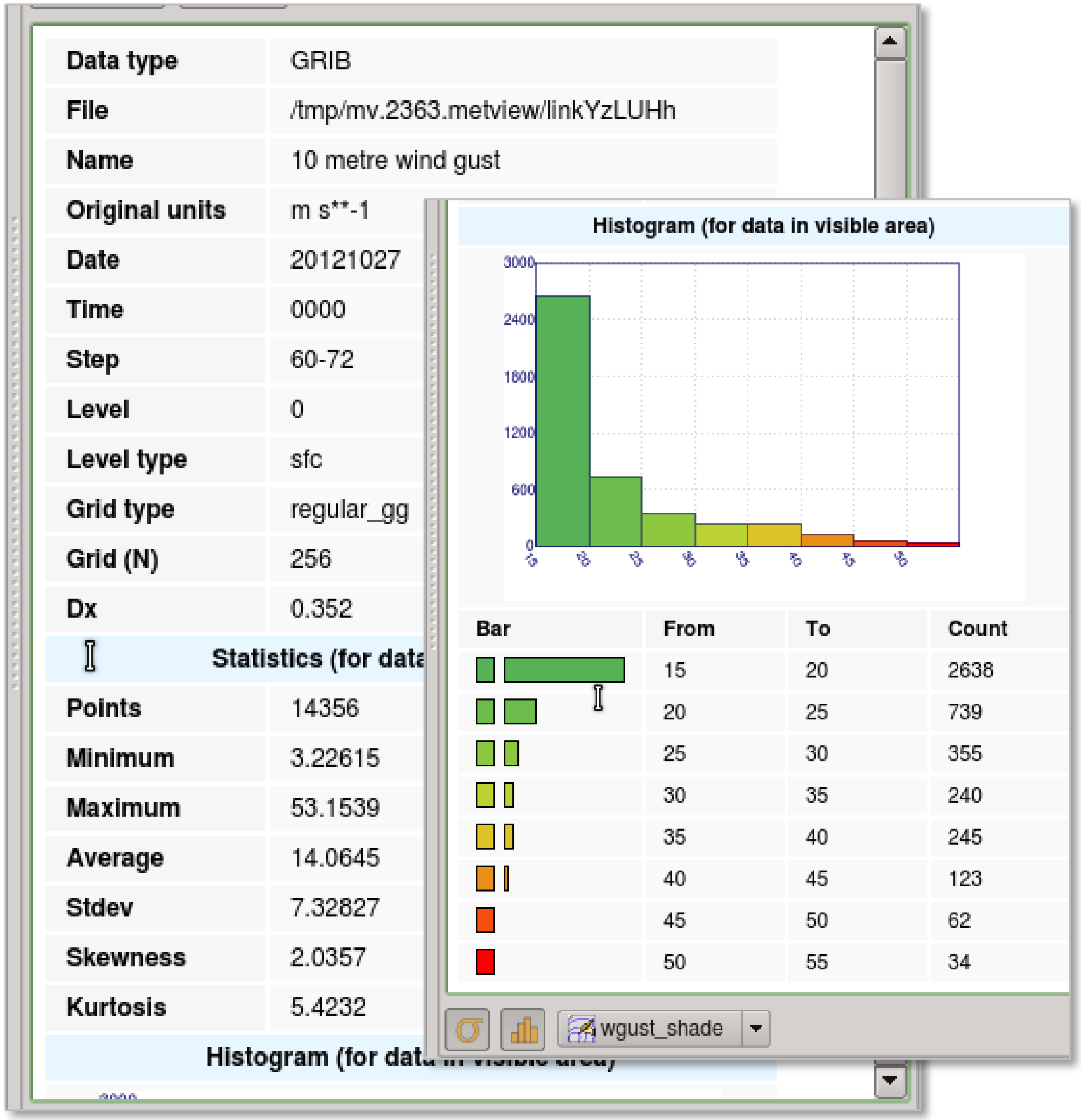
Lon	-72.338	Lat	36.351

Layer	Scaled Value	Value	Lon	Lat	Dist (km)
wgust.mv		54.008055 (m/s)	287.72	36.34	5.35
mslp.mv,	974.662500 (hpa)	97466.250000 (Pa)	287.72	36.34	5.35

Display Window - Layer Metadata



Sidebar with various tabs



Data type	GRIB
File	/tmp/mv.2363.metview/linkYzLUHh
Name	10 metre wind gust
Original units	m s**-1
Date	20121027
Time	0000
Step	60-72
Level	0
Level type	sfc
Grid type	regular_gg
Grid (N)	256
Dx	0.352

Statistics (for data

Points	14356
Minimum	3.22615
Maximum	53.1539
Average	14.0645
Stdev	7.32827
Skewness	2.0357
Kurtosis	5.4232

Histogram (for data in visible area)

Bar	From	To	Count
🟩	15	20	2638
🟩	20	25	739
🟩	25	30	355
🟨	30	35	240
🟨	35	40	245
🟧	40	45	123
🟥	45	50	62
🟥	50	55	34

Main features

4) Macro language

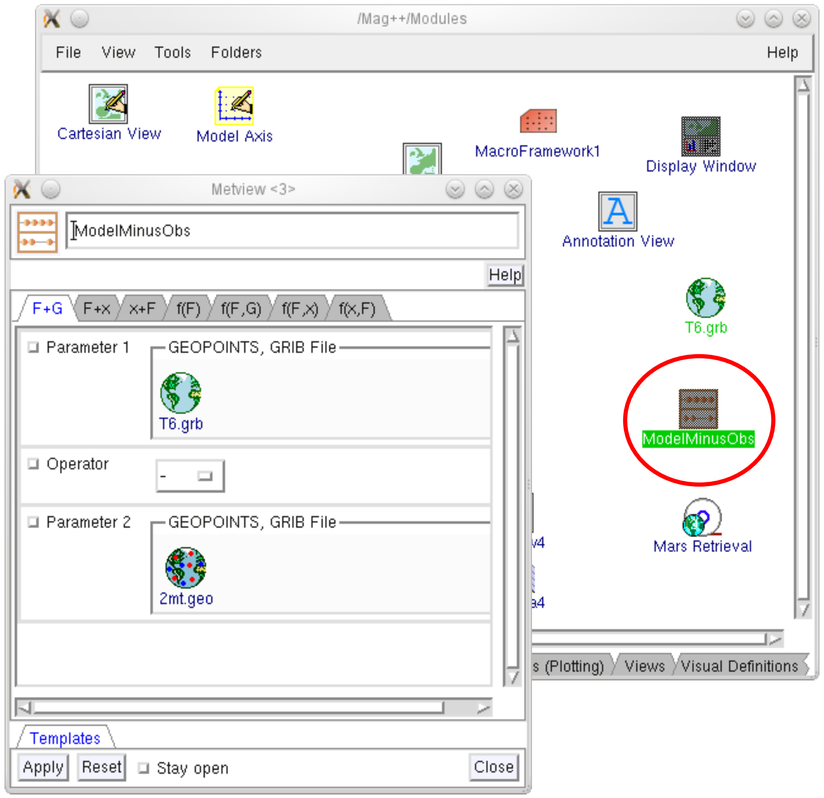
- Powerful meteorologically oriented language
- Simple script language + modern computer language
- Extensive list of functions
- Interfaces with Fortran/C/C++ code
- Outputs:
  - Derived data
  - Interactive plotting window
  - Multiple plots
- Customised editor
- Run in batch or interactive modes

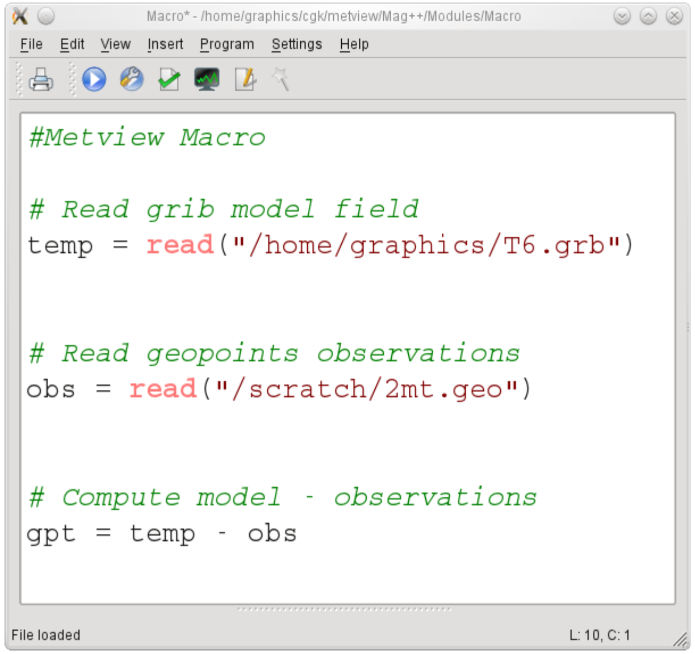
```
# Read a grib file
temp = read ( “/home/graphics/temp.grb” )

# Re-scaling field
if  threshold > 0  then
     temp = temp – 273.5
     a = integrate ( temp )
end if

# Compute the gradient
q = gradientb ( temp )

# Save field
write ( "/home/graphics/gradient.grb“ , q )

# Plot field
plot ( [ps,svg], q )
```

Main features

5) Strong synergy between Icons & Macros

- **Every icon can be translated into a Macro command**



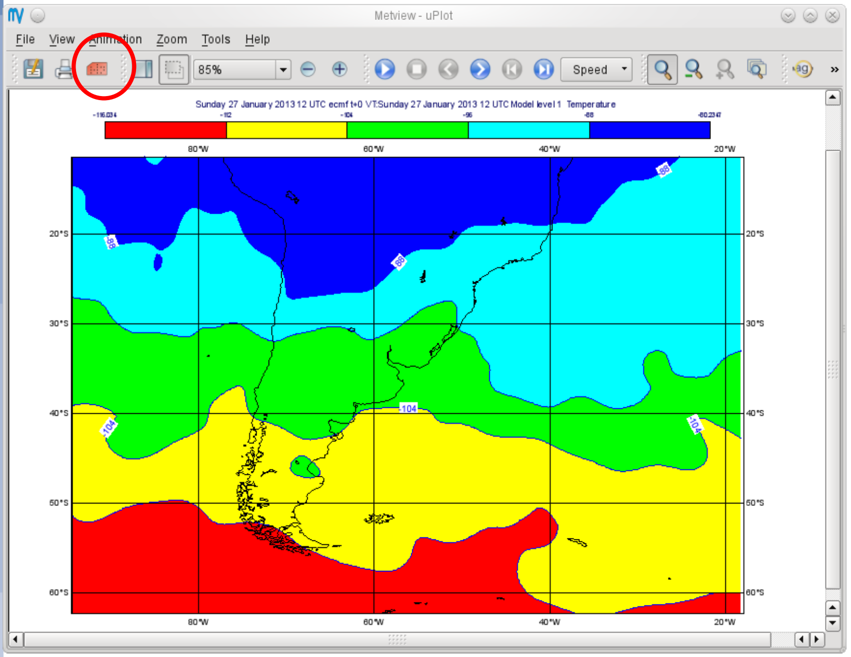

Macro* - /home/graphics/cgk/metview/Mag++/Modules/Macro
File Edit View Insert Program Settings Help
File loaded
L: 10, C: 1

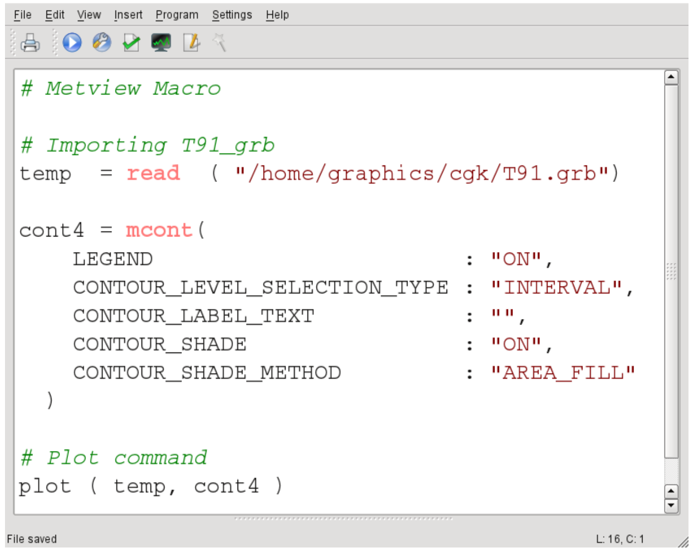
```
#Metview Macro

# Read grib model field
temp = read("/home/graphics/T6.grb")


# Read geopoints observations
obs = read("/scratch/2mt.geo")


# Compute model - observations
gpt = temp - obs
```

Main features

5) Strong synergy between Icons & Macros

- Plots can be translated into a Macro program


Metview - uPlot
File View Animation Zoom Tools Help
85%
Speed
Sunday 27 January 2013 12 UTC ecmf t+0 VT:Sunday 27 January 2013 12 UTC Model level 1 Temperature
80°W
60°W
40°W
20°W
20°S
30°S
40°S
50°S
60°S

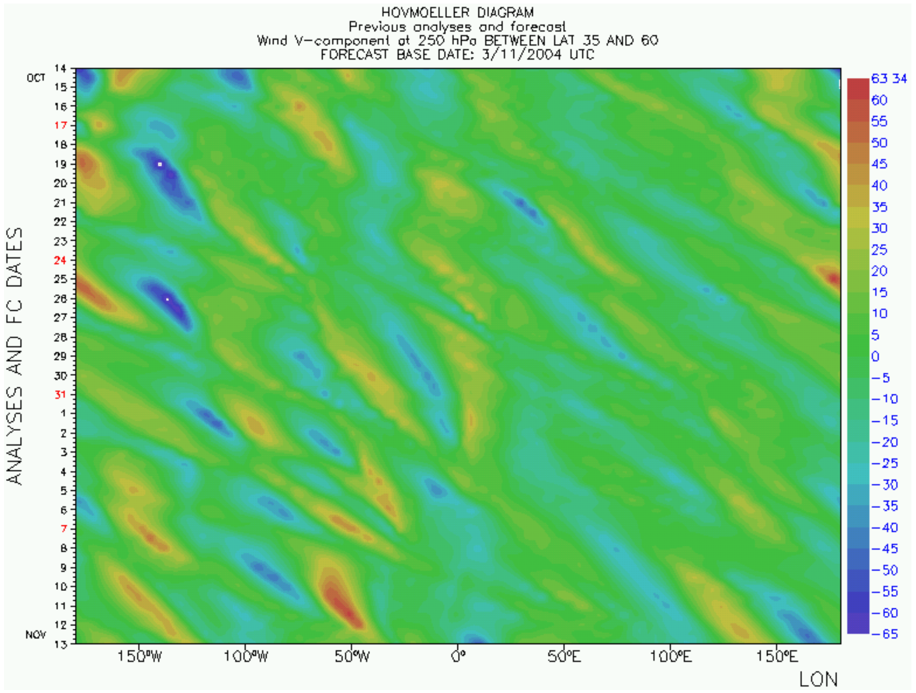


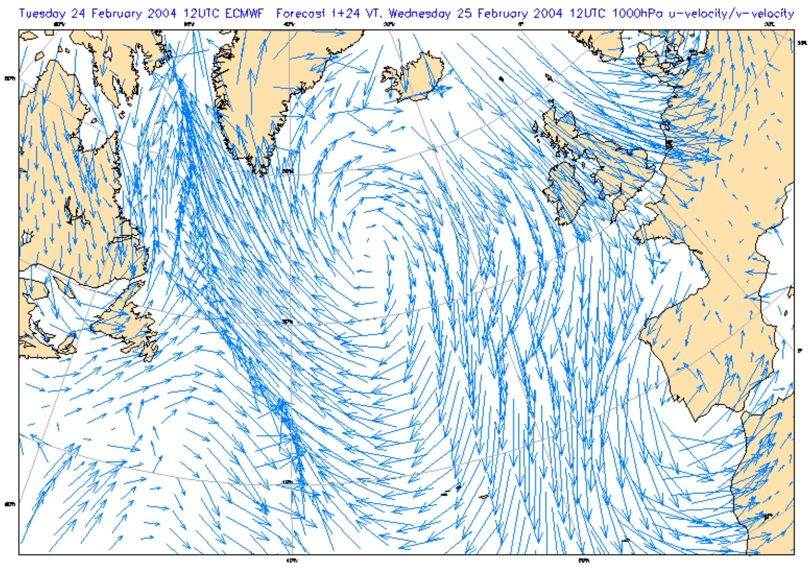
```
# Metview Macro

# Importing T91_grb
temp   = read   ( "/home/graphics/cgk/T91.grb")

cont4  = mcont(
    LEGEND                          : "ON",
    CONTOUR_LEVEL_SELECTION_TYPE    : "INTERVAL",
    CONTOUR_LABEL_TEXT              : "",
    CONTOUR_SHADE                   : "ON",
    CONTOUR_SHADE_METHOD            : "AREA_FILL"
  )

# Plot command
plot ( temp, cont4 )
```

Main features

6) **Can produce a variety of meteorological charts**

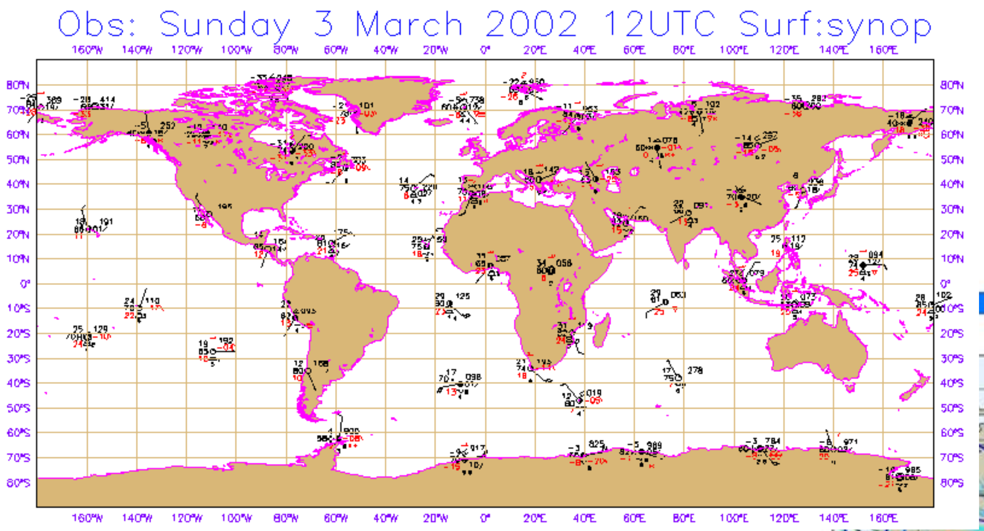
► **Rich set of visualisation attributes**

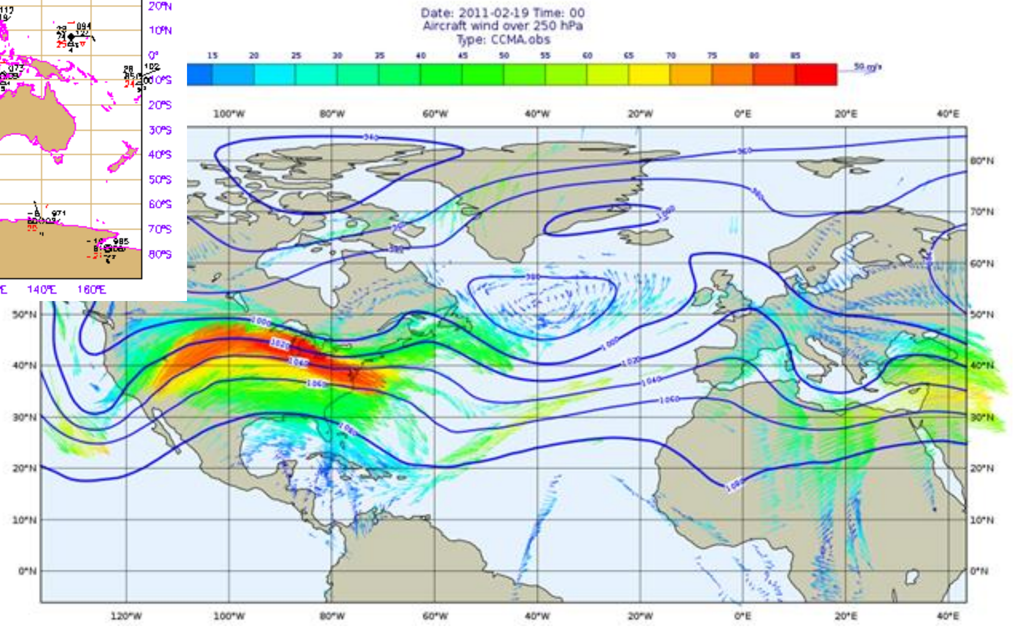



Tuesday 24 February 2004 12UTC ECMWF Forecast t+24 VT: Wednesday 25 February 2004 12UTC 1000hPa u-velocity/v-velocity

Main features

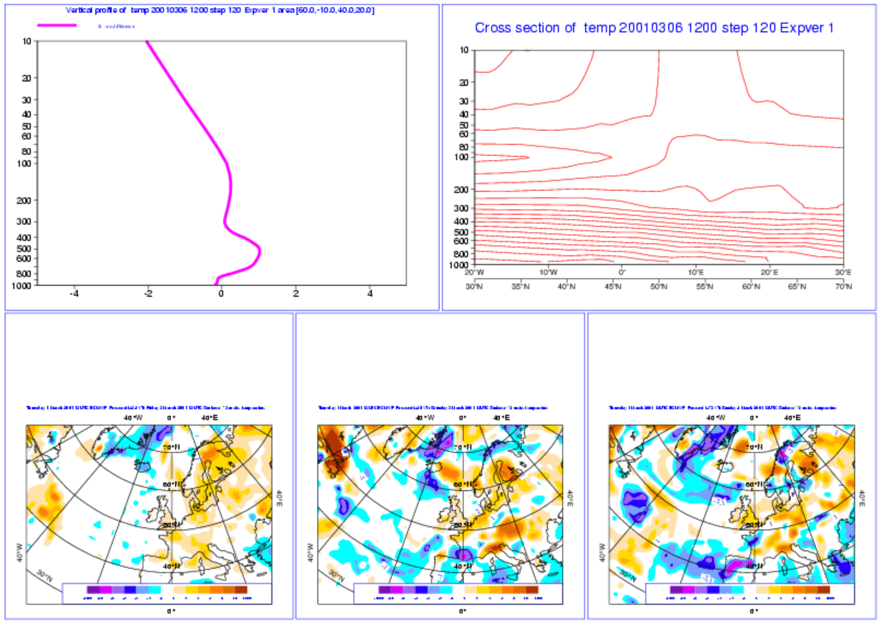
6) **Can produce a variety of meteorological charts**

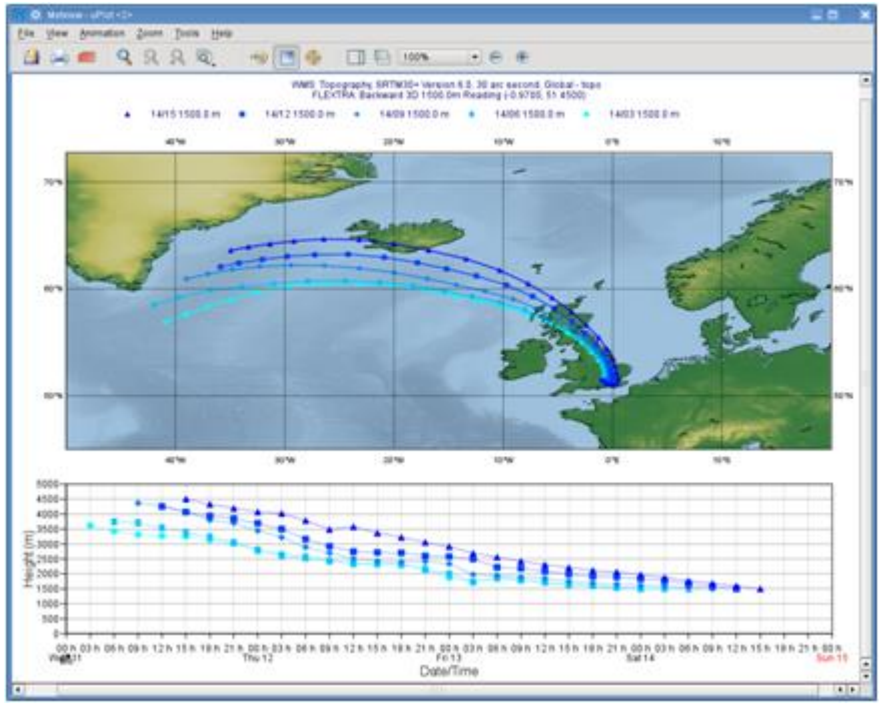




Main features

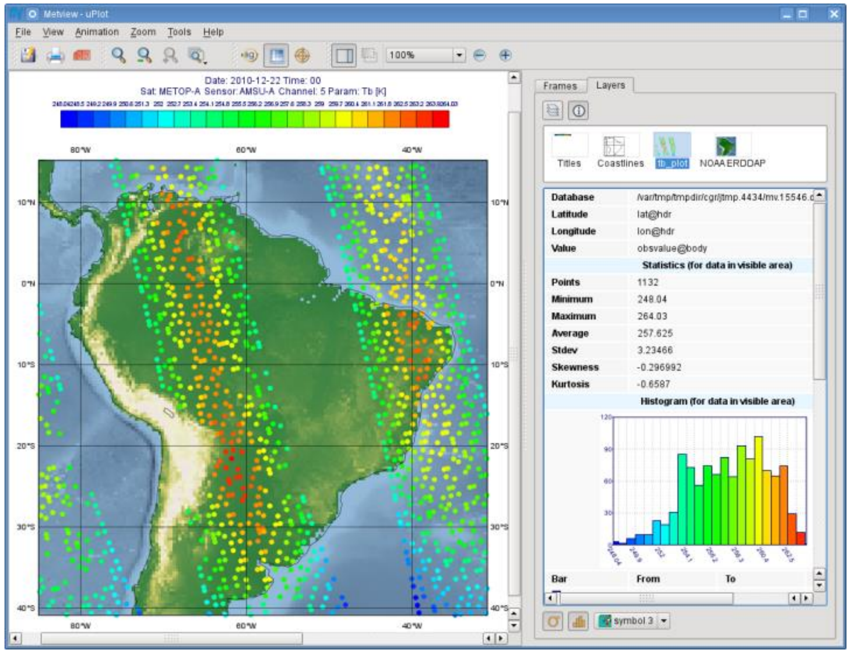
6) **Can produce a variety of meteorological charts**



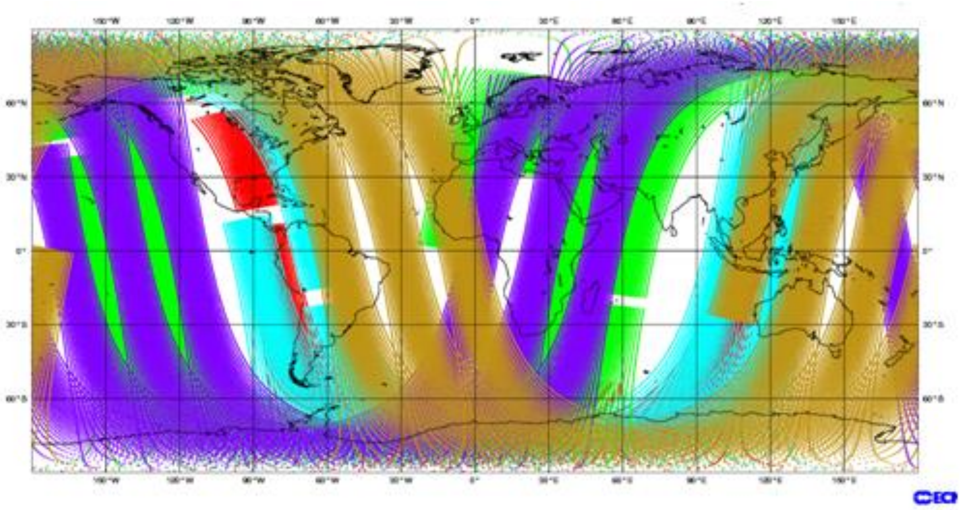

File View Animation Zoom Tools Help
100%
WMS Topography SRTM30+ Version 6.0, 30 arc second Global - topo
FLEXTRA Backward 3D 1500.0m Reading (-0.9795, 51.4500)
Height (m)
Date/Time

Main features

6) Can produce a variety of meteorological charts

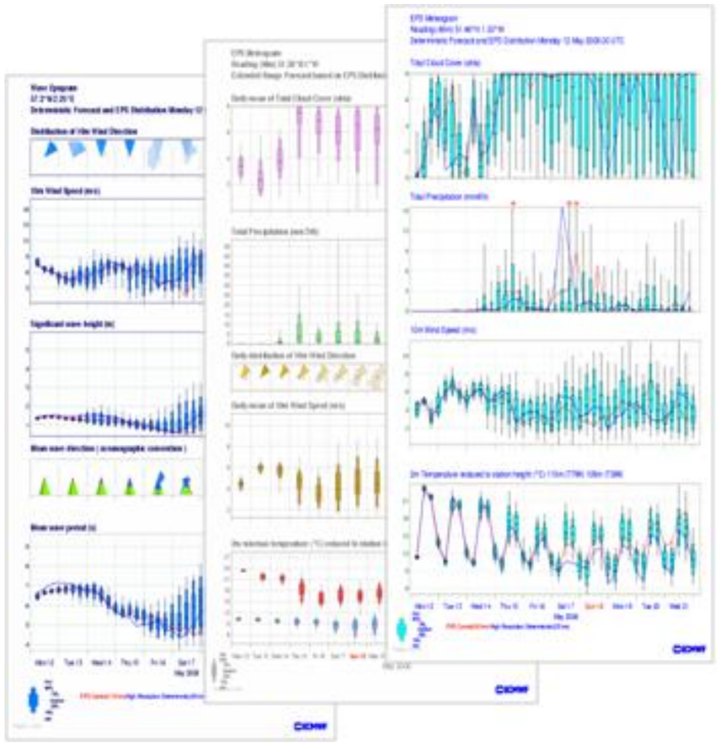


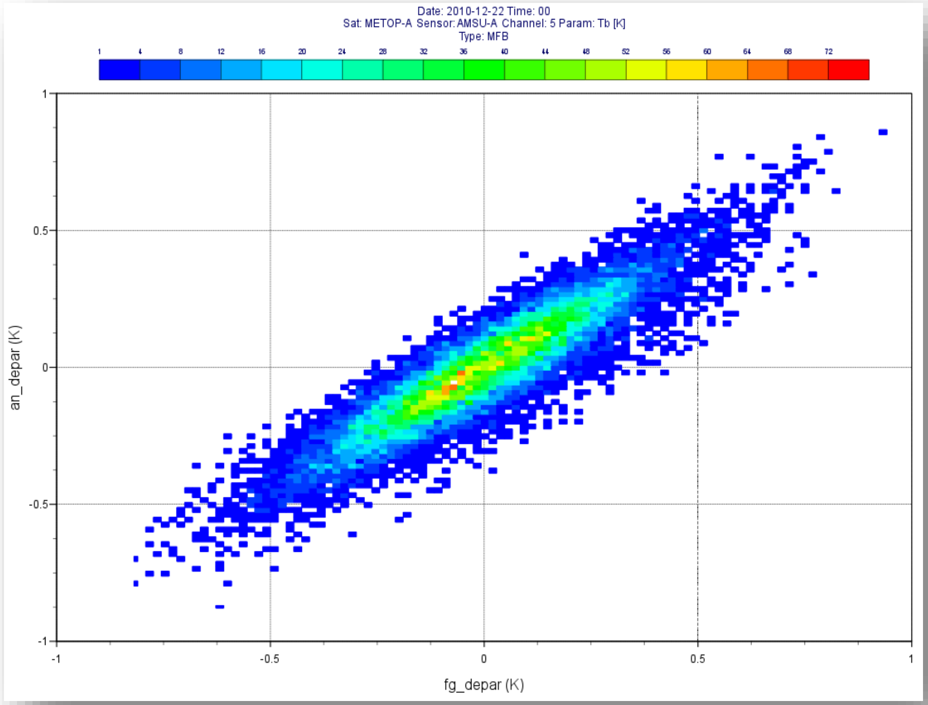
Database	/var/tmp/tmpdir/cgr/jtmp.4434/mv.15546.d
Latitude	lat@hdr
Longitude	lon@hdr
Value	obsvalue@body
	Statistics (for data in visible area)
Points	1132
Minimum	248.04
Maximum	264.03
Average	257.625
Stdev	3.23466
Skewness	-0.296992
Kurtosis	-0.6587
	Histogram (for data in visible area)



Main features

6) **Can produce a variety of meteorological charts**

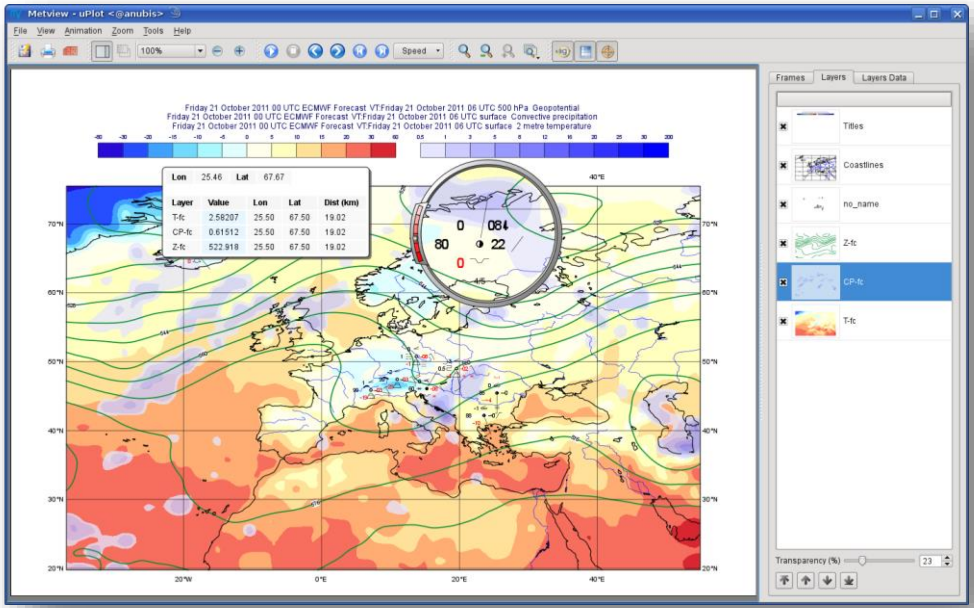




Main features

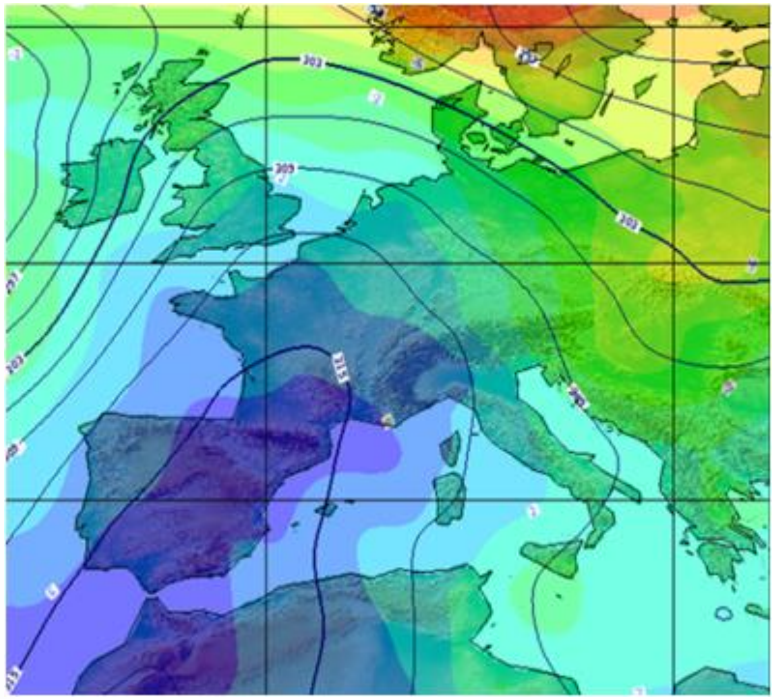
6) **Can produce a variety of meteorological charts**

- ► **Easy to overlay different data sets**


Metview - uPlot <@anubis>
File View Animation Zoom Tools Help
100%
Speed
Frames Layers Layers Data
Titles
Coastlines
no_name
Z-fc
CP-fc
T-fc
Transparency (%) 23

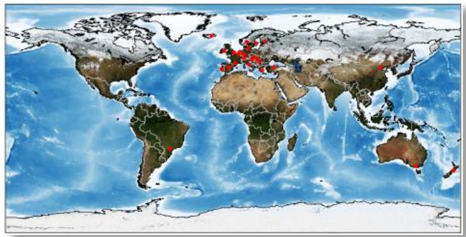
Friday 21 October 2011 00 UTC ECMWF Forecast VT:Friday 21 October 2011 06 UTC 500 hPa Geopotential
Friday 21 October 2011 00 UTC ECMWF Forecast VT:Friday 21 October 2011 06 UTC surface Convective precipitation
Friday 21 October 2011 00 UTC ECMWF Forecast VT:Friday 21 October 2011 06 UTC surface 2 metre temperature

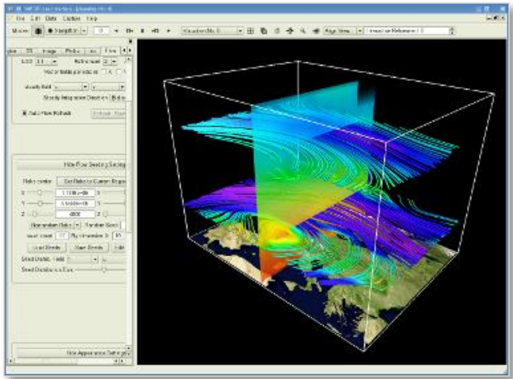
Lon	25.46	Lat	67.67	
Layer	**Value**	**Lon**	**Lat**	**Dist (km)**
T-fc	2.58207	25.50	67.50	19.02
CP-fc	0.61512	25.50	67.50	19.02
Z-fc	522.918	25.50	67.50	19.02



Who uses Metview?

- **Used internally at ECMWF by researchers and operational analysts**
  - **To assess the quality of Observations/Forecast**
  - **To develop new (graphical) products**
  - **For general research activities, publications**
- **Brazil (INPE)**
- **OpenIFS**
- **Member States researchers and forecasters (local installations and remotely on our ecgate server)**
- **Other national weather services and Universities**
- **Commercial customers of ECMWF products**
- **Input for 3D applications (VAPOR, Met3D)**





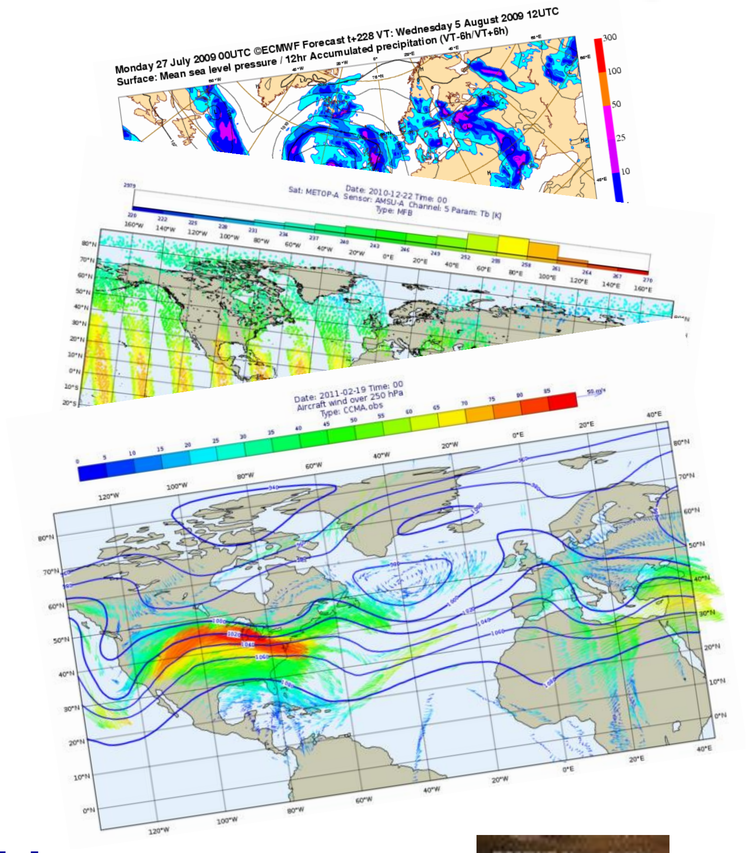
For more information …

email us:

- **Metview: metview@ecmwf.int**

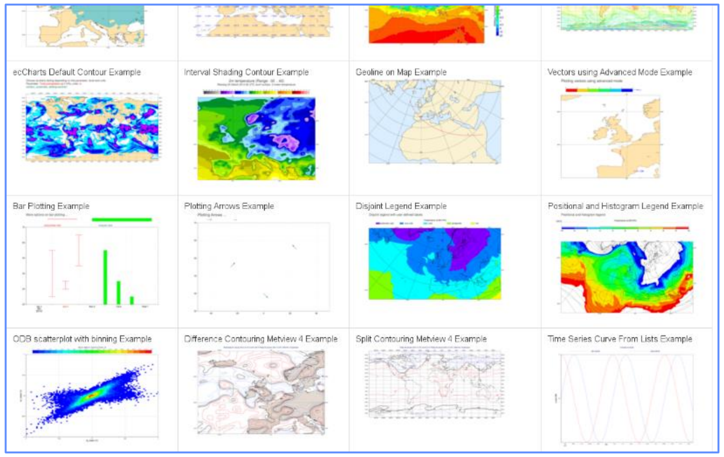
visit our web pages:

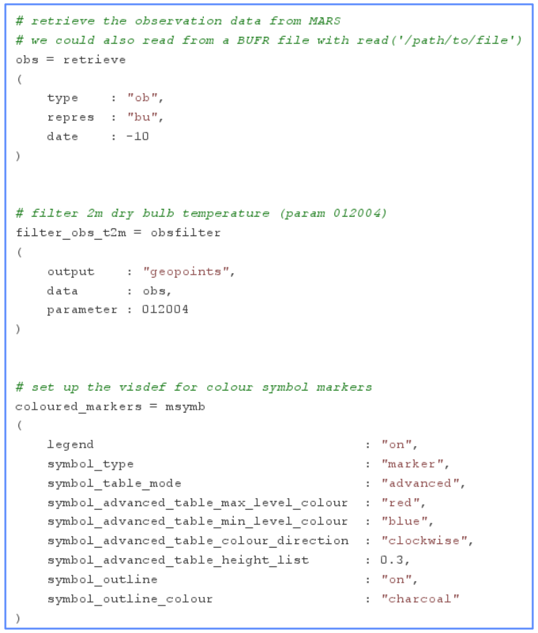
- **https://software.ecmwf.int/metview**
- **Download**
  - **Source code, RPMs, virtual machine**
- **Documentation and tutorials available**
- **Metview articles in recent ECMWF newsletters**


Monday 27 July 2009 00UTC ©ECMWF Forecast t+228 VT: Wednesday 5 August 2009 12UTC
Surface: Mean sea level pressure / 12hr Accumulated precipitation (VT-6h/VT+6h)
Sat: METOP-A Sensor: AMSU-A Channel: 5 Param: Tb [K]
Date: 2010-12-22 Time: 00
Type: MFB
Date: 2011-02-19 Time: 00
Aircraft wind over 250 hPa
Type: CCMA obs

Tutorial structure

- **Each part has:**
  - **Introductory slides**
  - **The tutorial itself**
    - **We are here to help and answer questions!**
  - **Extra tasks if you have finished quickly**
  - **If you still have extra time, you could examine the Metview Gallery:**
    - **https://software.ecmwf.int/metview/Gallery**


ecCharts Default Contour Example
Interval Shading Contour Example
Geoline on Map Example
Vectors using Advanced Mode Example
Bar Plotting Example
Plotting Arrows Example
Disjoint Legend Example
Positional and Histogram Legend Example
ODB scatterplot with binning Example
Difference Contouring Metview 4 Example
Split Contouring Metview 4 Example
Time Series Curve From Lists Example



```
# retrieve the observation data from MARS
# we could also read from a BUFR file with read('/path/to/file')
obs = retrieve
(
    type    : "ob",
    repres  : "bu",
    date    : -10
)


# filter 2m dry bulb temperature (param 012004)
filter_obs_t2m = obsfilter
(
    output    : "geopoints",
    data      : obs,
    parameter : 012004
)


# set up the visdef for colour symbol markers
coloured_markers = msymb
(
    legend                                  : "on",
    symbol_type                             : "marker",
    symbol_table_mode                       : "advanced",
    symbol_advanced_table_max_level_colour  : "red",
    symbol_advanced_table_min_level_colour  : "blue",
    symbol_advanced_table_colour_direction  : "clockwise",
    symbol_advanced_table_height_list       : 0.3,
    symbol_outline                          : "on",
    symbol_outline_colour                   : "charcoal"
)
```

A quick tour of Metview

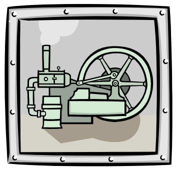
Fernando Ii

Software Applications Team

Metview Principles

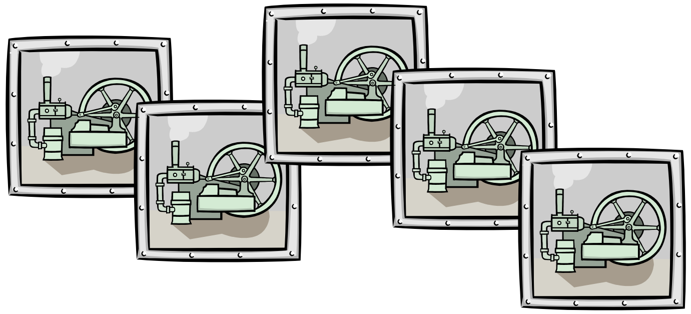
- **First Metview Principle:**

  "Everything in Metview is an Icon"



- **Second Metview Principle:**

  "Every Metview Task is a sequence of actions on icons"



Metview Desktop

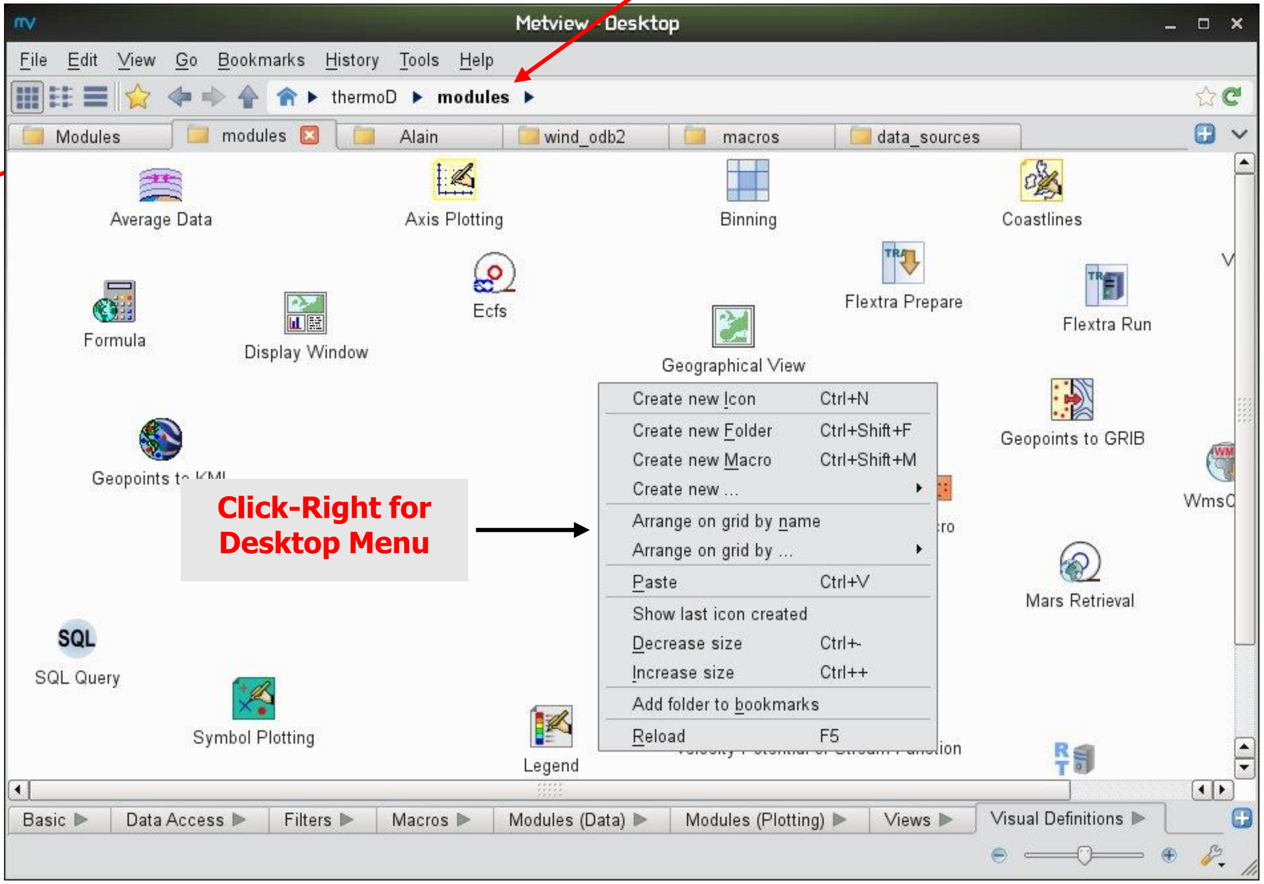
Navigation

View styles

Bookmarks

Icon Drawers

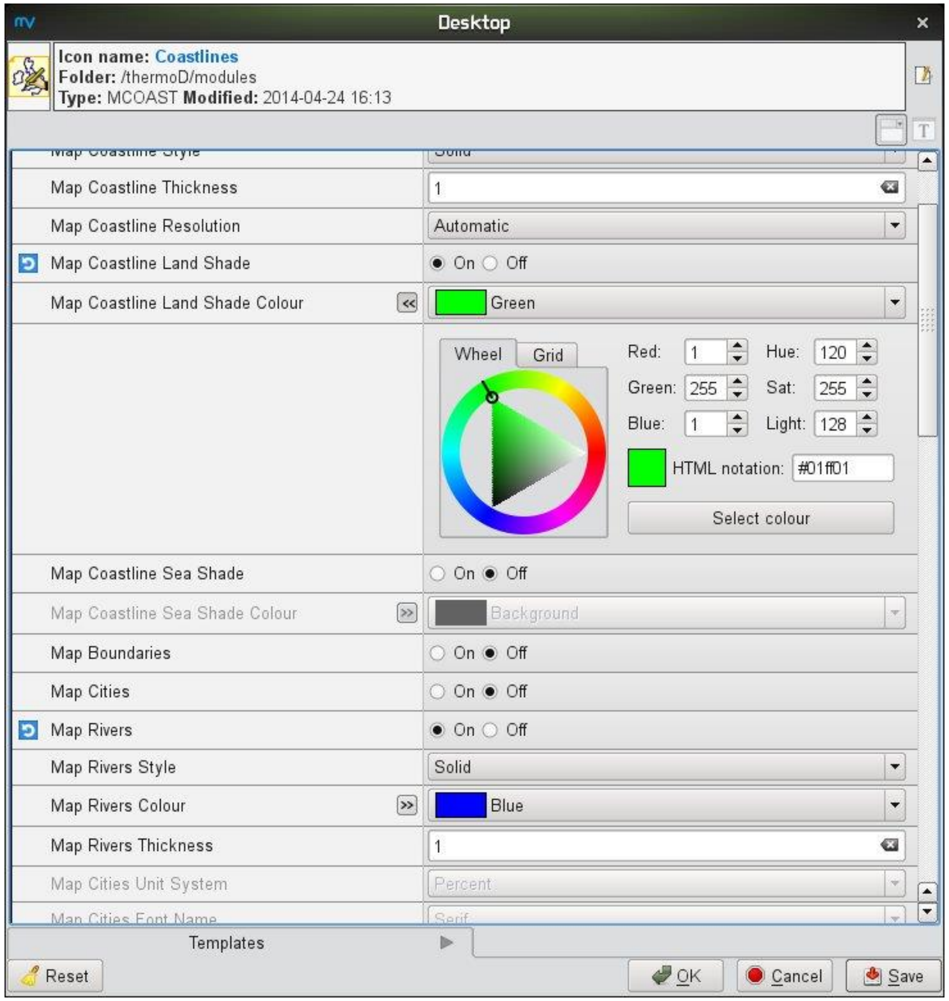
Icon size



Icon Standard Editor

Input area

Save/Exit area



Map Coastline Style	Solid
Map Coastline Thickness	1
Map Coastline Resolution	Automatic
Map Coastline Land Shade	[✓ On] Off
Map Coastline Land Shade Colour	🟩 Green
	Red: 1 Hue: 120 Green: 255 Sat: 255 Blue: 1 Light: 128 HTML notation: #01ff01
Map Coastline Sea Shade	On [✓ Off]
Map Coastline Sea Shade Colour	Background
Map Boundaries	On [✓ Off]
Map Cities	On [✓ Off]
Map Rivers	[✓ On] Off
Map Rivers Style	Solid
Map Rivers Colour	🟦 Blue
Map Rivers Thickness	1
Map Cities Unit System	Percent
Map Cities Font Name	Serif

Input element: Alphanumeric Field

Input element: Colour Menu

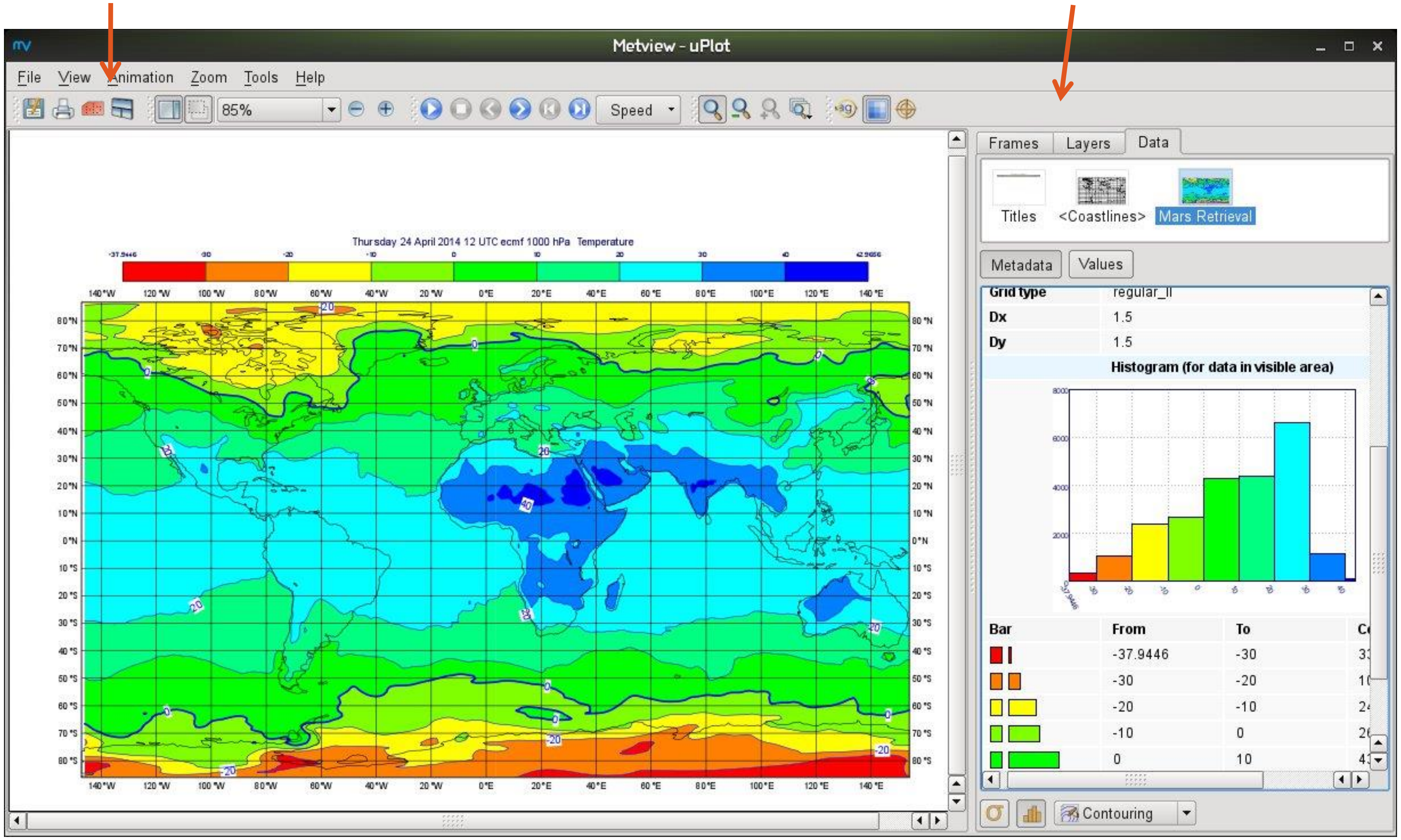
Input element: Toggle option

Input element: Option Menu

Display Window

Controls

Metadata

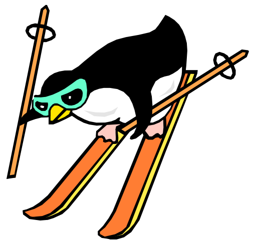


Grid type	regular_ll
Dx	1.5
Dy	1.5

Histogram (for data in visible area)

Bar	From	To	C
🟥	-37.9446	-30	3
🟧	-30	-20	1
🟨	-20	-10	2
🟩	-10	0	2
🟩	0	10	4

Desktop Behaviour (1)

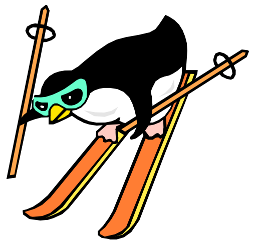


- **KDE settings relevant to Metview: (personal preference)**

1) Change the window behaviour

- **KDE menu (icon at bottom-left)**
- **System Settings**
- **Window behaviour**
- **Window behaviour**
- **Set *Focus stealing prevention level* to "*None*"**
- **Set *Policy* to "Focus Follows Mouse"**
- **Disable *Click raises active window***
- **Apply and close the dialog**

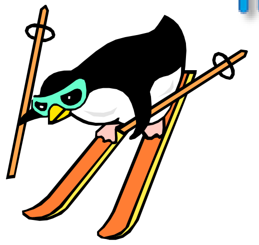
Desktop Behaviour (2)



2) Change the desktop behaviour

- KDE menu (icon at bottom-left)
- System Settings
- Desktop
- Screen Edges
- Disable the settings
  - *Maximise windows by dragging…*
  - *Tile windows by dragging....*
- Apply and close the dialog

Starting Metview



- **To start Metview, please type the following command from an *xterm*:**

```
module swap metview/train
metview &
```

- **Please minimise the *xterm* but do not close it**

Metview Tutorial: A Simple Visualisation

Get the data and icons for the day

- **From a command line type:**

```
~trx/mv_data/get_day_1.sh
```

- **A new folder called “training” will appear in your Metview desktop**

- **A new folder called “day_1” will appear in your “training” folder**

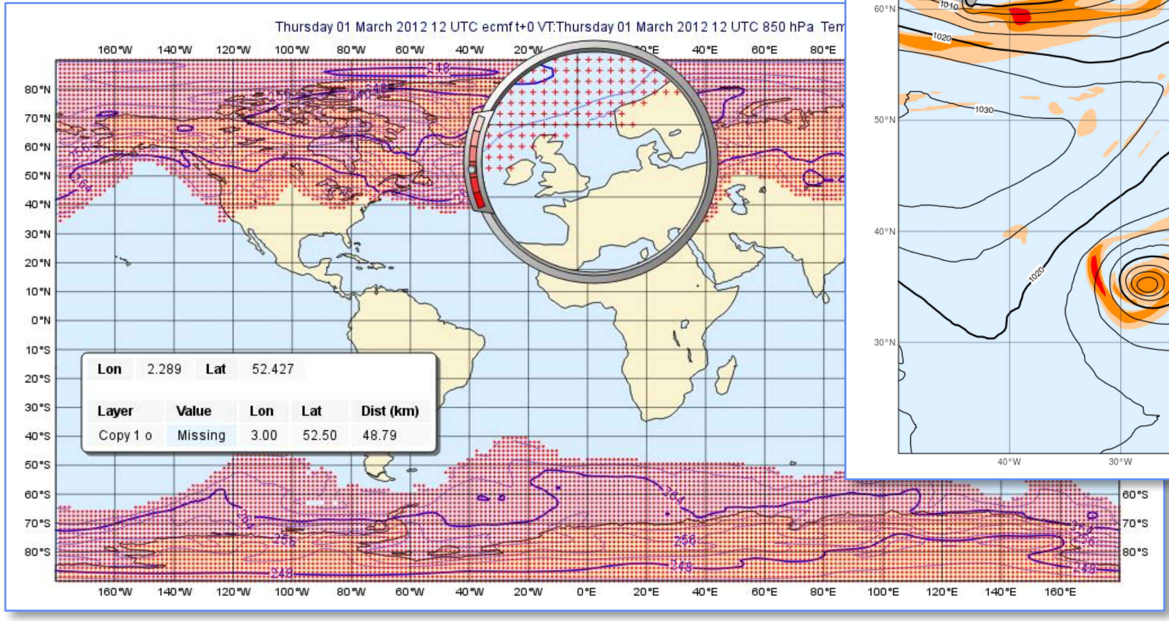
- **Please do exercise “A Simple Visualisation” in the provided sub-folder “a simple visualisation”**

Additional Notes

- Metview scans its open folders for new files every 8 seconds
- ‘View | Reload’ forces an immediate rescan (F5)
- Deleted icons go into the Wastebasket – right-click, Empty to finally delete icons from there
- Layer meta-data reflects the selected area

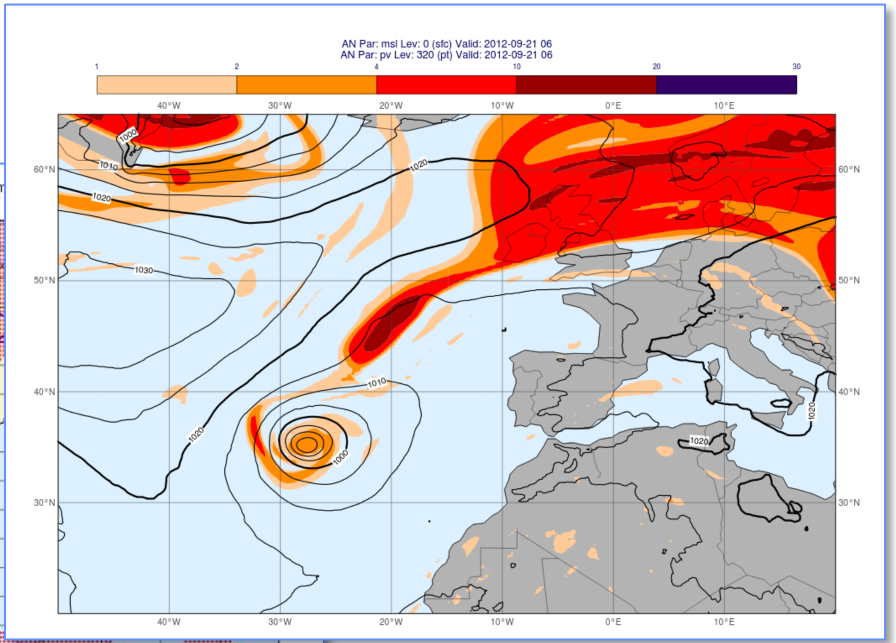
Customising your plot

- **Almost every aspect of the plot can be customised**
- **For example:**
  - **Coastlines, isolines, grid labels, titles, legend, …**

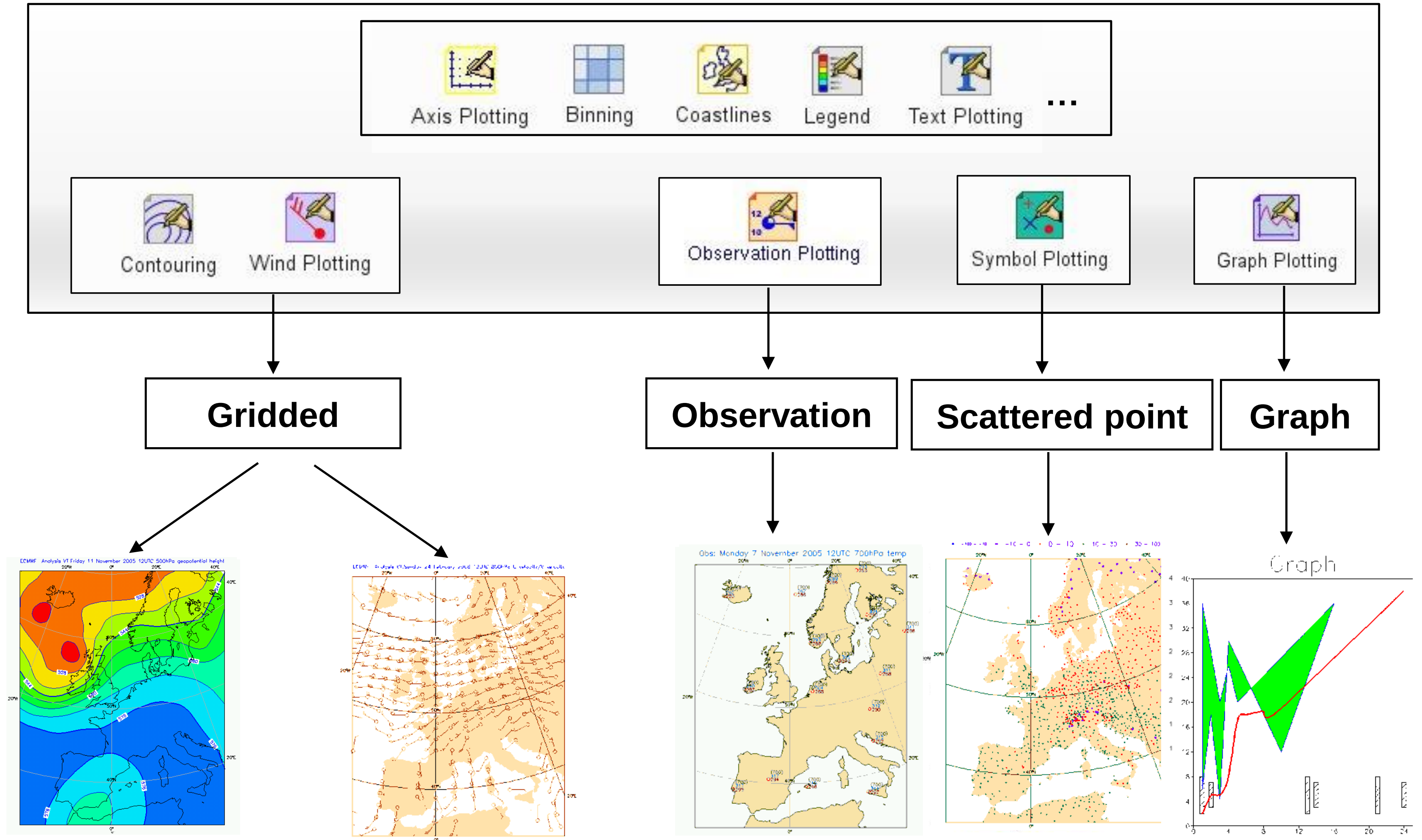

Thursday 01 March 2012 12 UTC ecmf t+0 VT:Thursday 01 March 2012 12 UTC 850 hPa Tem

Lon	2.289	Lat	52.427

Layer	Value	Lon	Lat	Dist (km)
Copy 1 o	Missing	3.00	52.50	48.79

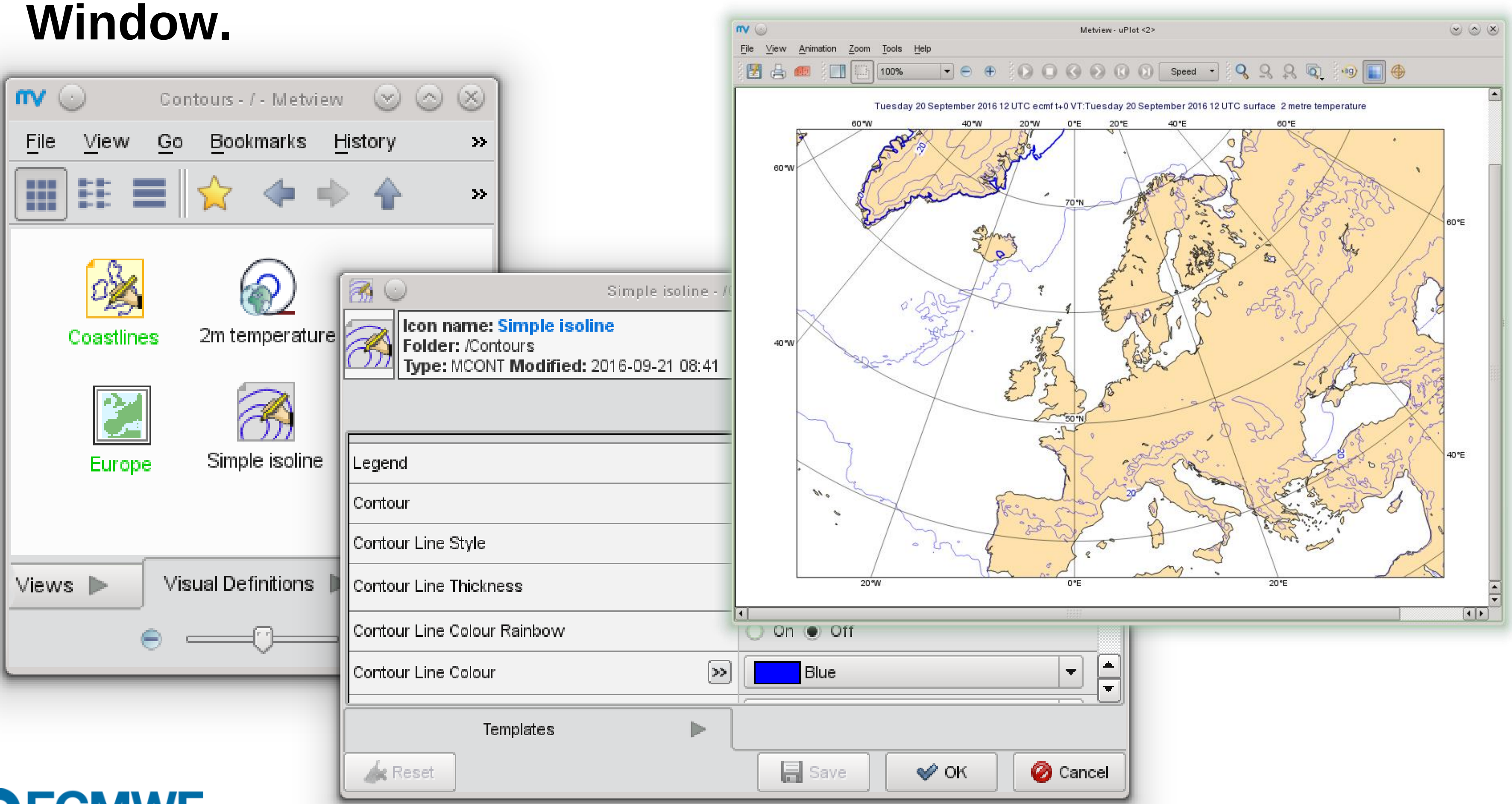

AN Par: msl Lev: 0 (sfc) Valid: 2012-09-21 06
AN Par: pv Lev: 320 (pt) Valid: 2012-09-21 06
1
2
4
10
20
30

Visual Definition (*visdef*)



What do you need to know about contouring?

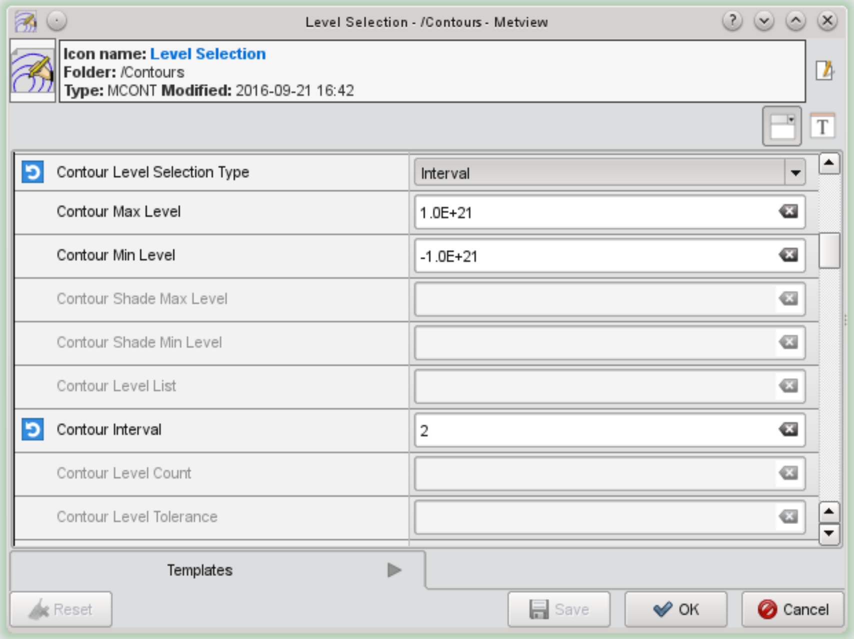
- **Visualise a data icon will use a default Contour icon**
- **To tailor your visualisation you need to create a Contour icon, select your option and drop it in the visualisation Window.**



Icon name: Simple isoline
Folder: /Contours
Type: MCONT **Modified:** 2016-09-21 08:41

Legend	
Contour	
Contour Line Style	
Contour Line Thickness	
Contour Line Colour Rainbow	On [✓ Off]
Contour Line Colour	Blue

How to select the levels you want to display?



Icon name: Level Selection
Folder: /Contours
Type: MCONT **Modified:** 2016-09-21 16:42

Contour Level Selection Type	Interval
Contour Max Level	1.0E+21
Contour Min Level	-1.0E+21
Contour Shade Max Level	
Contour Shade Min Level	
Contour Level List	
Contour Interval	2
Contour Level Count	
Contour Level Tolerance	

contour_level_selection_type

- count (default)

Choose the number of isolines you want by setting *contour_level_count*.

Magics will always try to pick up reasonable values.

- interval

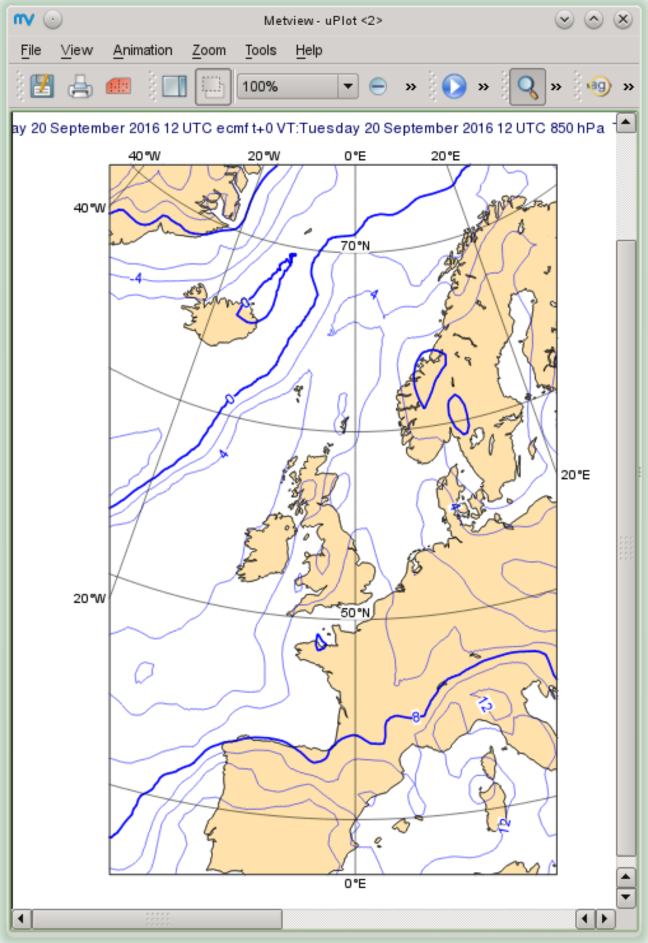
Choose the interval you want between 2 isolines by setting *contour_interval*.

contour_level_reference will be used as a base.

- level list

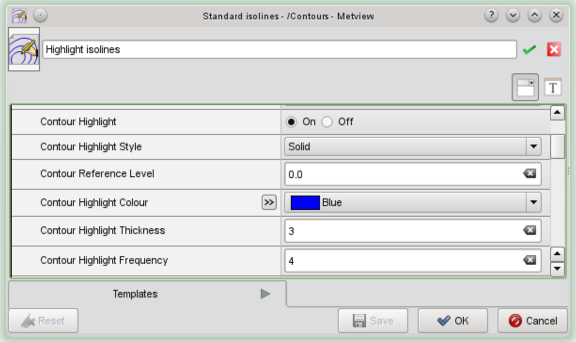
Explicitly define the list of levels you want by setting *contour_level_list*

How to style the isolines?


Metview - uPlot <2>
File View Animation Zoom Tools Help
100%
ay 20 September 2016 12 UTC ecmf t+0 VT:Tuesday 20 September 2016 12 UTC 850 hPa
40°W
20°W
0°E
20°E
70°N
50°N
-4
4
8
12

- A line in Magics has always 3 properties : colour/thickness/style
  - Colour : “red”, “RGB(0.,0.5,1.)
  - Style : solid, dash, dot …

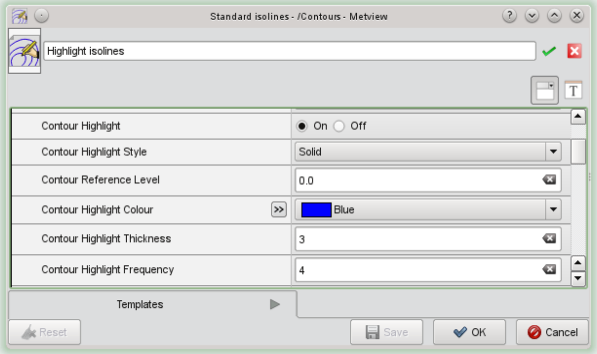
- Standard isolines



Highlight isolines

Contour Highlight	[✓ On] Off
Contour Highlight Style	Solid
Contour Reference Level	0.0
Contour Highlight Colour	🟦 Blue
Contour Highlight Thickness	3
Contour Highlight Frequency	4

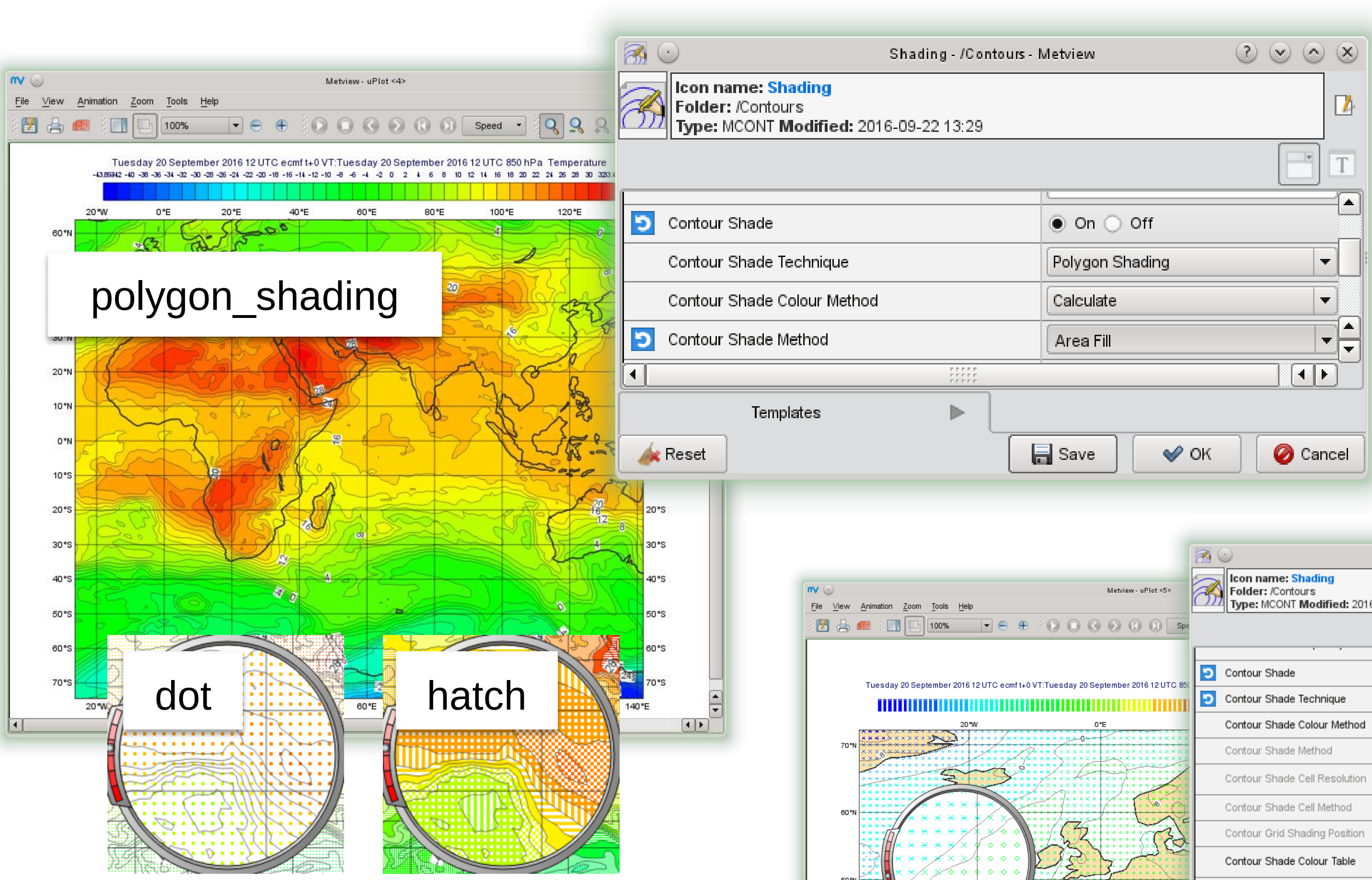
- Highlighted isolines



Highlight isolines

Contour Highlight	[✓ On] Off
Contour Highlight Style	Solid
Contour Reference Level	0.0
Contour Highlight Colour	🟦 Blue
Contour Highlight Thickness	3
Contour Highlight Frequency	4

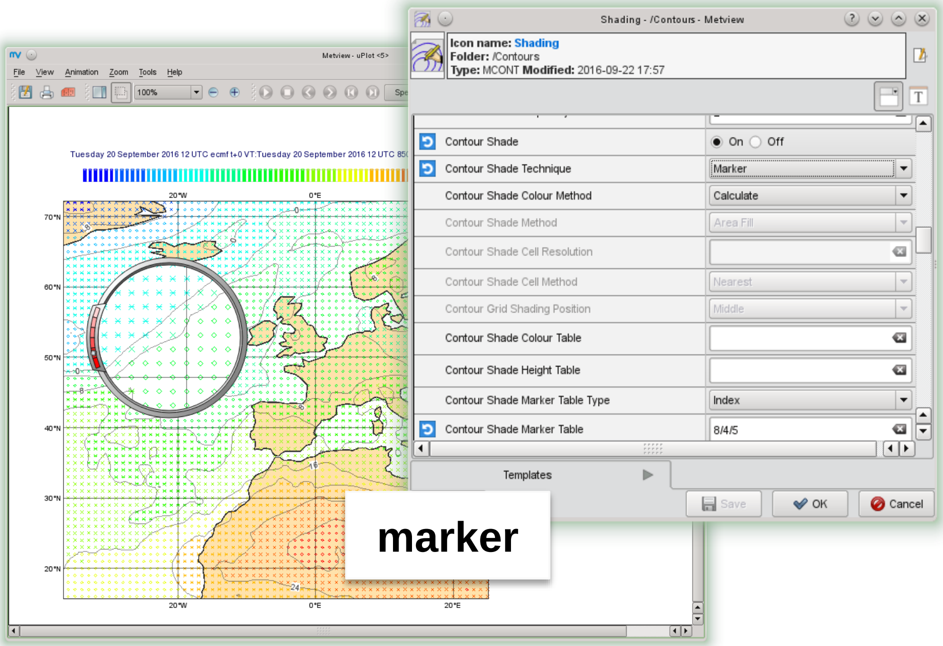
What are the shading options?


Metview - uPlot <4>
File View Animation Zoom Tools Help
Tuesday 20 September 2016 12 UTC ecmf t+0 VT:Tuesday 20 September 2016 12 UTC 850 hPa Temperature
polygon_shading
dot
hatch
Templates
Reset
Save
OK
Cancel

Shading - /Contours - Metview

Icon name: Shading
Folder: /Contours
Type: MCONT **Modified:** 2016-09-22 13:29

Contour Shade	[✓ On] Off
Contour Shade Technique	Polygon Shading
Contour Shade Colour Method	Calculate
Contour Shade Method	Area Fill

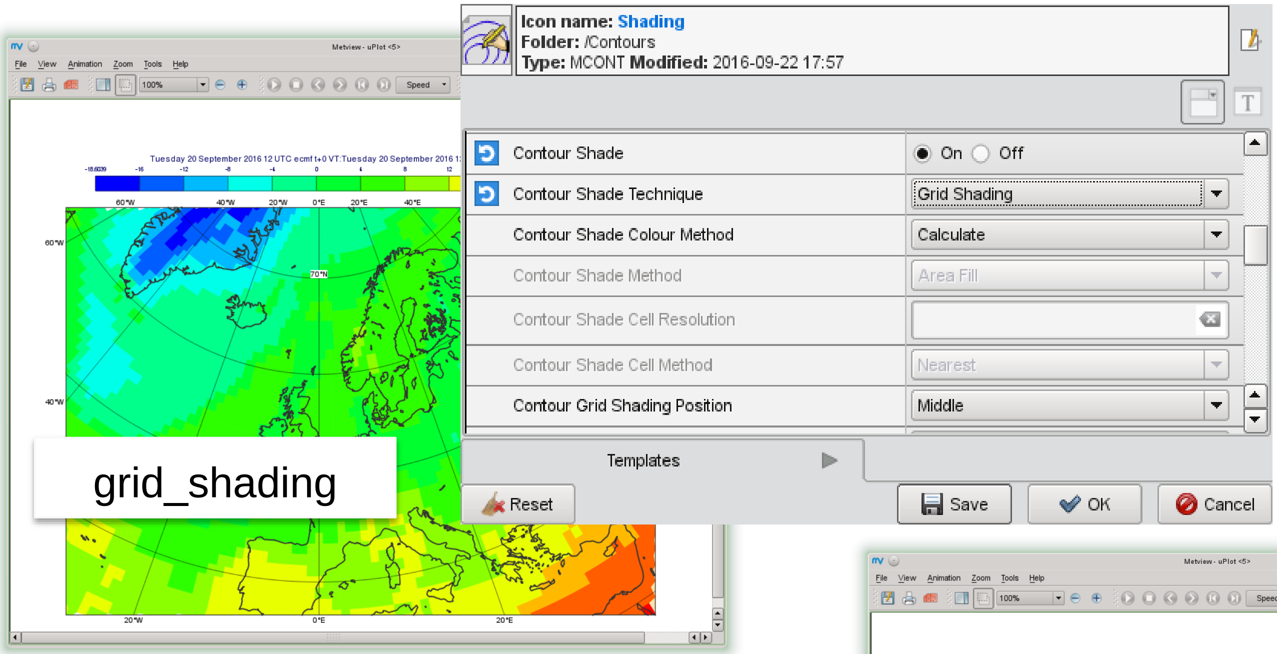

Metview - uPlot <5>
File View Animation Zoom Tools Help
Tuesday 20 September 2016 12 UTC ecmf t+0 VT:Tuesday 20 September 2016 12 UTC 850
marker
Templates
Save
OK
Cancel

Shading - /Contours - Metview

Icon name: Shading
Folder: /Contours
Type: MCONT **Modified:** 2016-09-22 17:57

Contour Shade	[✓ On] Off
Contour Shade Technique	Marker
Contour Shade Colour Method	Calculate
Contour Shade Method	Area Fill
Contour Shade Cell Resolution	
Contour Shade Cell Method	Nearest
Contour Grid Shading Position	Middle
Contour Shade Colour Table	
Contour Shade Height Table	
Contour Shade Marker Table Type	Index
Contour Shade Marker Table	8/4/5

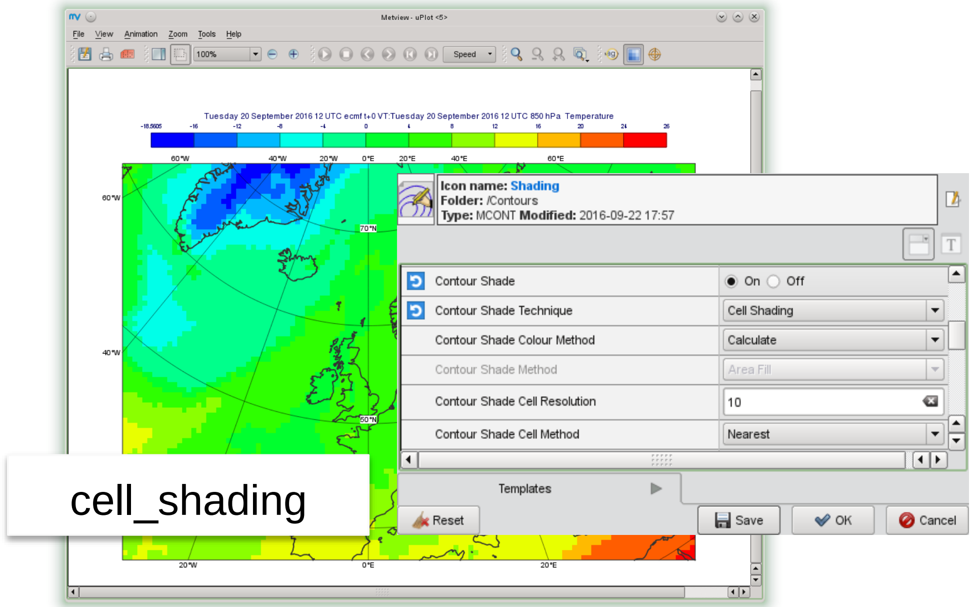
What are the shading options?



grid_shading

Icon name: Shading
Folder: /Contours
Type: MCONT Modified: 2016-09-22 17:57

Contour Shade	[✓ On] Off
Contour Shade Technique	Grid Shading
Contour Shade Colour Method	Calculate
Contour Shade Method	Area Fill
Contour Shade Cell Resolution	
Contour Shade Cell Method	Nearest
Contour Grid Shading Position	Middle

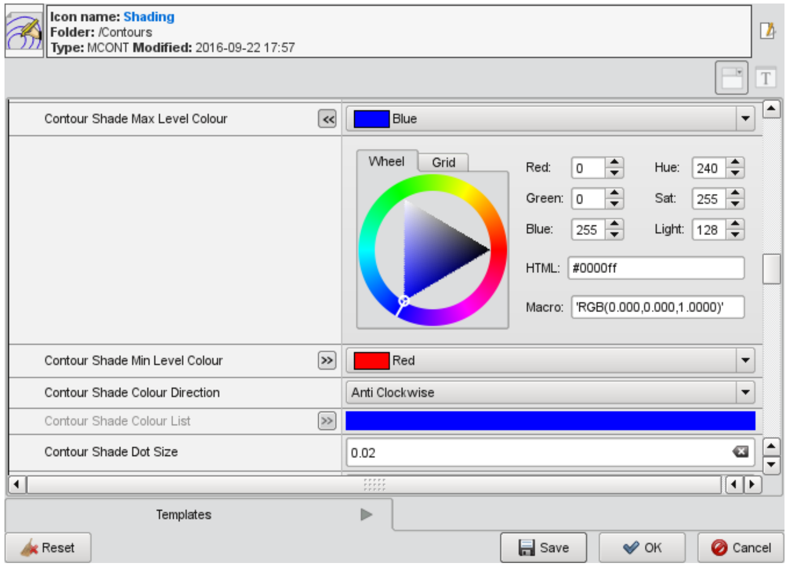


cell_shading

Icon name: Shading
Folder: /Contours
Type: MCONT Modified: 2016-09-22 17:57

Contour Shade	[✓ On] Off
Contour Shade Technique	Cell Shading
Contour Shade Colour Method	Calculate
Contour Shade Method	Area Fill
Contour Shade Cell Resolution	10
Contour Shade Cell Method	Nearest

How to define the colour map to use ?


Icon name: Shading
Folder: /Contours
Type: MCONT Modified: 2016-09-22 17:57

Contour Shade Max Level Colour: Blue

Red: 0 Hue: 240
Green: 0 Sat: 255
Blue: 255 Light: 128
HTML: #0000ff
Macro: 'RGB(0.000,0.000,1.0000)'

Contour Shade Min Level Colour: Red
Contour Shade Colour Direction: Anti Clockwise
Contour Shade Colour List
Contour Shade Dot Size: 0.02

contour_shade_colour_method

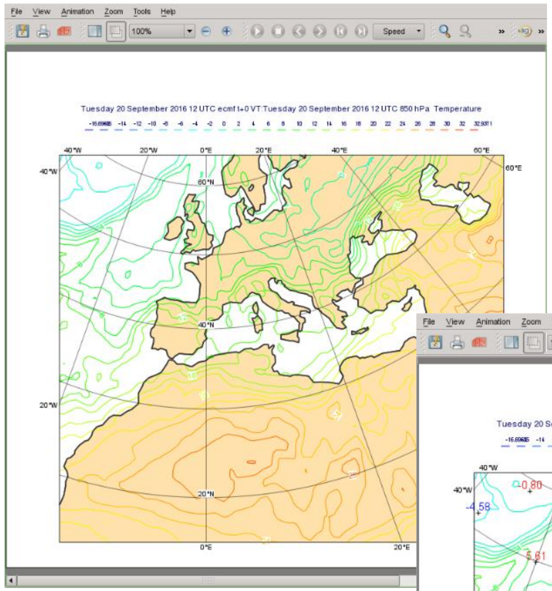
- list

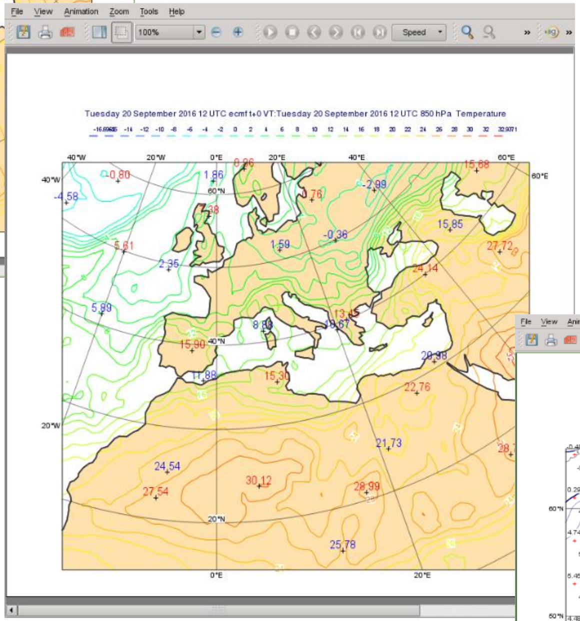
Explicitly define the list of colours you want by setting *contour_shade_colour_list.*

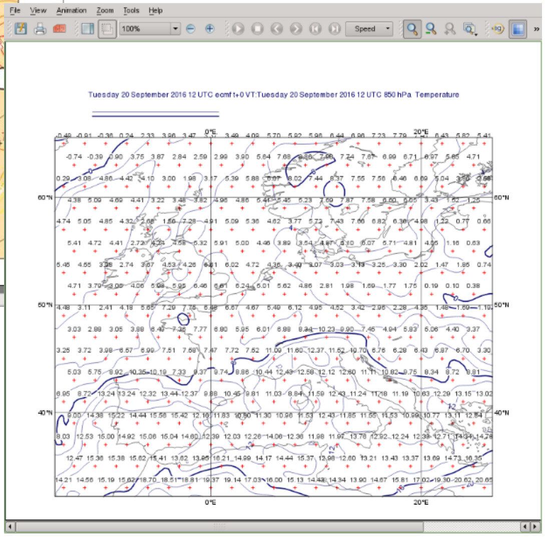
- calculate

Magics will calculate a list of colours from *contour_shade_min_level_colour* to *contour_shade_max_level_colour*. *contour_shade_colour_direction* will inform Magics on the direction you want to use the HSL wheel.

Other nice features…


File View Animation Zoom Tools Help
100%
Speed
Tuesday 20 September 2016 12 UTC ecmf t+0 VT:Tuesday 20 September 2016 12 UTC 850 hPa Temperature


File View Animation Zoom Tools Help
100%
Speed
Tuesday 20 September 2016 12 UTC ecmf t+0 VT:Tuesday 20 September 2016 12 UTC 850 hPa Temperature


File View Animation Zoom Tools Help
100%
Speed
Tuesday 20 September 2016 12 UTC ecmf t+0 VT:Tuesday 20 September 2016 12 UTC 850 hPa Temperature

Additional Notes

- **Contouring often has automatic unit conversion – can be deactivated in the *Contour* icon**
- **Cursor data – shows both scaled and non-scaled values**

Metview Tutorial: Customising your plot

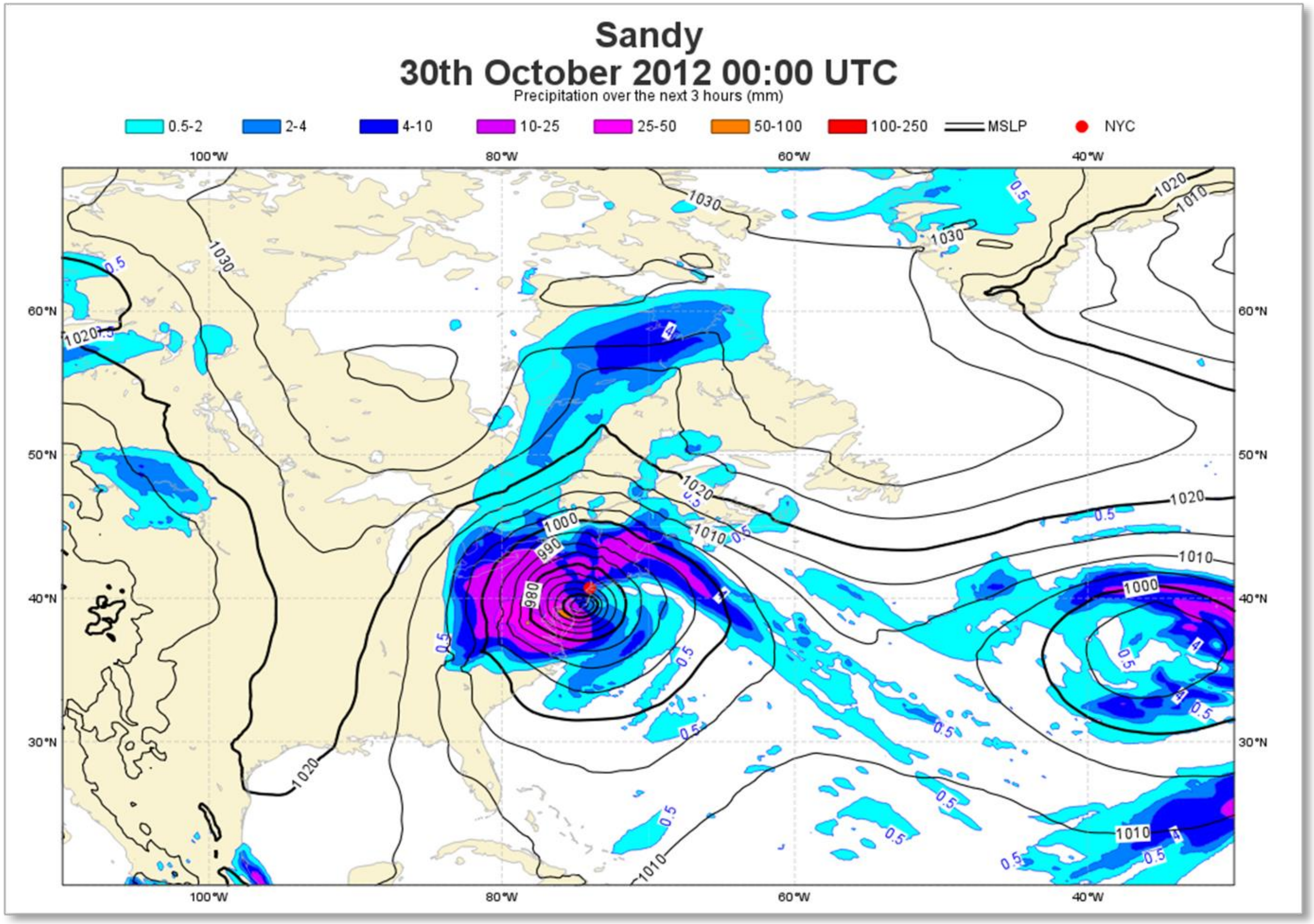
- **Please do “Customising your plot” in the provided sub-folder “customising your plot”**

Case Study: Contouring Hurricane Sandy

Iain Russell

Software Applications Team

Metview Tutorial: Case Study – Contouring Sandy



- **Please do exercise "Case Study: Plotting Hurricane Sandy on a Map" in the provided sub-folder "contouring sandy"**

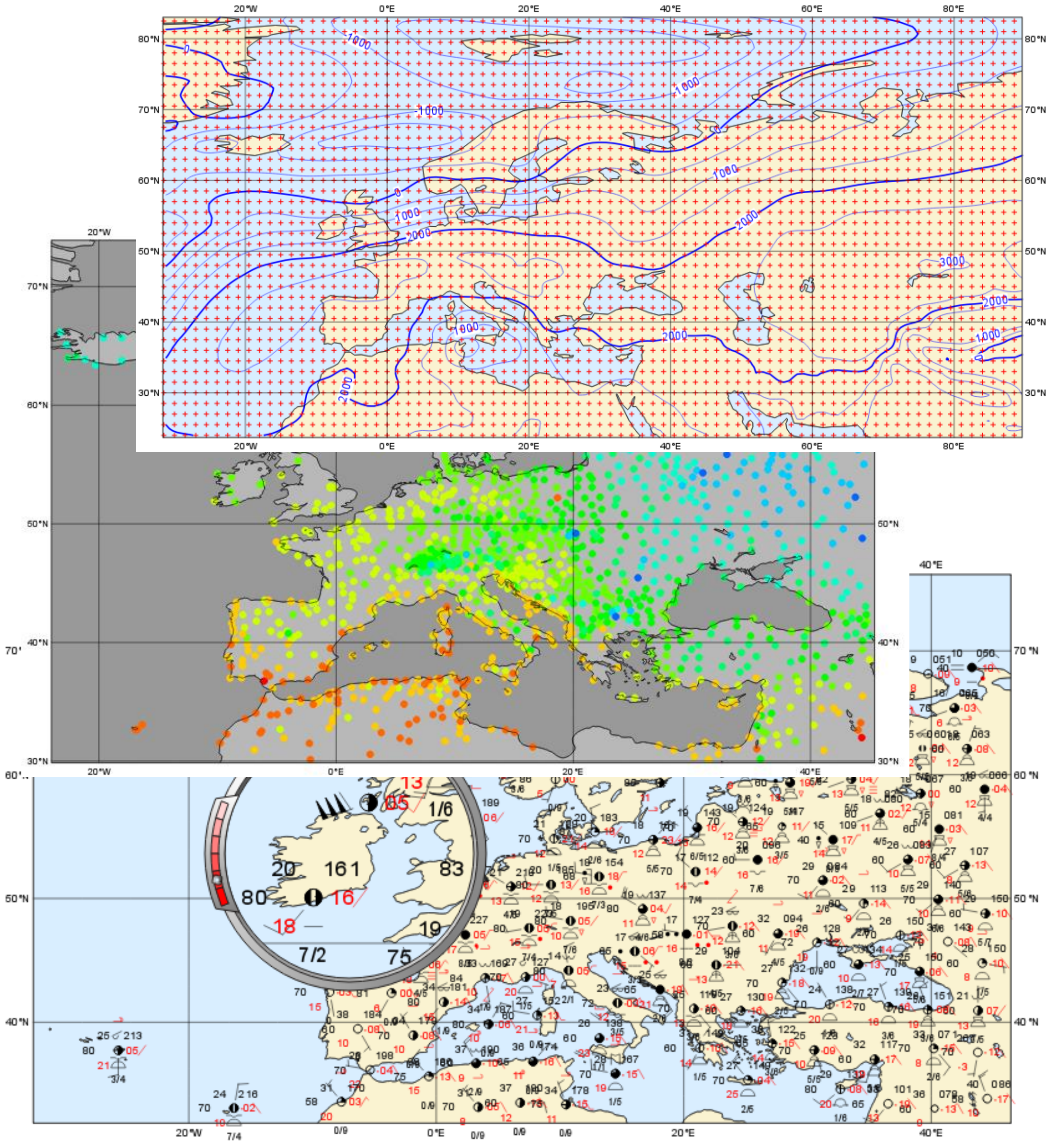
Data in Metview – Part 1

Sándor Kertész

Software Applications Team

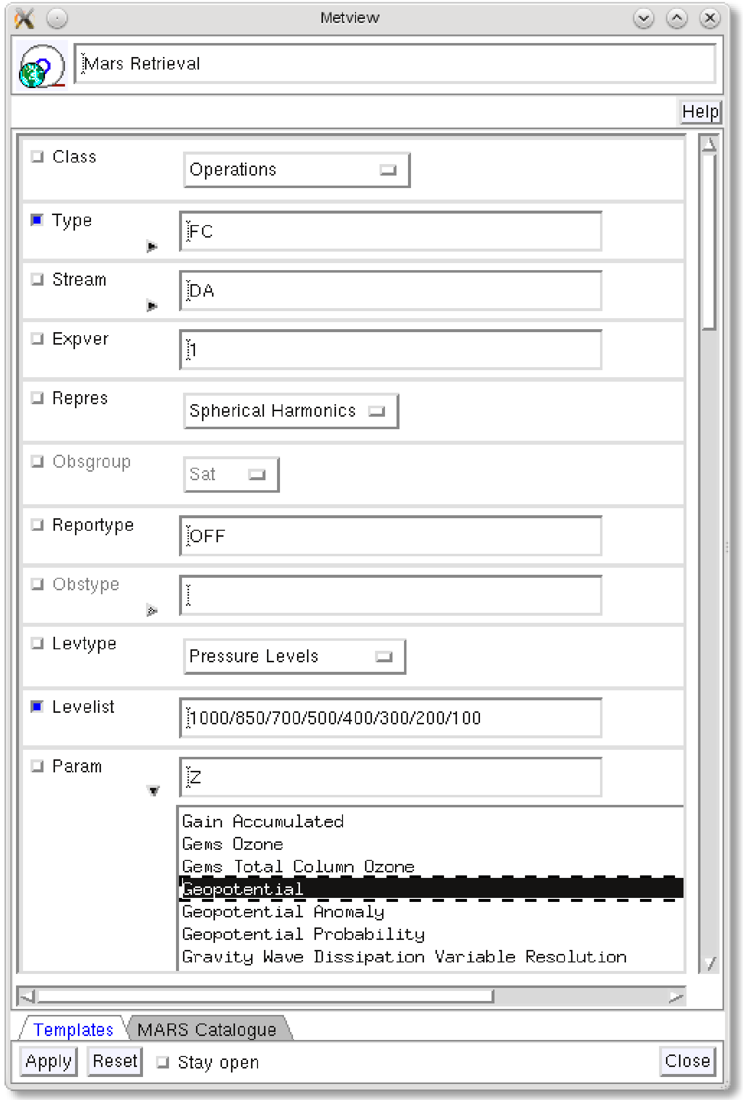
Data Types – Part 1

- **GRIB: gridded geographical data**
- **Geopoints: scattered geographical data**
- **BUFR: scattered observation data**



Metview and MARS

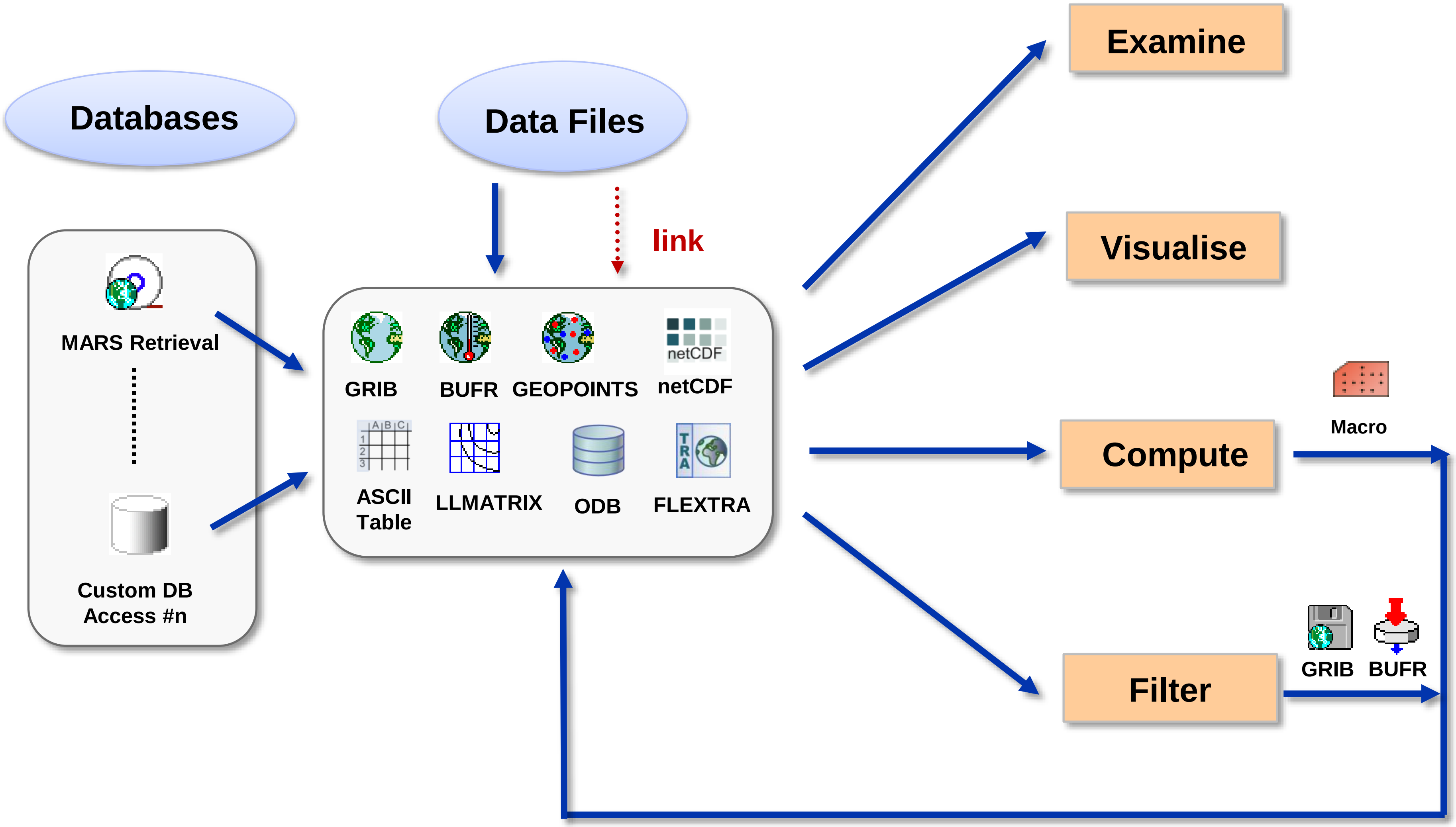
- **Metview incorporates a MARS client module**
  - **Built from same source code**
  - **All processing options are available**
  - **Direct access to local MARS archive, or through the Web API for external access**
- **All MARS parameters can be accessed**
- **Metview caches retrieved data**
- **Metview can examine, visualise and process any data formats in MARS**



Class	Operations
■ Type	FC
Stream	DA
Expver	1
Repres	Spherical Harmonics
Obsgroup	Sat
Reportype	OFF
Obstype	
Levtype	Pressure Levels
■ Levelist	1000/850/700/500/400/300/200/100
Param	Z

```
Gain Accumulated
Gems Ozone
Gems Total Column Ozone
Geopotential
Geopotential Anomaly
Geopotential Probability
Gravity Wave Dissipation Variable Resolution
```

Data handling in Metview



Metview Tutorial: Data Part 1

Get the data and icons for the day

- **From a command line type:**

```
~trx/mv_data/get_day_2.sh
```

- **A new folder called “day_2” will appear in your “training” folder**

- **Please do exercise “Data Part 1” in the provided sub-folder “data 1”**

Additional Notes (1)

- **What data is stored in MARS?**
  - **WebMars catalogue: http://www.ecmwf.int/en/forecasts/datasets**
- **MARS language syntax**
  - **List of values: 0/12/24/36/48**
  - **Range of values: 0/TO/48/BY/12**
- **MARS date format**
  - **Specific dates, e.g. 20090303**
  - **Relative dates, e.g. –1 (yesterday)**

Processing data in Metview

Iain Russell

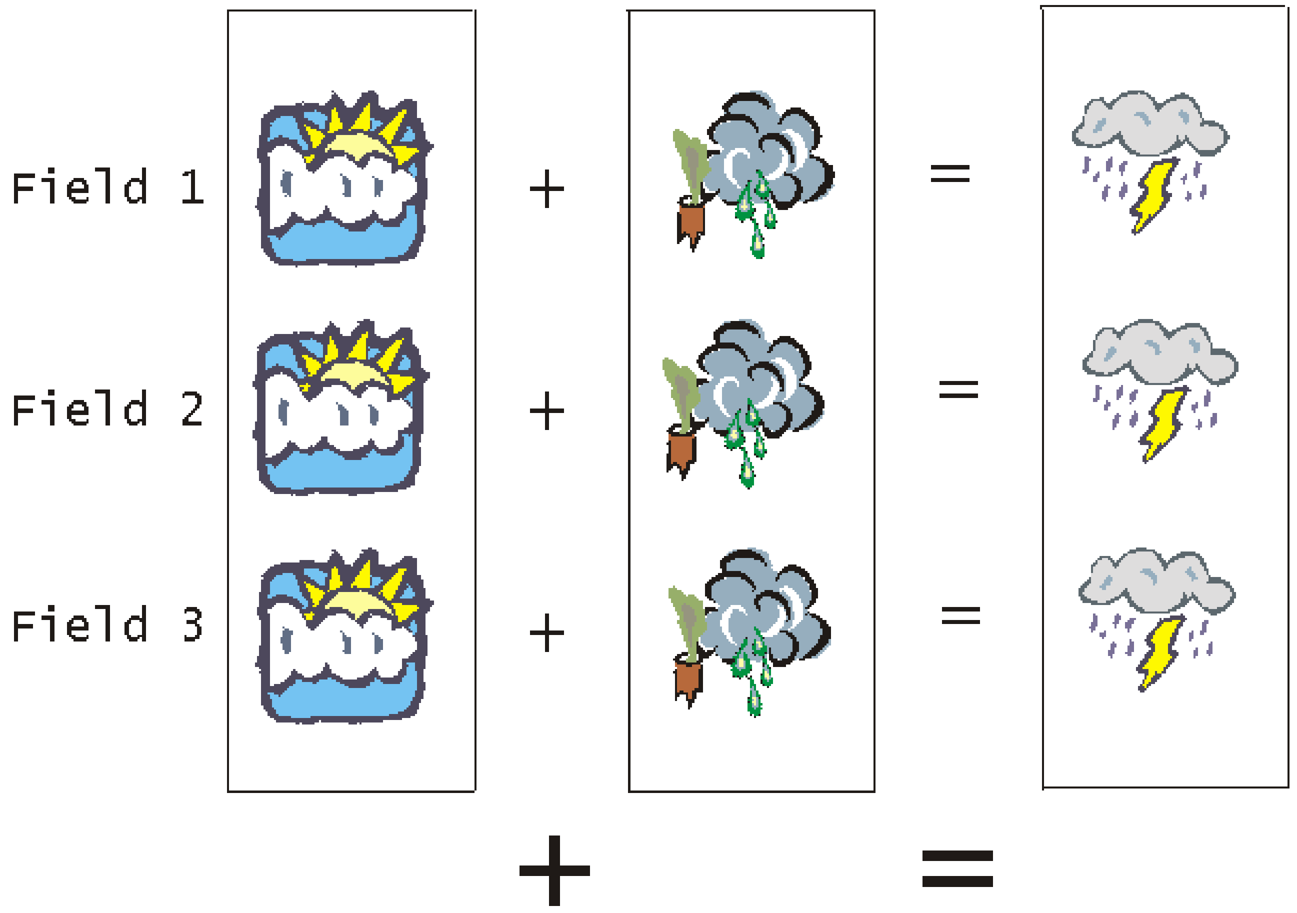
Software Applications Team

Processing Data - Fieldsets

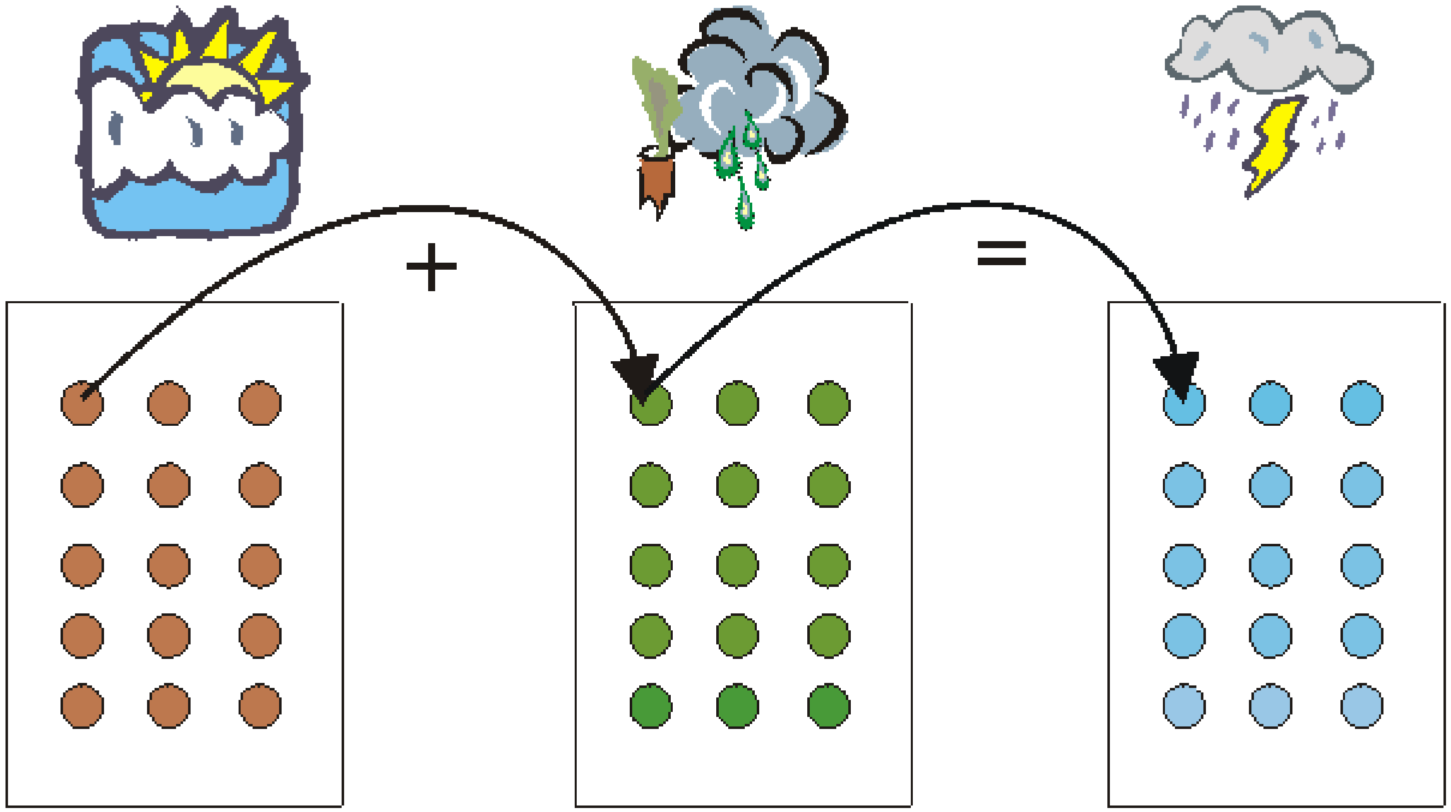
- **Definition**
  - **Entity composed of several meteorological fields, (e.g. output of a MARS retrieval).**
- **Read from and written to GRIB files**
- **Operations and functions on fieldsets**
  - **Operations on two fieldsets are carried out between each pair of corresponding values within each pair of corresponding fields. The result is a new fieldset.**

```
result = fieldset_1 + fieldset_2
```

Processing Data - Fieldsets



Processing Data - Fieldsets

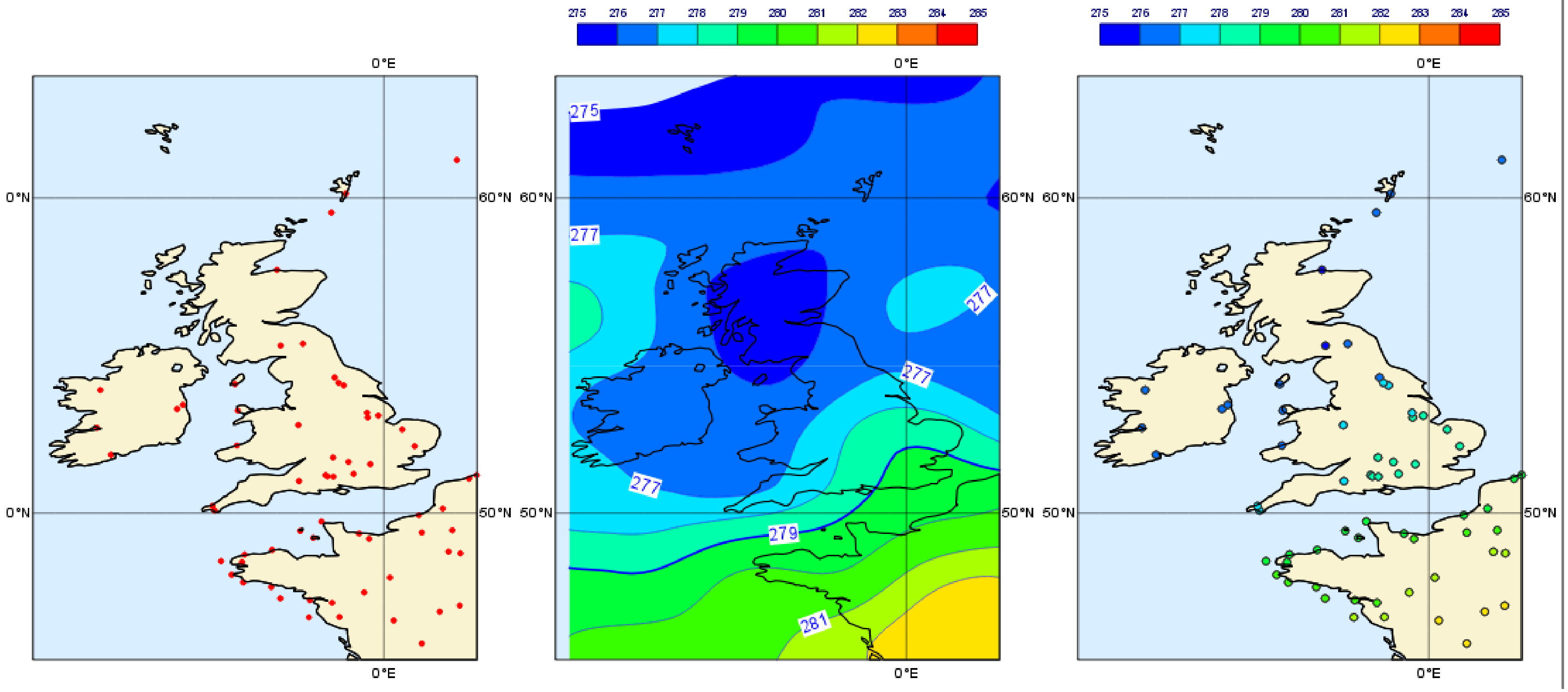


Processing Data - Fieldsets

- **Lots more functionality available for fieldsets, for example:**
  - **Interpolation between grids**
  - **Conversion between fields and scattered points**
  - **Extraction of sub-areas**
  - **Computations**
  - **Filtering**

Processing Data – Fieldsets and Geopoints

- **Operations between fields and geopoints**
  - **First, the field values are interpolated onto the geopoints locations**
  - **Then computations are done in ‘geopoints space’**



Metview Tutorial: Processing Data

- **Please do “Processing Data” in the provided sub-folder “processing data”**

Additional Notes

- **Extracting fields from fieldsets**
  - **fieldset [number]**
  - **fieldset [number,number]**
  - **fieldset [number,number,number]**
- **Examples :**

```
y = x[2]       # copies field 2 of x into y
y = x[3,8]     # copies fields 3,4,5,6,7 and 8
y = x[1,20,4]  # copies fields 1, 5, 9, 13 and 17
```

- **Concatenating fields :**

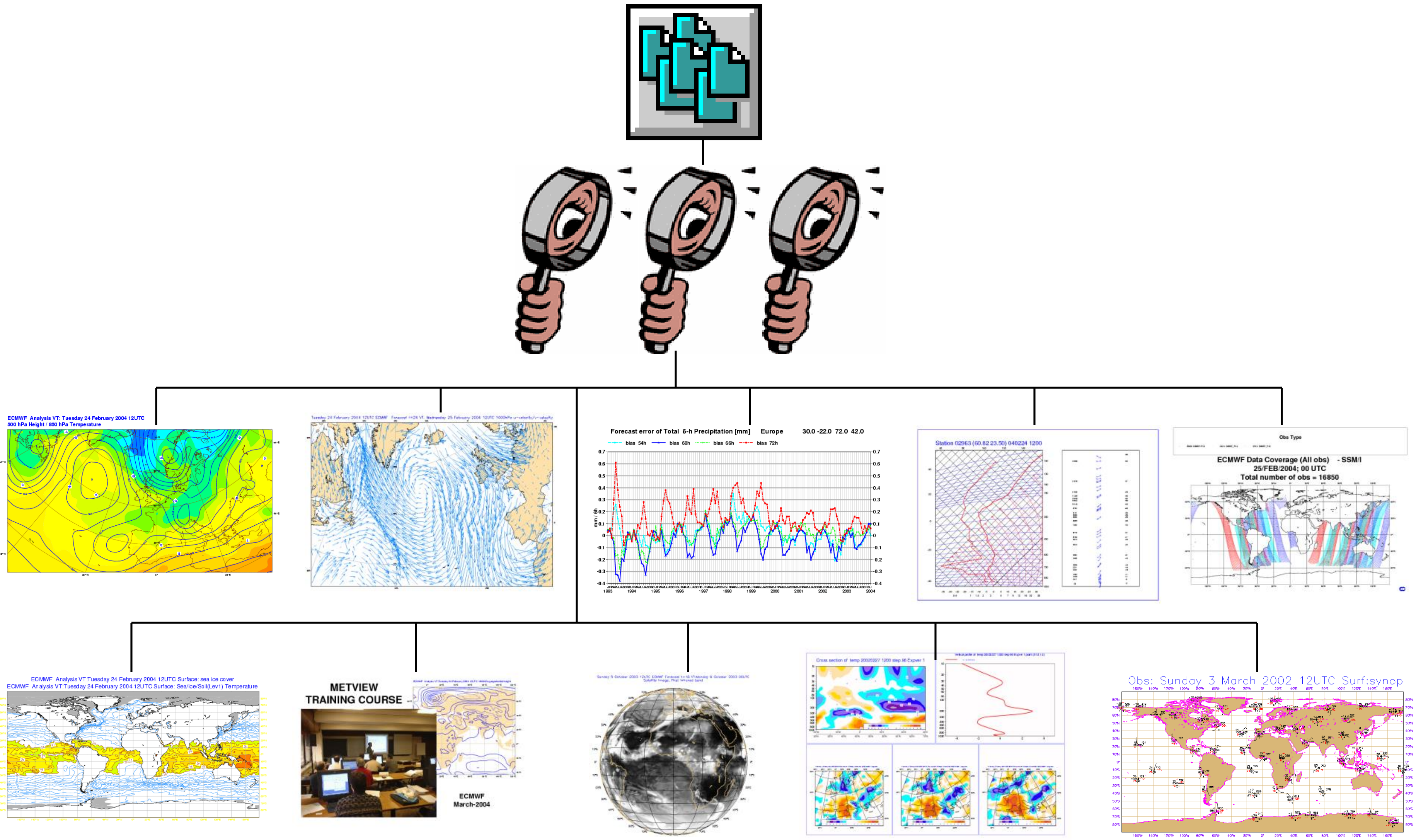
```
a = fs1 & fs2 & fs3
```

Views and layout

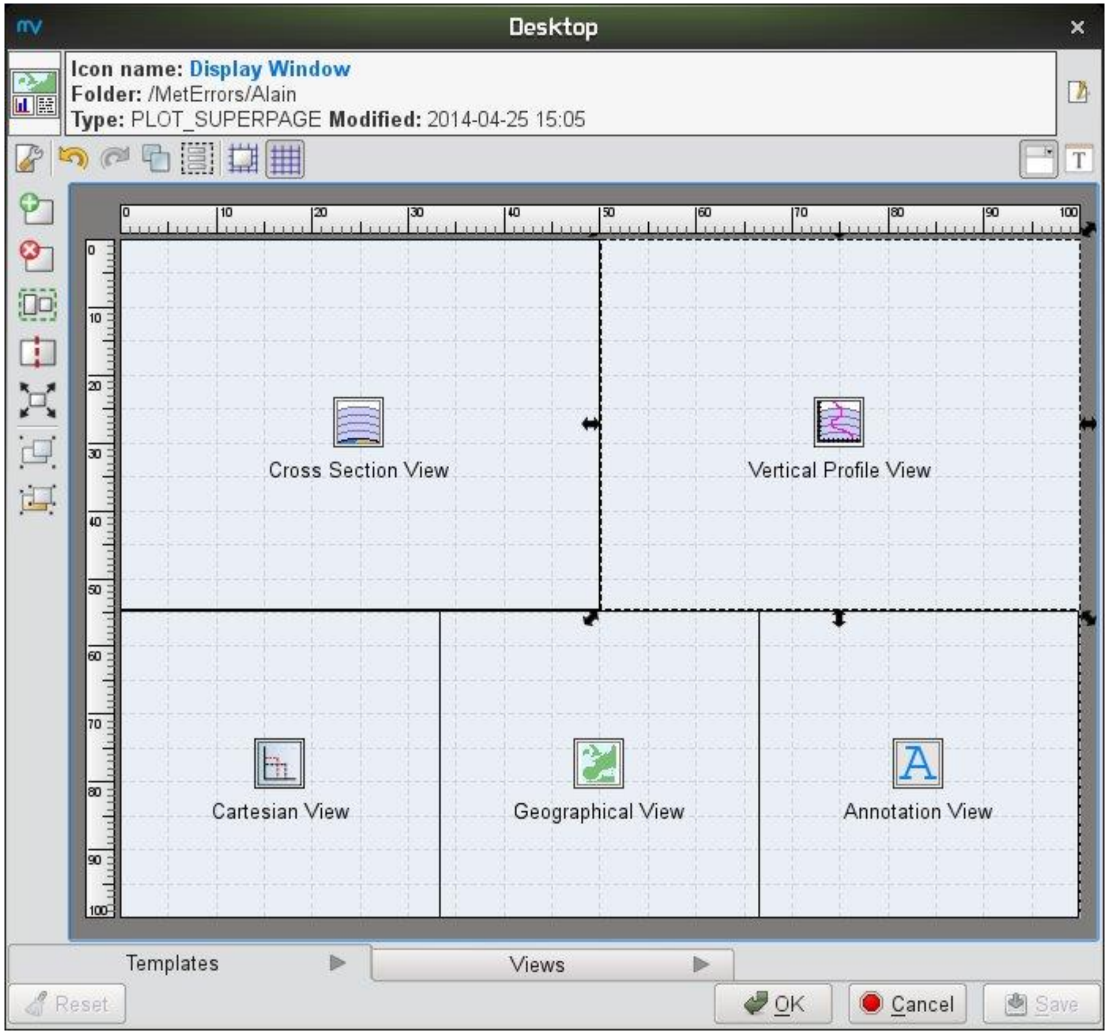
Fernando Ii

Software Applications Team

The VIEW concept


ECMWF Analysis VT: Tuesday 24 February 2004 12UTC
500 hPa Height / 850 hPa Temperature
Forecast error of Total 6-h Precipitation [mm] Europe 30.0 -22.0 72.0 42.0
bias 54h
bias 60h
bias 66h
bias 72h
Obs Type
ECMWF Data Coverage (All obs) - SSM/I
25/FEB/2004; 00 UTC
Total number of obs = 16850
ECMWF Analysis VT:Tuesday 24 February 2004 12UTC Surface: sea ice cover
ECMWF Analysis VT:Tuesday 24 February 2004 12UTC Surface: Sea/Ice/Soil(Lev1) Temperature
METVIEW
TRAINING COURSE
ECMWF
March-2004
Obs: Sunday 3 March 2002 12UTC Surf:synop

Display Window icon – layout editor


Desktop

Icon name: Display Window
Folder: /MetErrors/Alain
Type: PLOT_SUPERPAGE **Modified:** 2014-04-25 15:05

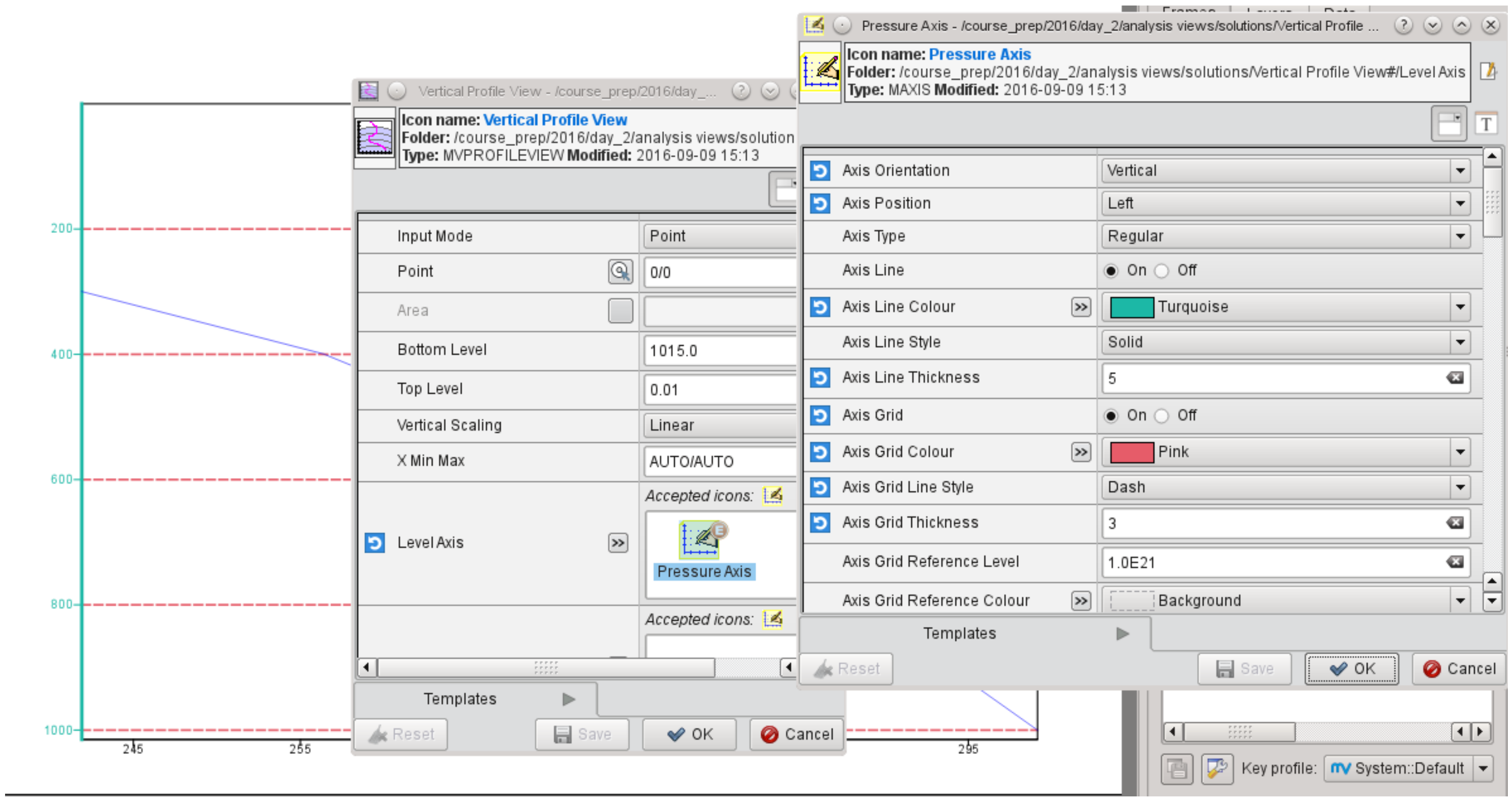
Metview Tutorial: Views and Layout

- **Please do exercise “Analysis Views” in the provided sub-folder “analysis views”**

- **Please do exercise “Layout in Metview”, also in the provided sub-folder “analysis views”**

Part 4 – Additional Notes

- **Many options are common to all views (position, …)**
- **Axis Plotting icons can be used to modify the look of the axes (e.g. fonts, titles, colours, etc)**



Vertical Profile View - /course_prep/2016/day_...

Icon name: Vertical Profile View
Folder: /course_prep/2016/day_2/analysis views/solution
Type: MVPROFILEVIEW **Modified:** 2016-09-09 15:13

Input Mode	Point
Point	0/0
Area	
Bottom Level	1015.0
Top Level	0.01
Vertical Scaling	Linear
X Min Max	AUTO/AUTO
Level Axis	*Accepted icons:* Pressure Axis
	Accepted icons:

Pressure Axis - /course_prep/2016/day_2/analysis views/solutions/Vertical Profile ...

Icon name: Pressure Axis
Folder: /course_prep/2016/day_2/analysis views/solutions/Vertical Profile View#/Level Axis
Type: MAXIS **Modified:** 2016-09-09 15:13

Axis Orientation	Vertical
Axis Position	Left
Axis Type	Regular
Axis Line	[✓ On] Off
Axis Line Colour	🟩 Turquoise
Axis Line Style	Solid
Axis Line Thickness	5
Axis Grid	[✓ On] Off
Axis Grid Colour	🟥 Pink
Axis Grid Line Style	Dash
Axis Grid Thickness	3
Axis Grid Reference Level	1.0E21
Axis Grid Reference Colour	Background

Part 4 – Additional Notes (2)

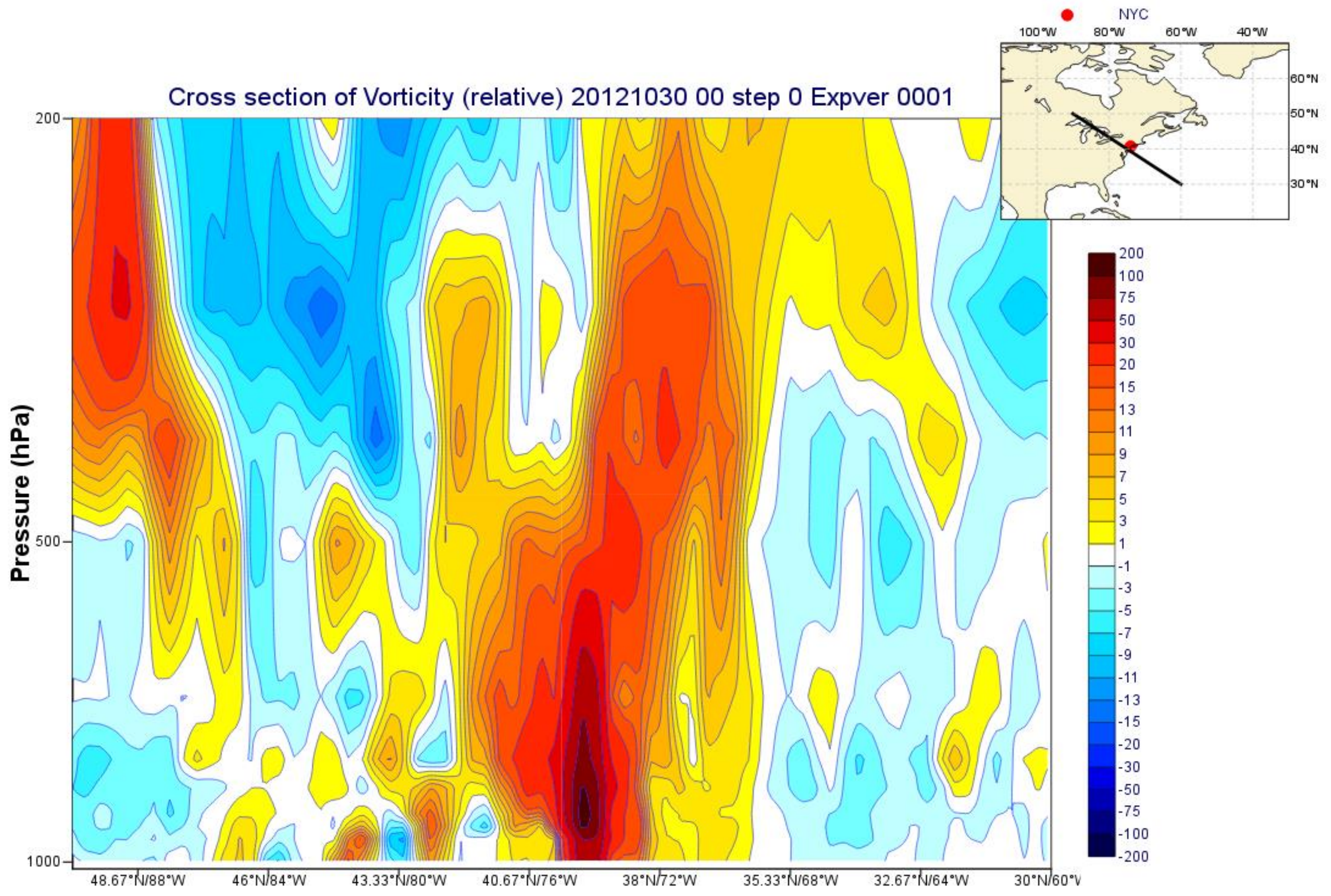
- **Views vs Data icons**
- **If you just want to plot, e.g. a Cross Section, use the Cross Section View icon**
- **If you want to store the actual computed cross section data (which is in netCDF format), use the Cross Section Data icon**
  - **Can write the file, or perform further computations and then plot**
- **Same goes for Average, Vertical Profile and Hovmoeller**

Case study: Cross section of Hurricane Sandy

Iain Russell

Software Applications Team

Metview Tutorial: Case Study – Cross Section of Sandy



- **Please do "Case Study: Cross Section of Sandy" in the provided sub-folder "sandy cross section"**

Data in Metview – Part 2

Iain Russell

Software Applications Team

More Data Formats

- **NetCDF**
  - **Multi-dimensional arrays (matrices, lines, points)**
- **ASCII tables**
  - **E.g. CSV – columns of values**
- **Other ASCII**
- **ODB (Observation Database, ECMWF)**

Visualisers

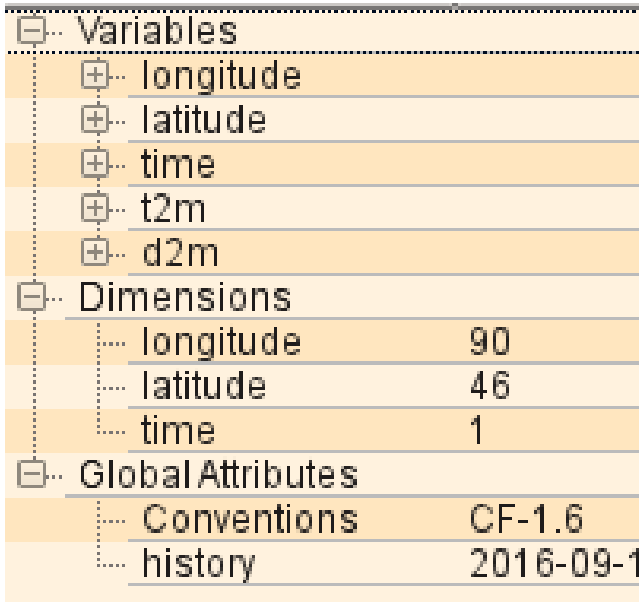
- GRIB is 'easy' to plot
  - Standardised meta-data – geographic coordinates, resolution, etc
- Some other formats (e.g. netCDF) are more versatile and can contain matrices, scattered points, multiple variables, etc
  - users need to tell Metview what to plot

→ visualiser icons

Handling NetCDF Data (1)

- **Macro computations are performed on the *current* variable**

```
a = read('file.nc')
setcurrent(a, 't2m')
b = a - 5
```

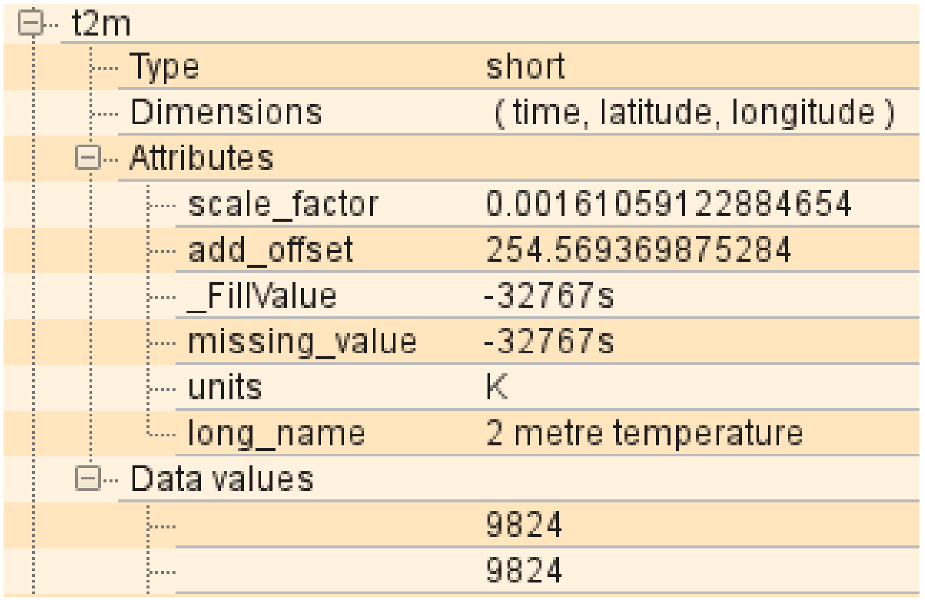


Variables	
longitude	
latitude	
time	
t2m	
d2m	
Dimensions	
longitude	90
latitude	46
time	1
Global Attributes	
Conventions	CF-1.6
history	2016-09-1

- **- result is a netCDF based on file.nc, but with the values for t2m reduced by 5**

Handling NetCDF Data (2)

- **Can extract the values for the current variable using the `values()` function**
- **Will apply `scale_factor` and `add_offset` if present**
- **Missing values will become `vector_missing_value` and will be ignored in computations**
- **Times will be translated into `date` type variables**
- **These behaviours are configurable**



t2m	
Type	short
Dimensions	(time, latitude, longitude)
Attributes	
scale_factor	0.00161059122884654
add_offset	254.569369875284
_FillValue	-32767s
missing_value	-32767s
units	K
long_name	2 metre temperature
Data values	
	9824
	9824

These behaviours are new to Metview 5

Metview Tutorial: Data Part 2

Get the data and icons for the day

- **From a command line type:**

```
~trx/mv_data/get_day_3.sh
```

- **A new folder called “day_3” will appear in your “training” folder**

- **Please do exercise “Data Part 2” in the provided sub-folder “data 2”**

Additional Notes

- **Note the different plot types available in the Visualiser icons**
  - **Allow a range of ways to interpret and plot data, e.g. geographic, x/y, matrices, vector pairs, …**
  - **Choose the plot type before entering other parameters!**

Handling time in Metview

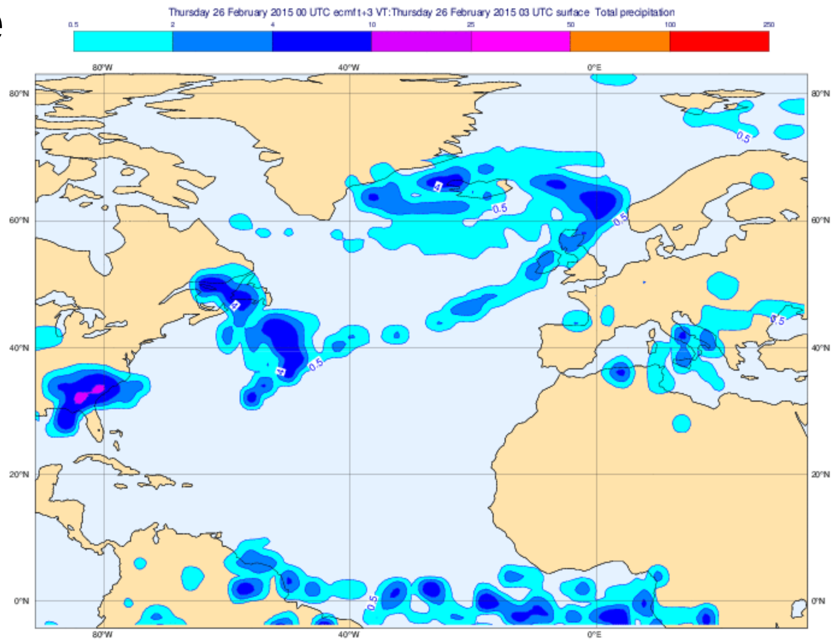
Iain Russell

Software Applications Team

Handling Time in Metview

- Note that time itself is multi-dimensional!
  - Run time (base time) and forecast step
  - Run time + step = valid time
  - Also analysis data which has step=0

- Can extract time from most data types – in Macro , we have the *date* variable type


Thursday 26 February 2015 00 UTC ecmf t+3 VT:Thursday 26 February 2015 03 UTC surface Total precipitation
0.5
2
4
10
25
50
100
250
80°W
40°W
0°E
80°N
60°N
40°N
20°N
0°N
0.5
4

Handling Time in Metview

- **Dates defined as a built-in type - year, month, day, hour, minute and second.**
- **Dates can be created as literals using :**
  - **yyyy-mm-dd**
  - **yyyy-DDD**
  - **where : yr, yyyy - 4 digit yr, mm - 2 digit month, dd - 2 digit day, DDD - 3 digit Julian day.**
- **The time can be added using :**
  - **HH:MM or HH:MM:SS**
  - **E.g.**

```
start_date = 2003-03-20 12:01
```

Handling Time in Metview

- **Date arithmetic works with ‘1' being a day**

```
d1 = 2015-12-31
d2 = d1 + 1
print (d2)
   2016-01-01 00:00:00
```

Handling Time in Metview

- **Function `date()` creates dates from numbers:**

```
d1        = date(20080129)

today     = date(0)

yesterday = date(-1)
```

- **Hour, minute and second components are zero.**
- **To create a full date, use decimal dates:**

```
d = date(20080129.5)
```

or

```
d = 2008-01-29 + 0.5
```

or

```
d = 2008-01-29 + hour(12)
```

Handling Time in Metview

- **Note that numbers passed to Metview modules are automatically converted to dates:**

```
r = retrieve(date : -1, ...)

r = retrieve(date : 20070101, ...)
```

Handling Time in Metview

- **Loops on dates using a for loop:**

```
 for d = 2007-01-01 to 2007-03-01 do
      ... # each step is 1 day
 end for

for d = 2007-01-01 to 2007-03-01 by 2 do
      ... # each step is 2 days
end for

for d = 2007-01-01 to 2007-03-01 by hour(6) do
      print(d)
       ... # each step is 6 hours
end for
```

Data Overlay

- Multi-data visualisations, e.g. T+Z,…
  - When are different data overlaid in the same plot?
- Default data overlay rules
- Need more control? – Use the Data Overlay Setting

Metview Tutorial: Handling Time

- **Please do exercise “Handling Time in Metview” in the provided sub-folder “time”**

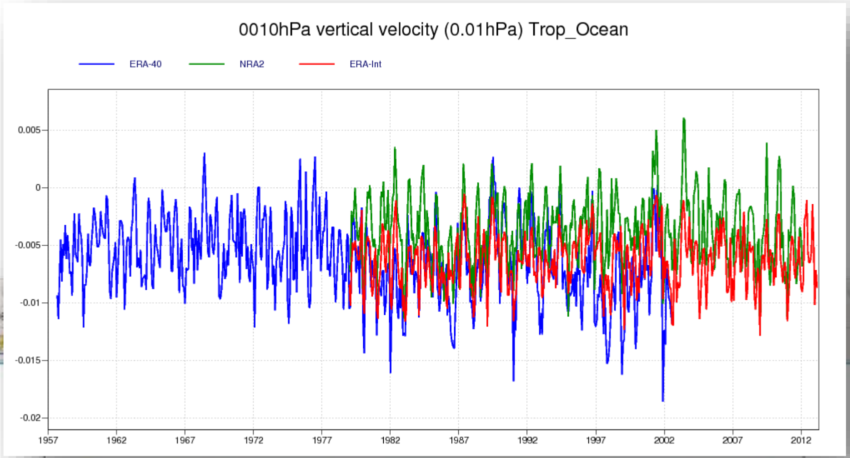
Graph Plotting

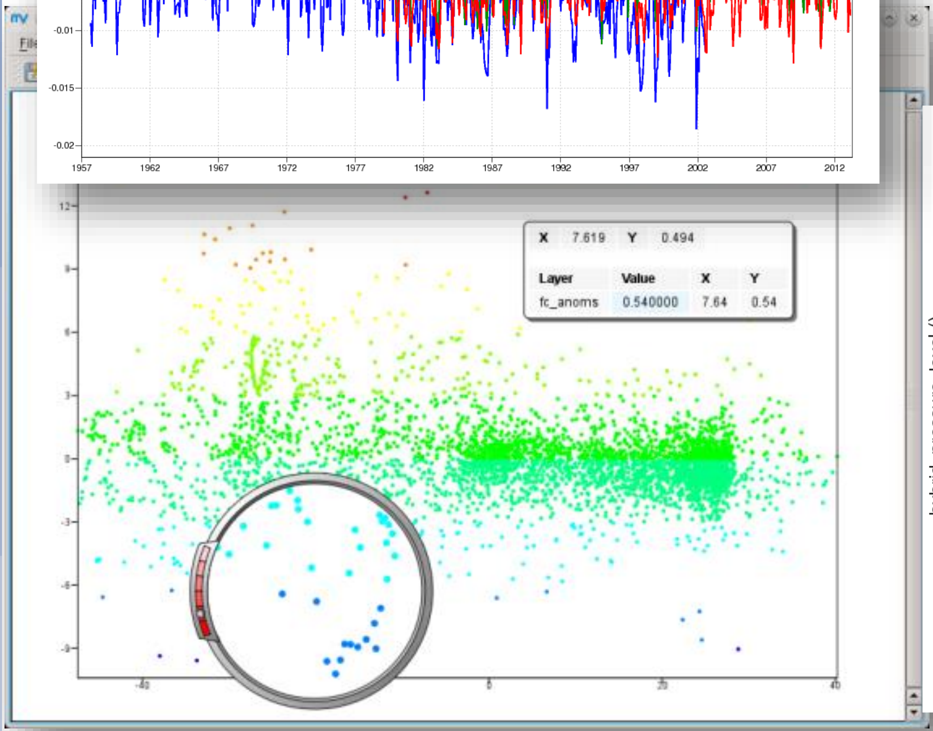
Iain Russell

Software Applications Team

Graph Plotting

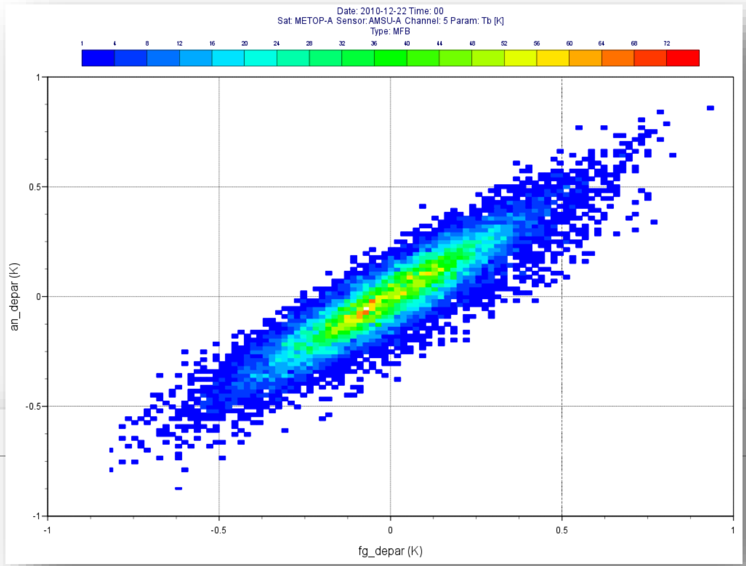
- **It’s not only geographical plots!**

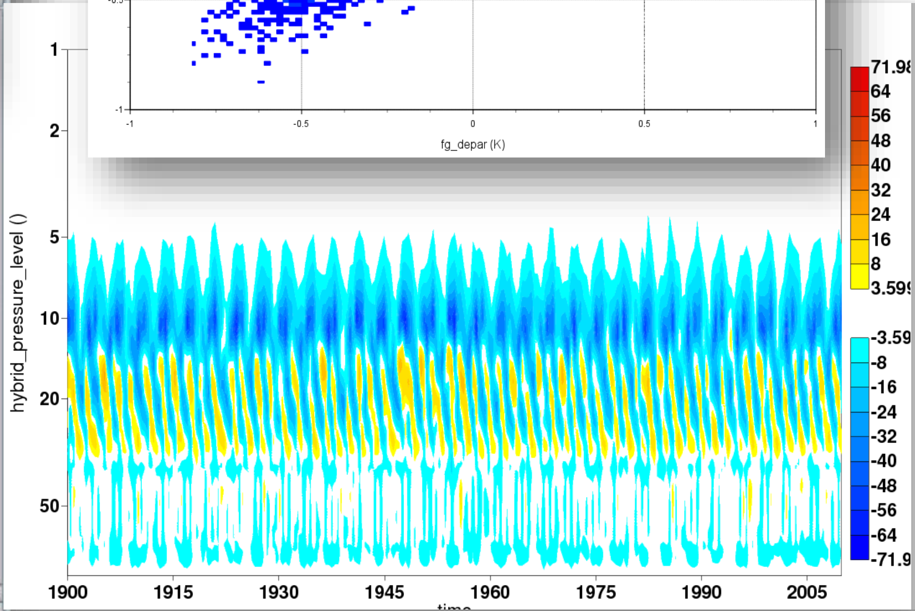




X 7.619 Y 0.494

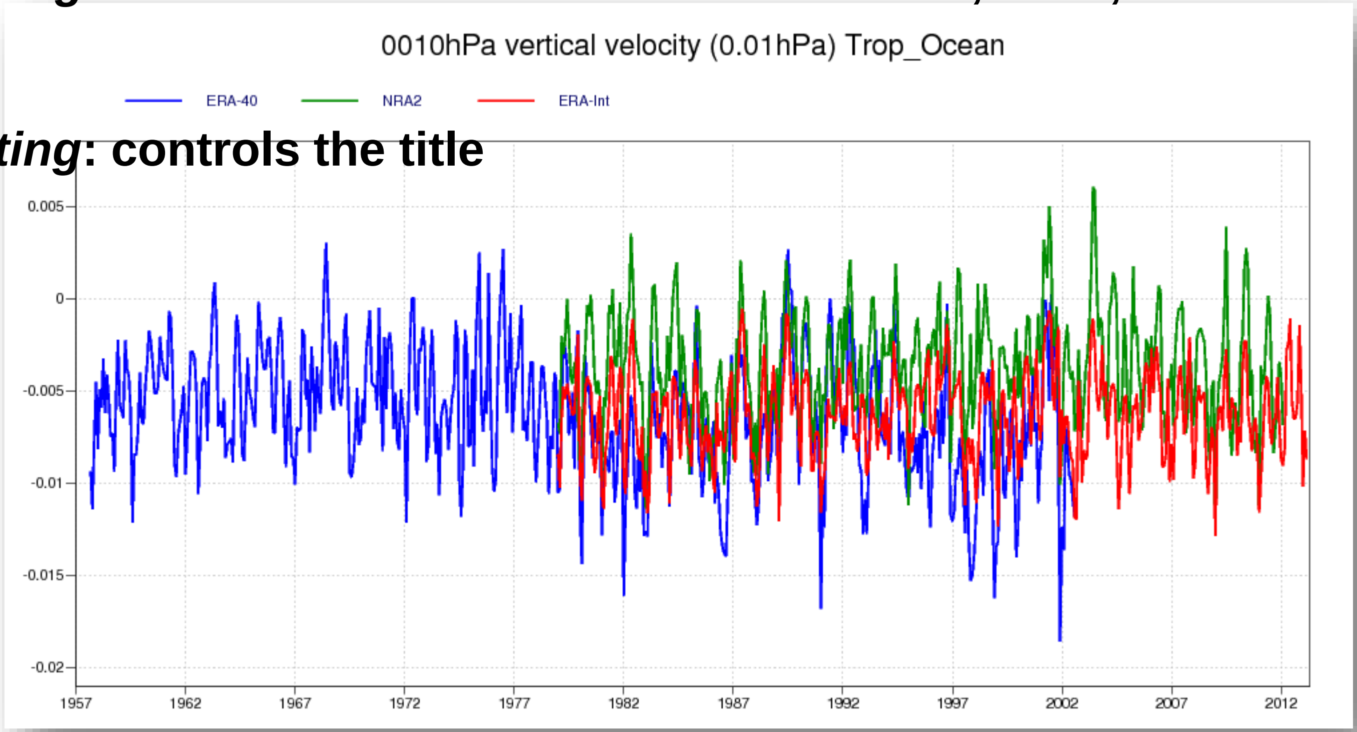
Layer	Value	X	Y
fc_anoms	0.540000	7.64	0.54





Graph Plotting: The Elements

- **Graph plots consist of:**
  - ***Cartesian View*: defines the coordinate system**
  - **Data!**
  - **Visdefs for the data**
  - ***Axis Plotting*: defines the look of the axes – colours, titles, etc.**
  - ***Legend***
  - ***Text Plotting*: controls the title**

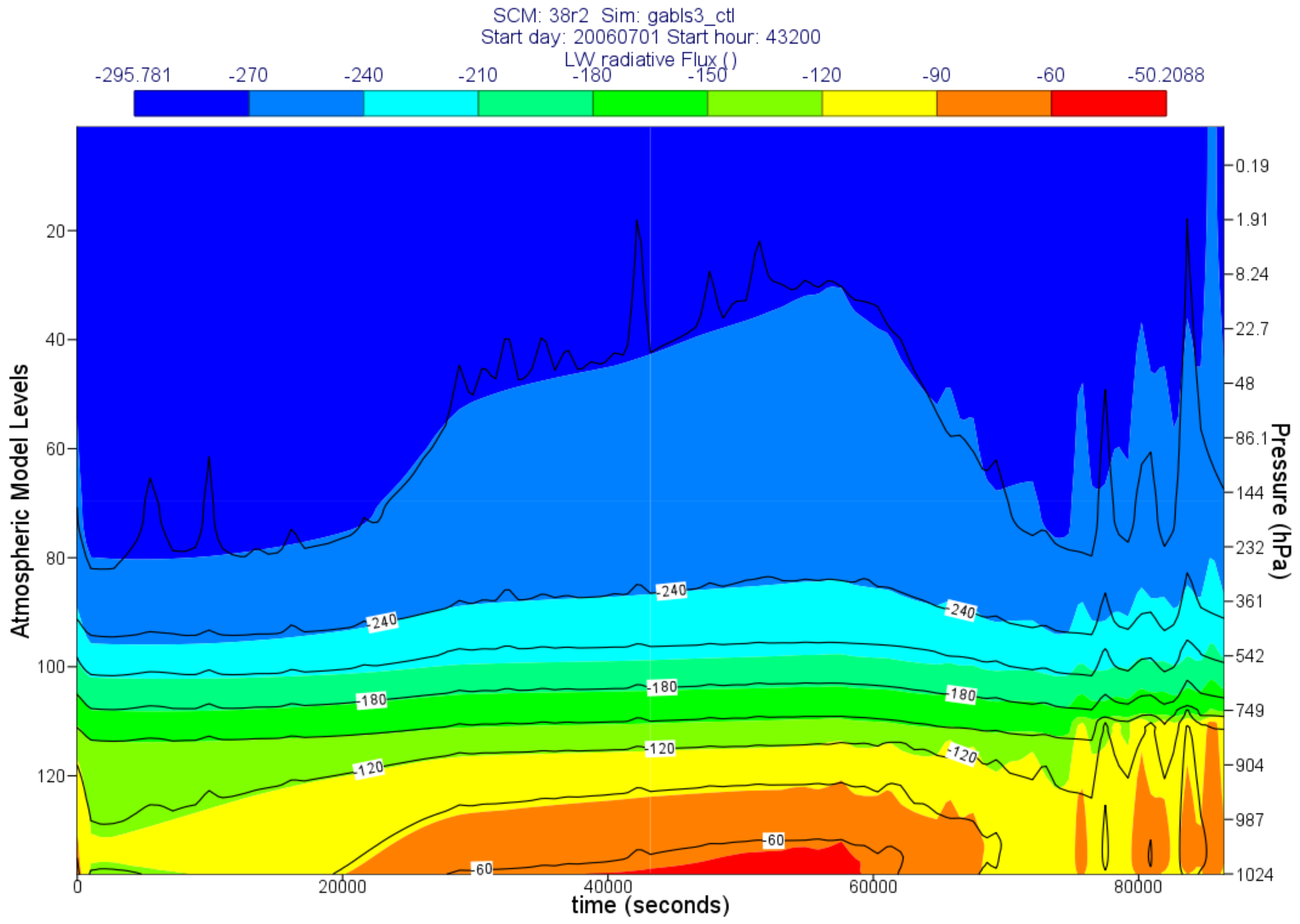


Metview Tutorial: Graph Plotting

- **Please do exercise “Graph Plotting in Metview” in the provided sub-folder “graph”**

Additional Notes

- **It is possible to use two Cartesian View icons to produce a ‘double’ axis**

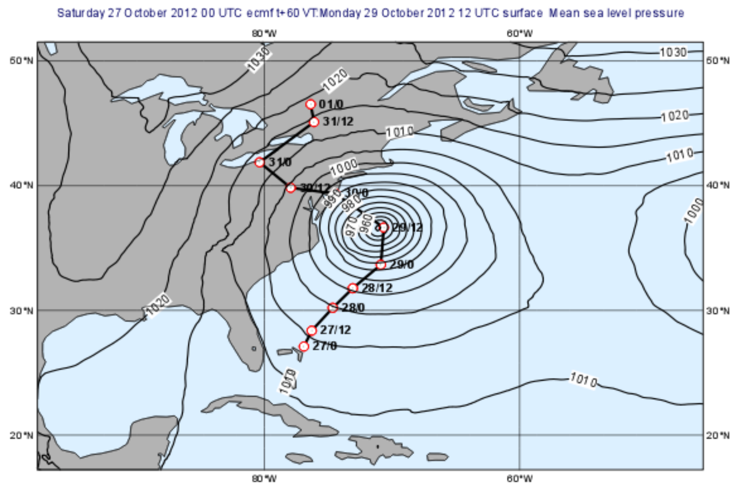


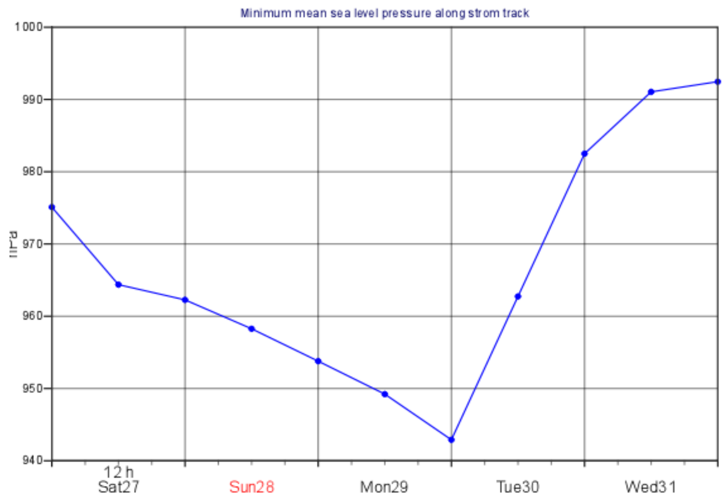
Case Study: Plotting the track of Hurricane Sandy

Sándor Kertész

Software Applications Team

Metview Tutorial: Case Study – Plotting the track of Hurricane Sandy


Saturday 27 October 2012 00 UTC ecmf t+60 VT:Monday 29 October 2012 12 UTC surface Mean sea level pressure
80°W
60°W
50°N
40°N
30°N
20°N
1030
1020
1010
1000
01/0
31/12
31/0
30/12
30/0
29/12
29/0
28/12
28/0
27/12
27/0



- **Please do exercise "Case Study: Plotting the track of Hurricane Sandy" in the provided sub-folder "sandy track"**

Titles and workflow

Iain Russell

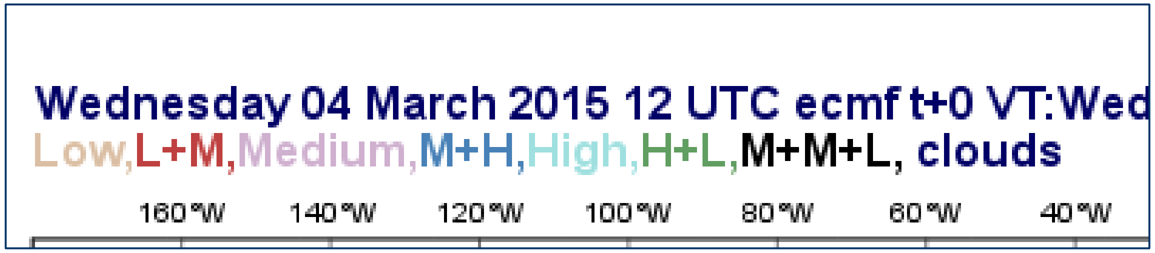
Software Applications Team

Optimising Your Workflow

- How to store icons for easy re-use
- Changing the defaults

Titles

- **Automatic text for fields**
- **User-defined text**
  - **With or without elements extracted from the data**
  - **Styled using HTML notation**



Wednesday 04 March 2015 12 UTC ecmf t+0 VT:Wed
Low,L+M,Medium,M+H,High,H+L,M+M+L, clouds

Metview Tutorial: Workflow and Titles

Get the data and icons for the day

- **From a command line type:**

```
~trx/mv_data/get_day_4.sh
```

- **A new folder called "day_4" will appear in your "training" folder**

- **Please do exercise "Optimising Your Workflow" in the provided sub-folder "workflow"**
- **Please do exercise "Customising Your Plot Title" in the provided sub-folder "titles"**

Macro in-depth

Iain Russell

Software Applications Team

Macro Essentials - Variables

- No need for declaration
- Dynamic typing

```
a = 1           # type(a) = 'number'
a = 'hello'     # type(a) = 'string'
a = [4, 5]      # type(a) = 'list'
a = |7, 8|      # type(a) = 'vector'
```

Macro Essentials - Variables

- **Scope and Visibility**
  - **Variables inside functions are local**
- **Functions cannot see ‘outside’ variables**

```
x = 9                  # cannot see y here

function func

     y = 10            # cannot see x here

end func

                       # cannot see y here
```

Macro Essentials - Variables

- **Scope and Visibility**
  - **… unless a variable is defined to be ‘global’**

```
global g1 = 9               # cannot see y1 here
function func
     y1 = 10 + g1           # can see g1 here
end func
                            # cannot see y1 here
```

Macro Essentials - Variables

- **Scope and Visibility**
  - **… a better solution is to pass a parameter**
  - **… that way, the function can be reused in other macros**

```
x = 9


func(x)      # x is passed as a parameter


function func (t : number) #t adopts value of x
    y1 = 10 + t           # y1 = 10 + 9
end func
```

Macro Essentials - Variables

- **Destroying variables automatically**
  - **When they go out of scope**

```
function plot_a
   a = retrieve(...)
   plot(a)
end plot_a


# Main routine
plot_a()   # a is created and destroyed
```

Macro Essentials - Variables

- **Destroying variables manually**
  - **Set to zero**
  - **(Variables can ‘hold’ lots of data, either in memory or in temporary files)**

```
a = retrieve(...)
plot(a) # we have finished with ’a’ now
a = 0
b = retrieve(...)
plot(b)
```

Macro Essentials - Strings

- `'Hello'` **is the same as** `"Hello"`

- **Concatenate strings with strings, numbers and dates using the `'&'` operator**

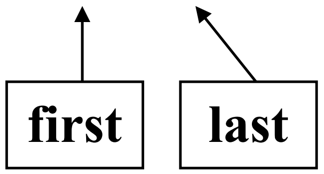
  eg. `"part1_" & "part2_" & 3`

  produces `"part1_part2_3"`

- **Obtain substrings with** `substring()`

  e.g. `substring ("Metview", 2, 4)`

  produces `"etv"`



Macro Essentials - Strings

- Split a string into parts using `parse()`
- Creates a list of substrings

```
n = parse("z500.grib", ".")
print ("name = ", n[1], " extension = ",
n[2])
```

- prints the following string :

```
name = z500 extension = grib
```

Macro Essentials - Lists

- **Ordered, heterogeneous collection of values. Not limited in length. List elements can be of any type, including lists. List are built using square brackets, and can be initialised with `nil`:**

```
l = [3,4,"foo","bar"]


l = nil
l = l & [2,3,[3,4]]
l = l & ["str1"] & ["str2"]
europe = [35,-12.5,75,42.5]    # S, W, N, E
```

Macro Essentials - Lists

- **Accessing List Elements**
- **Indexes start at 1**

```
mylist = [10,20,30,40]
a = mylist[1]      # a = 10
b = mylist[2,4]    # b = [20,30,40] (m to n)
c = mylist[1,4,2] # c = [10,30] (step 2)
```

Macro Essentials - Lists

- **Useful List Functions**

```
num_elements = count (mylist)
sorted = sort (mylist)
  # can provide custom sorting function

 if (2 in mylist) then
    …
 end if
```

Macro Essentials - Lists

- **Useful List Functions**

```
mylist = ['b', 'a', 'a', 'c']

# find occurrences of 'a' in list
 index   = find(mylist, 'a') # 2
 indexes = find(mylist, 'a', 'all') # [2,3]

 # return list of unique members
 reduced = unique(mylist) # ['b', 'a', 'c']
```

Macro Essentials - Lists

- **List Operations**
- **Operators acting on lists will act on each list element, returning a list of results**

-
  ```
  a = [3, 4]
  b = a + 5   # b is now [8, 9]
  c = a * b   # c is now [24, 36]
  ```
- **Lists are general-purpose, and are not recommended for handling large amounts (thousands) of numbers – for that, use *vectors* (see later)**

Macro Essentials - Vectors

- **Ordered, array of numbers. Much more efficient than lists for high volumes of numeric data. Vectors are built using the vertical bar symbol, and can be initialised with `nil`:**

```
v = |7, 8, 9|


v = nil # start from nil and append
v = v & |4.4, 5.5, 3.14| & |8, 9|


v = vector(10000) # pre-allocate space
v[1] = 4 # assign values to indexes
```

Macro Essentials - Vectors

- **Can extract data arrays from most data types into vector variables:**

```
v = values(fieldset)
v = values(geopoints)
v = values(netcdf) # takes from current variable
v = values(table, ‘col2’)
v = values(odb, ‘precip’)
```

Macro Essentials - Vectors

- **Assigning/replacing a range of values at once:**

```
v = |10,20,30,40|
v[2] = |99,99| # v is now |10,99,99,40|
```

Macro Essentials - Vectors

- **Operations and functions are applied to each element:**

```
x = |3, 4, 5|
y = x + 10 # y is now |13, 14, 15|

c = cos(x)

u = |7.3, 4.2, 3.6|
v = |-4.4, 1.1, -2.1|
spd = sqrt((u*u) + (v*v))
```

Macro Essentials - Vectors

- **Accessing vector elements**
- **Indexes start at 1**

```
v = |10,20,30,40|
a = v[1]          # a = 10
b = v[2,4]        # b = |20,30,40| (m to n)
c = v[1,4,2]      # c = |10,30| (step 2)
d = v[1,4,2,2]    # d = |10,20,30,40|
                  # (take 2 at each step)
```

Macro Essentials – Loops, Tests & Functions

- The for, while, repeat, loop statements
  - See ‘Metview Macro Syntax’ handout
- The if/else, when, case statements
  - See ‘Metview Macro Syntax’ handout
- Function declarations
  - See ‘Metview Macro Syntax’ handout

Macro Essentials - Functions

- **Multiple versions**
  - **Can declare multiple functions with the same name, but with different parameter number/types.**

```
function fn_test ()

function fn_test (param1: string)

function fn_test (param1: number)
```

  - **Correct one will be chosen according to the supplied parameters**

Macro Essentials - Definitions

- **A collection of named items (members)**
- **Eg**

```
   a = (x : 1, y : 2) # create definition


   c = a.x            # get value of 'x'
     or
   c = a["x"]
```

- **Like a struct in 'C' or a dictionary in Python**

Macro Essentials - Definitions

- **Icon-functions take definitions:**

```
acoast = mcoast(
     map_coastline_resolution       :    "high",
     map_coastline_colour           :    "red",
     map_grid_colour                :    "grey",
     map_grid_longitude_increment   :    10,
     map_label_colour               :    "grey",
     map_coastline_land_shade       :    "on",
     map_coastline_land_shade_colour:    "cream"
     )
```

Macro Essentials - Definitions

```
param_def = ( param : "Z",
              type  : "FC",
              date  : -1,
              step  : 24 )

# retrieve as LL grid or not according to user
# choice
if (use_LL = "yes") then
  param_def.grid = [1.5,1.5]
end if

Z_ret = retrieve (param_def)
```

Macro Essentials - Definitions

```
common_input = ( levtype : "PL",
                 levelist : 850,
                 time : 12,
                 grid : [2.5,2.5],
                 type : "AN" )

Uan = retrieve ( common_input,
                 date   : -1,
                 param : "U" )

Van = retrieve ( common_input,
                 date   : -2,
                 param : "V" )
```

Fortran and C in Macro - Introduction

- **Users can write their own Macro functions in Fortran or C/C++, extending the Macro language**

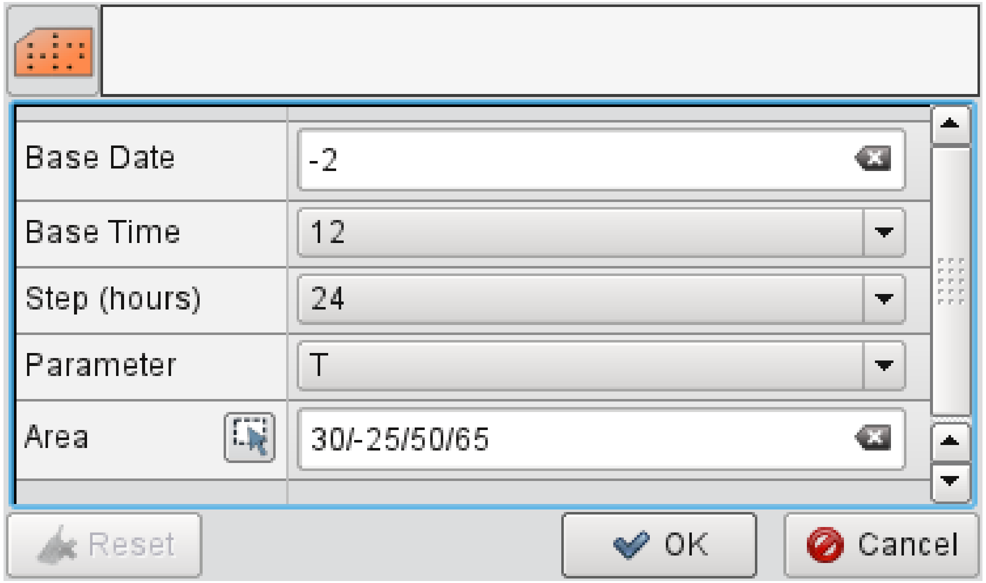
- **Used in tasks which cannot be achieved by macro functions. Or use existing FORTRAN/C code to save time.**

- **FORTRAN/C-Metview macro interfaces support input data of types GRIB, number, string and vector. BUFR, images and matrices are waiting implementation.**

- **See examples in solutions folder**

User Interfaces in Macro - Introduction

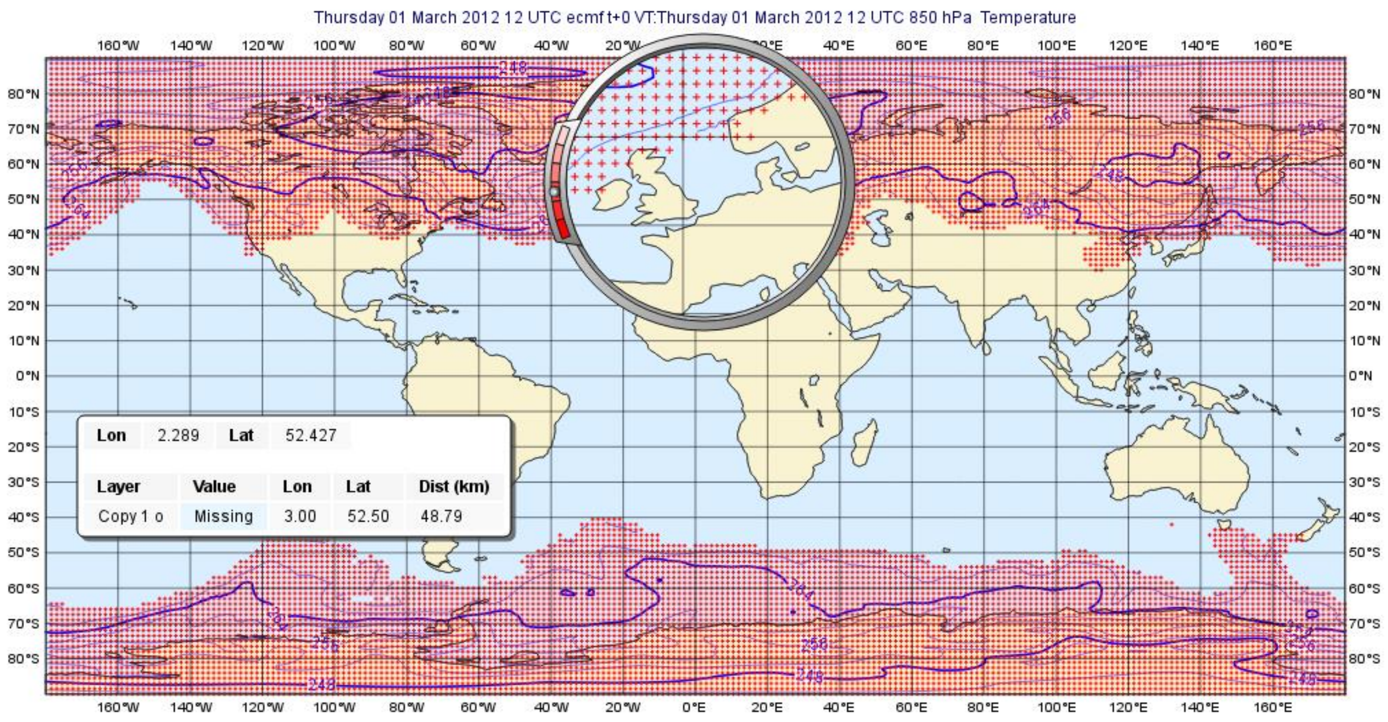
- **Users can write their own simple user interfaces in Macro**
- **One way to allow other users to call a macro and change its parameters without modifying code or passing command-line arguments**
- **See example in solutions folder**



Base Date	-2
Base Time	12
Step (hours)	24
Parameter	T
Area	30/-25/50/65

Missing Values

- **Fields and other data can have missing values**
- **Be aware of this!**
- **They can also be used to mask data, returning a specific subset of points**



Lon	2.289	Lat	52.427	
Layer	**Value**	**Lon**	**Lat**	**Dist (km)**
Copy 1 o	Missing	3.00	52.50	48.79

Metview Tutorial: Macro

- **Please do exercise “Organising Macros” in the provided sub-folder “organising macros”**

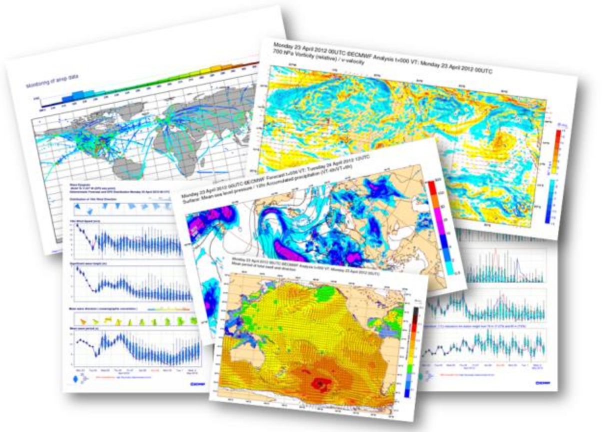
- **Please do exercise “Missing Values and Masks” in the provided sub-folder “missing values”**

Publishing graphical output

Stephan Siemen

Magics developer

The role of Magics in Metview

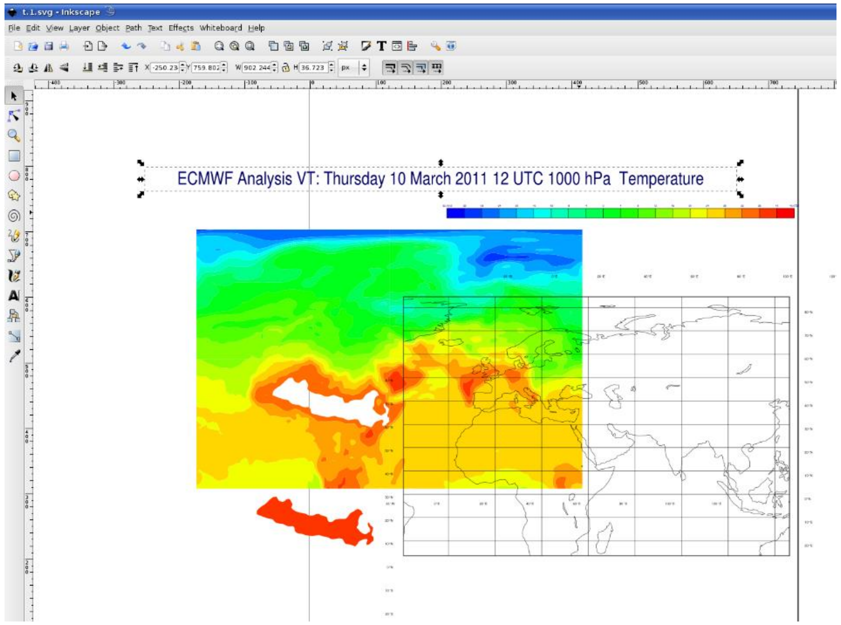


- **Magics is ECMWF’s plotting package**
  - **Tuned to visualise meteorological data, but has no data processing capabilities**
  - **GRIB, BUFR, ODB, CSV**
  - **APIs: Python, Fortran and Metview**
    - **The Python interface very similar to macro language**

      from Magics.macro import *
  - **Used also for the web (ecCharts, Metgrams)**
  - **Magics is now called alone at ECMWF 800K – 1Mill. times a day!**
- **Everything graphical in Metview is done through Magics**
- **There is a bit of historical legacy in names and settings**
  - **PostScript was for long time the main format**

Output formats

- **PDF**
  - **Static vector format for printing, web & archive**
- **PNG**
  - **Raster format for web**
- **PostScript & EPS**
  - **Vector format for printing**
- **SVG**
  - **Vector format for web**
- **KML/KMZ**
  - **Format for Google Earth & Maps**
- **Qt**
  - **Used by Metview only**


t.1.svg - Inkscape
File Edit View Layer Object Path Text Effects Whiteboard Help
ECMWF Analysis VT: Thursday 10 March 2011 12 UTC 1000 hPa Temperature

Pages and page layout

- **Settings are historical**
  - **From a time were most output was for printing**
- **Concept of superpage and page**
  - **Superpage is overall page**
    - **Change (paper) output size here**
  - **Pages can be pages (as in books) or sub pages within a superpage**
    - **Holds the view**
    - **Can be positioned within a superpage**
- **You can of course always use the Display Window editor and drop this in the Macro Editor**

Metview Tutorial: Working with Graphical Output

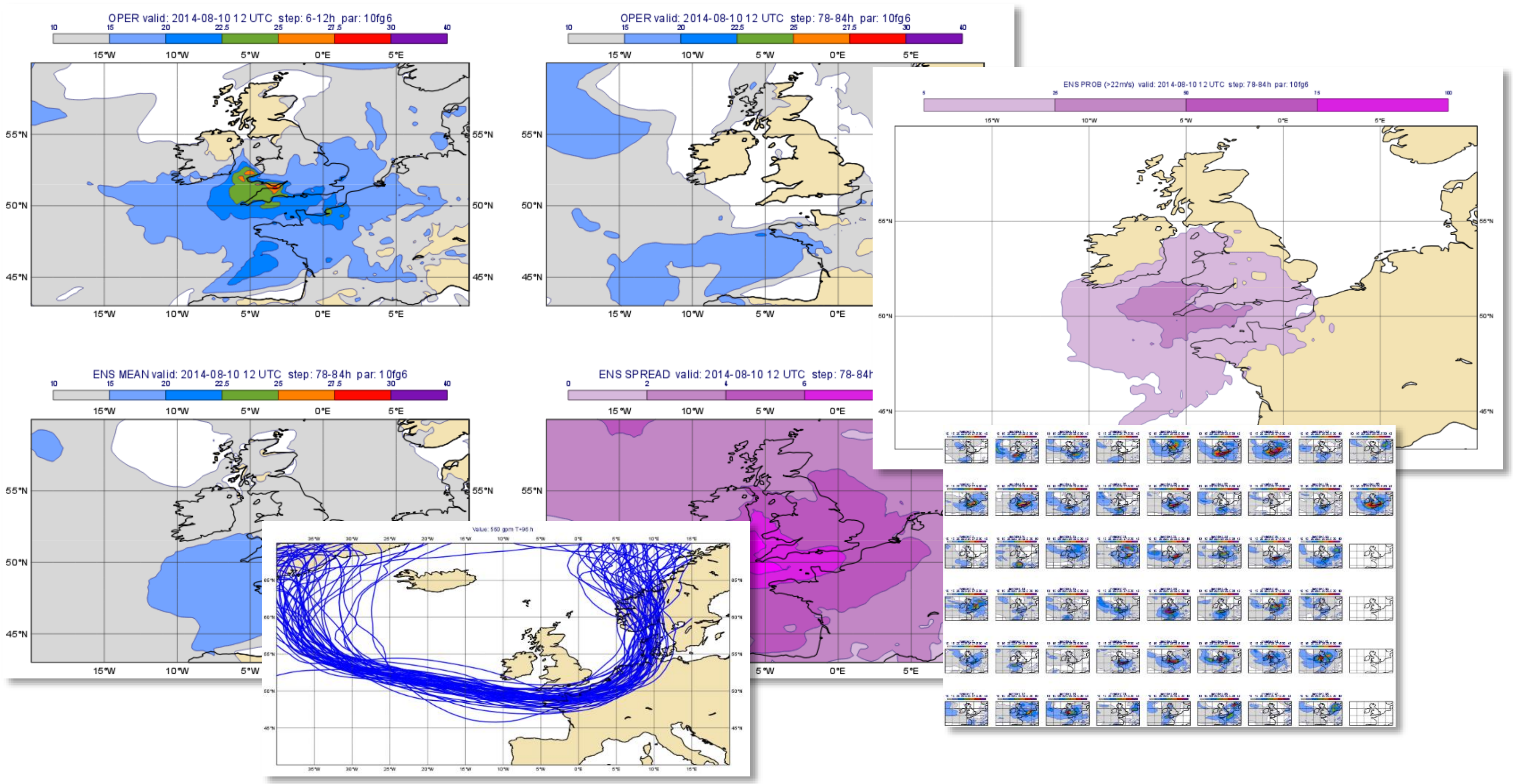
- **Please do exercise “Working with graphical output” in the provided sub-folder “graphics formats”**

Case Study: Ensemble forecast

Sándor Kertész

Software Applications Team

Metview Tutorial: Case Study – Ensemble Forecast



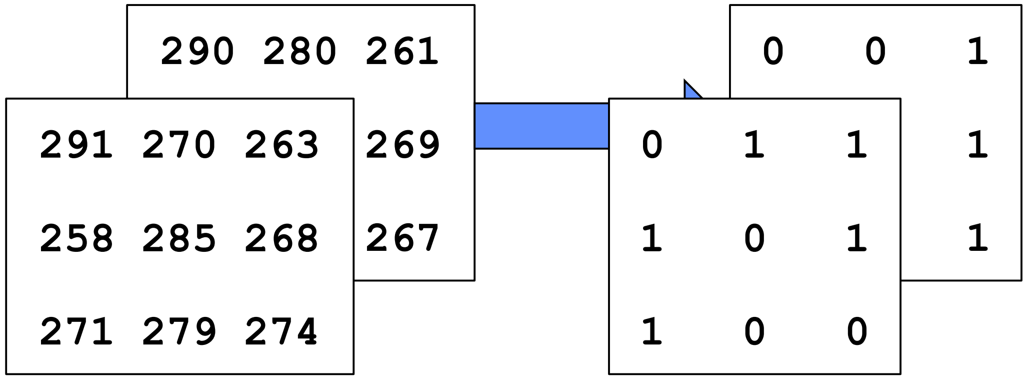
OPER valid: 2014-08-10 12 UTC step: 6-12h par: 10fg6
OPER valid: 2014-08-10 12 UTC step: 78-84h par: 10fg6
ENS PROB (>22m/s) valid: 2014-08-10 12 UTC step: 78-84h par: 10fg6
ENS MEAN valid: 2014-08-10 12 UTC step: 78-84h par: 10fg6
ENS SPREAD valid: 2014-08-10 12 UTC step: 78-84h
Value: 560 gpm T+96 h

Calculations with Ensemble Data

- If variable ens is a fieldset containing all ensemble members for temperature at a given time
- So ens has 51 fields…
- mean(ens) returns the ensemble mean (field)
- stdev(ens) returns the ensemble spread (field)

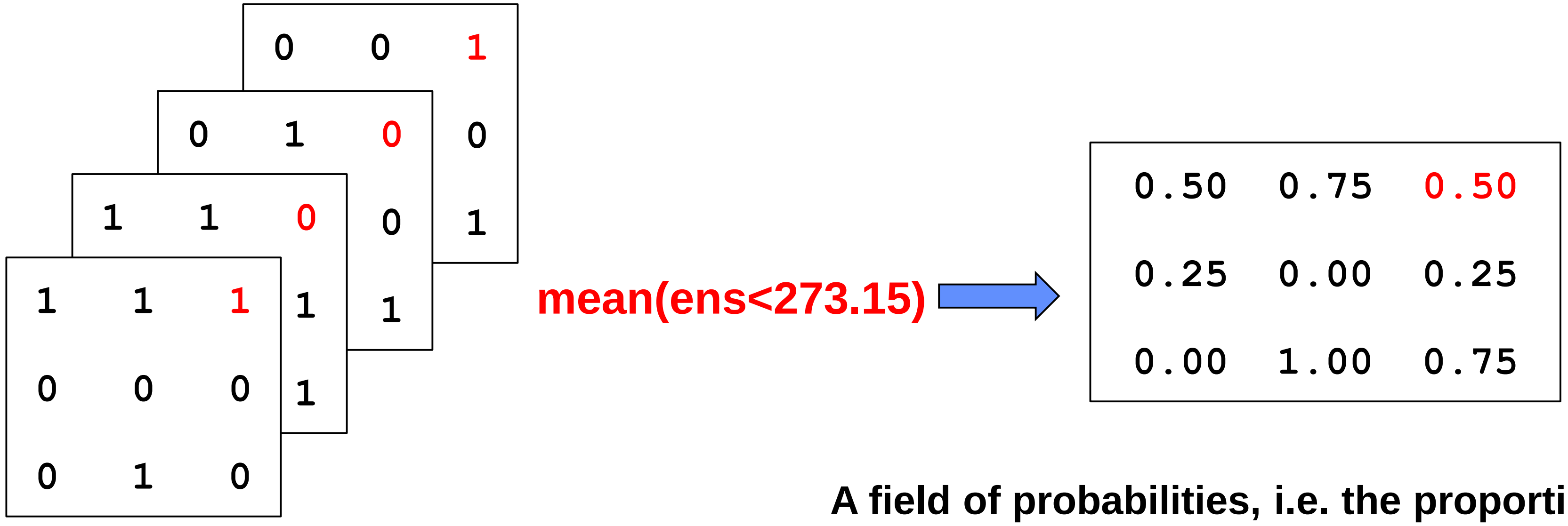
Calculations with Ensemble Data

- **Probability of freezing temperature**
- **ens < 273.15**
  - **Returns a set of 51 fields where all the grid values are 1s and 0s**



Calculations with Ensemble Data

- **So looking 'through' each grid point, for each point we have 51 values, a set of 1s and 0s**



A field of probabilities, i.e. the proportion of ensemble members which say 'yes'

Calculations with Ensemble Data

- **Other examples:**

- **Wind gust data:**
  - **mean(ens > 22) – probability of high wind gusts**

- **Precipitation data:**
  - **mean(ens > 0.02) – probability of more than 20mm of rain (if data are in metres)**

- **Temperature data:**
  - **mean(ens >= (40+273.15)) – probability of temperatures reaching 40C**

Calculations with Ensemble Data

- **Please do exercise “Case Study: Ensemble Forecast” in the provided sub-folder “ensemble forecast”**

Running Metview in batch mode

Iain Russell

Software Applications Team

Running Metview in Batch Mode

- **Can run macros directly from the command line:**
  - **metview –b <macro_name> [arg1 arg2 …]**

- **Arguments passed after the name of the macro to run**
- **Can also get environment variables**

- **The macro can detect whether it was run from the command line or the user interface**
  - **Can do different things in each case**

Running Metview in Batch Mode

Get the data and icons for the day

- **From a command line type:**

```
~trx/mv_data/get_day_5.sh
```

- **A new folder called "day_5" will appear in your "training" folder**

- **Please do exercise "Running Metview in Batch Mode" in the provided sub-folder "batch"**

Submitting Batch jobs on ecgate (separate presentation)

Xavi Abellan

User Support

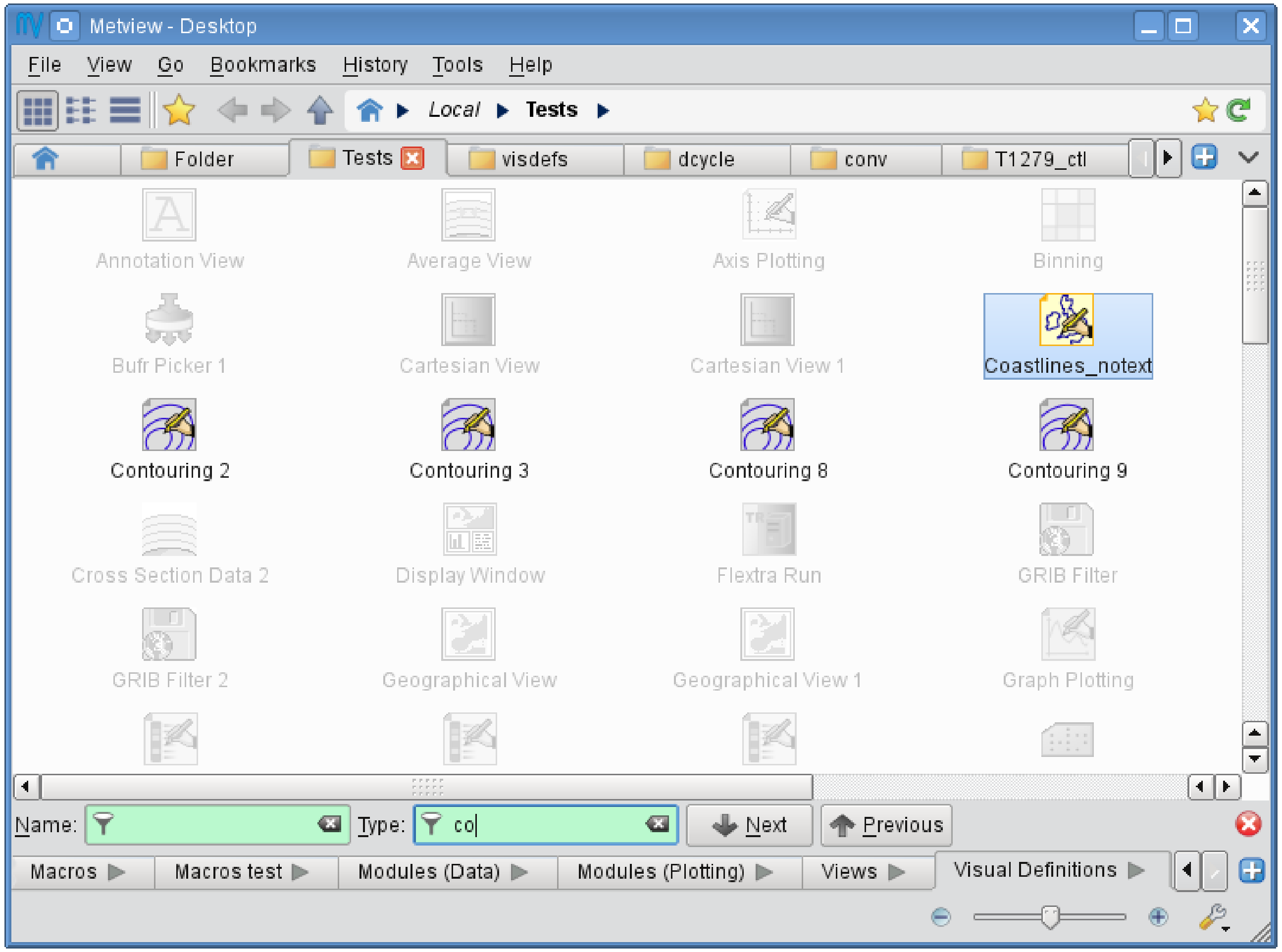
Exploring Metview

Iain Russell

Software Applications Team

Exploring Metview (1)

- **User interface**
  - **More advanced features**



Exploring Metview (2)

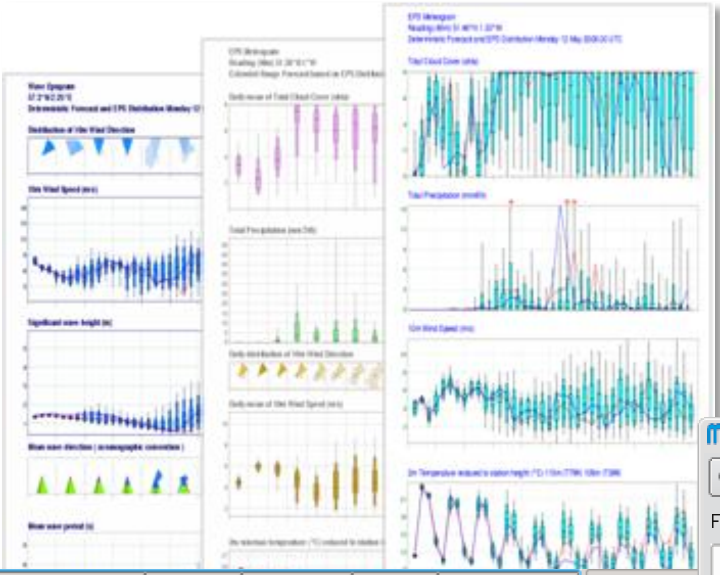
- **Mail -** exchange icons by email
- **Archive** - convert a group of icons into an archive
- **Monitor – to monitor and control tasks**
  - **Check the progress of long tasks**
  - **Abort a misbehaving Metview process**
- **Station – search Station Database**
  - **Access Metview database of 10,000 WMO stations**

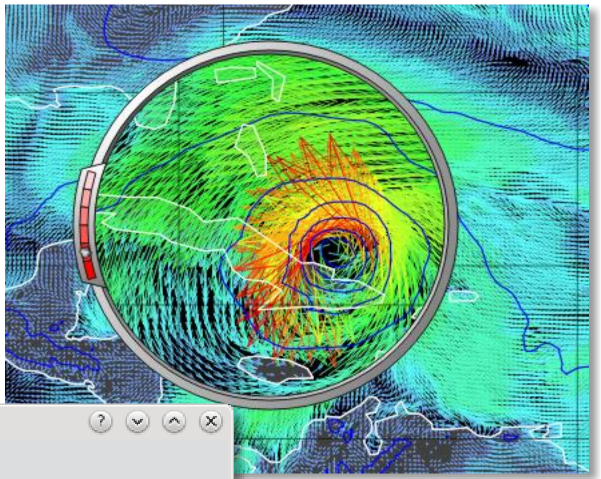
Exploring Metview (3)

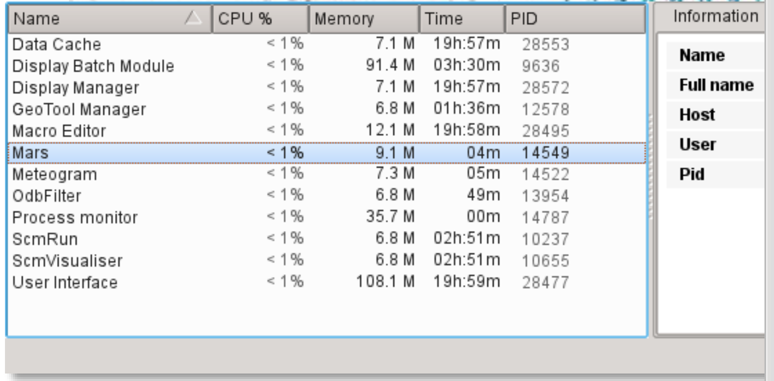
- **Command-line magic:**
  - `metview -p <file>`
    - **To plot the given file using the ecCharts style**
  - `metview -e <grib | bufr | odb> <file>`
    - **To start up a data examiner**
  - `metview -slog`
    - **provides output logs to the terminal**
  - `metview -h`
    - **To see the available options**

Exploring Metview (4)

- **More applications, plot types and tools**



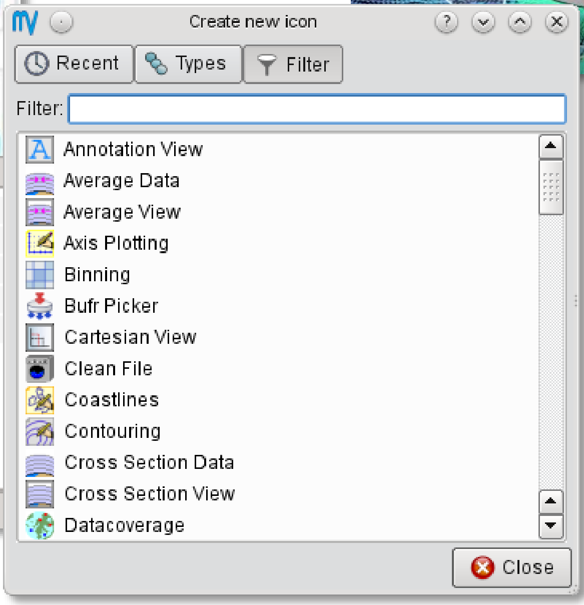




Name	CPU %	Memory	Time	PID
Data Cache	< 1%	7.1 M	19h:57m	28553
Display Batch Module	< 1%	91.4 M	03h:30m	9636
Display Manager	< 1%	7.1 M	19h:57m	28572
GeoTool Manager	< 1%	6.8 M	01h:36m	12578
Macro Editor	< 1%	12.1 M	19h:58m	28495
Mars	< 1%	9.1 M	04m	14549
Meteogram	< 1%	7.3 M	05m	14522
OdbFilter	< 1%	6.8 M	49m	13954
Process monitor	< 1%	35.7 M	00m	14787
ScmRun	< 1%	6.8 M	02h:51m	10237
ScmVisualiser	< 1%	6.8 M	02h:51m	10655
User Interface	< 1%	108.1 M	19h:59m	28477

Information

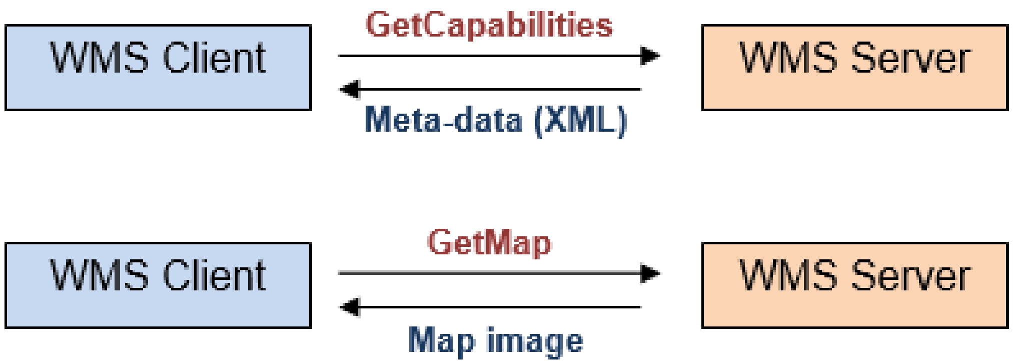
Name
Full name
Host
User
Pid



- Annotation View
- Average Data
- Average View
- Axis Plotting
- Binning
- Bufr Picker
- Cartesian View
- Clean File
- Coastlines
- Contouring
- Cross Section Data
- Cross Section View
- Datacoverage

OGC Web Map Service (WMS)

- **Established standard to exchange geographical maps**
  - **Web based**
  - **Server provides a *getCapabilities* document which describes which layers its provides**
  - **Allows supporting clients to request maps and overlay**

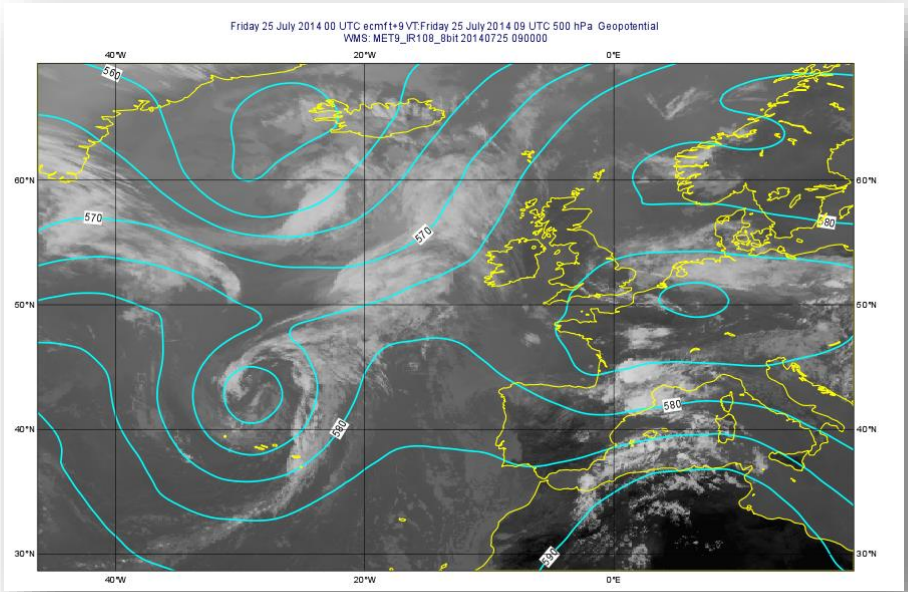


- **Also ECMWF has a WMS server integrated with ecCharts**
  - **Allows user clients to overlay forecast maps**
  - **Is secured through a “token” you need to request**

Metview's WMS client

- **Metview has a WMS client**
  - **You only need to give it the URL to a *getCapabilities* document**
  - **The client will build the user interface to select the layer**
  - **Fits well in Metview's service oriented design**
- **Metview will allow the overlay with any other content**

- **We have a separate tutorial on WMS and Metview if you want to learn more**
- **Be aware that the WMS services are not always reachable**
  - **You might not want to rely on an external server**



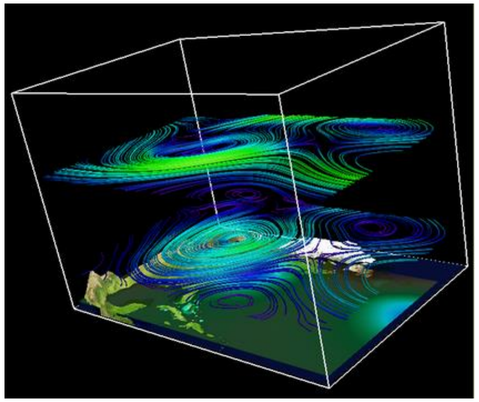
Friday 25 July 2014 00 UTC ecmf t+9 VT:Friday 25 July 2014 09 UTC 500 hPa Geopotential
WMS: MET9_IR108_8bit 20140725 090000
40°W
20°W
0°E
60°N
50°N
40°N
30°N
560
570
570
580
580
580
580

Example of a WMS layer from EUMETSAT overlaid with Metview coastlines and contours

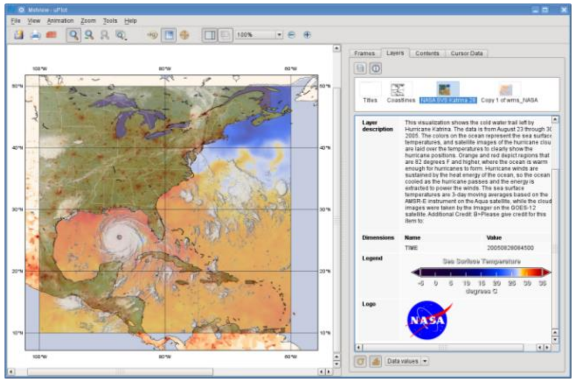
Exploring Metview: Tutorials

- **Many other tutorials available, including:**

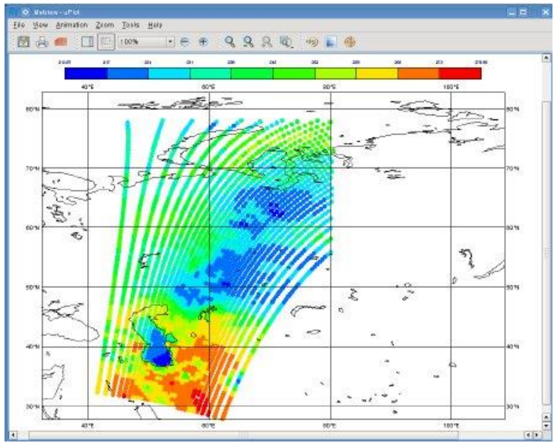
VAPOR

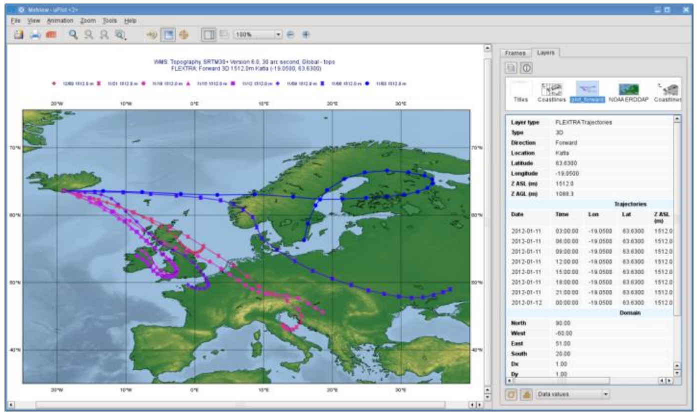


WMS



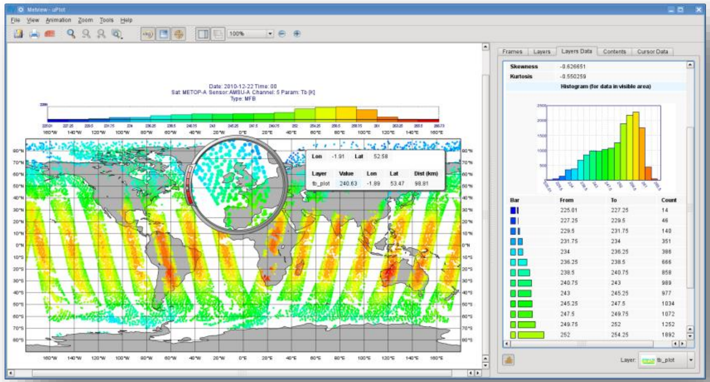
BUFR





FLEXTRA

ODB



https://software.ecmwf.int/metview

Metview Tutorial: Titles and Workflow

- **Please do exercise "Working with Folders and Icons" in the provided sub-folder "explore"**
- **Please do exercise "Exploring Metview" in the provided sub-folder "explore"**

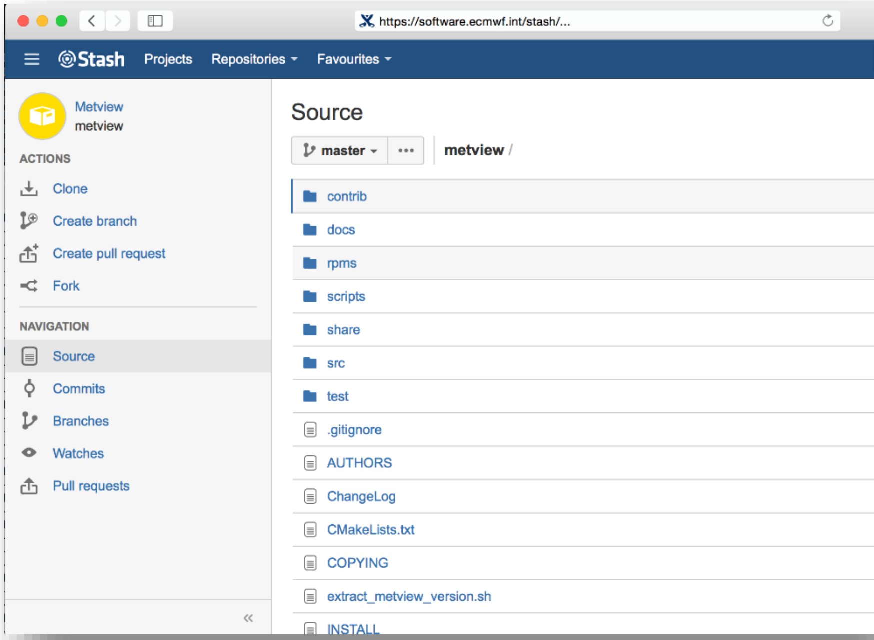
Closing remarks

Deploying Metview

- **Compilation from source**
  - The installation has been made much easier over recent years
    - CMake based build system is now the same for all ECMWF packages
    - Please give us feedback to improve it
  - Support for Mac OS X 10.10 was added
  - You can build batch-only version, with less dependencies
- **Used of pre-built versions**
  - More and more communities start building binary packages of Metview
    - *Opensuse Build Service (OBS)* offers RPMs for OpenSuse, SLES, Redhat, CentOS and Fedora
    - Ubuntu 15.04 will have Metview in its Science repository
  - Virtual machines are available to start quickly with Metview
    - On *SuseStudio* this can be used to build appliances for cloud services
  - There are plans to provide a Metview Docker image

How to take it further …

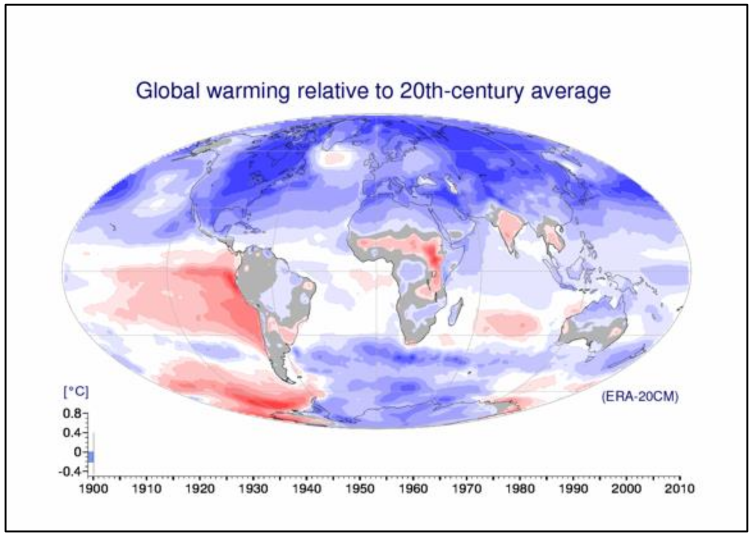
- **ECMWF has since last year a new Software Strategy**
  - Infrastructure will be put in place for external co-operation
  - Open up source code through Git
  - We will have unified build system
- **Metview has already code co-operations**
  - We have area in source code for contributions
  - We work on documenting how to extend Metview (with own modules)



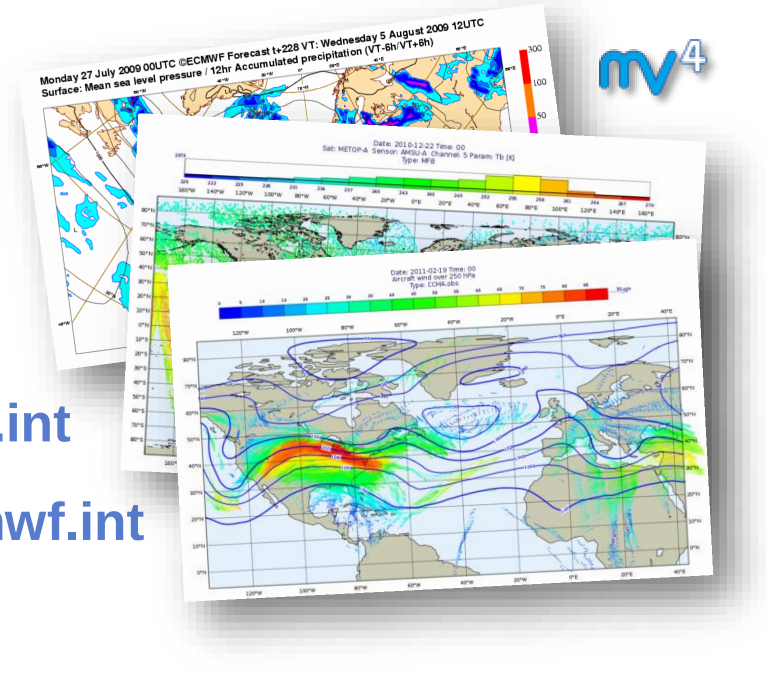
- **We build at ECMWF a comprehensive test system**
  - If you want make sure features not to regress, contribute a test
    - Needs to be small and self contained

Keep in touch ….

…. also when things work ;-)



For more information …



Support queries:

- **JIRA:** **https://software.ecmwf.int**
- **Support:** **software.support@ecmwf.int**

email us:

- **Metview:** **metview@ecmwf.int**

visit our web pages:

- **https://software.ecmwf.int/metview**
  - ➢ **Download**
  - ➢ **Documentation and tutorials available**
  - ➢ **Metview articles in recent ECMWF newsletters**

Hope you enjoyed the course and found it useful!

Please leave us your feedback ☺