EUMETSAT
H SAF
SUPPORT TO OPERATIONAL HYDROLOGY AND WATER MANAGEMENT

How to work with RZSM products: From download to visualization

EUMETSAT Event week

5th November 2019

David Fairbairn, Patricia de Rosnay, Phil Browne, Lars Isaksen and Clement Albergel

Contents

EUMETSAT H SAF
SUPPORT TO OPERATIONAL HYDROLOGY AND WATER MANAGEMENT

1. Introduction;
2. Requirements for running metview-python and other libraries
3. Downloading and visualizing H14 and H27 data record product
   - Downloading the grib files from the ftp
   - Reading the grib data using metview-python
   - Plotting the grib data directly using metview-python
   - Converting to regular grid and plotting using cartopy
   - Plotting a time series of the H27 data record
   - Converting to netCDF with regional data
   - Plotting regional netCDF data with cartopy
4. Summary

Table of current products

EUMETSAT H SAF
SUPPORT TO OPERATIONAL HYDROLOGY AND WATER MANAGEMENT

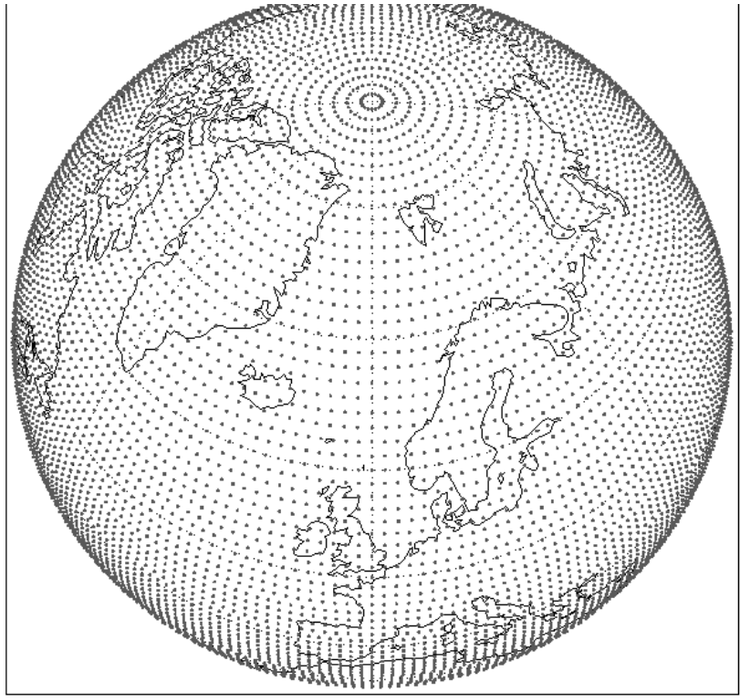
NRT and data record RSM products

Product name	Type	Period	Obs assimilated	DA system	Resolution	QC flags
H14	NRT	2012 onwards	25 km sampled ASCAT-A/B SSM products (H102/H103)	Regular updates of ECMWF LDAS (38R1-46R1)	25 km	**Yes since 4/10/18 (1=normal, 2=frozen risk, 3=outide nominal range)**
H27	Data record	1992-2014	ERS 1/2 (1992-2006) and ASCAT-A (2007-2014) reprocessed SSM	41R1 of ECMWF LDAS	16 km	**None**
H140	Data record	2015-2016	ASCAT-A (2015-2016) reprocessed SSM	43R3 of ECMWF LDAS	16 km	**None**

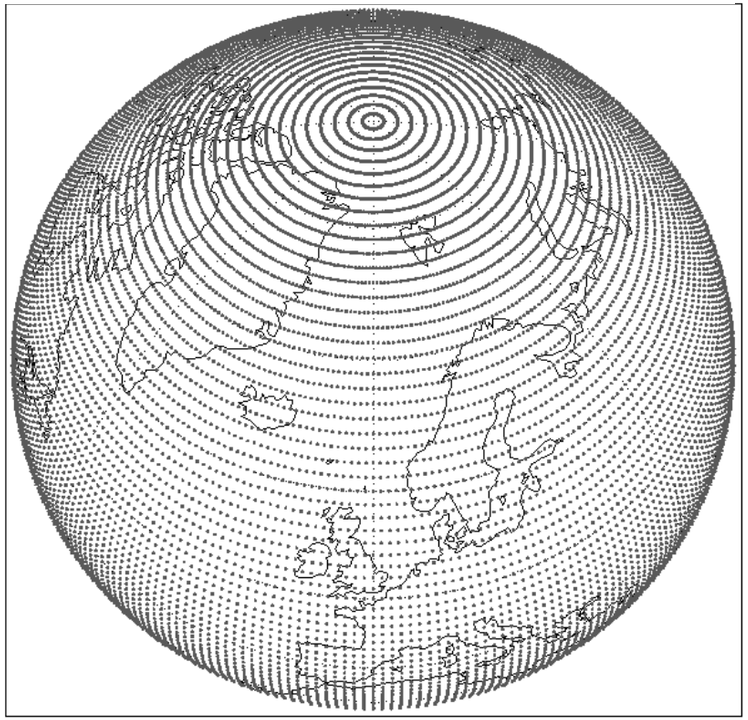
Format

- H14 is available in grib format on a linear reduced Gaussian grid (T799~25 km resolution);
- H27/H140 are available in grib format at (T1279~16 km resolution).
- The reduced Gaussian grid maintains approximately equidistant grid-point distances between the poles and the equator (unlike regular lat/lon grid):

Reduced Gaussian grid:



Regular lat/lon grid:



For more info, see https://confluence.ecmwf.int/display/FCST/Gaussian+grids

Requirements for examples

EUMETSAT H SAF
SUPPORT TO OPERATIONAL HYDROLOGY AND WATER MANAGEMENT

- python version 3.6
- The following libraries: *metview, pandas, pylab, matplotlib, numpy, cartopy*
- A miniconda script for linux/mac will be provided on github, which sets up a local environment and downloads the relevant libraries for all the soil moisture examples: https://github.com/H-SAF/eumetrain_sm_week_2019
- The following examples can be run in jupyter notebook (Exercise_2 from github above)
- More information about metview-python: https://github.com/ecmwf/metview-python

Documentation

The documentation can be found via http://hsaf.meteoam.it/user-documents.php:

- Product user manual
- Algorithm theoretical baseline document
- Product validation report
- If you have any queries, please contact the H SAF helpdesk: us_hsaf@meteoam.it

Downloading the data from ftp

- H14, H27 and H140 daily grib files can be downloaded via the H-SAF ftp: ftp://ftphsaf.meteoam.it/products
- First register with H SAF: http://hsaf.meteoam.it/user-registration.php to obtain username and password
- Easy to download data from ftp using software such as filezilla https://filezilla-project.org/ .
- Alternatively, to download example file (H14 for 30/05/2018) from the terminal type:

ftp user@ftphsaf.meteoam.it

password: (type your password)

cd /products/h14/h14_cur_mon_grib

get h14_20190530_0000.grib.bz2

exit

bzip2 -d h14_20190530_0000.grib.bz2

Reading grib data in metview-python

EUMETSAT H SAF
SUPPORT TO OPERATIONAL HYDROLOGY AND WATER MANAGEMENT

```
import metview as mv #for reading and plotting grib files
import numpy as np #for grid/data manipulation
import matplotlib.pyplot as plt #For plotting
```

```
SM =mv.read("h14_2019053000.grib") #Read in grib data for 30/5/2019:

SWI1=mv.read(data=SM,param="40.228") #1st layer (0-7 cm depth)
SWI2=mv.read(data=SM,param="41.228") #2nd layer (7-28 cm depth)
SWI3=mv.read(data=SM,param="42.228") #3rd layer (28-100 cm depth)
SWI4=mv.read(data=SM,param="43.228") #4th layer (100-289 cm)
qc=mv.read(data=SM,param="200.228") #Quality control flag (1=normal, 2=risk of frozen conditions,
#3=outside nominal range)

RZSM = SWI1
#Extract root-zone SM values from depth integrating the 4 layers
RZSM = RZSM.set_values(SWI1.values()*0.07 + SWI2.values()*0.21 + SWI3.values()*0.72)
```

Plotting grib data in metview-python

EUMETSAT H SAF SUPPORT TO OPERATIONAL WATER

```
#Create contouring for metview plot
cont_pc = mv.mcont(contour = "off",
          legend = "on",
          contour_shade = "on",
          contour_level_selection_type = "level_list",
          contour_level_list = [0.0, 0.125, 0.25, 0.375, 0.5, 0.625, 0.75, 0.875, 1.0],
          contour_label_format = "(F4.1)",
          contour_label = "off",
          contour_label_height = 0.40,
          contour_label_colour = "charcoal",
          contour_label_frequency = 10,
          contour_highlight                = "off",
          contour_shade_method = "area_fill",
          contour_shade_colour_method = "gradients",
          contour_gradients_colour_list = ['burgundy','red','orange','yellow','green','cyan','sky','blue','navy'],
          contour_gradients_step_list = [20,20,20,20,20,20,20,20])
```

```
#Create legend:
my_legend = mv.mlegend(legend_display_type = "continuous",
                  legend_entry_border='off',
                  legend_text_colour = "charcoal",
                  legend_values_list= [0.0, 0.2, 0.4, 0.6, 0.8, 1.0],
                  legend_text_composition = 'user_text_only',
                  legend_text_font_size = 1,
                  legend_user_maximum = "on",
)
```

Plotting grib data in metview-python

EUMETSAT H SAF
SUPPORT TO OPERATIONAL HYDROLOGY AND WATER MENT

```
# shaded land to make the points stand out more
grey_land_shading = mv.mcoast(
    map_coastline_land_shade         = "on",
    map_coastline_land_shade_colour = "grey",
    map_grid_latitude_increment      = 40,
    map_grid_longitude_increment     = 80,
    map_grid_colour                  = "charcoal",
    map_grid = "off",
    map_label = "off")
```

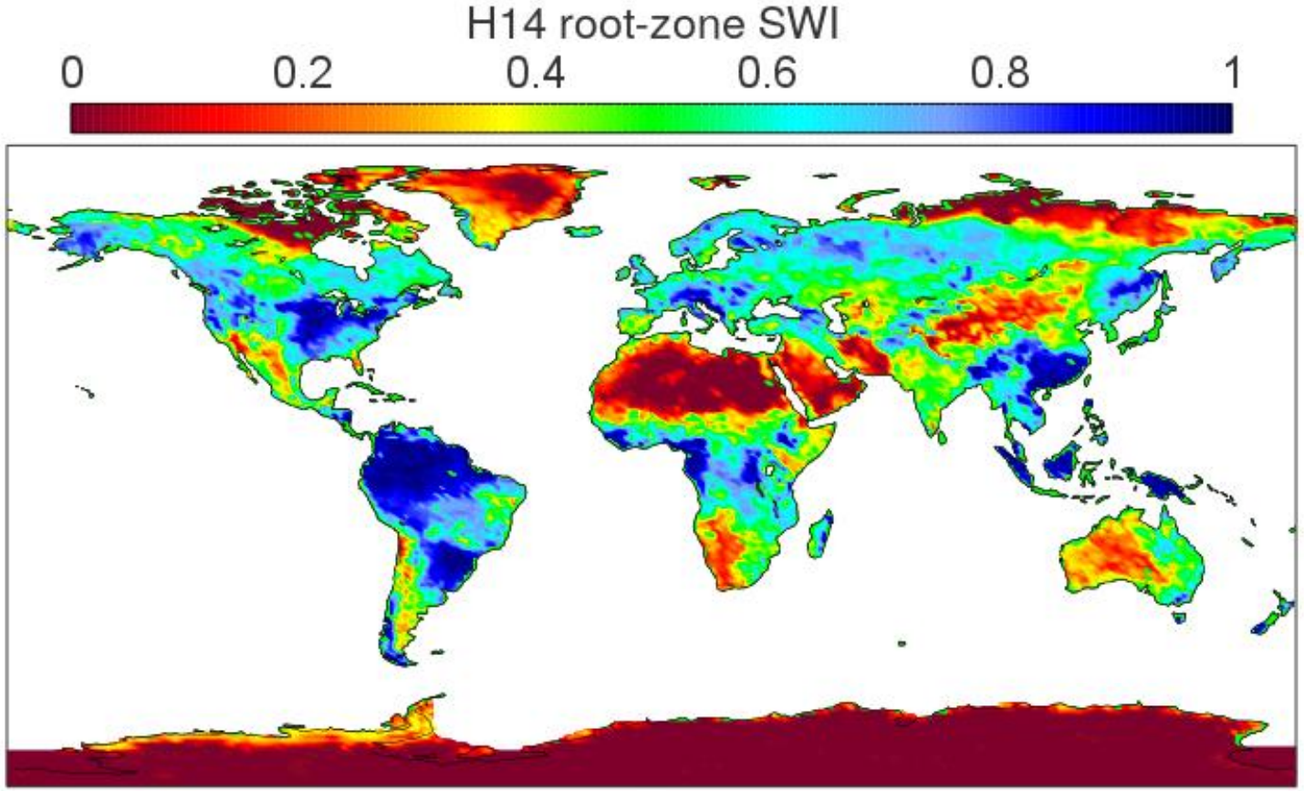
```
#Extract global domain [upper lon, lower lon, upper lat, lower lat]
area_view = mv.geoview(
    map_area_definition = 'corners',
    area = [-89,-179,89,179],
    coastlines = grey_land_shading
)
```

```
#Create title
title = mv.mtext(text_lines = ["<font size='1'>H14 root-zone SWI</font>"],
          text_justification = "centre",
          text_font_size = 0.5,
          text_colour = "charcoal")
```

```
#Create plot
mv.setoutput('jupyter')
#mv.setoutput(mv.png_output(output_name = 'H14_SWI_20181231')) # To save as png image
dw = mv.plot_superpage(pages = mv.mvl_regular_layout(area_view,1,1,1,1))
mv.plot(dw[0], RZSM, cont_pc, my_legend,title)
```

Plotting grib data in metview-python

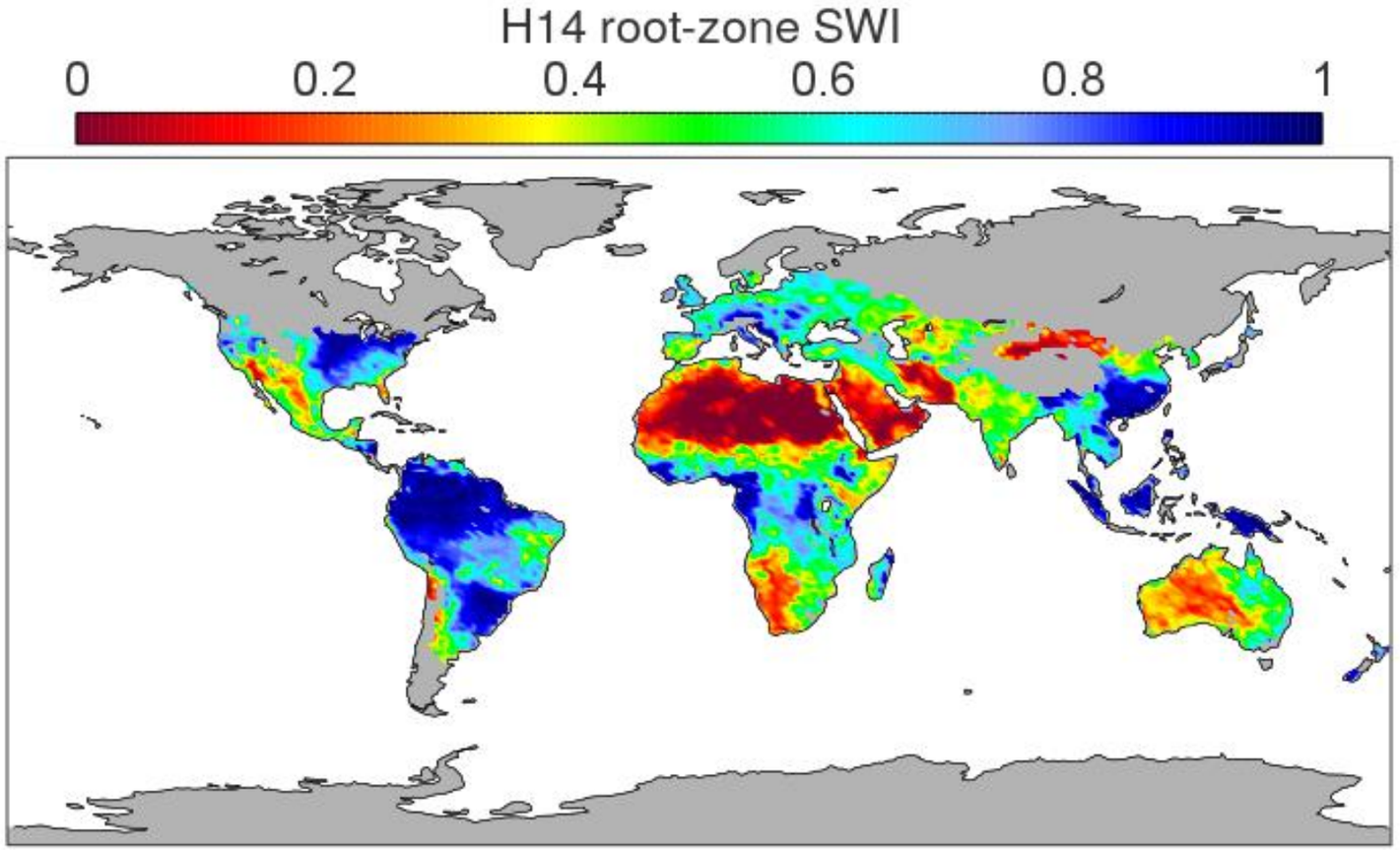
EUMETSAT
H SAF
SUPPORT TO OPERATIONAL HYDROLOGY AND WATER MANAGEMENT



Masking suspect data

EUMETSAT H SAF
SUPPORT TO OPERATIONAL HYDROLOGY AND WATER MANAGEMENT

```
#Mask suspect data
masked_data=RZSM.values(); masked_data[qc.values()>1.0]=np.nan
RZSM = RZSM.set_values(masked_data)
```



Conversion to regular grid

EUMETSAT H SAF
SUPPORT TO OPERATIONAL HYDROLOGY AND WATER MANAGEMENT

```
import matplotlib.pyplot as plt
import cartopy.crs as ccrs
import metview as mv

filename='h14_2019053000.grib'
SM = mv.read(filename)

SM1 = mv.read(data=SM,param='40.228') #1st layer (0-7 cm depth)
SM2 = mv.read(data=SM,param='41.228') #2nd layer (7-28 cm depth)
SM3 = mv.read(data=SM,param='42.228') #3rd layer (28-100 cm depth)

RZSM = SM1
RZSM = RZSM.set_values(SM1.values()*0.07 + SM2.values()*0.21 + SM3.values()*0.72) #Depth integration for RZSM

RZSM_ll = mv.read(data=RZSM,grid=[0.25,0.25]) #Convert grid to regular lat-lon of 0.25 degrees by 0.25 degrees
```

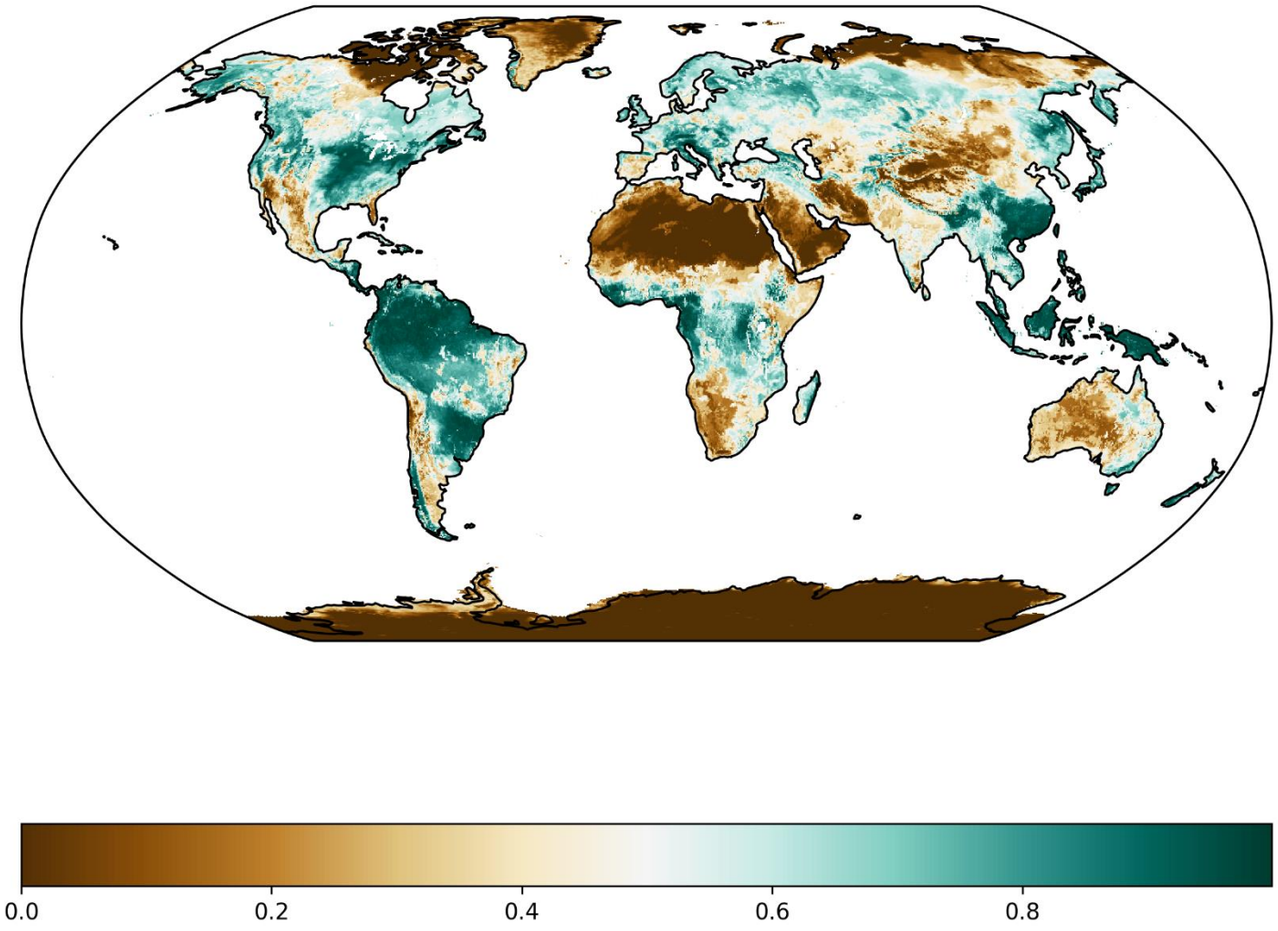
Plot with cartopy

EUMETSAT H SAF
SUPPORT TO OPERATIONAL HYDROLOGY AND WATER MANAGEMENT

```
[[Nj,Ni]] = mv.grib_get(RZSM_ll,['Nj','Ni']) #Extract number of points in lat/lon
Nj=int(Nj)
Ni=int(Ni)

fig = plt.figure(figsize=(10, 10))
cmap=plt.cm.get_cmap('BrBG') #Choose colour map type
RZSM_np = RZSM_ll.values().reshape(Nj,Ni)
ax = plt.axes(projection=ccrs.Robinson()) #Choose projection
ax.coastlines()
#Plot image with cartopy:
im = ax.imshow(RZSM_np,origin='upper',cmap=cmap,extent=[0.00001,360.,-90,90],transform=ccrs.PlateCarree())
plt.gcf().colorbar(im,ax=ax,orientation='horizontal'); plt.title("H14 root-zone SWI",fontsize=20)
#plt.savefig('Cartopy_plot.png',dpi=300,layout='tight') #To save in png format
plt.show()
plt.close()
```

Plot with cartopy

EUMETSAT H SAF
SUPPORT TO OPERATIONAL HYDROLOGY AND WATER MANAGEMENT



Plot time series over Europe

EUMETSAT H SAF
SUPPORT TO OPERATIONAL HYDROLOGY AND WATER MANAGEMENT

```
import metview as mv #for reading and plotting grib files
import numpy as np #for grid/data manipulation
import pandas as pd #Needed for time series
import pylab as pl #mainly for plotting

analysis_period = ['2001-01-01','2010-12-31']

data_range = pd.date_range(analysis_period[0], analysis_period[1])

area = [30,-10.87,60,30] #Europe domain with which to integrate [upper lon, lower lon, upper lat, lower lat]


SM_df_layer=pd.DataFrame(pl.empty((data_range.size),data_range)) #Pandas dataframe for time series
```

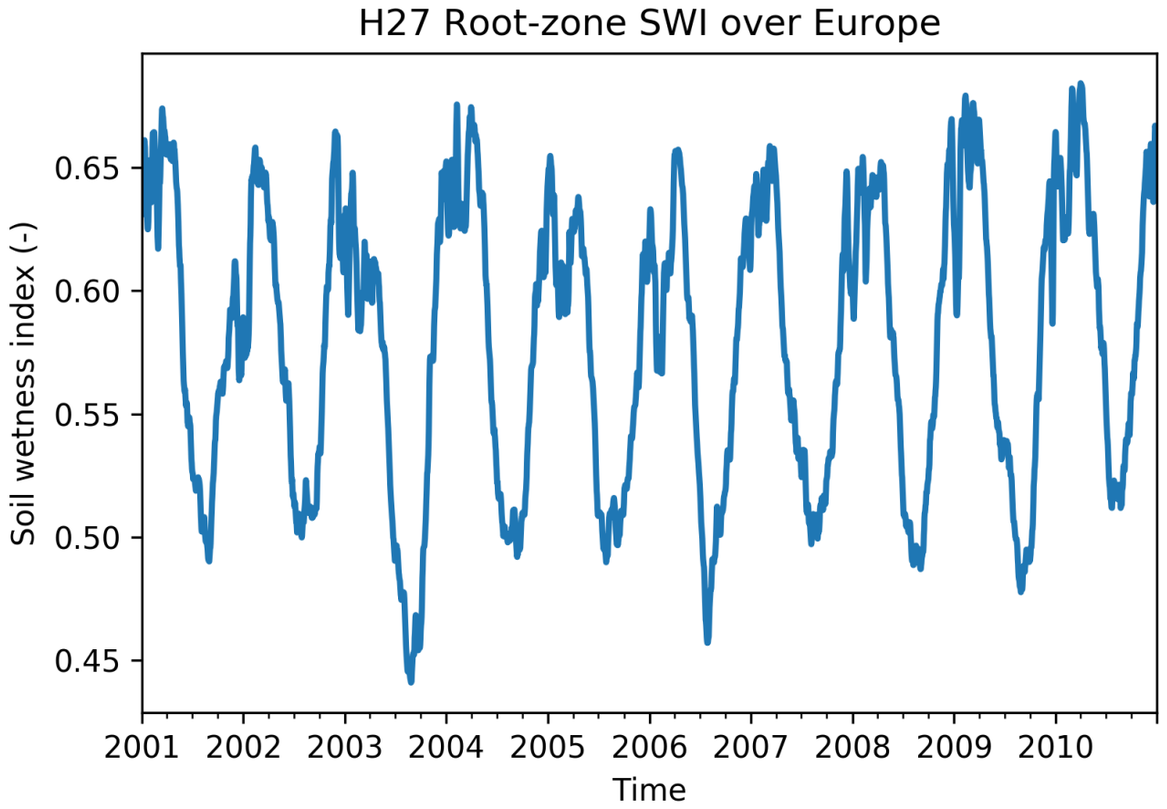
Plot time series over Europe

EUMETSAT H SAF
SUPPORT TO OPERATIONAL HYDROLOGY AND WATER MANAGEMENT

```
for days, d in zip(data_range,range(len(data_range))): #Loop over days in time series:

  SM = mv.read("/h27_"+str(days.year)+'%02d'%(days.month)+'%02d'%(days.day)+'00.grib')

  SM1 = mv.read(data=SM,param='40.228') #1st layer (0-7 cm depth)
  SM2 = mv.read(data=SM,param='41.228') #2nd layer (7-28 cm depth)
  SM3 = mv.read(data=SM,param='42.228') #3rd layer (28-100 cm depth)

  SM1_layer = mv.integrate(SM1, area) #Average soil moisture over European domain
  SM2_layer = mv.integrate(SM2, area)
  SM3_layer = mv.integrate(SM3, area)

  SM_df_layer.iloc[d] = (SM1_layer*0.07)+(SM2_layer*0.21)+(SM3_layer*0.72) #Depth integrated SWI
```

```
#Plot time series
SM_df_layer.index=data_range
SM_df_layer.iloc[:].plot(legend='False',linewidth=2.0)
pl.gca().get_legend().remove()
pl.xlabel('Time'); pl.ylabel('Soil wetness index (-)'); pl.title('H27 Root-zone SWI over Europe');
pl.savefig('H27_time_series.png',dpi=300,layout='tight')
```

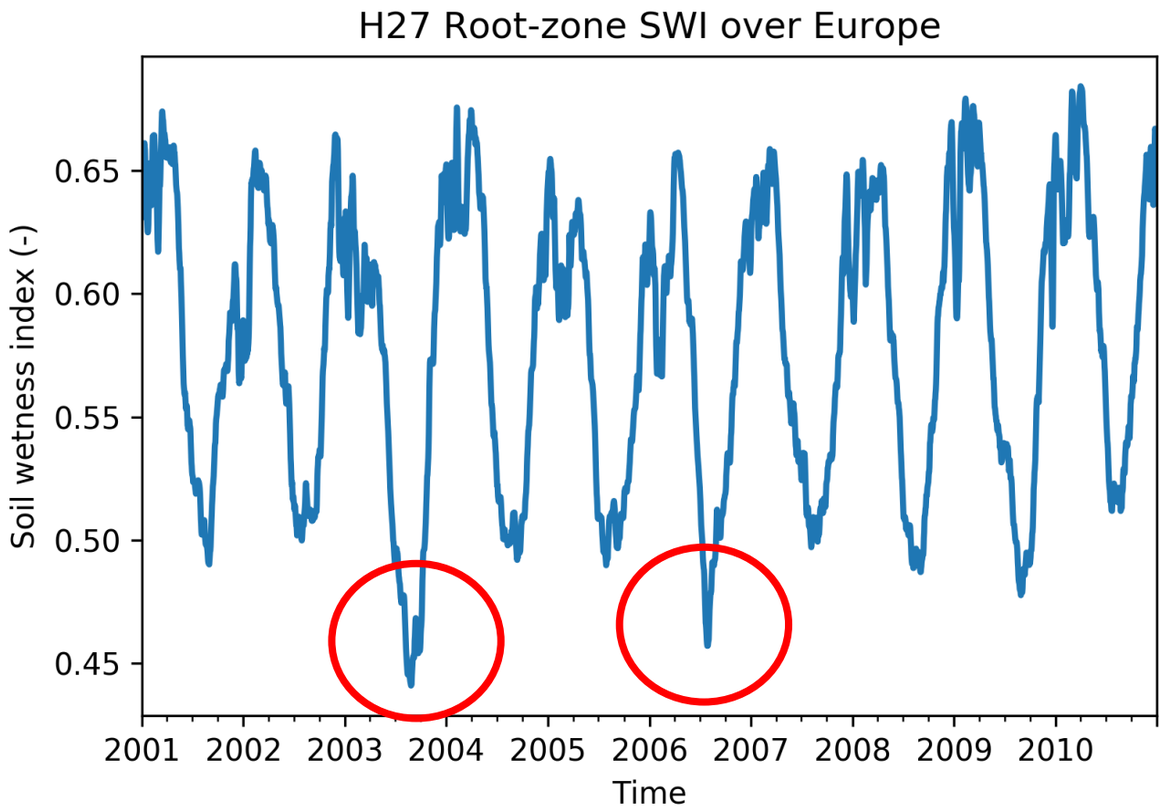
Plot time series over Europe




```
#To plot a time series for a particular gridpoint, simply use the "nearest_gridpoint" function
SM1_point = mv.nearest_gridpoint(SWI1, [60, 16]) #Value of SM1 at nearest point to 60 degrees East, 16 North
print(SM_1_point)
```

```
0.5507599115371704
```

Plot time series over Europe



Hot and dry summers of 2003 and 2006

```
#To plot a time series for a particular gridpoint, simply use the "nearest_gridpoint" function
SM1_point = mv.nearest_gridpoint(SWI1, [60, 16]) #Value of SM1 at nearest point to 60 degrees East, 16 North
print(SM_1_point)
```

```
0.5507599115371704
```

Convert grib to netCDF with regional data

EUMETSAT H SAF
SUPPORT TO OPERATIONAL HYDROLOGY AND WATER MANAGEMENT

- The following examples use the CDO tool, which is free to download from:
https://code.mpimet.mpg.de/projects/cdo

- Information about latitude and longitude points for reduced Gaussian grid:
https://www.ecmwf.int/en/forecasts/documentation-and-support/gaussian_n400

- For our grid (T799), there are 1600 regular latitude points (and 800 longitude points). We can now convert our reduced Gaussian grid to regular lat/lon:

```
module load cdo

#To convert to regular lat/lon
cdo -R remapcon,r1600x800 -setgridtype,regular h14_2019053000.grib h14_2019053000_r.grib
#Then convert the file from grib to netcdf:
cdo -f nc copy h14_2019053000_r.grib h14_2019053000.nc
#Then extract the Italian domain from the global netcdf file (lon0=5, lon1=19, lat0=36, lat1=48):
cdo -sellonlatbox,5,19,36,48 h14_2019053000.nc h14_2019053000_Italy.nc
#File is vastly reduced (h14_2019053000.nc ~ 25 mb, h14_2019053000_Italy.nc ~ 70 kb)!
```

Regional plot of netCDF

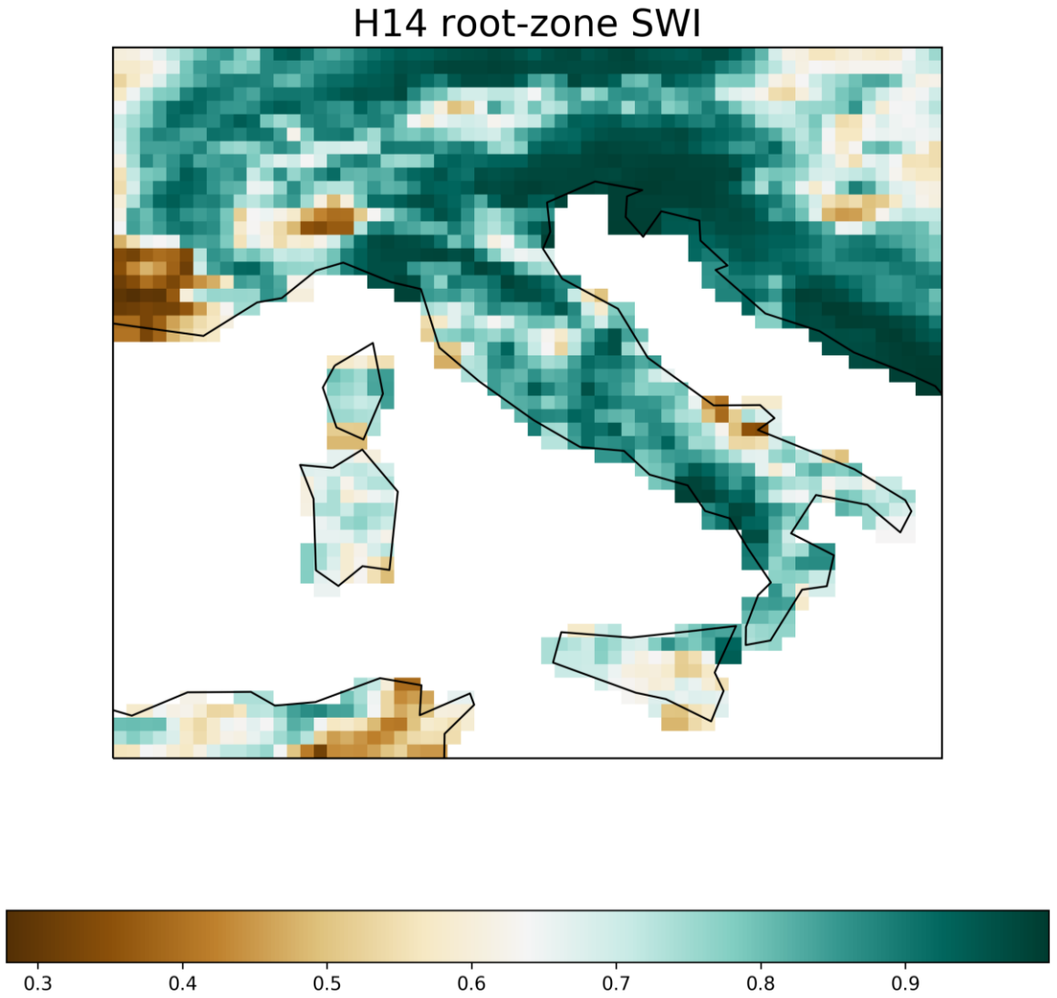
EUMETSAT H SAF
SUPPORT TO OPERATIONAL AND WATER NT

```
import matplotlib.pyplot as plt
import cartopy.crs as ccrs
import metview as mv
from netCDF4 import Dataset # For reading netCDF
```

```
SM="h14_2019053000_Italy.nc"
SM_data=Dataset(SM) #Read netCDF data

lon=SM_data.variables["lon"][:] #Read in lon coordinates
lat=SM_data.variables["lat"][:] #Read in lat coordinates

SWI_l1=SM_data.variables["var40"][:] #1st layer (0-7 cm depth)
SWI_l2=SM_data.variables["var41"][:] #2nd layer (7-28 cm depth)
SWI_l3=SM_data.variables["var42"][:] #3rd layer (28-100 cm depth)
SWI_l4=SM_data.variables["var43"][:] #4th layer (100-289 cm)

SWI_RZ=(SWI_l1*0.07) + (SWI_l2*0.21) + (SWI_l3*0.72) #Depth integrated SWI
print (np.shape(SWI_RZ))

fig = plt.figure(figsize=(10, 10))

cmap=plt.cm.get_cmap('BrBG')

ax = plt.axes(projection=ccrs.PlateCarree())
im = ax.imshow(SWI_RZ[0],cmap=cmap,extent=[5,19,36,48])
plt.gcf().colorbar(im,ax=ax,orientation='horizontal'); plt.title("H14 root-zone SWI",fontsize=20)
ax.coastlines()
plt.savefig('regional_netCDF.png',dpi=300,layout='tight')
```

Regional plot of netCDF

EUMETSAT H SAF
SUPPORT TO OPERATIONAL HYDROLOGY AND WATER MANAGEMENT



Summary

EUMETSAT H SAF
SUPPORT TO OPERATIONAL HYDROLOGY AND WATER MANAGEMENT

- There are 3 different root-zone SWI products:
  - H14: Global near-real-time 25 km resolution
  - H27: Global data record (1992-2014)
  - H140: Global data record (2015-2016)
- The metview library directly reads and plots grib data from its original non-Gaussian grid
- Metview has other features, including conversion to regular grid, nearest gridpoint extraction and integrating over geographical region
- The CDO tool can convert grib files to regular grids in netCDF format and extract local regions
- Cartopy is a good alternative to metview for regular grids

References

- Albergel, C., 2011 Product Validation Report (PVR-14). URL: http://hsaf.meteoam.it/documents/PVR/SAF_HSAF_PVR-14.pdf. Last accessed April 2018.
- E. Artinyan, 2012: Assimilation of small-scale surface soil moisture (H-SAF H08) into a coupled SVAT and hydrological model system. 2012 Eumetsat Meteorological satellite conference, 3-7 September, 2012, Sopot, Poland********* poster presentation
- Baguis, P. and Roulin, E., 2017. Soil Moisture Data Assimilation in a Hydrological Model: A Case Study in Belgium Using Large-Scale Satellite Data. Remote Sensing, 9(8), p.820.
- Dorigo, W. A., Wagner, W., Hohensinn, R., Hahn, S., Paulik, C., Xaver, A., Gruber, A., Drusch, M.,Mecklenburg, S., van Oevelen, P., Robock, A., and Jackson, T.: The International Soil MoistureNetwork: a data hosting facility for global in situ soil moisture measurements, *Hydrol. Earth Syst.Sci.*, 15, 1675-1698, https://doi.org/10.5194/hess-15-1675-2011,2011.
- D. Fairbairn, P. de Ronsay, and P. Browne, “The new stand-alone surface analysis at ECMWF: Implications for land-atmosphere DA coupling,” *J. Hydrometeor*, 2019. https://doi.org/10.1175/JHM-D-19-0074.1
- Koster, R.D., Sud, Y.C., Guo, Z., Dirmeyer, P.A., Bonan, G., Oleson, K.W., Chan, E., Verseghy, D., Cox, P., Davies, H. and Kowalczyk, E., 2006. GLACE: the global land–atmosphere coupling experiment. Part I: overview. Journal of Hydrometeorology, 7(4), pp.590-610.
- Massari, C., Brocca, L., Ciabatta, L., Moramarco, T., Gabellani, S., Albergel, C., De Rosnay, P., Puca, S. and Wagner, W., 2015. The use of H-SAF soil moisture products for operational hydrology: flood modelling over Italy. Hydrology, 2(1), pp.2-22.
- Struzik, P. and Kępińska-Kasprzak, M., 2016. Use of conventional and satellite data for estimation of evapotranspiration spatial and temporal pattern. Meteorology Hydrology and Water Management. Research and Operational Applications, 4.