



*Meteorologisk
institutt
met.no*

EMEP MSC-W NCL Plot Examples

EMEP MSC-W Model Training Course
AMVB <alvaro.valdebenito@met.no>

NCL: www.ncl.ucar.edu

<http://dx.doi.org/10.5065/D6WD3XH5>



NCL is an interpreted language designed specifically for scientific data analysis and visualization.

Portable, robust and free, NCL is available as binaries or open source

Supports netCDF3/4, GRIB1/2, HDF-SDS, HDF4-EOS, binary, shapefiles, and ASCII files

Numerous analysis functions are built-in

High quality graphics are easily created and customized with hundreds of graphic resources

Many example scripts and their corresponding graphics are available

Release Information

Current Version: 6.1.2
Release Date: February 2011

NCL version 6.1.2 released
Version 6.1.2 is now available on the ESG.

NCL has a DOI
Please see how to cite NCL in your publications.

©2013 UCAR | Privacy Policy | Terms of Use | Contact the Webmaster | Sponsored by NSF

- Interpreted language
- Many examples and already coded functions
- Behind most of EMEP MSC-W verification plots

Meteorological Input

ftp://ftp.met.no/projects/emep/mscw-oscr_ncl.tgz



- Read met files
 - 12 monthly fields
 - PS & LL projections
 - Plot SRF variables
 - surface_pressure
 - temperature_2m
 - Plot derived variables (mean 3 lower levels)
 - precipitation
 - wind vector
- mscw-osrc_ncl.tgz
- met_plot/**met_plot.ncl**
 - test_data/meteco2003_mets_elected_??_nc
 - emep_local/share/NCL/
 - emep_util.ncl
 - emep_util_io.ncl
 - emep_plot_setup.ncl

Meteorological Input

ftp://ftp.met.no/projects/emep/mscw-osrc_ncl.tgz



```
$ tar -xzf mscw-osrc_ncl.tgz
$ cd mscw-osrc_ncl/
$ export EMEPLOCAL=$PWD/emep_local
$ export TESTDATA=$PWD/test_data
# plot to screen (default)
$ ncl met_plot/met_plot.ncl
# plot to pdf file
$ ncl met_plot/met_plot.ncl \
    plot=True plot@type=\"pdf\"
```



Source code

```
1  load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_code.ncl"
2  load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/gsn_csm.ncl"
3  load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/contributed.ncl"
4  load "$EMEPLOCAL/share/NCL/emep_util.ncl"
5  load "$EMEPLOCAL/share/NCL/emep_util_io.ncl"
6  load "$EMEPLOCAL/share/NCL/emep_plot_setup.ncl"
7  ;-----
8  ; (if PS projection) u_wind and v_wind --> zonal and meridional winds
9  ;-----
10 procedure plot_fix_ps_wind(f:file,u[*][*]:numeric,v[*][*]:numeric)
    ■
    ■
    ■
29 ;-----
30 ; Paneled met. plots
31 ;-----
32 procedure met_plot(fname[1]:string,vname[1]:string,rec[*]:integer,lev[*]:integer,\
33                    wks[1]:graphic,plot[1]:logical)
    ■
    ■
    ■
```



Source code

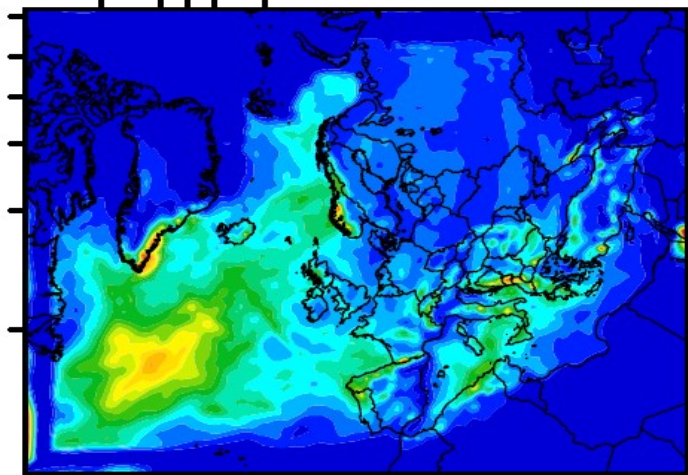
```
99 ;-----
100 ; Setup paneled met. plots
101 ;-----
102 procedure plot_met_12(fname[1]:string,plt[1]:logical)
   ■
   ■
   ■

131 ;-----
132 ; Main script
133 ;-----
134 TEST_DATA=getenv("TESTDATA")
135 begin
136 ;; Default plot settings
137   plot=True
138   set_attr(plot,"file"      ,"metdata")
139   set_attr(plot,"type"     ,"x11")
140   set_attr(plot,"panel_dims",(/4,3/))
141   set_attr(plot,"year"    ,"2003")
142   set_attr(plot,"title"   ,"Montly mean (daily)")
   ■
   ■
   ■
157 end
```

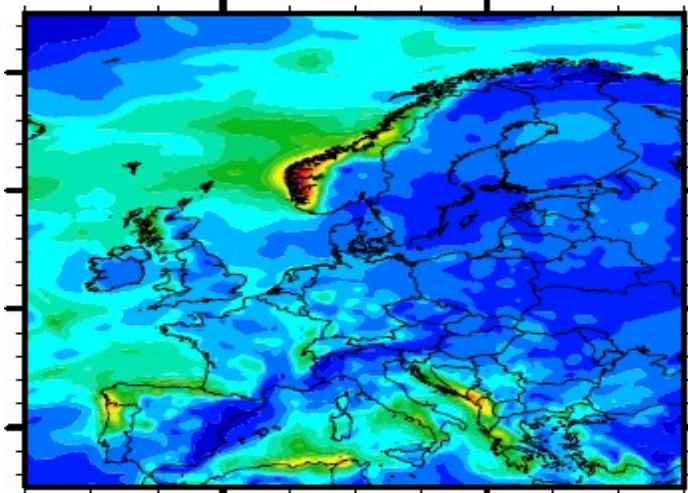


Precipitation

precipitation Jan 2003 mm



precipitation Jan 2003 mm



```
29 ; -----  
30 ; Paneled met. plots  
31 ; -----  
32 procedure met_plot(fname[1]:string,vname[1]:string,rec[*]:integer,lev[*]:integer,\  
33                      wks[1]:graphic,plot[1]:logical)
```

Average

Read

```
80 var=dim_avg_n_wrap(read_emep_met(fname,vname,rec(m),lev,plot),\  
81                      ispan(0,rank-3,1))
```

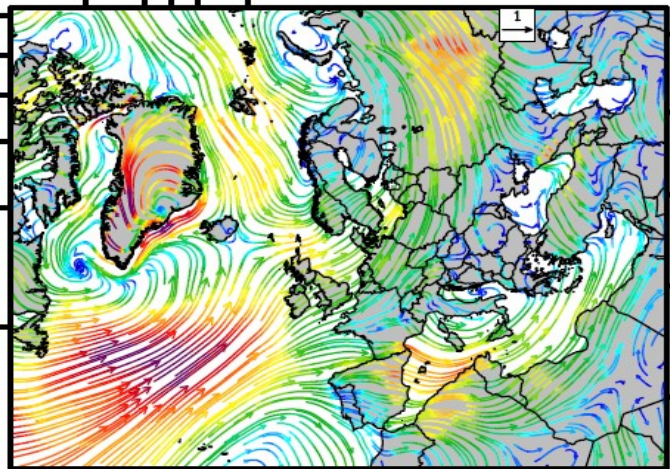
Plot

```
83 map(m)=gsn_csm_contour_map(wks,var,res)
```

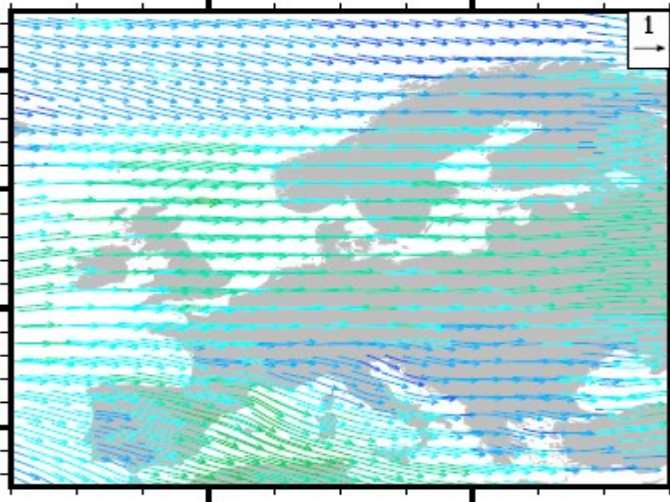


Wind vectors

wind speed Jan 2003 m/s



wind speed Jan 2003 m/s



```
29 ;-----  
30 ; Paneled met. plots  
31 ;-----  
32 procedure met_plot(fname[1]:string,vname[1]:string,rec[*]:integer,lev[*]:integer,\  
33                    wks[1]:graphic,plot[1]:logical)
```

Average Read

```
62 u=dim_avg_n_Wrap(read_emep_met(fname,"u_wind",rec(m),lev,plot),\  
63                 ispan(0,rank-3,1))  
64 v=dim_avg_n_Wrap(read_emep_met(fname,"v_wind",rec(m),lev,plot),\  
65                 ispan(0,rank-3,1))  
66 plot_fix_ps_wind(f,u,v)  
67 var=sqrt(u^2+v^2)
```

Plot

```
83 map(m)=gsn_csm_contour_map(wks,var,res)
```




Hourly Output

ftp://ftp.met.no/projects/emep/mscw-osrc_ncl.tgz

- Read hourly output
 - 8 hourly fields
 - LL projection
- SRF variables
 - NO₂ & PM₁₀
- Plot derived quantities
 - Daily max
 - Daily mean

mscw-os_ncl.tgz

- cwf_plot/**cwf_plot.ncl**
- test_data/CWF_20130416_hour.nc
- Stand alone example



Hourly Output

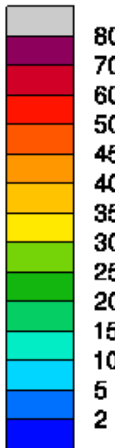
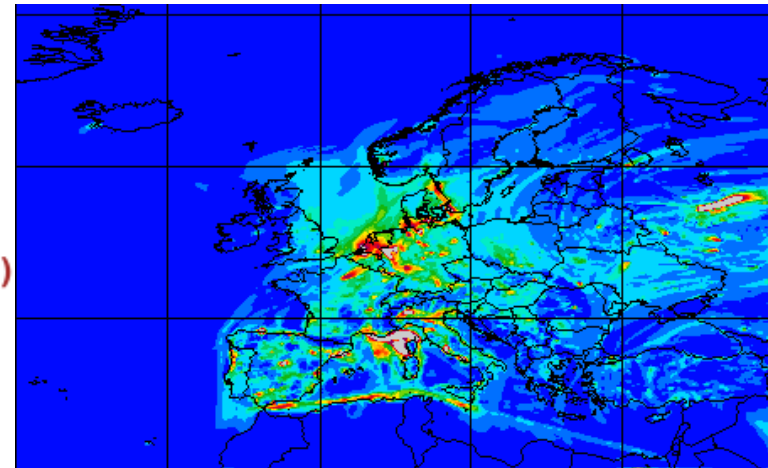
ftp://ftp.met.no/projects/emep/mscw-osrc_ncl.tgz

```
$ tar -xzf mscw-osrc_ncl.tgz
$ cd mscw-osrc_ncl/
$ export EMEPLOCAL=$PWD/emep_local
$ DATE=20130416
# plot to eps file
$ ncl cwf_plot/cwf_plot.ncl \
    date="\$DATE\" nday=1 \
    filename="\test_data/CWF_${DATE}_hour.nc\"
```

Surface Concentrations



```
19 ;*****
20 ; CWF-EMEP Read surface level from 3d variables
21 ;*****
22 if(isfilevar(f,"no2_3km")) then
23     N02 = f->no2_3km (:,0,,:)
24 end if
25 if(isfilevar(f,"pm10_3km")) then
26     PM10 = f->pm10_3km(:,0,,:)
27     if(isfilevar(f,"pm_h2o_3km")) then
28         PM10 = PM10 + f->pm_h2o_3km(:,0,,:)
29     end if
30 end if
■
■
■
```

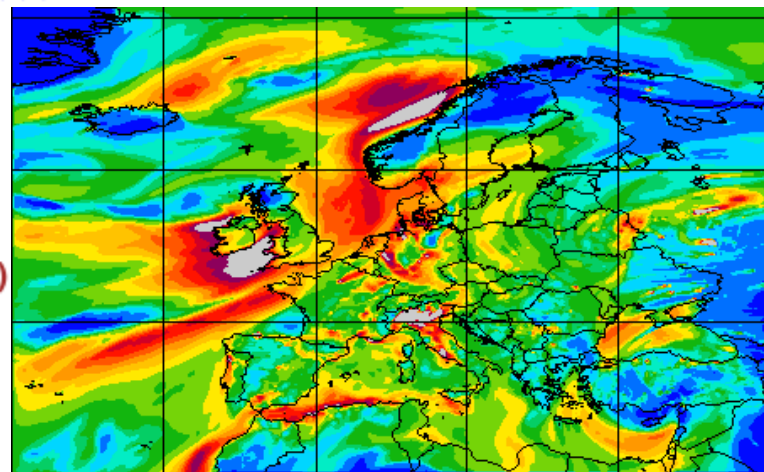


```
103 ;*****
104 ; N02 plot: daily maximum
105 ;*****
106 if(isvar("N02"))then
107     var = dim_max_n(N02(ii,::),0)
108     wks = gsn_open_wks ("eps", "no2_max_"+date+"_d"+iday)
109     gsn_define_colormap(wks, cmap)
110     res@cnLevels = (/2.0,5,10,15,20,25,30,35,40,45,50,60,70,80/)
111     map = gsn_contour_map(wks,var,res)
112 end if
```



Surface Concentrations

```
19 ;*****
20 ; CWF-EMEP Read surface level from 3d variables
21 ;*****
22 if(isfilevar(f,"no2_3km")) then
23     N02 = f->no2_3km (:,0,,:)
24 end if
25 if(isfilevar(f,"pm10_3km")) then
26     PM10 = f->pm10_3km(:,0,,:)
27     if(isfilevar(f,"pm_h2o_3km")) then
28         PM10 = PM10 + f->pm_h2o_3km(:,0,,:)
29     end if
30 end if
■
■
■
```



```
113 ;*****
114 ; PM10 plot: daily average
115 ;*****
116 if(isvar("PM10"))then
117     var = dim_avg_n(PM10(ii,::),0)
118     wks = gsn_open_wks ("eps", "pm10_avg_"+date+"_d"+iday)
119     gsn_define_colormap(wks, cmap)
120     res@cnLevels = (/2.0,4,6,8,10,15,20,25,30,35,40,45,50,55/)
121     map = gsn_contour_map(wks,var,res)
122 end if
```