

NCL

Regridding

using ESMF

Version: 2018/10/18



Contact: Karin Meier-Fleischer

Deutsches Klimarechenzentrum (DKRZ)
Bundesstrasse 45a
D-20146 Hamburg
Germany

Email: meier-fleischer@dkrz.de
<http://www.dkrz.de/>

Copyright: NCAR/DKRZ 2018

The NCAR Command Language (Version 6.5.0) [Software]. (2018).
Boulder, Colorado: UCAR/NCAR/CISL/VETS. <http://dx.doi.org/10.5065/D6WD3XH5>

<http://www.ncl.ucar.edu>

Content

1	Introduction	4
2	Grid Types	6
2.1	Rectilinear grid	6
2.2	Curvilinear grid	6
2.3	Unstructured grid	6
3	Regriding	7
3.1	Rectilinear grid – from coarse to higher resolution	7
3.2	Rectilinear to curvilinear grid	9
3.3	Curvilinear to rectilinear grid	11
3.4	Unstructured to rectilinear grid	13
3.5	Regrid to a given grid	15

1 Introduction

Regridding, remapping or interpolation of data values depending on the underlying grid means the computation method to transfer the data values to a new grid.

The upper figure on the right side shows two different rectilinear grids in one plot: a coarse rectilinear (black solid line) and a finer higher resolution rectilinear grid (red dashed line).

The plot in the middle figure displays a coarse rectilinear (red dashed line) and a curvilinear grid (black solid line).

And in the lower figure the difference between an unstructured grid (FESOM, black solid lines) and a rectilinear grid (T63, red dashed lines) is pictured.

NCL provides different functions to regrid the data using different methods. Here, we recommend to use the ESMF regridding software which is integrated in the NCL software package since version 6.1.0.

This document gives a brief outline about ESMF, further information can be found at the ESMF home page at <https://www.earthsystemcog.org/projects/esmf/>

'The [Earth System Modeling Framework](#) (ESMF) is "software for building and coupling weather, climate, and related models". The ESMF "ESMF_RegridWeightGen" tool has been incorporated into NCL for generating weights for interpolating (regridding) data from one grid to another.'

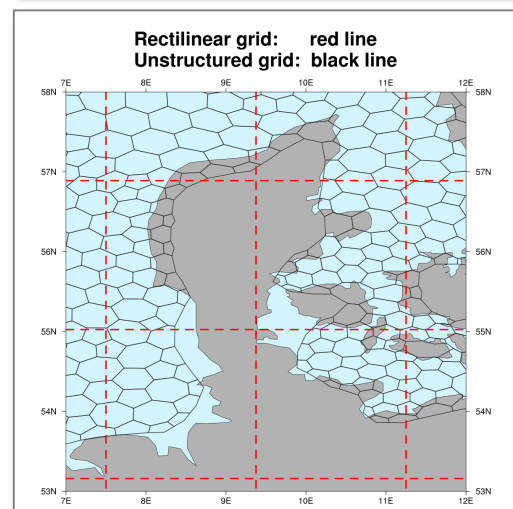
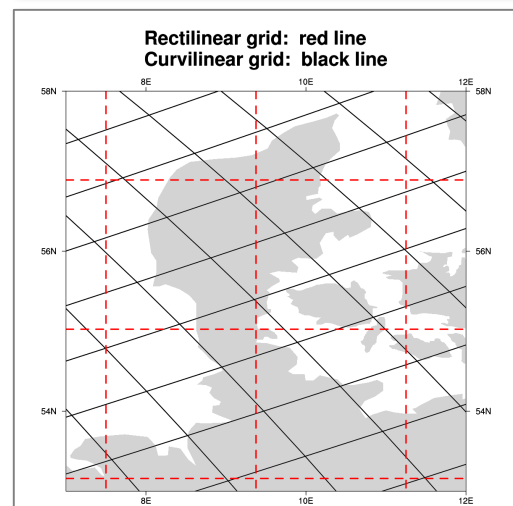
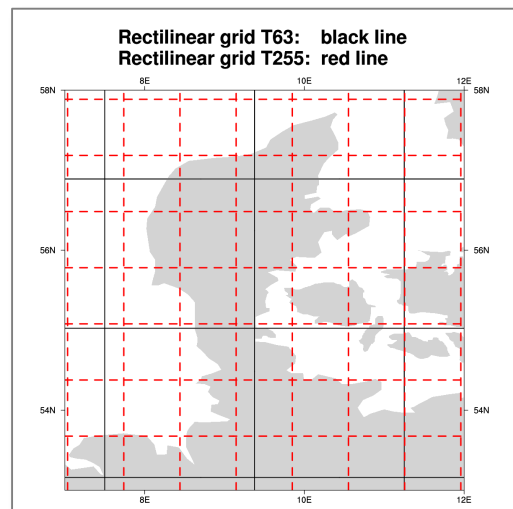
The basic steps of NCL/ESMF regridding involve:

1. Reading or generating the "source" grid.
2. Reading or generating the "destination" grid.
3. Creating special NetCDF files that describe these two grids.
4. Generating a NetCDF file that contains the weights. *
5. Applying the weights to data on the source grid, to interpolate the data to the destination grid.
6. Copying over any metadata to the newly regridded data.

* This is the most important step. Once you have a weights file, you can skip steps #1-4 if you are interpolating data on the same grids.

There are different regridding methods available and it is the decision of the user which one to choose: bilinear, patch, conservative or nearest-neighbor interpolation.

"bilinear" - the algorithm used by this application to generate the bilinear weights is the standard one found in many textbooks. Each destination point is mapped to a



location in the source mesh, the position of the destination point relative to the source points surrounding it is used to calculate the interpolation weights.

"patch" - this method is the ESMF version of a technique called "patch recovery" commonly used in finite element modeling. It typically results in better approximations to values and derivatives when compared to bilinear interpolation.

"conserve" - this method will typically have a larger interpolation error than the previous two methods, but will do a much better job of preserving the value of the integral of data between the source and destination grid.

The "conserve" method requires that the corners of the lat/lon grid be input. NCL will try to provide a guess at the corners if you don't provide them, but this could likely result in failure.

"neareststod" / "nearestdtos" - the nearest neighbor methods work by associating a point in one set with the closest point in another set. If two points are equally close then the point with the smallest index is arbitrarily used (i.e. the point with that would have the smallest index in the weight matrix).

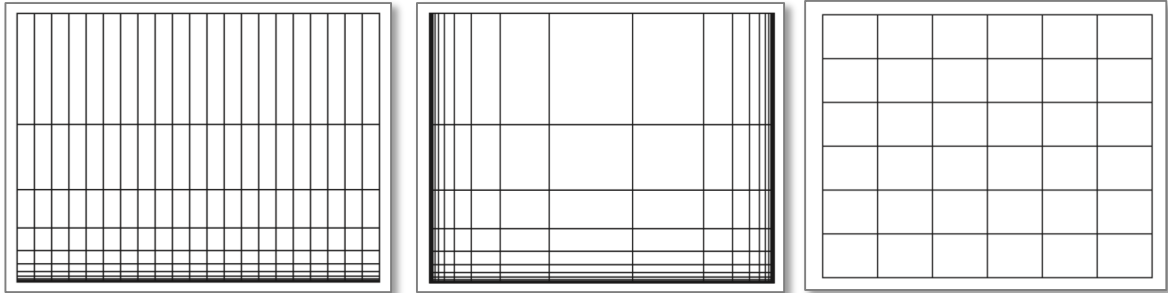
There are two versions of this type of interpolation available in the regrid weight generation application. One of these is the nearest source to destination method ("**neareststod**"). In this method each destination point is mapped to the closest source point. The other of these is the nearest destination to source method ("**nearestdtos**"). In this method each source point is mapped to the closest destination point.

See also: ESMF home <https://www.earthsystemcog.org/projects/esmf/>
 NCL ESMF <http://ncl.ucar.edu/Applications/ESMF.shtml>

2 Grid Types

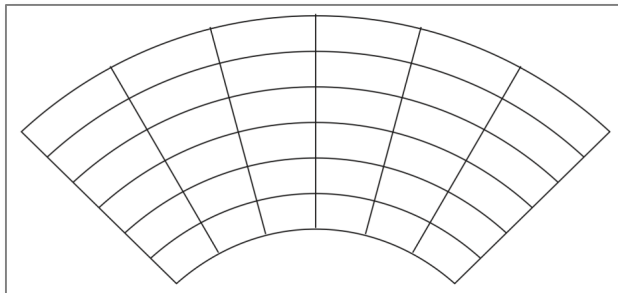
2.1 Rectilinear grid

A 2-dimensional rectilinear grid have parallel grid axes, the x-axis values are monotonic increasing and perpendicular to the monotonic-increasing y-axis values. The x- and y-axis coordinates are 1-dimensional $x(l)$ and $y(l)$.



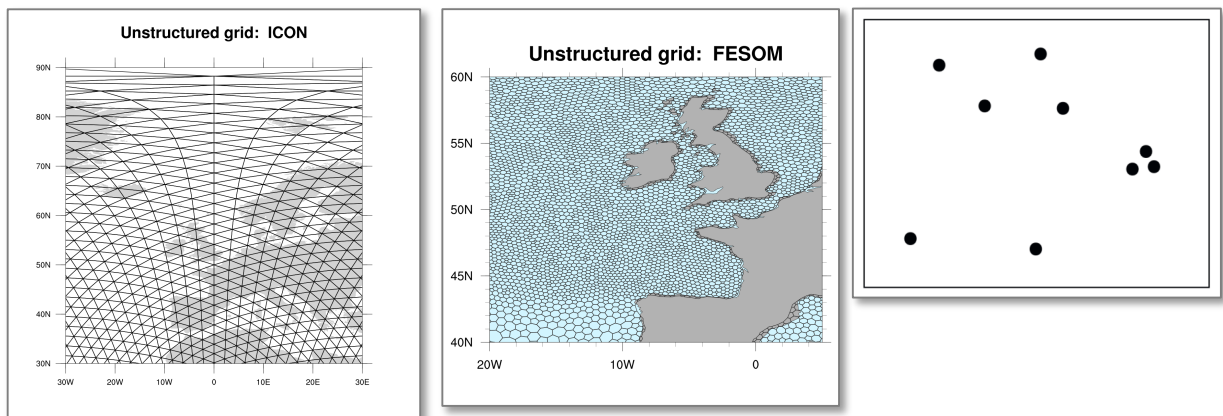
2.2 Curvilinear grid

A curvilinear grid is characterized by curved coordinate lines. The x- and y-axis coordinates are 2-dimensional $x(i,j)$ and $y(i,j)$.



2.3 Unstructured grid

An unstructured or irregular grid can have shapes such as triangle or tetrahedral in an irregular pattern defined by latitude and longitude vertices and the number of vertices for each cell. It also can represent irregular distributed points (point(x,y)).



3.2 Rectilinear to curvilinear grid

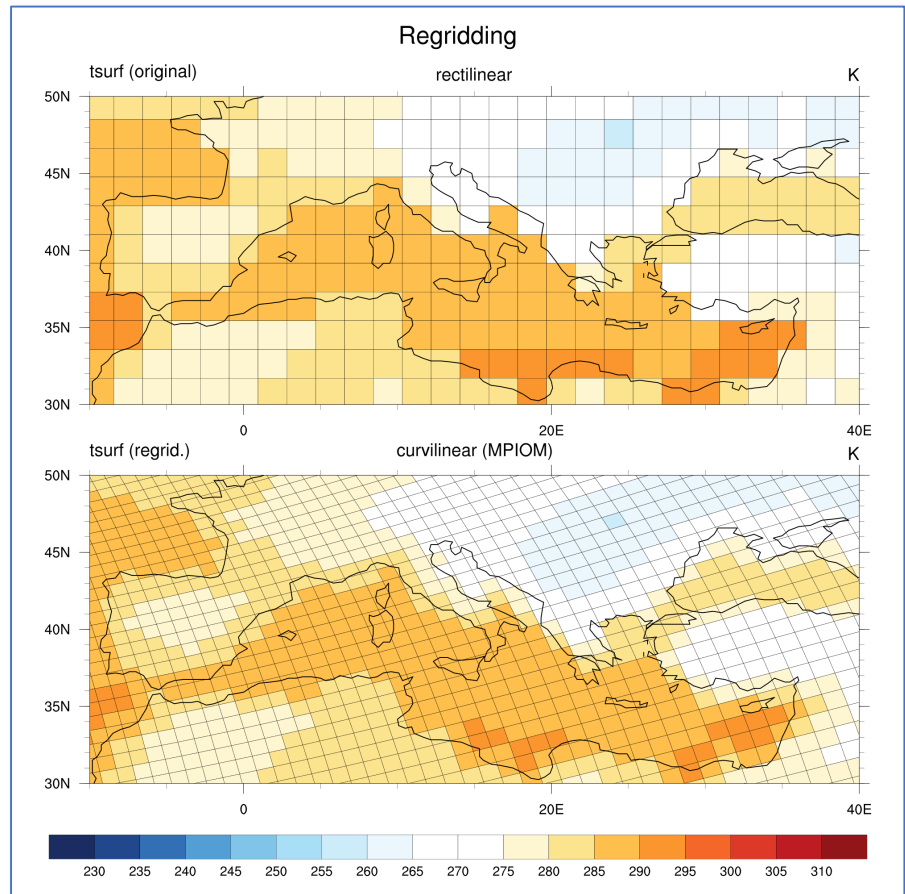
Sometimes for comparison purposes it is necessary to regrid the data variables on an underlying rectilinear grid to a curvilinear grid which is demonstrated in the next example. The input coordinates latitude and longitude are 1-dimensional arrays there again the output curvilinear coordinates are 2-dimensional arrays.

Original grid: T63

- rectilinear
- lat 96 x lon 192
- 1.875°

Output grid: MPIOM

- curvilinear
- y = 220
- x = 256
- lat(y, x)
- lon(y, x)



NCL example script:

```
load "$NCARG_ROOT/lib/ncarg/nclscripts/esmf/ESMF_regridding.ncl"

begin
  fili      = "rectilinear_grid_2D.nc"           ;-- data file name
  f         = addfile(fili,"r")                 ;-- open data file
  var       = f->tsurf(0,::,:)                  ;-- read variable

  ;-- read the destination grid lat/lon arrays
  dstfili   = "curvilinear_ocean.nc"           ;-- destination grid file
  d         = addfile(diri+dstfili,"r")        ;-- open grid file
  dst_lat   = d->lat                             ;-- 2D latitudes
  dst_lon   = d->lon                             ;-- 2D longitudes
  dims      = dimsizes(dst_lat)                 ;-- size of lat/lon
  nlat     = dims(0)                             ;-- number of latitudes
  nlon     = dims(1)                             ;-- number of longitudes

  ;-- set ESMF resources
  Opt       = True
  Opt@InterpMethod = "bilinear"                 ;-- interpolation method
  Opt@SrcFileName   = "T63_SCRIP_bilinear.nc"   ;-- source file name
```

```

Opt@DstFileName      = "WorldCurvilinear_SCRIP_bilin.nc" ;-- destination file
Opt@WgtFileName      = "T63toWorldCurvilinear_bilinear.nc"
                    ;-- name of weights file, which will be generated
Opt@ForceOverwrite   = True                               ;-- force overwrite
Opt@DstMask2D        = where(ismissing(dvar),0,1)          ;-- create mask from
                    ;-- destination
Opt@DstGridType      = "curvilinear"                      ;-- destination grid
Opt@DstTitle         = "World Grid Curvilinear Resolution bilinear"
                    ;-- destination title
Opt@DstGridLon       = dst_lon                            ;-- set destination lon
Opt@DstGridLat       = dst_lat                            ;-- set destination lat

;-- call ESMF_regrid
var_regrid = ESMF_regrid(var,Opt)                        ;-- regrid variable
var_regrid!0 = "y"                                       ;-- named coordinate dimension 0
var_regrid!1 = "x"                                       ;-- named coordinate dimension 1

delete_VarAtts(var_regrid,(/"lat2d", "lon2d"/)) ;-- delete attributes

;-- assign a output netcdf file for the new regridded data name of output file
ofile = "regridd_rectilin_to_curvilinear_bilin_wgts_destgrid_ESMF.nc"
system("rm -rf "+ofile) ;-- delete ofile if it exists
fout = addfile(ofile, "c") ;-- open new file

;-- start to define output file settings
setfileoption(fout,"DefineMode",True) ;-- explicitly declare file def mode

;-- create global attributes of output file
fAtt = True ;-- assign file attributes
fAtt@Conventions = "CF-1.5"
fAtt@title = "Regrid T63 to curvilinear grid (MPIOM)"
fAtt@source_file = fili
fAtt@creation_date = systemfunc ("date")
fAtt@history = \
    "ncl NCL_Advanced_regrid_rectilin_to_curvilinear_bilin_wgts_ESMF.ncl"
fileattdef(fout,fAtt) ;-- copy file attributes

;-- predefine the coordinate variables and their dimensionality
dimNames = (/ "y", "x" /) ;-- curvilinear grid: dimensions not lat/lon
dimSizes = (/ nlat, nlon /) ;-- dimension size of destination y/x
dimUnlim = (/ False, False /)
filedimdef(fout,dimNames,dimSizes,dimUnlim) ;-- define dimensions

;-- predefine the the dimensionality of the variables to be written out
filevardef(fout,"lat",typeof(dst_lat),getvardims(dst_lat)) ;-- variable lat
filevardef(fout,"lon",typeof(dst_lon),getvardims(dst_lon)) ;-- variable lon
filevardef(fout,"var",typeof(var_regrid),getvardims(var_regrid))
;-- copy attributes associated with var to output file
;-- copy variable attributes
filevarattdef(fout,"lat", dst_lat) ;-- copy attributes from destination lat
filevarattdef(fout,"lon", dst_lon) ;-- copy attributes from destination lon
filevarattdef(fout,"var", var_regrid) ;-- copy var_regrid attributes

;-- explicitly exit file definition mode (not required)
setfileoption(fout,"DefineMode",False)

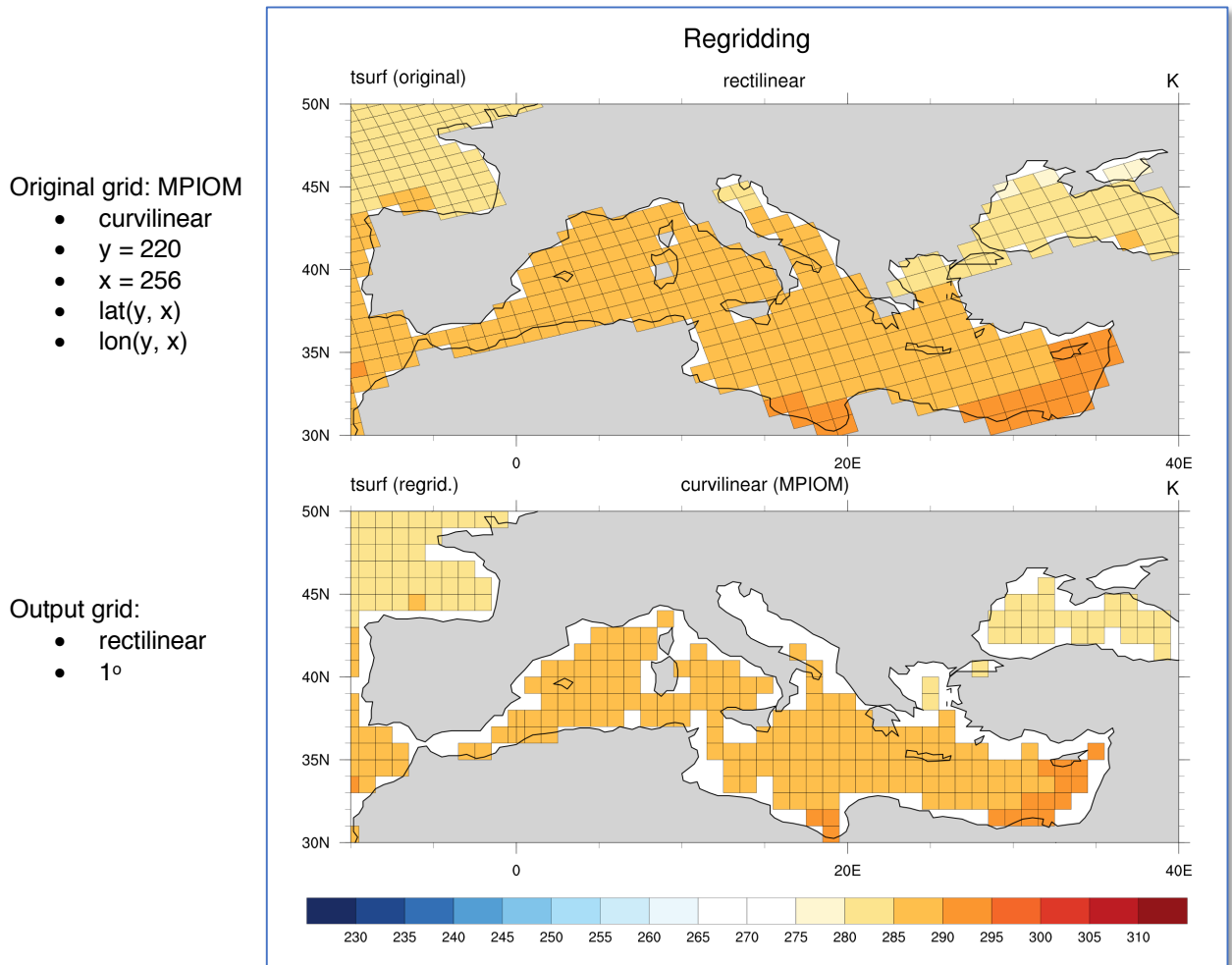
;-- output only the data values since the dimensionality and such have been
;-- predefined; the "(/ /)" syntax tells NCL to only output the data values
fout->var = (/var_regrid/) ;-- write variable to new netCDF file
fout->lat = (/dst_lat/) ;-- write lat to new netCDF file
fout->lon = (/dst_lon/) ;-- write lon to new netCDF file

end

```

3.3 Curvilinear to rectilinear grid

Also, the other way around is important. The next example shows how to regrid variables from an underlying curvilinear grid to a 1° rectilinear grid.



NCL example script:

```
load "$NCARG_ROOT/lib/ncarg/nclscripts/esmf/ESMF_regridding.ncl"

begin
  fili = "curvilinear_ocean.nc"

  sfile      = addfile(fili,"r")      ;-- open data file
  thetao    = sfile->thetao(0,0,::,:) ;-- read variable
  thetao@lat2d = sfile->lat          ;-- tells ESMF what kind of source
                                       ;-- grid
  thetao@lon2d = sfile->lon          ;-- tells ESMF what kind of source
                                       ;-- grid

  ;-- set resources
  Opt      = True
  Opt@InterpMethod = "bilinear"      ;-- interpolation method
  Opt@SrcFileName  = "Curvilin_SCRIP_bilinear.nc" ;-- source file name
  Opt@DstFileName  = "World1deg_SCRIP_bilinear.nc" ;-- destination file
  Opt@WgtFileName  = "CurvilitoWORLD_1x1_bilinear.nc" ;-- name of weights file, which will be generated
  Opt@ForceOverwrite = True          ;-- force overwrite
end
```

```

Opt@SrcMask2D      = where(.not. ismissing(thetao),1,0) ;-- if data contains
                                     ;-- missing values
Opt@DstGridType    = "1deg"                ;-- Destination grid
Opt@DstTitle       = "World Grid 1x1-degree Resolution bilinear"
                                     ;-- destination title
Opt@DstLLCorner    = (/ -89.5d,  0.0d /)    ;-- destination lower
                                     ;-- left corner
Opt@DstURCorner    = (/ 89.5d, 359.0d /)    ;-- destination upper
                                     ;-- right corner
;-- call ESMF_regrid
thetao_regrid = ESMF_regrid(thetao,Opt)

nlon = dimsizes(thetao_regrid&lon)
nlat = dimsizes(thetao_regrid&lat)

;-- assign a output netcdf file for the new regrided data;
;-- (npoints = 180x360)
ofile = "regridded_curvilinear_bilinear_1x1deg_thetao_ESMF.nc"
system("rm -rf "+ofile)                ;-- delete file if exists
fout = addfile(ofile,"c")                ;-- open new file

;-- start to define output file settings
setfileoption(fout,"DefineMode",True)
                                     ;-- explicitly declare file definition mode

;-- create global attributes of the file
fAtt      = True                        ;-- assign file attributes
fAtt@Conventions = "CF-1.5"
fAtt@project_id = "DKRZ NCL Advanced Workshop - Regridding"
fAtt@source_file = fili
fAtt@creation_date = systemfunc ("date")
fAtt@history = \
    "ncl NCL_Advanced_regrid_curvilinear_to_rectilinear_weights_ESMF.ncl"
fileattdef(fout,fAtt)                ;-- copy file attributes

;-- predefine the coordinate variables and their dimensionality
dimNames = (/ "lat", "lon" /)
dimSizes = (/ nlat, nlon /)
dimUnlim = (/ False, False /)
filedimdef(fout,dimNames,dimSizes,dimUnlim)                ;-- define dimensions

;-- predefine the the dimensionality of the variables to be written out
filevardef(fout,"lat",typeof(thetao_regrid&lat),getvardims(thetao_regrid&lat))
filevardef(fout,"lon",typeof(thetao_regrid&lon),getvardims(thetao_regrid&lon))
filevardef(fout,"thetao",typeof(thetao_regrid),getvardims(thetao_regrid))

;-- copy attributes associated with each variable to the file
filevarattdef(fout,"lat",thetao_regrid&lat)                ;-- copy lat attributes
filevarattdef(fout,"lon",thetao_regrid&lon)                ;-- copy lon attributes
filevarattdef(fout,"thetao",thetao_regrid)
                                     ;-- copy thetao_regrid attributes

;-- explicitly exit file definition mode (not required)
setfileoption(fout,"DefineMode",False)

;-- output only the data values since the dimensionality and such have been
;-- predefined; the "(/ /)" syntax tells NCL to only output the data values

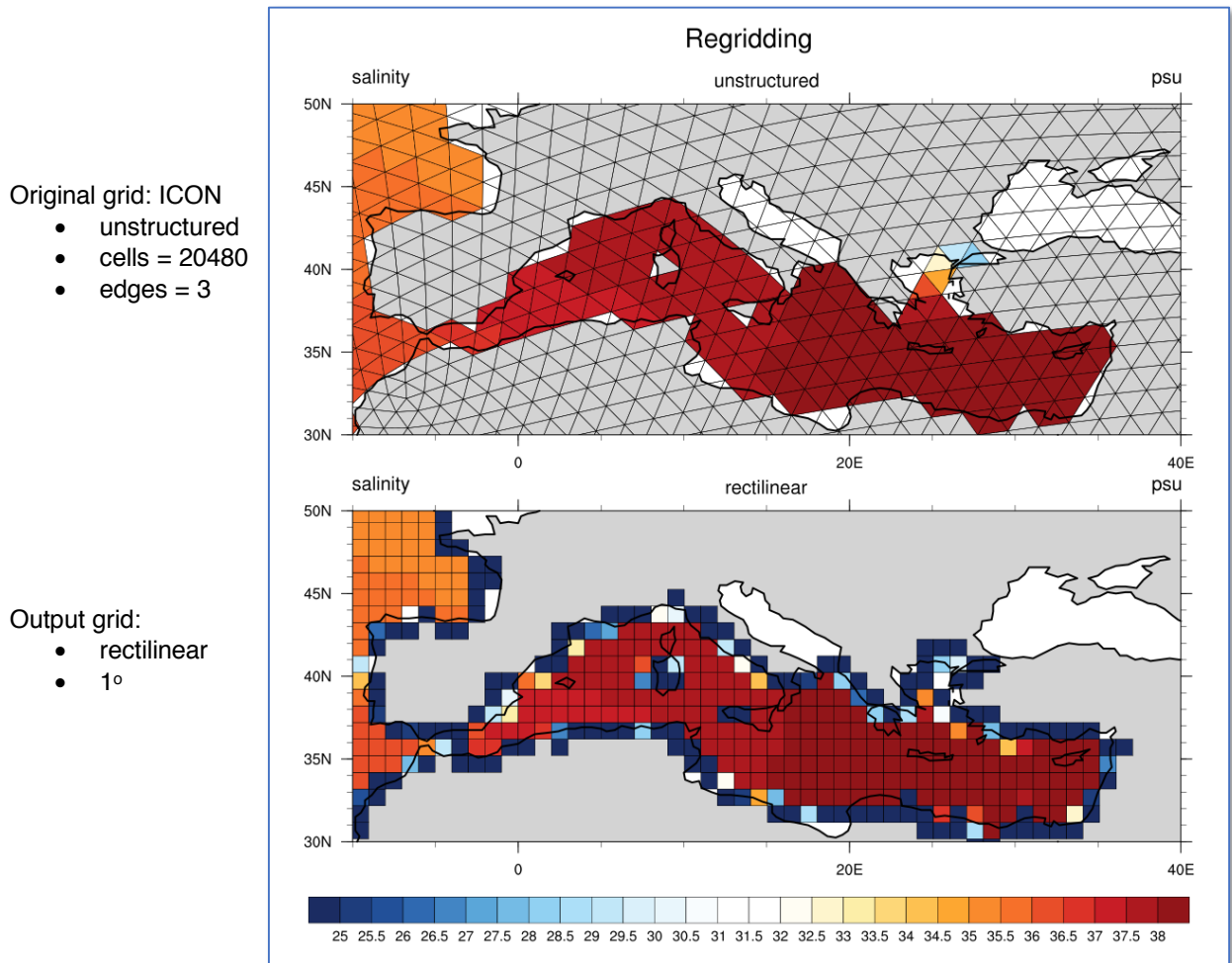
fout->thetao = (/thetao_regrid/)                ;-- write variable to new netCDF file
fout->lat     = (/thetao_regrid&lat/)            ;-- write lat to new netCDF file
fout->lon     = (/thetao_regrid&lon/)            ;-- write lon to new netCDF file

end

```

3.4 Unstructured to rectilinear grid

Unstructured data like ICON or FESOM model output can be easily interpolated to a rectilinear grid with the following example script.



NCL example script:

```
load "$NCARG_ROOT/lib/ncarg/nclscripts/esmf/ESMF_regridding.ncl"

begin
  rad2deg = 45./atan(1.) ;-- radians to degrees

  f      = addfile("triangular_grid_ICON.nc","r")
  var    = f->S(time|0,depth|0,ncells|:) ;-- read variable

  x      = f->clon * rad2deg ;-- cell center, lon
  y      = f->clat * rad2deg ;-- cell center, lat
  x!0    = "lon" ;-- set named dimension lon
  y!0    = "lat" ;-- set named dimension lat
  x@units = "degrees_east" ;-- set lon units
  y@units = "degrees_north" ;-- set lat units

  ;-- set ESMF resources
  Opt      = True
  Opt@InterpMethod = "bilinear" ;-- interpolation method
  Opt@ForceOverwrite = True ;-- force overwrite
  Opt@SrcFileName = "Unstruct_SCRIP_bilinear.nc";-- source file name
```

```

Opt@SrcInputFileName = diri+fili                ;-- optional, but good idea
Opt@SrcRegional      = False
Opt@SrcGridLat       = y
Opt@SrcGridLon        = x
Opt@WgtFileName      = "UnstructtoWORLD_1x1_bilinear.nc"
                                     ;-- name of weights file
Opt@DstFileName      = "Worldldeg_SCRIP_bilinear.nc" ;-- destination file name
Opt@DstGridType      = "rectilinear"                ;-- destination grid
Opt@DstTitle         = "World Grid 1x1-degree Resolution bilinear"
                                     ;-- dest. title

Opt@DstRegional      = False
Opt@DstGridLon       = fspan(-180.,180.,360)
Opt@DstGridLat       = fspan(-90.,90.,180)          ;-- call ESMF_regrid

var_regrid = ESMF_regrid(var,Opt)                ;-- do the regridding
nlon = dimsizes(var_regrid&lon)                  ;-- dim size new lon
nlat = dimsizes(var_regrid&lat)                  ;-- dim size new lat

;-- assign a output netcdf file for the new regridded data (npoints = 180x360)
system("rm -rf regridded_unstructured_to_rectilinear_bilinear_ESMF.nc")
fout = addfile("regrid_unstructured_to_rectilin_bilin_ESMF.nc", "c")

;-- start to define output file settings
setfileoption(fout,"DefineMode",True)           ;-- explicitly declare file def. mode

;-- create global attributes of the file
fAtt          = True                               ;-- assign file attributes
fAtt@Conventions = "CF-1.5"
fAtt@source_file = fili
fAtt@creation_date = systemfunc ("date")
fAtt@history = \
    "ncl NCL_Advanced_regrid_unstruct_to_rectilin_bilinear_wgts_ESMF.ncl"
fileattdef(fout,fAtt)                             ;-- copy file attributes

;-- predefine the coordinate variables and their dimensionality
dimNames = (/ "lat", "lon" /)
dimSizes = (/ nlat, nlon /)
dimUnlim = (/ False, False /)
filedimdef(fout,dimNames,dimSizes,dimUnlim)

;-- predefine the the dimensionality of the variables to be written out
filevardef(fout, "lat", typeof(var_regrid&lat), getvardims(var_regrid&lat))
filevardef(fout, "lon", typeof(var_regrid&lon), getvardims(var_regrid&lon))
filevardef(fout, "S",  typeof(var_regrid),      getvardims(var_regrid))

;-- copy attributes associated with each variable to the file
filevarattdef(fout,"lat", var_regrid&lat)         ;-- copy lat attributes
filevarattdef(fout,"lon", var_regrid&lon)         ;-- copy lon attributes
filevarattdef(fout,"S",  var_regrid)             ;-- copy var_regrid attributes

;-- explicitly exit file definition mode (not required)
setfileoption(fout,"DefineMode",False)

;-- output only the data values since the dimensionality and such have
;-- been predefined; the "( / )" syntax tells NCL to only output the
;-- data values to the predefined locations on the file.
fout->S = (/var_regrid/)                          ;-- write variable to new netCDF file
fout->lat = (/var_regrid&lat/)                     ;-- write lat to new netCDF file
fout->lon = (/var_regrid&lon/)                     ;-- write lon to new netCDF file

end

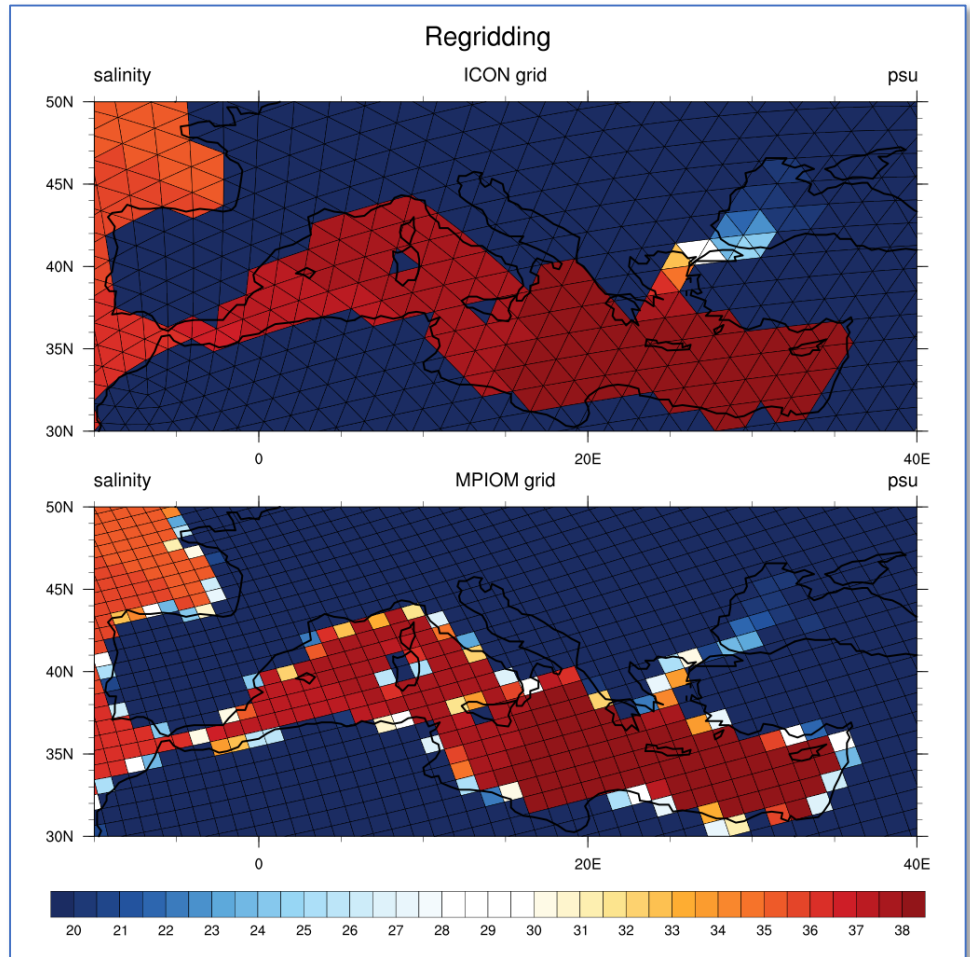
```

3.5 Regrid to a given grid

To compare two different data sets in most cases it is required to have both data sets on the same grid. How to interpolate one data file from its underlying grid to the grid of the other file is shown by the next example script.

- Original grid: ICON
- unstructured
 - cells = 20480
 - edges = 3

- Output grid: MPIOM
- curvilinear



NCL example script:

```
load "$NCARG_ROOT/lib/ncarg/nclscripts/esmf/ESMF_regridding.ncl"

begin
  rad2deg = 45./atan(1.) ;-- radians to degrees

  diri = "../..data/" ;-- data directory
  fili = "triangular_grid_ICON.nc" ;-- unstructured data file
  grid = "curvilinear_ocean.nc" ;-- use grid from file
  ofile = "regrid_rectilin_bilin_ICON_S_ESMF_destgrid.nc" ;-- output file

;-- read destination grid data
  g = addfile(diri+grid,"r") ;-- open data file
  lat2d = g->lat ;-- read 2D latitudes
  lon2d = g->lon ;-- read 2D longitudes

;-- read data
  f = addfile(diri+fili,"r")
  var = f->S(time|0,depth|0,ncells|:)
  ;-- set variable with dims: (time,depth,ncells)
```

