

EUROCLIMA+

# Generación de escenarios climáticos en Centroamérica

Curso de capacitación sobre extracción y utilización de datos CORDEX

Utilidades de manipulación e interpolación de ficheros

NetCDF

Antonio Ángel Serrano de la Torre

25 de febrero al 1 de marzo de 2019

San Salvador



Financiado por la Unión Europea



Scale 1:12,500,000  
Lambert Conformal Conic Projection, 9°N and 17°N  
0 Kilometers  
100 300 Miles

# Tareas comunes con Netcdf

- Creación. Escribo un NetCDF a disco.
- Exploración. Mirar qué información hay dentro de un NetCDF
- Modificación. P.ej.: modificar algún atributo.
- Operar con sus datos.
- Exportarlo a ascii.

# Creación de ficheros Netcdf

- He generado unos datos como resultados de ciertos cálculos que he hecho y los quiero guardar a disco.
- Puedo guardarlos a disco utilizando:
  - el mismo lenguaje de programación con el que he hecho los cálculos, p. ej.: R, c, Fortran, Python.  
Todos estos lenguajes utilizan las api de NetCDF, que tienen que estar instaladas en el sistema.
  - utilidades en línea de comandos: ncgen, cdo, etc. Aquí veremos el ncgen, aunque con fines más pedagógicos que prácticos.
- Es más práctico usar el lenguaje de programación con el que he creado los datos, que una utilidad, pero la utilidad puede ser más pedagógica.

# Exploración de ficheros Netcdf

- Me han dado unos datos en NetCDF y tengo que conocer cómo están organizados por dentro.
- Puedo explorarlos utilizando:
  - el mismo lenguaje de programación con el que pienso hacer cálculos con los datos, p. ej.: R, c, Fortran, Python.  
Todos estos lenguajes utilizan las api de NetCDF, que tienen que estar instaladas en el sistema.
  - utilidades en línea de comandos: ncdump, cdo, nco, etc.
  - un programa que represente gráficamente los datos.
- Es más práctico y rápido usar una utilidad en línea de comandos o un lenguaje interpretado (como R) que un lenguaje de programación no interpretado.

# Modificación de ficheros Netcdf

- Me han dado unos datos en NetCDF y tengo que corregir alguno de sus aspectos. P.ej.: modificar un atributo, poner longitudes en el rango  $[-180, 180]$  o en el rango  $[0, 360]$ .
- Se puede hacer con un lenguaje de programación, pero es más práctico usar una utilidad en línea de comandos,
- Utilidades para esto: nco, cdo, ...

# Operar con los datos de ficheros Netcdf

- Si los cálculos son poco comunes, convendrá usar un lenguaje de programación.
- Para cálculos más comunes, es más práctico y rápido usar una utilidad como cdo, que está muy optimizada.

# Utilidades comunes de NetCDF

- *ncgen* y *ncdump*: creación y exploración de ficheros NetCDF. Se distribuyen junto con las api de NetCDF.
- Utilidades *nco* (*del inglés NetCDF Operators*): diversas operaciones con ficheros NetCDF. Se compone de varias utilidades. Proyecto colaborativo en SourceForge (<http://nco.sourceforge.net/>)
- Utilidad *cdo* (*Climate Data Operators*) : diversas operaciones con ficheros NetCDF. Puede operar sobre NetCDF y sobre Gribs. Desarrollado por el Instituto Max Planck.

# Creación de ficheros NetCDF: *ncgen*

- La utilidad *ncgen* permite crear un fichero NetCDF a partir de su especificación en el lenguaje *cdl*:  
**`ncgen -b NombFich.cdl`**
- El parámetro “-b” indica que debe crearse un fichero NetCDF.
- *ncgen* también puede crear código C y Fortran.
- Más información en  
[https://www.unidata.ucar.edu/software/netcdf/docs/netcdf\\_utilities\\_guide.html#guide\\_ncgen](https://www.unidata.ucar.edu/software/netcdf/docs/netcdf_utilities_guide.html#guide_ncgen)

# Exploración de ficheros NetCDF: *ncdump*

- La utilidad *ncdump* hace un volcado de un fichero NetCDF a texto en lenguaje *cdl*:  
**`ncdump NombFich.nc`**
- Si el fichero es muy largo, se puede:
  - Volcar solamente la información de cabecera (omitiendo la sección data):  
**`ncdump -h NombFich.nc`**
  - Combinarlo con el comando *less*:  
**`ncdump NombFich.nc | less`**
- También puede traducir los valores temporales a formato *humano*:  
**`ncdump -t NombFich.nc | less`**
- Por último, nos permite conocer la versión del formato NetCDF de un fichero:  
**`ncdump -k NombFich.nc`**
- Más información en  
[https://www.unidata.ucar.edu/software/netcdf/docs/netcdf\\_utilities\\_guide.html#ncdump\\_guide](https://www.unidata.ucar.edu/software/netcdf/docs/netcdf_utilities_guide.html#ncdump_guide)

# Exploración de ficheros NetCDF: utilidades *nco*

- `ncks`. Puede imprimir en pantalla, o copiar a otro NetCDF, un subconjunto de datos del fichero de partida.

```
ncks -H -v temp -d lon,203.0 -d lat,19.5 in.nc
```

-H indica que se se imprima en pantalla. La variable a imprimir se especifica tras el parámetro “-v”. Tras el parámetro “-d” se indica la longitud y la latitud. `ncks` extraerá el punto más cercano al que le proporcionemos. Si no se proporciona -H, se puede añadir al final un nombre de fichero nc de salida, que será creado.

```
ncks -v temp,presure in.nc out.nc
```

Extrae las variables *temp* y *pressure* del fichero *in.nc* y los escribe en el fichero *out.nc*.

```
ncks -H -v lon in.nc
```

Proporciona todos los valores de la longitud contenidos en el fichero.

# Exploración de ficheros NetCDF: utilidades *nco* (obtener valor de un atributo)

Hay que hacer un script, que yo haré en bash. También se puede utilizar *ncdump* o *nco*. Yo usaré *nco*:

```
NombFich="in.nc"
NombVar="time"
NombAtr="units"
# Obtiene los atributos como texto con una línea por atributo
attributes="$(ncks -m -v $NombVar $NombFich)"
while IFS= read -r line; do # Buscamos nuestro atributo, línea a línea
  if [[ "$line" == *"$NombVar:$NombAtr"* ]]; then
    AtrBuscado=${line#*=} # Obtiene el valor del atributo, a la derecha del =
    # La siguiente línea sólo sirve si el atributo es un
    # texto y queremos quitarle las comillas que lo rodean
    AtrBuscado=$(echo "$AtrBuscado" | cut -d'"' -f 2)
    # La siguiente línea elimina los espacios al principio y al final
    AtrBuscado=$(echo $AtrBuscado | xargs)
  fi
done <<< "$attributes"
```

# Exploración de ficheros NetCDF: utilidad *cdo*

- *cdo*:
  - *cdo infon nomb\_fich.nc*. Listado de pasos de tiempo con algunas estadísticas.
  - *cdo sinfon nomb\_fich.nc*. Más información acerca de la estructura del fichero.
  - *cdo showname nomb\_fich.nc*. Muestra los nombres de todas las variables contenidas.
  - *cdo showdate nomb\_fich.nc*. Muestra las fechas contenidas, en formato “humano”. Si las queremos en una sola columna, se puede combinar con comandos de linux:  
***cdo showdate nomb\_fich.nc | xargs -n1***
  - *cdo showatts nomb\_fich.nc*. Muestra todos los atributos, globales y de variables.
  - *cdo diff nomb\_fich\_1.nc nomb\_fich\_2.nc*. Compara dos ficheros. Mejor que el comando *diff* de linux.
  - *cdo griddes nomb\_fich.nc*. Genera una descripción, en formato texto, de la malla espacial. Además de estar en formato “humano”, se puede usar como entrada en otros operadores de *cdo*.

# Exploración de ficheros NetCDF: gráficos con *Panoply*

- Panoply:
  - Graficar un campo sobre un mapa.
  - No permite personalizar los gráficos al detalle, pero sirve para obtener un gráfico rápidamente.

# Modificación de ficheros NetCDF: edición de atributos

- Para la edición de atributos se suele usar una de las utilidades *nco*, concretamente, *ncatted*.
- *ncatted* edita los atributos (ya sean globales o de cualquier variable). Puede añadir un valor a un atributo existente, crear un atributo, borrarlo, modificarlo o sobrescribirlo.

```
ncatted -a long_name,temp,c,c,"temperatura en superficie" FichModificar.nc
```

# Modificación de ficheros NetCDF: edición de atributos

Las llamadas más usuales a *ncatted* siguen el siguiente patrón:

```
ncatted -a nomb_attr,nomb_var,modo,tipo_attr,valor_attr  
NombFich
```

Donde:

-a significa que lo que viene inmediatamente después, es una descripción de atributo.

Todo lo que viene entre “-a” y el nombre de fichero que queremos a modificar, es una descripción de atributo, que desglosamos en la siguiente pantalla.

# Modificación de ficheros NetCDF: edición de atributos

Descripción de atributo:

**nomb\_attr, nomb\_var, modo, tipo\_attr, valor\_attr**

Donde:

**nomb\_attr** es el nombre del atributo que queremos añadir, eliminar, etc.

Se puede poner una expresión regular.

**nomb\_var** es el nombre de la variable a la que pertenece el atributo que queremos añadir, eliminar, etc. Se puede poner una expresión regular. Si ponemos “global”, significa que el atributo es global (no pertenece a ninguna variable).

# Modificación de ficheros NetCDF: edición de atributos

Descripción de atributo:

**nomb\_atr, nomb\_var, modo, tipo\_atr, valor\_atr**

Donde:

**modo** es una de las letras **a, c, d, n, m, o**, que tienen el siguiente significado:

**a** para añadir el valor **valor\_atr** al atributo. Si el atributo no existe, es creado.

**c** para crear el atributo si no existe. Si ya existe, lo deja inalterado.

**d** para borrar el atributo. Si hemos omitido el nombre del atributo, se borrarán todos los atributos de la variable especificada. En este caso, la descripción del atributo comenzaría como “,nomb\_var,modo,...”

# Modificación de ficheros NetCDF: edición de atributos

Descripción de atributo:

**nomb\_attr, nomb\_var, modo, tipo\_attr, valor\_attr**

Donde:

**modo** es una de las letras **a, c, d, n, m, o**, que tienen el siguiente significado:

**a** para añadir el valor **valor\_attr** al atributo. ...

**c** para crear el atributo si no existe. Si ya existe, lo deja inalterado.

**d** para borrar el atributo. ...

**n** igual que **a**, pero si el atributo no existe no es creado.

**m** modifica el valor del atributo, siendo **valor\_attr** el nuevo valor

**o** sobrescribe el atributo con el nuevo valor **valor\_attr**

# Modificación de ficheros NetCDF: edición de atributos

Descripción de atributo:

**nomb\_attr, nomb\_var, modo, tipo\_attr, valor\_attr**

Donde:

**tipo\_attr** es el tipo del atributo que estamos creando, modificando, etc. si lo estamos borrando, no hace falta ponerlo, por lo que la descripción del atributo quedaría:

**nomb\_attr, nomb\_var, modo, , valor\_attr**

El tipo puede ser: f (real de precisión simple), d (real de doble precisión), i ó I (entero), s (entero corto), c (cadena de caracteres), b (entero byte), ub (entero byte sin signo), us (entero corto sin signo), etc. (

<http://nco.sourceforge.net/nco.html#ncatted-netCDF-Attribute-Editor> )

# Modificación de ficheros NetCDF: edición de atributos

Descripción de atributo:

**nomb\_atr, nomb\_var, modo, tipo\_atr, valor\_atr**

Donde:

**valor\_atr** es el valor del atributo que estamos creando, modificando, etc. Si lo estamos borrando, no hace falta ponerlo, por lo que la descripción del atributo quedaría:

**nomb\_atr, nomb\_var, modo, ,**

Además, cuando **valor\_atr** es de tipo carácter, lo ponemos entre comillas.

# Modificación de ficheros NetCDF: edición de atributos

Por ejemplo, para cambiar las unidades de la variable temp:

```
ncatted -a units,temp,o,c,"C" ejemplo_nc.nc
```

Para añadir un atributo a la variable temp:

```
ncatted -a standard_name,temp,c,c,"air_temperature"  
ejemplo_nc.nc
```

# Modificación de ficheros NetCDF: renombrar

- *ncrename*. Cambia el nombre de dimensiones, variables y atributos.
  - **ncrename -v pressure,presion in.nc** ← nombre de variable
  - **ncrename -d lon,longitude -v lon,longitude in.nc**  
cambia el nombre de la dimensión y de la variable de coordenada correspondiente (hay que renombrar ésta explícitamente). Recordemos que ambas deben tener siempre el mismo nombre.
  - **ncrename -a temp@units,unidades in.nc** ← nombre de atributo

# Modificación de ficheros NetCDF: reordenar dimensiones

- La mayor parte del software existente, espera encontrar las dimensiones de una variable en el orden (*time, level, lat, lon*). Si lo tenemos como (*time, level, lon, lat*), el software más moderno podría interpretarlo correctamente, pero si disponemos de un software antiguo, puede que tengamos que reordenar.

Si en el fichero *in.nc* la variable *temp* es *temp(time, lat, lon)*, la orden

```
ncpdq -v temp -a time,lon,lat in.nc out.nc
```

la convierte en *temp(time, lon, lat)*

- La mayor parte del software existente, espera encontrar la latitud en orden decreciente. Si no lo está y nuestro software lo requiere, podemos invertirla con la orden (nco):

```
ncpdq -v temp -a '-lat' in.nc out.nc
```

O también con la orden (cdo):

```
cdo invertlat in.nc out.nc
```

- También nos puede convenir cambiar longitudes del rango [0,360] al rango [-180,180]

```
cdo sellonlatbox,-180,180,-90,90 infile outfile
```

# Operaciones con NetCDF: selección

## Selección temporal:

- Seleccionar un periodo temporal entre dos fechas  
**cdo seldate, 2050-01-01, 2015-01-03 in.nc out.nc**
- Seleccionar los meses del calendario que le digamos. P.e., para seleccionar los meses de una estación seca desde noviembre hasta abril inclusive:  
**cdo selmon, 11, 12, 1, 2, 3, 4 FichEntrada FichSalida**  
si queremos que formen parte de la misma estación debemos cuidar que ésta no se vea cortada al comienzo ni al final del periodo de tiempo que hay en el fichero:  
**cdo selmon, 11, 12, 1, 2, 3, 4 -seldate, 2006-11-01, 2100-05-01 FichEntrada FichSalida**

# Operaciones con NetCDF: selección

## Selección espacial:

- Seleccionar un dominio espacial rectangular:

```
cdo sellonlatbox,lon1,lon2,lat1,lat2 infile outfile
```

Por ejemplo:

```
cdo sellonlatbox,-100,-95,30,40 infile outfile
```

también, como hemos visto anteriormente, sirve para cambiar longitudes del rango [0,360] al rango [-180,180]

```
cdo sellonlatbox,-180,180,-90,90 infile outfile
```

## Selección de niveles:

- **cdo sellevel,levels infle.nc outfile.nc**

Por ejemplo:

```
cdo sellevel,700 infile.nc outfile.nc
```

Extrae el nivel de 700 hPa (suponiendo que las unidades del nivel estén en hPa, cosa que hay que comprobar, por ejemplo, con ncdump).

# Operaciones con NetCDF: establecer regiones a missing

- Poner a *missing* lo que esté fuera del rectángulo que le digamos:  
**cdo maskbox, lon1, lon2, lat1, lat2 infile outfile**  
Por ejemplo:  
**cdo maskbox, -100, -95, 30, 40 infile outfile**
- Poner a *missing* lo que esté fuera de la región que le digamos.  
**cdo maskregion, fichregion.txt infile outfile**  
donde *fichregion.txt* es un fichero de texto con dos columnas (longitud y latitud) y una fila por cada punto del polígono que conforma la región. Puede haber varios polígonos en un mismo fichero, separándolos mediante una línea que solamente contenga el carácter “&”.

# Operaciones con NetCDF: aritmética

- Añadir una constante:

```
cdo addc, constante infile outfile
```

Por ejemplo, para convertir grados Kelvin a Celsius:

```
cdo addc,-273.15 infile outfile
```

O, si hay varias variables en el fichero, seleccionamos primero la variable con la que queremos operar:

```
cdo addc,-273.15 -selvar,temp infile outfile
```

Después habría que cambiar valor del atributo “units”:

```
cdo setattribute,temp@units=degC outfile outfile2
```

O, con las utilidades nco:

```
ncatted -a units,temp,o,c,"degC" outfile outfile2
```

- También se puede restar (subc), multiplicar (mulc), dividir (divc) por una constante (ver ayuda de cdo).
- Sumar dos variables de ficheros diferentes:
 

```
cdo add infile1 infile2 outfile
```
- También se pueden restar (sub), multiplicar (mul), dividir (div) dos variables de dos ficheros diferentes (ver ayuda de cdo).
- Aritmética con variables que están en el mismo fichero (utilidad nco)
 

```
ncap -O -s "windspeed=sqrt(u^2+v^2)" in.nc out.nc
```

 ”-O” significa sobrescribir fichero de destino. Tras la opción “-s”, se especifica la operación aritmética a realizar  
 Con cdo sería:
 

```
cdo sqrt -add -sqr -selvar,u infile.nc -sqr -selvar,v infile.nc outfile.nc
```
- Más operaciones: <https://code.mpimet.mpg.de/projects/cdo/wiki/Cdo#Documentation>

# Operaciones con NetCDF: estadística

- Obtener, para cada paso de tiempo, el valor máximo de todos los puntos de la malla bidimensional:  
**cdo fldmax infile outfile**
- Obtener, para cada paso de tiempo, el valor medio de todos los puntos de la malla bidimensional:  
**cdo fldmean infile outfile**
- También se puede obtener el valor mínimo (fldmin), la desviación estándar (fldstd), etc de todos los puntos de la malla bidimensional en cada paso de tiempo. Ver ayuda de cdo.
- Obtener, en cada punto de la malla, el valor medio de todos los pasos de tiempo:  
**cdo timmean infile outfile**
- Obtener, en cada punto de la malla, la desviación estándar de todos los pasos de tiempo:  
**cdo timstd infile outfile**
- También se pueden obtener más estadísticas. Ver la ayuda de cdo.

# Exportación de ficheros NetCDF a ascii

```
cdo outputf, format, ncol infile.nc > outfile.nc
```

- donde *format* es una cadena de formato que sigue las convenciones del lenguaje de programación C.

El formato más usual toma la forma:

```
%ancho.decimalesf
```

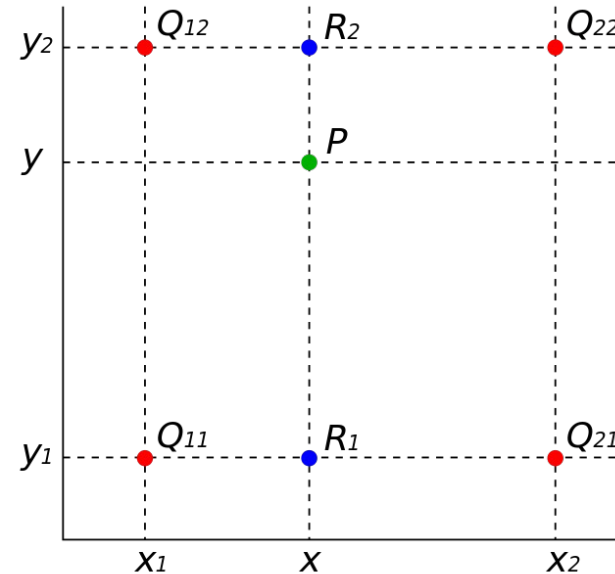
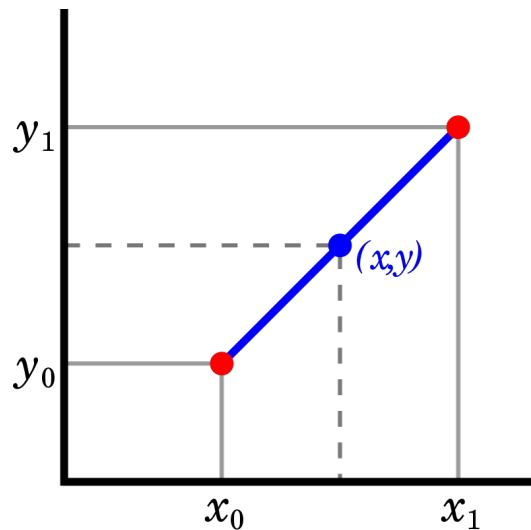
donde *ancho* es el numero de caracteres que ocupará cada número, incluyendo el signo y el punto decimal; *decimales* es el número de decimales a usar; y al final se le añade la letra “f”. Si el numero resulta tener un ancho menor que el especificado, se completa con blancos a la izquierda.

Por ejemplo: %10.4f es un numero que ocupará 10 caracteres de ancho y tendrá 4 decimales.

- El parámetro *ncol* es el número de columnas en las que se distribuirán los números.

# Operaciones con NetCDF: interpolación

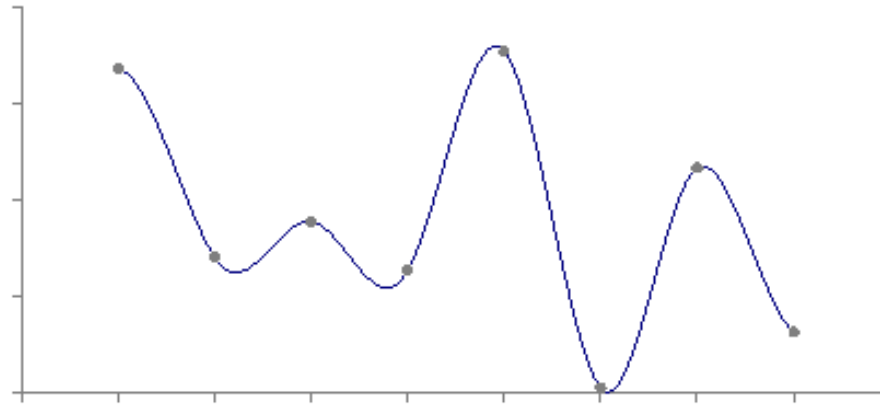
Intepolación bilineal. Es la extensión de la interpolación lineal a dos dimensiones.



Gráficos tomados de Wikipedia

# Operaciones con NetCDF: interpolación

Interpolación bicúbica. Es la extensión de la interpolación cúbica, a dos dimensiones.



Interpolación cúbica de una serie de puntos

Gráfico tomado de <http://paulbourke.net/miscellaneous/interpolation/>

# Operaciones con NetCDF: interpolación

Vecino más cercano. Asigna, a cada punto, el valor de su vecino más cercano.

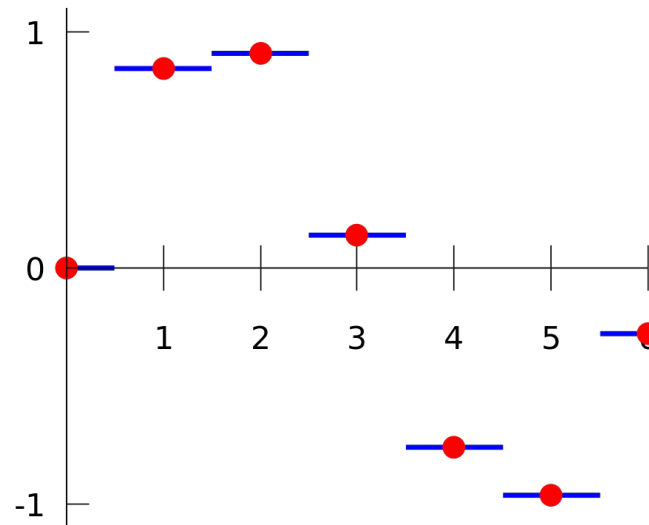
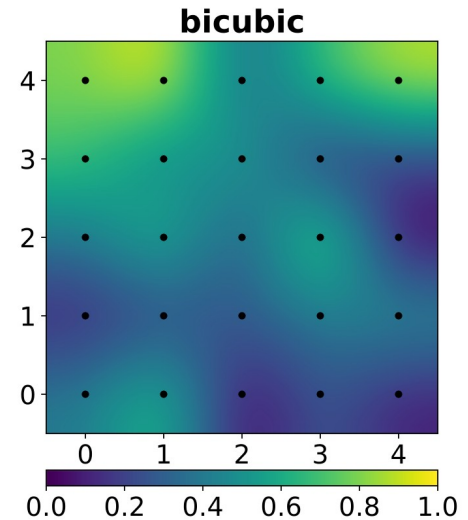
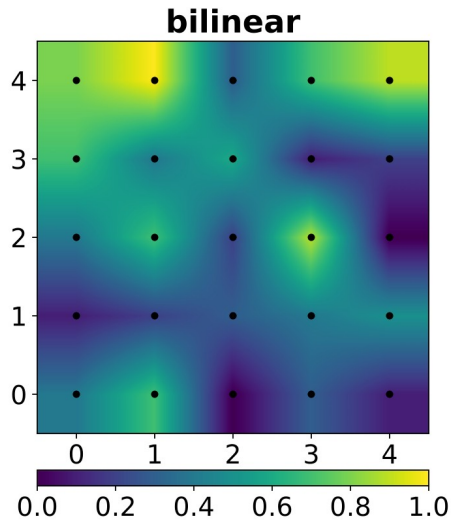
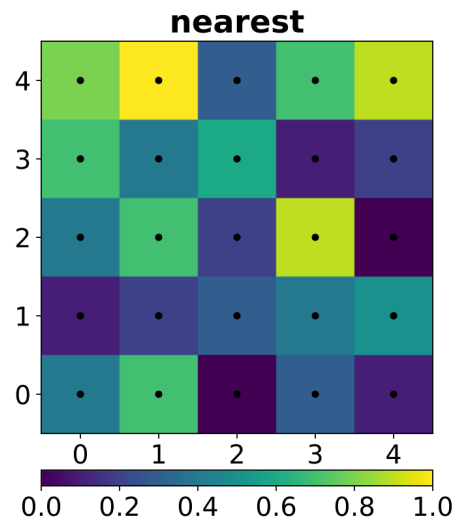


Gráfico tomado de Wikipedia

# Operaciones con NetCDF: interpolación

Comparación de la versión bidimensional de los tres métodos



Gráficos tomados de Wikipedia

# Operaciones con NetCDF: interpolación

- No es apropiado interpolar para aumentar la resolución de los datos.

# Operaciones con NetCDF: interpolación

En mi pueblo (a 0 km) no ha llovido.

A 100 km al norte han caído 800 mm

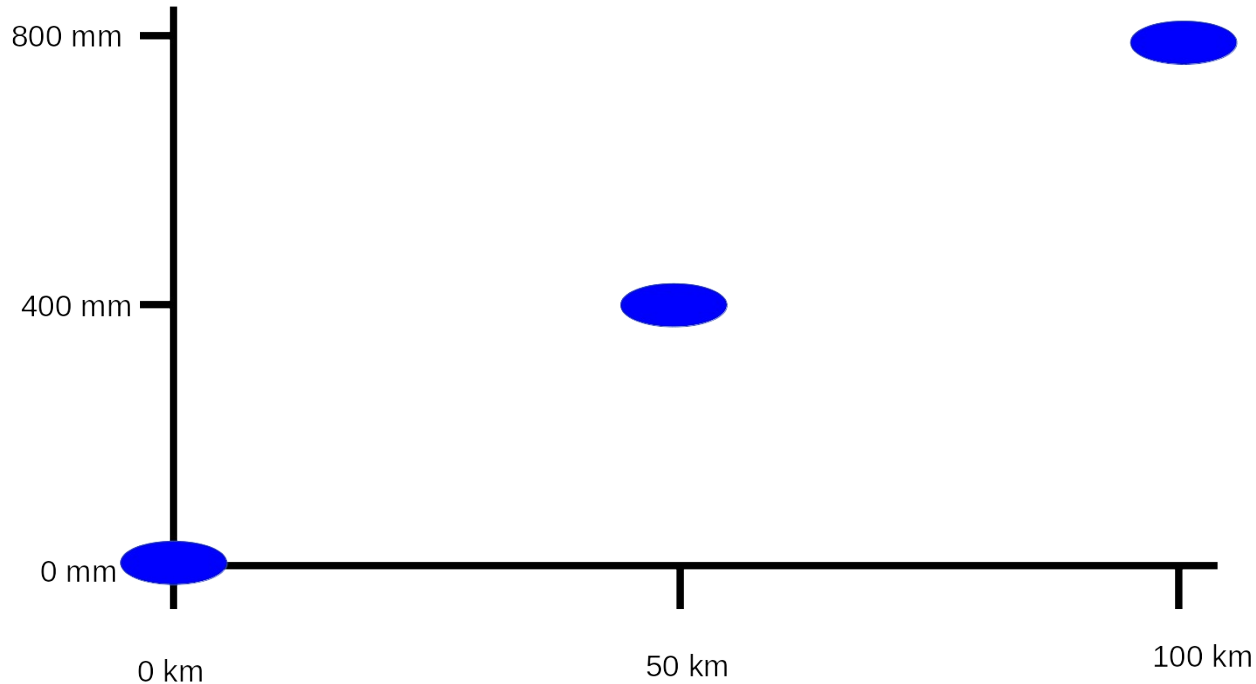


# Operaciones con NetCDF: interpolación

En mi pueblo (a 0 km) no ha llovido.

A 100 km al norte han caído 800 mm

Una interpolación lineal nos diría que han caído 400 mm a 50 km al norte, pero esto, generalmente, sería erróneo.



# Operaciones con NetCDF: interpolación

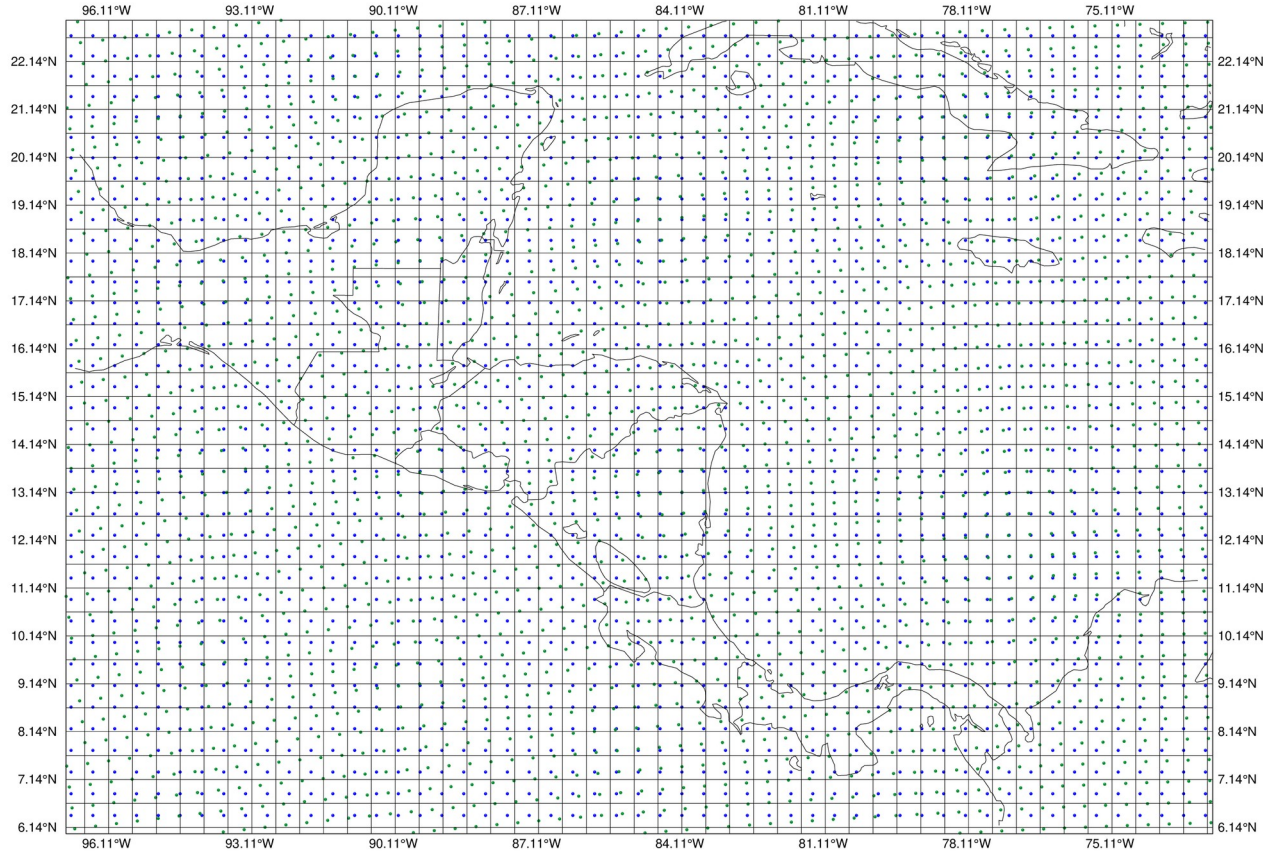
- No es apropiado interpolar para aumentar la resolución de los datos.
- Para aumentar la resolución de los datos, hay que regionalizar (dinámica o estadísticamente).
- La interpolación es apropiada para cambiar de una estructura de grid a otra cuando la resolución de ambas es similar.  
Por ejemplo, los datos del SMHI tienen unas coordenadas rotadas con  $0,44^\circ$  de resolución. Podemos interpolar a una rejilla regular de  $0,50^\circ$  de resolución.

# Operaciones con NetCDF: interpolación

- No es apropiado interpolar para aumentar la resolución de los datos.
- Para aumentar la resolución de los datos, hay que regionalizar (dinámica o estadísticamente).
- La interpolación es apropiada para cambiar de una estructura de grid a otra cuando la resolución de ambas es similar.
- Uno de los métodos de interpolación es la del vecino más cercano, que conserva la coherencia espacial, cosa que será apropiada dependiendo de la zona que estemos trabajando y de la resolución.
  - Para los campos en superficie, se aconseja conservar la naturaleza de la celda (mar o tierra). Es decir, para obtener el valor de un punto en la malla de destino que está en tierra, le asignamos el valor de su vecino más próximo (de la malla de origen) que también esté en tierra.

# Operaciones con NetCDF: interpolación

AZUL: grid ICTP  
VERDE: SMHI



Resoluciones:

- SMHI: 0,44°
- ICTP: 50 km (proyectado)
- Malla a la que se han interpolado todos los datos: 0,50°. Son los puntos de intersección de las líneas verticales y horizontales de color negro.

Las tres resoluciones son parecidas, por lo que la interpolación no está desaconsejada.

# Operaciones con NetCDF: interpolación

- Para interpolar con *cdo*, tenemos que especificar la malla de destino, en el mismo formato del operador *griddes*.

```
gridtype = lonlat
gridsize = 12
xsize    = 3
ysize    = 4
xname    = lon
xlongname = "longitude"
xunits   = "degrees_east"
```

```
yname    = lat
ylongname = "latitude"
yunits   = "degrees_north"
xfirst   = -30
xinc     = 1
yfirst   = 40
yinc     = 0.5
```

# Operaciones con NetCDF: interpolación

- Interpolación bilineal:  
`cdo remapbil,griddes.txt infile outfile`
- Interpolación bicúbica:  
`cdo remapbic,griddes.txt infile outfile`
- Interpolación por el vecino más próximo:  
`cdo remapnn,griddes.txt infile outfile`  
la terminación *nn* viene del inglés *Nearest Neighborhood*
- Más tipos de interpolaciones que se pueden hacer con *cdo*:  
<https://code.mpimet.mpg.de/projects/cdo/embedded/cdo.pdf>

# Bibliografía

- Herramientas *ncgen*, *ncdump* y otras que vienen de serie con la instalación de NetCDF:  
[https://www.unidata.ucar.edu/software/netcdf/docs/netcdf\\_utilities\\_guide.html](https://www.unidata.ucar.edu/software/netcdf/docs/netcdf_utilities_guide.html)
- Herramientas del paquete *nco*: <http://nco.sourceforge.net/>
- Documentación *cdo*:  
<https://code.mpimet.mpg.de/projects/cdo/wiki/Cdo#Documentation>
- Webs para aprender más sobre interpolación:
  - <http://paulbourke.net/miscellaneous/interpolation/>
  - [http://www.gisresources.com/types-interpolation-methods\\_3/](http://www.gisresources.com/types-interpolation-methods_3/)