**Manual NCL**

Clementine Junquas

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

<http://www2.mmm.ucar.edu/wrf/OnLineTutorial/Graphics/NCL/NCL_examples.htm>

Lanzar un script desde un terminal unix :

*ncl my\_script.ncl*

Estructura de un programa ncl :

*begin*

*…*

*end*

Comentarios “;”

**1. Matrices en NCL**

Definiciones

*A=new(2,float)*

*A=(/10.,20./)*  ; El “punto” para números “real”, sin el punto si es un “integer”

*B=new((/ntime,nlev,nlat,nlon/),float)*

*delete(A)* ; para borrar la matriz A

*delete( [/ A, B, C /] )* ; para borrar varias matrices

*free memory*; si falta memoria

Funciones utiles generales

*M=(/(/1,3/),(/7,2/)/)*

*M(0:,:)*  ; **Importante** : En NCL se cuentan las dimensiones a partir de 0, como en NCOs

*transpose(M)*  ; = M'

*dim\_sum(M)* ; = suma de cada linea (1era dimension)

*dim\_sum\_n(M,1)* ; = suma de cada columna (2da dimension)

*sum(M)* ; = todos los elementos

*dimsizes(M)*  ; = size(M) en matlab

*product(dimsizes(M))* ; = length(M) en matlab

*print(M)* ; para que aparezca la matriz en la pantalla (= *M* en matab)

*print(dimsizes(M))* ; para que aparezca las dim de M en la pantalla (=*size(M)* en matlab)

Funciones utiles en climatología

promediar= average en inglés

*dim\_avg(A(0,:))*

*dim\_avg\_n(A,x) ;* promedio sobre la dimension x

*reshape(A,(/2\*2/)) ; =*reshape en matlab

*permute*=X(lat|:,lon|:,lev|:)

*smth9* para suavizar orografia por ej.

*linint2* para hacer interpolación lat/lon,

ej : matB=linint2(lon,lat,matA, False, lon2,lat2,0) ; True if lon is cyclic

función divergence *= uv2dv\_cfd* ; si lat/lon grid

*squeeze* en matlab = *rm\_single\_dims(M)*

Para encontrar una coordenada : latx=ind\_nearest\_coord(xlat,lat,0) ; ej. latx=83, xlat=-72.5°, lat=matrice columna

ncrcat en NCO = *array\_append\_record(M1,M2,x)* ;= se juntan 2 matrices en la dimension x

crear archivo txt :

asciiwrite("qum\_vertint.txt",qum)

leer archivos .txt :

qum = asciiread("qum\_vertint.txt",(/nlat,nlon/),"float")

Loops

*if (i .gt. 0) then*

*…*

*end if*

*do i=0,ntimes-1*

*…*

*end do*

*do while (i .lt. 10)*

*…*

*end do*

*a .and. b*

*a .or. b*

*.not. A*

*a .eq. B ; equal*

*a .lt. B ; lower than*

*a .gt. B ; greater than*

*a .le. B ; lower or equal*

*a .ge. B ; greater or equal*

*a .ne. b ; not equal*

**2. Hacer gráficos**

Generalidades

*plot=gsn\_x(wks,M,Resources)* ; linea final para dibujar la figura

“wks”, “M” y “Resources” tienen que ser definidos antes :

Wks indica si el “plot” se va a registrar en un archivo o si va a aparecer a la pantalla (no pueden ser los dos en mismo tiempo) :

*wks = gsn\_open\_wks (type, “nombre del file”)*

el type tiene que ser definido antes, por ejemplo :

*type=”x11”*  ; para que aparezca el gráfico en la pantalla

o *type = “pdf”* ; para que el gráfico sea guardado como archivo pdf

o *type = “ps”*

M en este caso es la matriz (ej. campo de temp)

Para definir características a una matriz :

[*M@long\_name*](mailto:M@long_name)*=”Mi Matriz”*

“Resources” son las características de como se quiere que la variable aparezca en el gráfico.

Ejemplo para un contorno :

*Resources = True* ; el uso de la palabra “Resources” es arbitrario, puede ser “res”, o como quieren

[*Resources@xyLineColors*](mailto:Resources@xyLineColors) *= (/2/)* ; o “red”, “green”, etc..

[*Resources@xyLineThicknesses*](mailto:Resources@xyLineThicknesses) *= (/1./)*  ; =1 por defecto

[*Resources@tiMainString*](mailto:Resources@tiMainString) *= “title”*

[*Resources@tiXAxisString*](mailto:Resources@tiXAxisString) *= “X Axis title”* ; lo mismo para Y

[*Resources@tiMainFont*](mailto:Resources@tiMainFont) *= “Helvetica”*

[*Resources@tiXAxisFont*](mailto:Resources@tiXAxisFont) *= “Helvetica”* ; lo mismo para Y

*delete(Resources)* ; para empezar con nuevas opciones

*delete(plot)*

**Otras funcionalidades**

Convertir del formato GRIB a NETCDF :

*grib a netcdf con NCL : ncl\_convert2nc file.nc*

*netcdf a grib con CDO : cdo -f grb copy inflie.nc outfile.grb*

Cambiar una variable en un archivo netcdf :

*load "$NCARG\_ROOT/lib/ncarg/nclscripts/csm/gsn\_code.ncl"*

*begin*

*a = addfile("./met\_em.d01.2000-01-24\_12:00:00.nc",”w")*

*sst = a->SST* ; read a field

*sst = sst + 1* ; change the field

*a->SST = sst* ; write the field

*end*