

ANEXO B

MANUAL PARA LA CREACIÓN DE LAS MALLAS DE VIENTOS PARA EL DELFT 3D

INTRODUCCIÓN

En la creación de modelos hidrodinámicos es necesaria la utilización de diferentes fuentes de información, el modelo del mar Caribe utiliza una malla equidistante rectangular de vientos y presiones, la cual, en trabajos futuros podría contar con información de otras variables como la temperatura, la radiación solar, la humedad relativa y otras.

La información climatológica necesaria para la generación de estas mallas esta almacenada en archivos tipo netCDF (*.nc) provenientes del NARR y el NCAR, por lo que, los datos de estos archivos fueron convertidos al formato de entrada del Delft 3D y proyectados a la proyección cartográfica WGS 1984 PDC Mercator, con la cual fue construido el modelo.

En este anexo se describe el algoritmo con el que se generaron las mallas de vientos y presiones y se realiza un manual paso a paso de la utilización de este.

1. METODOLOGIA PARA LA CREACIÓN DE ARCHIVOS DE VARIACIÓN ESPACIO TEMPORAL PARA DEFLT 3D, UTILIZANDO INFORMACIÓN DEL NARR Y EL NCAR

Los datos de las variables oceanográficas y atmosféricas son comúnmente almacenados en archivos NetCDF (network Common Data Form), los cuales son un conjunto de interfaces dirigidas al almacenamiento y manejo de variables climatológicas, para uso científico. Estos formatos vectoriales fueron desarrollados por Glenn Davis, Russ Rew, Ed Hartnett, John Caron, Steve Emmerson, y Harvey Davies en el centro del programa Unidata en Boulder, Colorado.

Los programas NARR Y NCAR, almacenan los resultados de sus programas de recolección de datos y re análisis de diferentes variables en el formato que se acaba de mencionar, con el fin de construir las mallas rectangulares que representan los forzadores atmosféricos del modelo del mar Caribe, se convirtieron la información de los NetCDF al formato adecuado y a la proyección cartográfica adecuada, para que la función de entrada de datos ADD-ON del Delft 3D recibiera la información y luego la utilizara para forzar el modelo.

El modelo del mar Caribe utilizó los datos de las velocidades del vientos y de las presiones barométricas del NARR y NCAR, estos archivos tiene extensión '*.nc' y dentro de su estructura acumulan variables independientes y dependientes.

Para un mejor entendimiento de la estructura de estos archivos, se presenta como ejemplo la información de un NetCDF en cuatro dimensiones que varía en función de la y, x, la coordenada barométrica y el tiempo, como ejemplo se utiliza el archivo de temperaturas del aire air.200807.nc de la base de datos del NARR.

Estos archivos NetCDF del NARR y el NCAR, se pueden describir como una malla en cuatro dimensiones donde cada celda tiene coordenadas (y,x, level, t), las cuales son variables independientes, estas variables se listan a continuación:

- **Tiempo:** Son los tiempos para los cuales existen datos dentro del archivo, estos se almacenan dentro de la variable 'time'. Esta variable para este archivo, está dada en horas a partir del año 1800, aunque para cada fuente de información (NARR o NCAR) puede ser diferente el origen temporal de sus datos. Estos datos se encuentran almacenados en un vector unidimensional.
- **Coordenada barométrica:** Los diferentes niveles de presión están almacenados dentro de la variable 'level', esta variable se encuentra almacenada en un vector unidimensional.
- **x:** número de nodos de la malla del dominio del NARR o el NCAR, con respecto al origen, estos nodos se pueden contar en un eje paralelo al eje del Ecuador, la variable dentro del archivo se llama 'x'.
- **y:** número de nodos de la malla del dominio del NARR o el NCAR, con respecto al origen, estos nodos se pueden contar en un eje paralelo a los meridianos de la tierra, la variable dentro del archivo se llama 'y'.

Las variables independientes comienzan en 0, a partir del origen de la malla del NARR o el NCAR.

Variables dependientes:

- **Latitud y Longitud:** esta es la latitud o la longitud de cada nodo del archivo, para los archivos del NARR estas variables están almacenadas dentro de una matriz de tamaño 'x' por 'y', y están definidas por 'lat(y,x)' y 'lon(y,x)', mientras que en el caso del NCAR están almacenadas en vectores definidos por 'lat(y)' y 'lon(x)' ósea vectores unidimensionales.
- **Temperatura:** La temperatura o la variable principal de este archivo está definida por 'air(time,level,y,x)' y está almacenada en matrices de tamaño 'x' por 'y' por 'time'.

En vez de las temperaturas un archivo de estos puede contener cualquier tipo de variable como velocidades del viento, presiones atmosféricas, humedad relativa, radiación solar y otras.

Estos archivos se pueden visualizar según su tamaño con diferentes herramientas, en: Fortran 90, C++, Matlab, Excel y otros, para este caso en específico se eligió el Matlab como plataforma para convertir la información de los programas de re análisis al formato de entrada del Delft 3D, como se presenta a continuación:

1.1 PLANTEAMIENTO DEL PROBLEMA

El modelo del Mar Caribe, tiene como forzadores atmosféricos las componentes de la velocidad del viento y la presión atmosférica. La plataforma del Delft 3D tiene como protocolos de entrada para cada una de estas variables, el siguiente formato:

- Ejemplo 1 : formato de las velocidades en el sentido de las latitudes:

```
FileVersion = 1.02
filetype = meteo_on_equidistant_grid
NODATA_value = -9999.000
n_cols = 3
n_rows = 4
grid_unit = m
x_llcorner = 2553000
y_llcorner = 1245000
value_pos = centre
dx = 30000
dy = 40000
n_quantity = 1
quantity1 = x_wind
unit1 = m s-1
TIME = 0.0 hours since 2009-01-01 00:00:00 +00:00
10.0 15.0 20.0
10.5 15.5 20.5
11.0 16.0 21.0
11.5 16.5 21.5
TIME = 24.0 hours since 2009-01-01 00:00:00 +00:00
10.0 15.0 20.0
10.5 15.5 20.5
11.0 16.0 21.0
11.5 16.5 21.5
```

Este formato está conformado por el número de columnas y filas que va a tener la malla rectangular con la que se ingresaran los forzamientos al modelo (`n_cols` y `n_rows`), luego se ven la unidades para las distancias entre las celdas de la malla, las coordenadas de la esquina suroccidental de la malla y las distancias entre las celdas, el tipo de variable a

ingresar y las unidades de esta. Para más información sobre estos formatos ver el manual del Delft 3D “Simulation of multidimensional hydrodynamic flow and transport phenomena including sediments. User Manual”. Luego están los datos de la malla para cada uno de los tiempos en los que se va ingresar la variable.

Los archivos NetCDF del NARR o NCAR todos tienen más o menos la misma estructura del archivo de temperatura descrito anteriormente solo cambia la variable principal, por ejemplo temperatura, velocidades, humedad, etc. (ver <http://www.esrl.noaa.gov/psd/data/gridded/data.narr.html>).

Estos archivos contienen información, que se sale de los dominios del modelo del mar Caribe, lo que los hace pesados y difíciles de manejar completos. Por esto se decidió utilizar una biblioteca para Matlab llamada mexcdf, que permitió manipular estos archivos y solo extraer la información necesaria para ser convertida en la malla rectangular del Delft 3D, utilizando el algoritmo que se describe en el siguiente umeral.

1.2 DESCRIPCION DEL ALGORITMO

Se construyeron dos algoritmos en Matlab diseñados para extraer datos de los archivos NetCDF, ofrecidos en las direcciones electrónicas del NARR y el NCAR <http://www.esrl.noaa.gov/psd/data/gridded/data.narr.monolevel.html> y <http://www.esrl.noaa.gov/psd/data/gridded/data.ncep.reanalysis.pressure.html> .

Se debe aclarar que los archivos de vientos y presiones atmosféricas no varían en la vertical, ya que son valores dados para la superficie del agua, a comparación del archivo del ejemplo inicial.

A continuación se explica paso a paso el algoritmo para procesar los datos del NARR, este tiene pequeñas variaciones con respecto al algoritmo del NCAR, por lo que solo se explicara el primero.

Este es un algoritmo que parte de un dominio en forma de rectángulo, con vértices en coordenadas WGS 1984 en grados decimales. Como se puede ver en la subrutina siguiente.

Advertencia: Este algoritmo solo funciona en el sistema de coordenadas WGS 1984, por lo que siempre antes de comenzar a utilizar estos algoritmos se debe conocer el dominio de las coordenadas del archivo NetCDF que se va a utilizar. Las longitudes en los archivos del NARR varían entre -180° y 180° creciendo hacia el este, y los archivos del NCARR varían entre 0 y 360° creciendo hacia el este, ambos sistemas con el cero en el meridiano de Greenwich.

```
clear all
clc
close all
```

```
%Working window points input:
```

```
lon(1)=input('Specify point 1 Longitude: ');
lat(1)=input('Specify point 1 Latitude: ');
lon(2)=input('Specify point 2 Longitude: ');
lat(2)=input('Specify point 2 Latitude: ');
lon(3)=input('Specify point 3 Longitude: ');
lat(3)=input('Specify point 3 Latitude: ');
lon(4)=input('Specify point 4 Longitude: ');
lat(4)=input('Specify point 4 Latitude: ');
```

```
%Window raw limits definition
```

```
lonLowLimit=min(lon);
lonHiLimit=max(lon);
latLowLimit=min(lat);
latHiLimit=max(lat);
lonLowLimitg=lonLowLimit;
lonHiLimitg=lonHiLimit;
latLowLimitg=latLowLimit;
latHiLimitg=latHiLimit;
```

La subrutina a continuación le solicita al usuario el nombre del archivo NetCDF, la fecha de inicio y fin del intervalo de tiempo, la variable a extraer y el espaciamiento temporal con el que se deben extraer los datos. El algoritmo solo trabaja con intervalos de tiempo que sean múltiplos enteros del el intervalo mínimo de tiempo del archivo de origen (archivos del NARR 3 horas y NCAR 6 horas).

```
%extraction parameters fron netcdf file
```

```
infile=input('input file *.nc');
tvar=input('if variable is pressure pulse 1, vel. 2, relative hum.
3, Temp. 4, Shortwave rad. 5');
nc_dump(infile)
```

```

time1=datenum(input('Specify initial time [YYYY MM DD hh mm ss]
'));
time2=datenum(input('Specify final time [YYYY MM DD hh mm ss] '));
Dtime=input('Delta time in hours')/24;
lontotal=nc_varget(infile,'lon');
latotal=nc_varget(infile,'lat');
lontotaln=size(lontotal);
latotaln=size(latotal);
Var=input('What is the variable to be extract?');
tic

```

Con el nombre del archivo netCDF el programa utiliza el comando nc_dump, y muestra la estructura del archivo original, luego extrae la totalidad de los datos de las latitudes y longitudes de cada uno de los puntos dentro del dominio del NARR o el NCAR, utilizando el comando nc_varget.

%Coordinates conversion to meters

```

index=find(lon>180);
lon(index)=lon(index)-360;
indexp=find(lon>=-150 & lon<30);
indexn=find(lon<-150 & lon>30);
lon(indexp) = 6378137*(lon(indexp)+150)*pi/180; %Converts every value in lon positive
lon(indexn) = 6378137*(lon(indexn)-150)*pi/180; %Converts every value in lon negative
lat=6356752.314245*lat*pi/180; %Converts every value in lat
%Window raw limits definition in meters
lonLowLimit=min(lon);
lonHiLimit=max(lon);
latLowLimit=min(lat);
latHiLimit=max(lat);
x_llcorner=lonLowLimit;
y_llcorner=latLowLimit;

```

En la rutina anterior se proyectan las coordenadas de las esquinas del dominio en la proyección WGS 1984 PDC de Mercator. Y luego se define cual es la esquina Suroccidental de la malla generada para el Delf 3D.

%delta

```

dx=32463;
dy=32463;
lvmatrix=fix(abs(lonHiLimit-lonLowLimit)/dx)+1;
lvmatrix=fix(abs(latHiLimit-latLowLimit)/dy)+1;

```

En las líneas de código anterior se define el numero de filas y columnas de la malla para el Delft 3D, el programa utiliza un espaciamento uniforme de 32463 m, para datos del NARR y 324630 m para el NCAR o aproximadamente 0,25° Y 2,5°, despreciando las

variaciones debidas a los cambios en la curvatura de la tierra, debido a esta aproximación lineal el algoritmo es funcional solo para las zonas cercanas al ecuador.

```
%extract the data position in the netcdf file
cont=1;
for j=1:latotaln(1,1)
    for i=1:lontotaln(1,2)
        if lontotal(j,i)>lonLowLimitg && lontotal(j,i)<lonHiLimitg && latotal(j,i)>latLowLimitg
        && latotal(j,i)<latHiLimitg
            Vectub(cont,1) =lattotal(j,i);
            Vectub(cont,2) =lontotal(j,i);
            cont=cont+1;
        end
    end
end
```

Debido a que los datos del NARR y el NCAR fueron generados inicialmente con una malla cónica y luego proyectados a WGS 1984, los datos no están distribuidos en una malla rectangular. Por lo que es necesario identificar con la rutina anterior la posición de estos datos en el espacio con respecto a la malla de cálculo.

```
%Coordinates conversion to meters Vectub
Vectublat=Vectub(:,1);
Vectublon=Vectub(:,2);
index=find(Vectublon>180);
Vectublon(index)=Vectublon(index)-360;
indexp=find(Vectublon>=-150 & Vectublon<30);
indexn=find(Vectublon<-150 & Vectublon>30);
Vectublon(indexp) = 6378137*(Vectublon(indexp)+150)*pi/180; %Converts every value in
Vectublon positive
Vectublon(indexn) = 6378137*(Vectublon(indexn)-150)*pi/180; %Converts every value in
Vectublon negative
Vectublat=6356752.314245*Vectublat*pi/180; %Converts every value in Vectublat
LonVectub=length(Vectub);
```

Con lo anterior, las posiciones de los puntos del archivo de origen que se encuentran dentro del dominio de la malla de vientos para el Delft 3D, son proyectadas a las coordenadas WGS 1984 PDC de Mercator.

```
%Create *.txt
sdate = nc_varget(infile,'time') ;
sdate=sdate/24+datenum(1800,01,01);
outfilename=strcat('out_',infile);
fid=fopen(outfilename,'w');
fprintf(fid,'FileVersion = 1.02\r\n');
```



```

fprintf(fid,'filetype = meteo_on_equidistant_grid\r\n');
fprintf(fid,'NODATA_value = -9999.000\r\n');
fprintf(fid,'n_cols = %6.4f\r\n',lxmatrix);
fprintf(fid,'n_rows = %6.4f\r\n',lymatrix);
fprintf(fid,'grid_unit = m\r\n');
fprintf(fid,'x_llcorner = %6.4f\r\n',x_llcorner);
fprintf(fid,'y_llcorner = %6.4f\r\n',y_llcorner);
fprintf(fid,'value_pos = corner\r\n');
fprintf(fid,'dx = %6.4f\r\n',dx);
fprintf(fid,'dy = %6.4f\r\n',dy);
fprintf(fid,'n_quantity = 1\r\n');
fprintf(fid, strcat('quantity1 = ',Vardelft, '\r\n'));
fprintf(fid, strcat('unit1 = ',units, '\r\n'));

```

Como se acaba de mostrar, ya se ha creado el archivo de texto de salida y fue generado el encabezado del formato de entrada del Delft 3D.

La última parte el algoritmo realiza un ciclo que obtiene los datos de la variable de interés, en múltiples instantes de tiempo dentro del intervalo temporal y para los puntos dentro del archivo original que influyen la malla de vientos. Y luego promedia los valores cercanos a cada vértice de la malla de vientos, determinando así, los valores para cada celda de la malla rectangular.

```

chou=0;
for t=time1:Dtime:time2
    ts = find(sdate == t);
    hou=num2str(hour(t));
    fprintf(fid,'TIME = %1.0f',chou);
    fprintf(fid, strcat('.0 hours since ',datestr(time1, ' yyyy-mm-dd'), ' 00:00:00 +00:00\r\n'));
    chou=chou+Dtime*24;
    % Attention: SNCTOOLS indexing is zero-based:
    cont=1;
    for j=1:latotaln(1,1)
        for i=1:lontotaln(1,2)
            if lontotal(j,i)>lonLowLimitg && lontotal(j,i)<lonHiLimitg && latotal(j,i)>latLowLimitg
                && latotal(j,i)<latHiLimitg
                    Vectub(cont,3) = nc_varget(infile,Var,[ts-1 j-1 i-1],[1 1 1]);
                    cont=cont+1;
                end
            end
        end
    end

%Create the square net
matrix=zeros(lymatrix,lxmatrix);

```

```

for m=1:lymatrix
    for n=1:lxmatrix
        ubcy=latHiLimit-(m-1)*dy;
        ubcx=lonLowLimit+(n-1)*dx;
        cant=1;
        infcell=zeros(1);
        infcell(1)=1000i;
        for o=1:LonVectub
            if Vectublon(o)>=ubcx-dx/2 && Vectublon(o)<=ubcx+dx/2 && Vectublat(o)>=ubcy-
dy/2 && Vectublat(o)<=ubcy+dy/2;
                infcell(CORREA ARANGO, Iván Darío et al.)= Vectub(o,3);
                cant=cant+1;

            end
        end
        if infcell(1)==1000i
            for o=1:LonVectub
                if Vectublon(o)>=ubcx-dx && Vectublon(o)<=ubcx+dx && Vectublat(o)>=ubcy-dy
&& Vectublat(o)<=ubcy+dy;
                    infcell(CORREA ARANGO, Iván Darío et al.)= Vectub(o,3);
                    cant=cant+1;
                end
            end
        end
        if abs(mean(infcell))>50
            matrix(m,n)=-9999.000;
        else
            matrix(m,n)=mean(infcell);
        end
    end
end
end

```

Luego de creada la matriz de datos, se realiza un procedimiento donde se eliminan los datos erróneos, provenientes del archivo original cambiando el valor en estas celdas por el promedio de las celdas que lo rodean.

```

[IndRowprob,IndColprob]=find(matrix>50);
for intp=1:length(IndRowprob)
    if IndRowprob==1 && IndColprob==1
        matrix(IndRowprob(intp),IndColprob(intp))=
(matrix(IndRowprob(intp+1),IndColprob(intp))+
matrix(IndRowprob(intp),IndColprob(intp+1))+
matrix(IndRowprob(intp+1),IndColprob(intp+1)))/3;
    else
        if IndRowprob==1 && IndColprob>1 && IndColprob<lxmatrix

```

```

matrix(IndRowprob(intp),IndColprob(intp))=
(matrix(IndRowprob(intp+1),IndColprob(intp))+
matrix(IndRowprob(intp),IndColprob(intp+1))+
matrix(IndRowprob(intp+1),IndColprob(intp+1))+
matrix(IndRowprob(intp+1),IndColprob(intp-1))+ matrix(IndRowprob(intp),IndColprob(intp-
1)))/5;
else
if IndRowprob==1 && IndColprob==lmatrix
matrix(IndRowprob(intp),IndColprob(intp))=
(matrix(IndRowprob(intp+1),IndColprob(intp))+
matrix(IndRowprob(intp+1),IndColprob(intp-1))+ matrix(IndRowprob(intp),IndColprob(intp-
1)))/3;
else
if IndRowprob>1 && IndRowprob<lymatrix && IndColprob==lmatrix
matrix(IndRowprob(intp),IndColprob(intp))=
(matrix(IndRowprob(intp+1),IndColprob(intp))+
matrix(IndRowprob(intp+1),IndColprob(intp-1))+ matrix(IndRowprob(intp),IndColprob(intp-
1))+ matrix(IndRowprob(intp-1),IndColprob(intp-1))+ matrix(IndRowprob(intp-
1),IndColprob(intp)))/5;
else
if IndRowprob==lymatrix && IndColprob==lmatrix
matrix(IndRowprob(intp),IndColprob(intp))= (matrix(IndRowprob(intp-
1),IndColprob(intp))+ matrix(IndRowprob(intp-1),IndColprob(intp-1))+
matrix(IndRowprob(intp),IndColprob(intp-1)))/3;
else
if IndRowprob==lymatrix && IndColprob<lmatrix && IndColprob>1
matrix(IndRowprob(intp),IndColprob(intp))=
(matrix(IndRowprob(intp),IndColprob(intp+1))+ matrix(IndRowprob(intp-
1),IndColprob(intp+1))+ matrix(IndRowprob(intp-1),IndColprob(intp))+
matrix(IndRowprob(intp-1),IndColprob(intp-1))+ matrix(IndRowprob(intp),IndColprob(intp-
1)))/5;
else
if IndRowprob==lymatrix && IndColprob==1
matrix(IndRowprob(intp),IndColprob(intp))=
(matrix(IndRowprob(intp),IndColprob(intp+1))+ matrix(IndRowprob(intp-
1),IndColprob(intp+1))+ matrix(IndRowprob(intp-1),IndColprob(intp)))/3;
else
if IndRowprob<lymatrix && IndRowprob>1 && IndColprob==1
matrix(IndRowprob(intp),IndColprob(intp))=
(matrix(IndRowprob(intp+1),IndColprob(intp))+
matrix(IndRowprob(intp+1),IndColprob(intp+1))+
matrix(IndRowprob(intp),IndColprob(intp+1))+ matrix(IndRowprob(intp-
1),IndColprob(intp+1))+ matrix(IndRowprob(intp-1),IndColprob(intp)))/5;
else
matrix(IndRowprob(intp),IndColprob(intp))=
(matrix(IndRowprob(intp+1),IndColprob(intp))+
matrix(IndRowprob(intp+1),IndColprob(intp+1))+
matrix(IndRowprob(intp),IndColprob(intp+1))+ matrix(IndRowprob(intp-

```

```

1),IndColprob(intp+1))+ matrix(IndRowprob(intp-1),IndColprob(intp))+
matrix(IndRowprob(intp-1),IndColprob(intp-1))+ matrix(IndRowprob(intp),IndColprob(intp-
1))+ matrix(IndRowprob(intp+1),IndColprob(intp-1)))/8;
    end
    end
    end
    end
    end
    end
    end
    end
end
end
for rowfp=1:lymatrix
    for colfp=1:lxmatrix
        fprintf(fid,'%6.2f',matrix(rowfp,colfp));
        fprintf(fid,' ');
    end
    fprintf(fid,'\r\n');
end
end
fclose(fid);
elapsedtime=toc

```

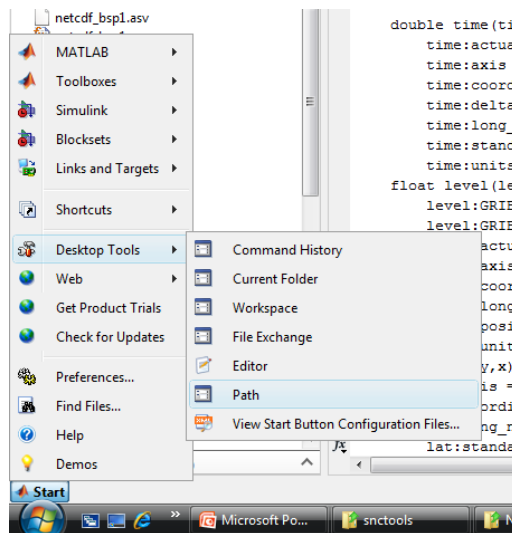
Y por último se escribe la información del archivo de entrada para el Deflt 3D.

2. INSTALACIÓN Y ENTRADA DE DATOS

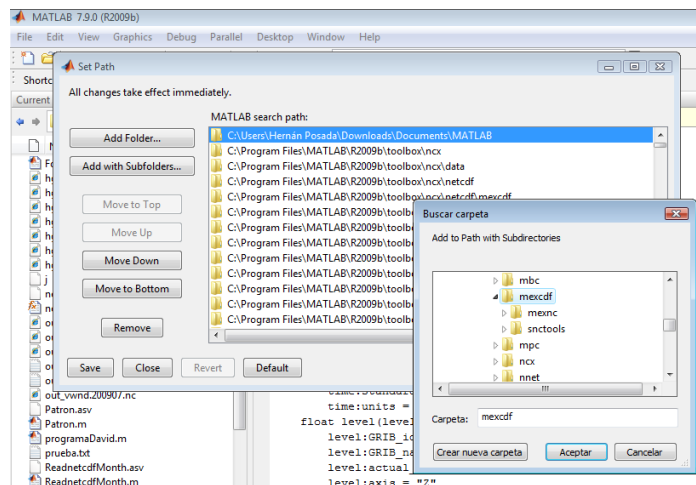
Los algoritmos para la creación de mallas ReadnetcdfTodasLasVarNAAR.m y ReadnetcdfTodasLasVarNCAR.m, funcionan utilizando las funciones de la biblioteca mexcdf. En este numeral se explicará la instalación de esta biblioteca en Matlab y luego como se deben ingresar los datos en los algoritmos para la creación de las mallas de viento.

2.1 INSTALACIÓN

- El primer paso es descargar la biblioteca mexcdf de <http://mexcdf.sourceforge.net/downloads/nc4.php>.
- El archivo se debe descomprimir en la carpeta C:\Program Files\MATLAB\R2009b\toolbox\.
- Ahora se debe abrir el Matlab e ir al botón Start y entrar en el menú Desktop Tools y luego en Path.



- En Path dar click en Add with Subfolders, y elegir la carpeta mexcdf ubicada en C:\Program Files\MATLAB\R2009b\toolbox\ y dar click en aceptar, luego un click en Save y después en Close.



- Ya instalada la biblioteca se deben digitar los siguientes comandos:
`Setpref('MEXNC',USE_TMW',true)`
`Setpref('SNCTOOLS','PRESERVE_FVD',false).`
- Luego se deben pegar en la carpeta donde está el archivo NetCDF las siguientes rutinas de Matlab:
`ReadnetcdfTodasLasVarNAAR.m` o `ReadnetcdfTodasLasVarNCAR.m`

2.2 ENTRADA DE DATOS

En Comand Window en Matlab, se deberá ejecutar uno de los dos algoritmos, según la procedencia de los datos. Como ejemplo se utilizara el archivo `vwnd.10m.2007.nc` del NARR.

El algoritmo pedirá las coordenadas de los vértices de la malla rectangular, estos valores deben estar en coordenadas WGS 1984 en grados decimales como se presenta en la gráfica a continuación:

Command Window

```
Specify point 1 Longitude: -84.5
Specify point 1 Latitude: 15.75
Specify point 2 Longitude: -84.5
Specify point 2 Latitude: 7.5
Specify point 3 Longitude: -63
Specify point 3 Latitude: 15.75
Specify point 4 Longitude: -63
Specify point 4 Latitude: 7.5
input file *.nc'vwnd.10m.2007.nc'
fx if variable is pressure pulse 1, vel. 2, relative hum. 3, Temp. 4, Shortwave rad. 5
```

- Después se debe escribir el nombre del archivo netCDF con su extensión como se muestra en la figura anterior y entre comillas simples.
- Y luego elegir la variable a extraer.
- Ahora el programa desplegará la descripción del archivo netCDF.

```
short vwnd(time,level,y,x), shape = [248 29 277 349]
  vwnd:GRIB_id = 34 s
  vwnd:GRIB_name = "VGRD"
  vwnd:_FillValue = -32767 s
  vwnd:actual_range = -67.414291 64.080887 f
  vwnd:add_offset = 47.660000 f
  vwnd:dataset = "NARR 3-hourly"
  vwnd:least_significant_digit = 1 s
  vwnd:level_desc = "Pressure Levels"
  vwnd:long_name = "3-hourly V-wind on Pressure Levels"
  vwnd:missing_value = 32766 s
  vwnd:parent_stat = "Other"
  vwnd:precision = 2 s
  vwnd:scale_factor = 0.010000 f
  vwnd:standard_name = "grid_northward_wind"
  vwnd:statistic = "Individual Obs"
  vwnd:units = "m/s"
  vwnd:unpacked_valid_range = -280.000000 350.000000 f
  vwnd:valid_range = -32766 30234 s
  vwnd:var_desc = "v-wind"
```

- En el paso siguiente el algoritmo pregunta cuál es el tiempo de inicio, para la extracción de datos, el valor de tiempo debe estar dentro del rango del archivo y en este formato [yyyy mm dd hh mm ss], por ejemplo [2007 01 01 03 00 00].
- Luego pide el tiempo final del intervalo temporal, en el mismo formato del anterior, por ejemplo [2007 01 01 12 00 00].

- En el paso siguiente se debe introducir el diferencial de tiempo en horas. Se debe introducir un valor entero, para el ejemplo 3.
- Luego se debe introducir el tipo de variable, para el ejemplo 'vwnd'.

Ahora el usuario debe esperar a que el programa realice todas las operaciones, esto puede demorarse varias horas dependiendo: del procesador del computador, de la cantidad de datos de la malla de origen y del intervalo temporal.

Cuando en la ventana de comando aparezca el tiempo transcurrido durante el proceso (Elapsedtime=###). En la carpeta activa en Matlab se encontrará un archivo con el nombre out_XXX.txt (XXX es igual al nombre del archivo netCDF), para el ejemplo out_vwnd.10m.2007.nc (los archivos del ejemplo se entregan con este anexo e la versión digital).