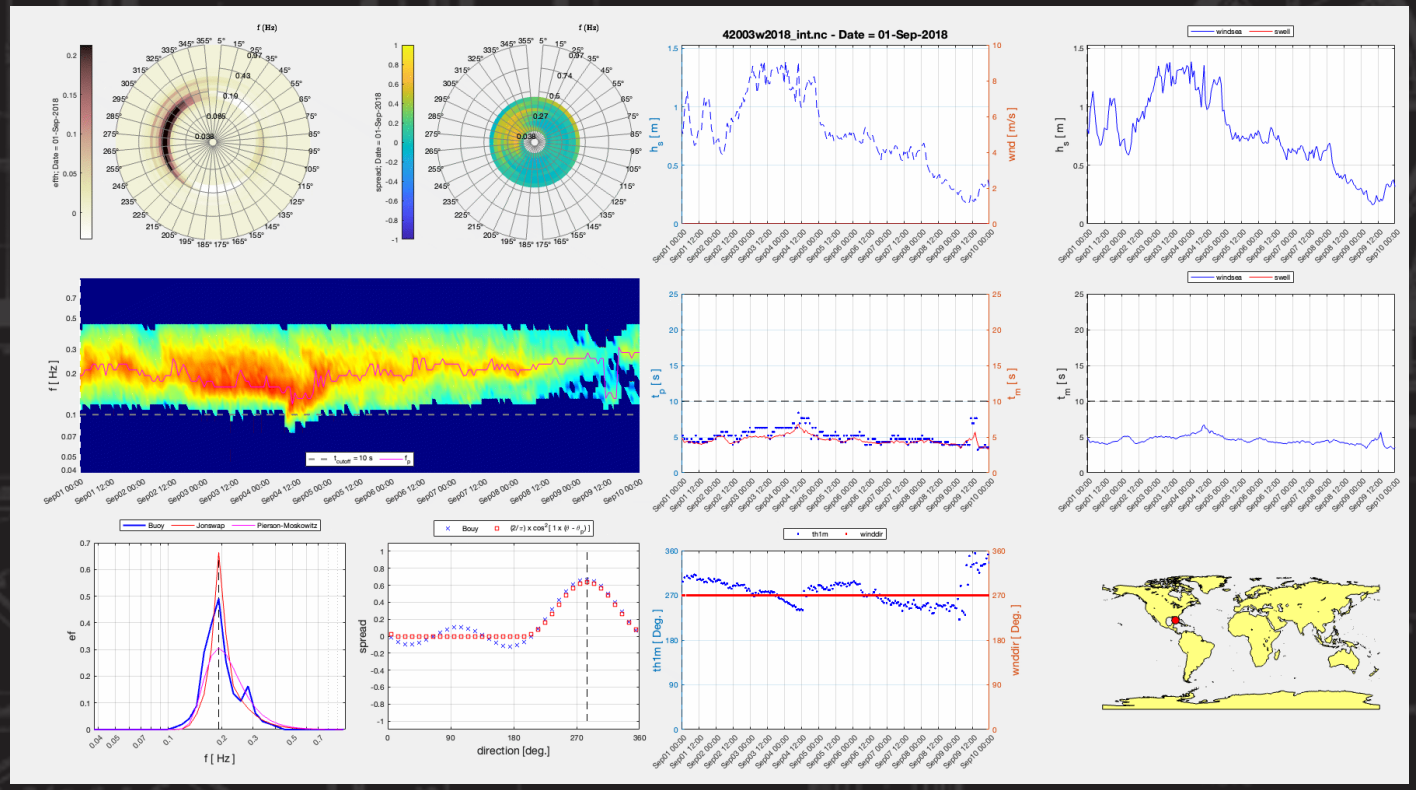
LECTURE 07:
BOUNDARY CONDITIONS

Lecturer:Ali Abdolali, PhD.

ERDC Wind Waves Training



U.S. ARMY

US Army Corps of Engineers®

WW3 VS OBS

WW3 spectral discretization is in log scale, while most obs are provided with equal increment

The spectral resolution in WW3 is defined in ww3_grid.nml(inp)

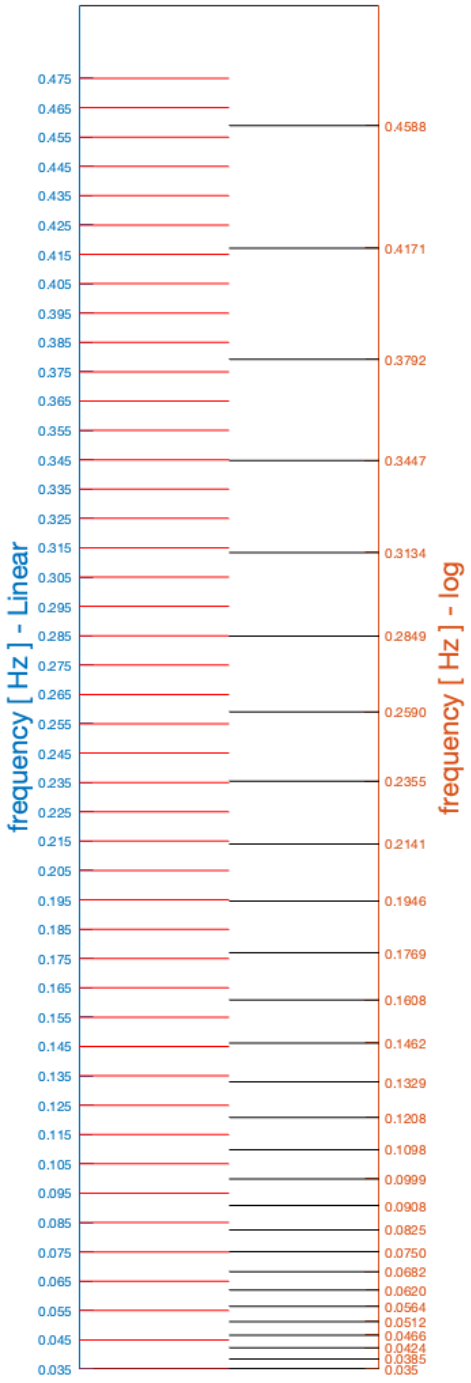
```
! ------------------------------------------------------------- !
! Define the spectrum parameterization via SPECTRUM_NML
namelist
! ------------------------------------------------------------- !
&SPECTRUM_NML
  SPECTRUM%XFR        =  1.1
  SPECTRUM%FREQ1      =  0.035
  SPECTRUM%NK         =  35
  SPECTRUM%NTH        =  36
/
```

First increment

First frequency

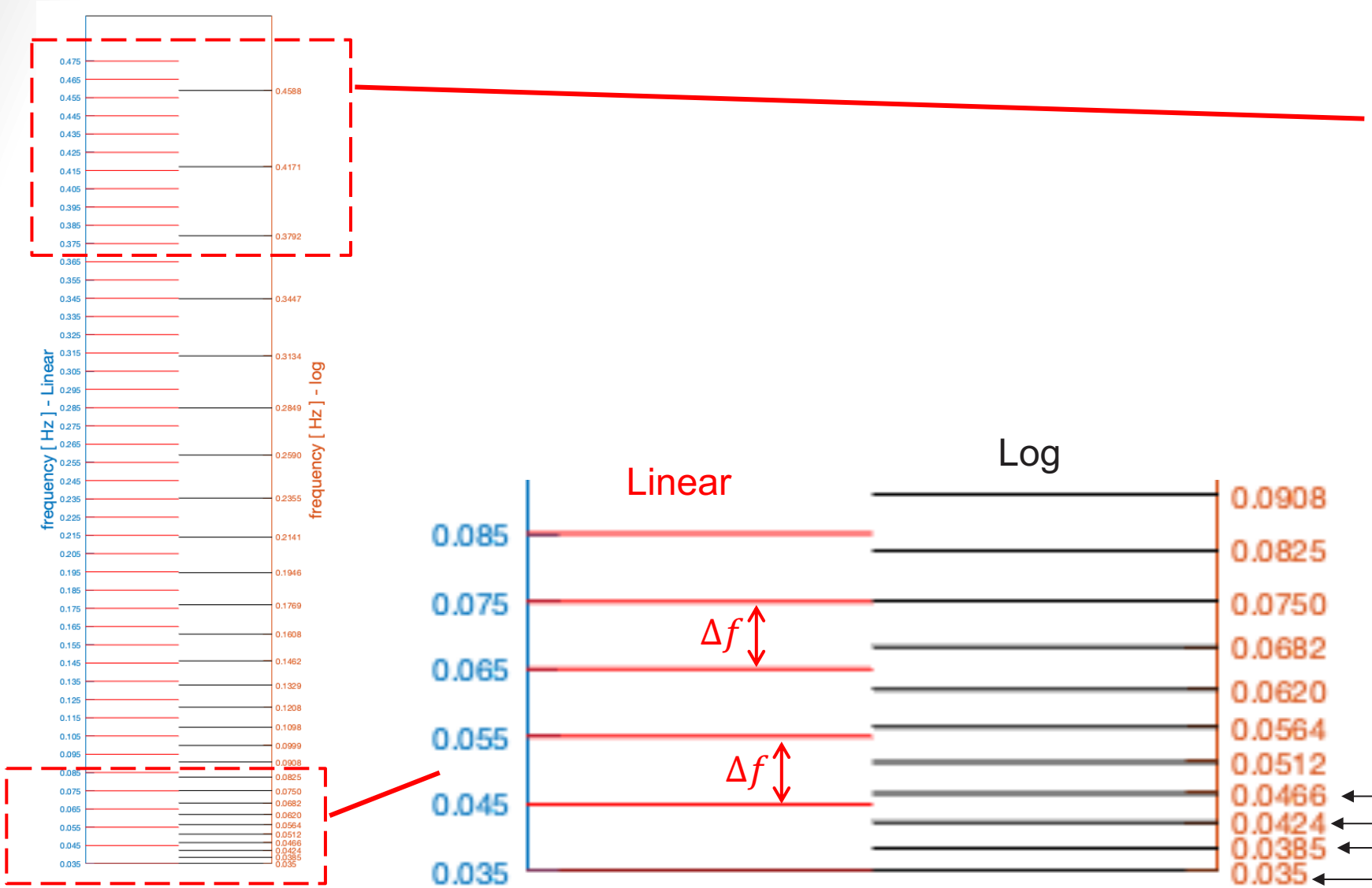
Number of frequency bins

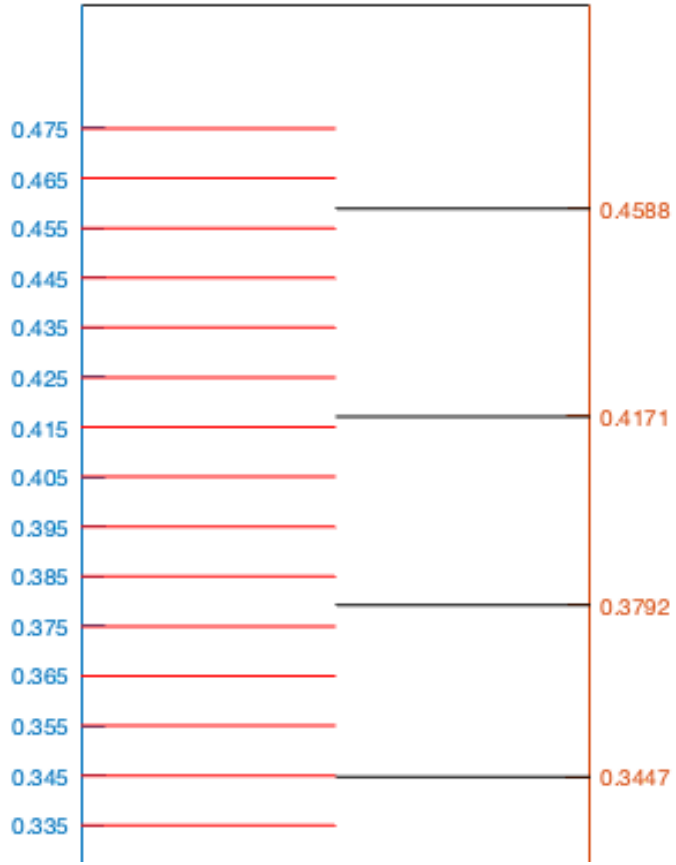
Number of directional bins



The recommended frequency increments are 1.07 and 1.1

WW3 VS OBS


frequency [Hz] - Linear
frequency [Hz] - log
0.475
0.465
0.455
0.445
0.435
0.425
0.415
0.405
0.395
0.385
0.375
0.365
0.355
0.345
0.335
0.325
0.315
0.305
0.295
0.285
0.275
0.265
0.255
0.245
0.235
0.225
0.215
0.205
0.195
0.185
0.175
0.165
0.155
0.145
0.135
0.125
0.115
0.105
0.095
0.085
0.075
0.065
0.055
0.045
0.035
0.4588
0.4171
0.3792
0.3447
0.3134
0.2849
0.2590
0.2355
0.2141
0.1946
0.1769
0.1608
0.1462
0.1329
0.1208
0.1098
0.0999
0.0908
0.0825
0.0750
0.0682
0.0620
0.0564
0.0512
0.0466
0.0424
0.0385
0.035


0.475
0.465
0.455
0.445
0.435
0.425
0.415
0.405
0.395
0.385
0.375
0.365
0.355
0.345
0.335
0.4588
0.4171
0.3792
0.3447

Linear
Log
0.085
0.075
0.065
0.055
0.045
0.035
Δf
Δf
0.0908
0.0825
0.0750
0.0682
0.0620
0.0564
0.0512
0.0466
0.0424
0.0385
0.035
Inc^3*f$_0$
Inc^2*f$_0$
Inc*f$_0$
First frequency (f$_0$)

HOW TO PROVIDE BOUNDARY CONDITIONS:

1- From a larger scale model:

In this case, the spectral resolution of the spectrum from larger scale simulation should match the local scale one, defined in ww3_grid.nml(inp). If not, you can interpolate them.

2- From observations:

- The full spectrum from observations such as NDBC can be used, however, it should be interpolated to match the model spectral resolution.
- The bulk stats (significant wave height, peak period and mean wave direction) from observations and Jonswap or Pietson Moskowitz assumption can be used to generate synthetic boundary conditions.

What format:

WW3 has two programs that pre-process ascii and netcdf formats:

- ww3_bounc: netcdf
- ww3_bound: ascii

WW3_BOUNC: PROCESS NETCDF FILES

```
! -------------------------------------------------------------- !
! WAVEWATCH III - ww3_bounc.nml - Boundary input post-processing       !
! -------------------------------------------------------------- !
! -------------------------------------------------------------- !
! Define the input boundaries to preprocess via BOUNC_NML namelist
!
! * namelist must be terminated with /
! * definitions & defaults:
!     BOUND%MODE              = 'WRITE'          ! ['WRITE'|'READ']
!     BOUND%INTERP            = 2                ! interpolation [1(nearest),2(linear)]
!     BOUND%VERBOSE             = 1              ! [0|1|2]
!     BOUND%FILE              = 'spec.list'      ! input _spec.nc listing file
! -------------------------------------------------------------- !
&BOUNC_NML
/
! -------------------------------------------------------------- !
! WAVEWATCH III - end of namelist                                !
! -------------------------------------------------------------- !
```

If you have more than one spec files, listed in spec.list, you can define either linear or nearest point for interpolation.

WW3 SPEC FILE (NETCDF)

Navigate to *boundary_condtion/bounc* and use ncdisp('bound.nc') command in matlab or ncdump –h bound.nc in terminal:

```
Dimensions:
        time      = 25    (UNLIMITED)
        station   = 1
        string16  = 16
        frequency = 32
        direction = 36
Variables:
   time
        Size:       25x1
        Dimensions: time
        Datatype:   double
        Attributes:
                long_name     = 'julian day (UT)'
                standard_name = 'time'
                units         = 'days since 1990-01-01 00:00:00'
                conventions   = 'Relative julian days with decimal part (as parts of the day)'
                axis          = 'T'
   station
        Size:       1x1
        Dimensions: station
        Datatype:   int32
        Attributes:
                long_name  = 'station id'
                _FillValue = -2147483647
                axis       = 'X'
   string16
        Size:       16x1
        Dimensions: string16
        Datatype:   int32
        Attributes:
                long_name  = 'station_name number of characters'
                _FillValue = -2147483647
                axis       = 'W'
   station_name
        Size:       16x1
        Dimensions: string16,station
        Datatype:   char
        Attributes:
                long_name  = 'station name'
                content    = 'XW'
                associates = 'station string16'
```

```
longitude
        Size:       1x25
        Dimensions: station,time
        Datatype:   single
        Attributes:
                long_name     = 'longitude'
                standard_name = 'longitude'
                globwave_name = 'longitude'
                units         = 'degree_east'
                scale_factor  = 1
                add_offset    = 0
                valid_min     = -180
                valid_max     = 360
                _FillValue    = 9.969209968386869e+36
                content       = 'TX'
                associates    = 'time station'
   latitude
        Size:       1x25
        Dimensions: station,time
        Datatype:   single
        Attributes:
                long_name     = 'latitude'
                standard_name = 'latitude'
                globwave_name = 'latitude'
                units         = 'degree_north'
                scale_factor  = 1
                add_offset    = 0
                valid_min     = -90
                valid_max     = 180
                _FillValue    = 9.969209968386869e+36
                content       = 'TX'
                associates    = 'time station'
   frequency
        Size:       32x1
        Dimensions: frequency
        Datatype:   single
        Attributes:
                long_name     = 'frequency of center band'
                standard_name = 'sea_surface_wave_frequency'
                globwave_name = 'frequency'
                units         = 's-1'
                scale_factor  = 1
                _FillValue    = 9.969209968386869e+36
                axis          = 'Y'
```

```
direction
        Size:       36x1
        Dimensions: direction
        Datatype:   single
        Attributes:
                long_name     = 'sea surface wave to direction'
                standard_name = 'sea_surface_wave_to_direction'
                globwave_name = 'direction'
                units         = 'degree'
                scale_factor  = 1
                add_offset    = 0
                valid_min     = 0
                valid_max     = 360
                _FillValue    = 9.969209968386869e+36
                axis          = 'Z'
   efth
        Size:       36x32x1x25
        Dimensions: direction,frequency,station,time
        Datatype:   single
        Attributes:
                long_name     = 'sea surface wave directional variance spectral
density'
                standard_name =
'sea_surface_wave_directional_variance_spectral_density'
                globwave_name = 'directional_variance_spectral_density'
                units         = 'm2 s rad-1'
                scale_factor  = 1
                add_offset    = 0
                valid_min     = 0
                valid_max     = 1.000000020040877e+20
                _FillValue    = 9.969209968386869e+36
                content       = 'TXYZ'
                associates    = 'time station frequency direction'
   dpt
        Size:       1x25
        Dimensions: station,time
        Datatype:   single
        Attributes:
                long_name     = 'depth'
                standard_name = 'depth'
                globwave_name = 'depth'
                 valid_min    = -100
                valid_max     = 10000
                _FillValue    = 9.969209968386869e+36
                 associates   = 'time station'
```

wnd
Size: 1x25
Dimensions: station,time
Datatype: single
Attributes:
long_name = 'wind speed at 10m'
standard_name = 'wind_speed'
globwave_name = 'wind_speed'
units = 'm s-1'
scale_factor = 1
add_offset = 0
valid_min = 0
valid_max = 100
_FillValue = 9.969209968386869e+36
content = 'TX'
associates = 'time station'

wnddir
Size: 1x25
Dimensions: station,time
Datatype: single
Attributes:
long_name = 'wind direction'
standard_name = 'wind_from_direction'
globwave_name = 'wind_from_direction'
units = 'degree'
scale_factor = 1
add_offset = 0
valid_min = 0
valid_max = 360
_FillValue = 9.969209968386869e+36
content = 'TX'
associates = 'time station'

cur
Size: 1x25
Dimensions: station,time
Datatype: single
Attributes:
long_name = 'sea water speed'
standard_name = 'sea_water_speed'
globwave_name = 'sea_water_speed'
units = 'm s-1'
scale_factor = 1
add_offset = 0
valid_min = 0
valid_max = 100
_FillValue = 9.969209968386869e+36
content = 'TX'
associates = 'time station'

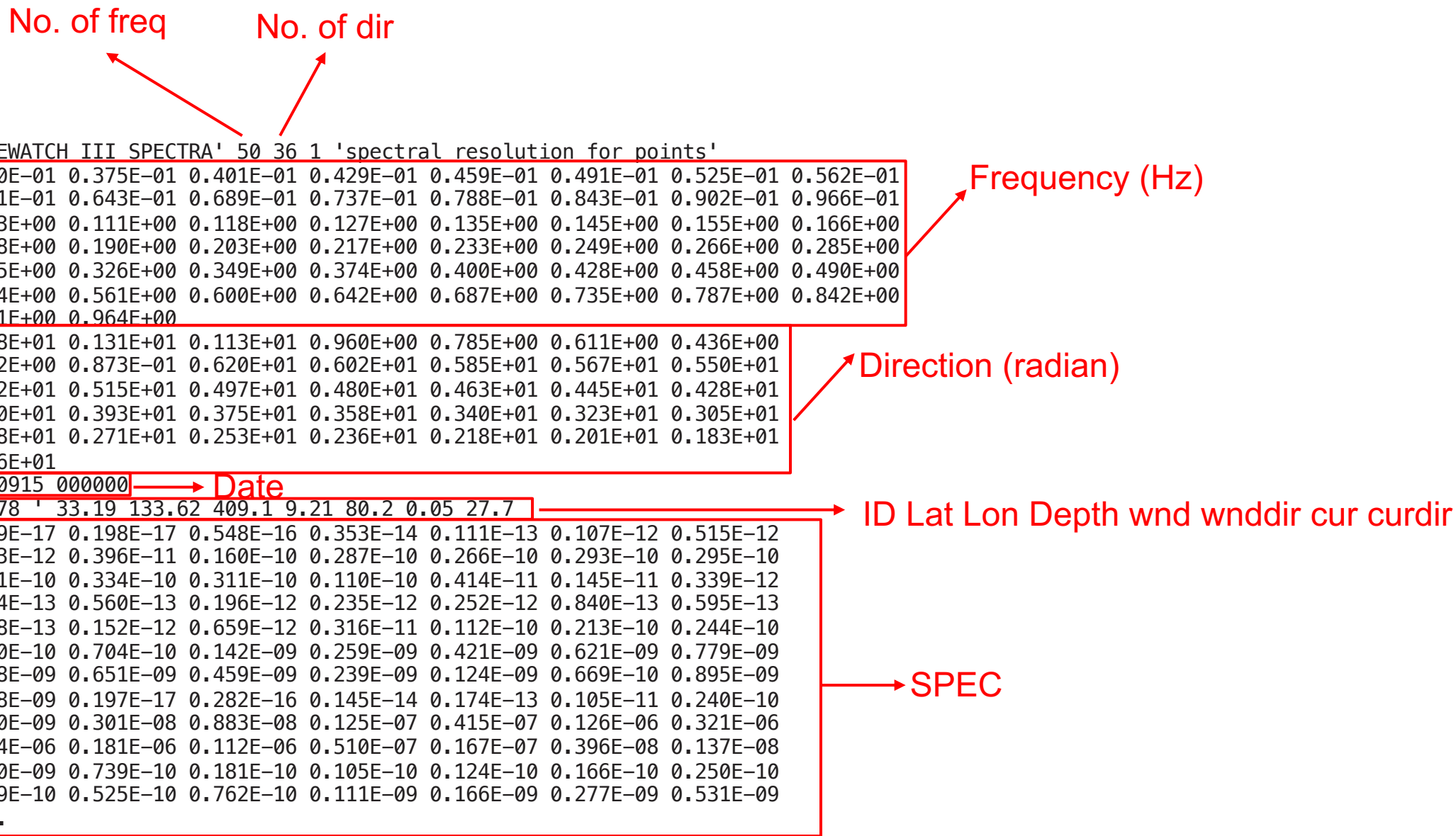
curdir
Size: 1x25
Dimensions: station,time
Datatype: single
Attributes:
long_name = 'direction from of sea water velocity'
standard_name = 'direction_of_sea_water_velocity'
globwave_name = 'direction_of_sea_water_velocity'
units = 'degree'
scale_factor = 1
add_offset = 0
valid_min = 0
valid_max = 360
_FillValue = 9.969209968386869e+36
content = 'TX'
associates = 'time station'

WW3_BOUND: PROCESS ASCII FILES

```
! ------------------------------------------------------------------------------ !
! WAVEWATCH III - ww3_bound.nml - Boundary input post-processing                 !
! ------------------------------------------------------------------------------ !
! ------------------------------------------------------------------------------ !
! Define the input boundaries to preprocess via BOUND_NML namelist
!
! * namelist must be terminated with /
! * definitions & defaults:
!     BOUND%MODE              = 'WRITE'          ! ['WRITE'|'READ']
!     BOUND%INTERP            = 2                ! interpolation [1(nearest),2(linear)]
!     BOUND%VERBOSE             = 1              ! [0|1|2]
!     BOUND%FILE              = 'spec.list'      ! input _spec.spc listing file
! ------------------------------------------------------------------------------ !
&BOUND_NML
/
! ------------------------------------------------------------------------------ !
! WAVEWATCH III - end of namelist                ------------------------------- !
! ------------------------------------------------------------------------------ !
```

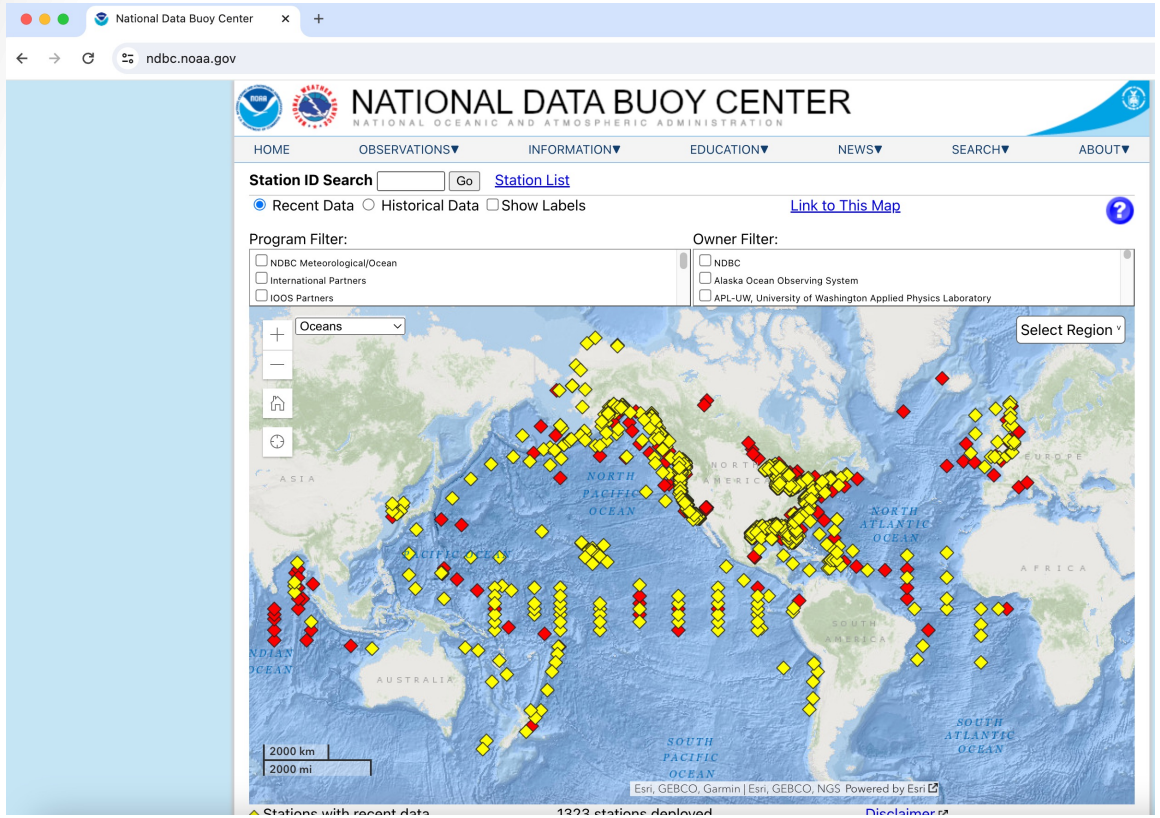
If you have more than one spec files, listed in spec.list, you can define either linear or nearest point for interpolation.

WW3 SPEC FILE (ASCII)

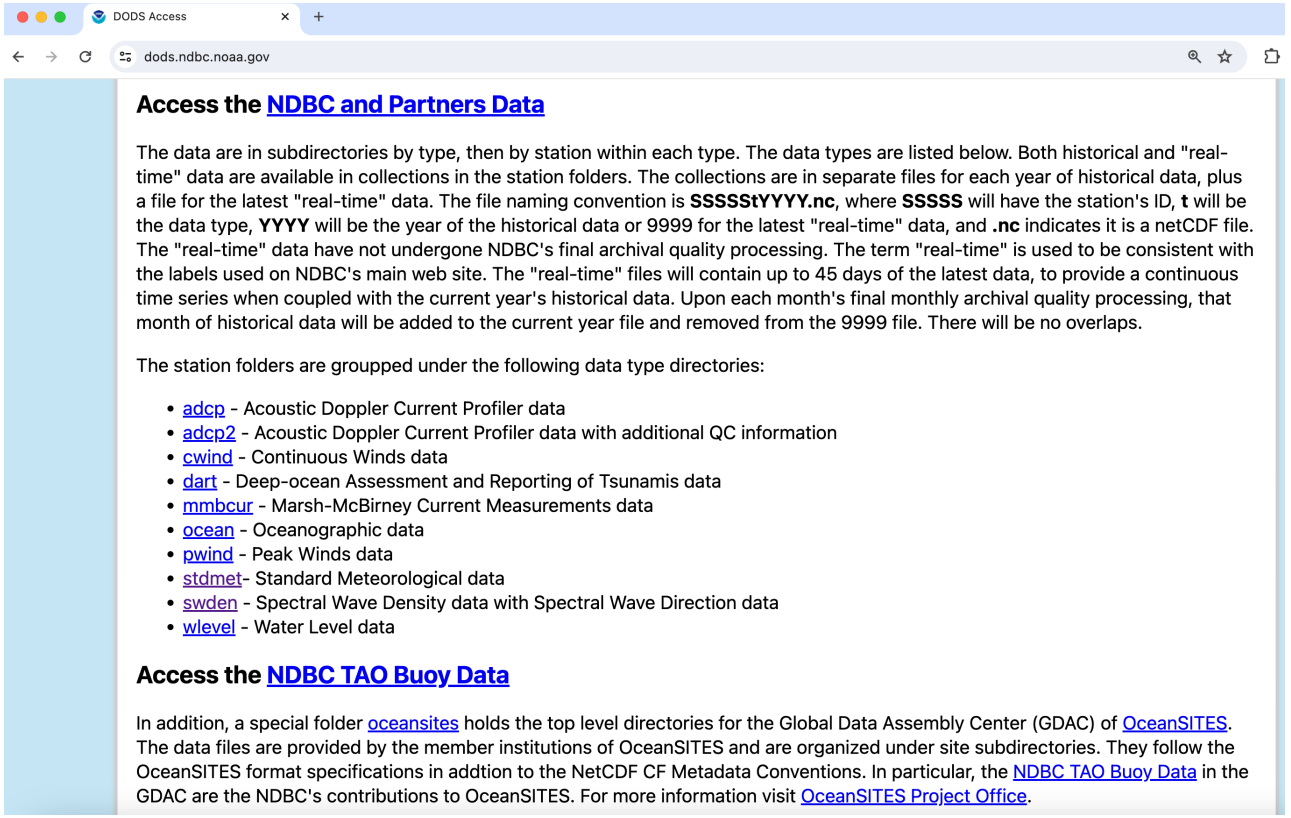


BUOY DATA

https://www.ndbc.noaa.gov/



https://dods.ndbc.noaa.gov/



BUOY DATA

α_1: mean wave direction
α_2: principle wave direction
R_1, R_2: parameters which describe the directional spreading about the main direction. These two parameters are the first and second normalized polar coordinates of the Fourier coefficients and are nondimensional.
θ: Azimuth angle measured clockwise from true North to the direction wave is from.
$D(\theta, f) = \frac{1}{\pi}[0.5 + R_1 \cos(\theta - \alpha_1)] + R_2 \cos(2(\theta - \alpha_2))$: Direction Spread
$E(\theta, f) = S(f)D(\theta, f)$: Directional Spectral Energy
$D(\theta, f)$ can take on negative values because of the trigonometric sine and cosine functions. There are several approaches to prevent or deal with the negative values. For more information and discussion of some approaches see: Use of advanced directional wave spectra analysis methods, M. D. Earle, K. E. Steele, and D. W. C. Wang, Ocean Engineering, Volume 26, Issue 12, December 1999, Pages 1421-1434.

NDBC - FULL SPEC (NETCDF)

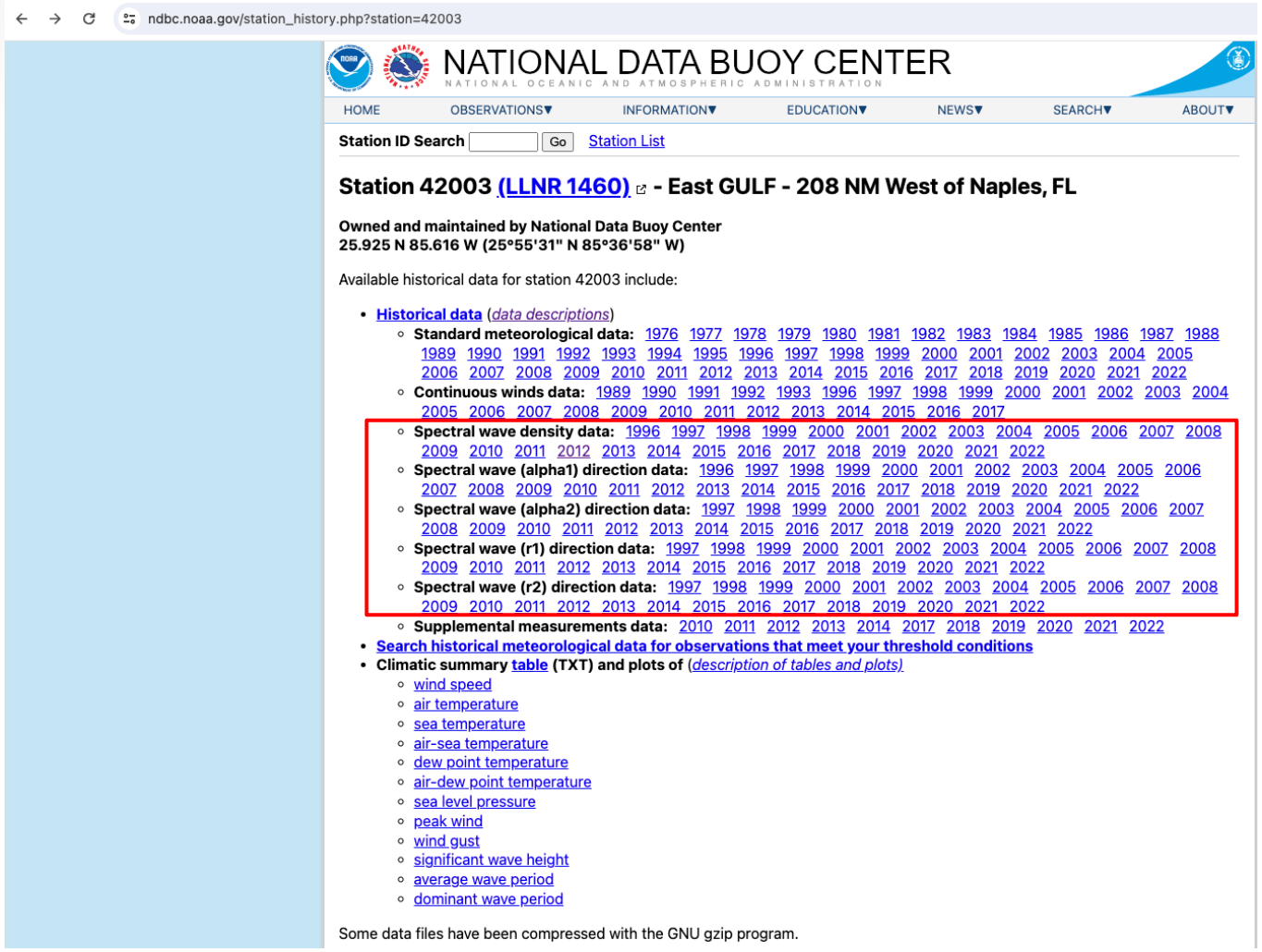
Navigate to *boundary_condtion/bounc* and use ncdisp('42003w2018.nc') command in matlab or ncdump –h 42003w2018.nc in terminal

```
Dimensions:
        time      = 7341  (UNLIMITED)
        frequency = 47
        latitude  = 1
        longitude = 1
Variables:
   time
        Size:       7341x1
        Dimensions: time
        Datatype:   int32
        Attributes:
                long_name     = 'Epoch Time'
                short_name    = 'time'
                standard_name = 'time'
                units         = 'seconds since 1970-01-01 00:00:00 UTC'
   frequency
        Size:       47x1
        Dimensions: frequency
        Datatype:   single
        Attributes:
                long_name     = 'Frequency'
                short_name    = 'frequency'
                standard_name = 'frequency'
                units         = 'Hz'
   latitude
        Size:       1x1
        Dimensions: latitude
        Datatype:   single
        Attributes:
                long_name     = 'Latitude'
                short_name    = 'latitude'
                standard_name = 'latitude'
                units         = 'degrees_north'
   longitude
        Size:       1x1
        Dimensions: longitude
        Datatype:   single
        Attributes:
                long_name     = 'Longitude'
                short_name    = 'longitude'
                standard_name = 'longitude'
                units         = 'degrees_east'
```

```
spectral_wave_density
        Size:       1x1x47x7341
        Dimensions: longitude,latitude,frequency,time
        Datatype:   single
        Attributes:
                long_name     = 'Spectral Wave Density'
                short_name    = 'swden'
                standard_name = 'spectral_wave_density'
                units         = '(meter * meter)/Hz'
                _FillValue    = 999
   mean_wave_dir
        Size:       1x1x47x7341
        Dimensions: longitude,latitude,frequency,time
        Datatype:   int32
        Attributes:
                long_name     = 'Mean Wave Direction'
                short_name    = 'alpha1'
                standard_name = 'mean_wave_direction'
                units         = 'degrees_true'
                _FillValue    = 999
   principal_wave_dir
        Size:       1x1x47x7341
        Dimensions: longitude,latitude,frequency,time
        Datatype:   int32
        Attributes:
                long_name     = 'Principal Wave Direction'
                short_name    = 'alpha2'
                standard_name = 'principal_wave_direction'
                units         = 'degrees_true'
                _FillValue    = 999
   wave_spectrum_r1
        Size:       1x1x47x7341
        Dimensions: longitude,latitude,frequency,time
        Datatype:   single
        Attributes:
                long_name     = 'First Normalized Polar Coordinate Derived from the Fourier Coefficients'
                short_name    = 'r1'
                standard_name = 'wave_spectrum_r1'
                _FillValue    = 999
   wave_spectrum_r2
        Size:       1x1x47x7341
        Dimensions: longitude,latitude,frequency,time
        Datatype:   single
        Attributes:
                long_name     = 'Second Normalized Polar Coordinate Derived from the Fourier Coefficients'
                short_name    = 'r2'
                standard_name = 'wave_spectrum_r2'
                _FillValue    = 999
```

NDBC - FULL SPEC (ASCII)

https://www.ndbc.noaa.gov/station_history.php?station=42003



NDBC – BULK STATISTICS (NETCDF)

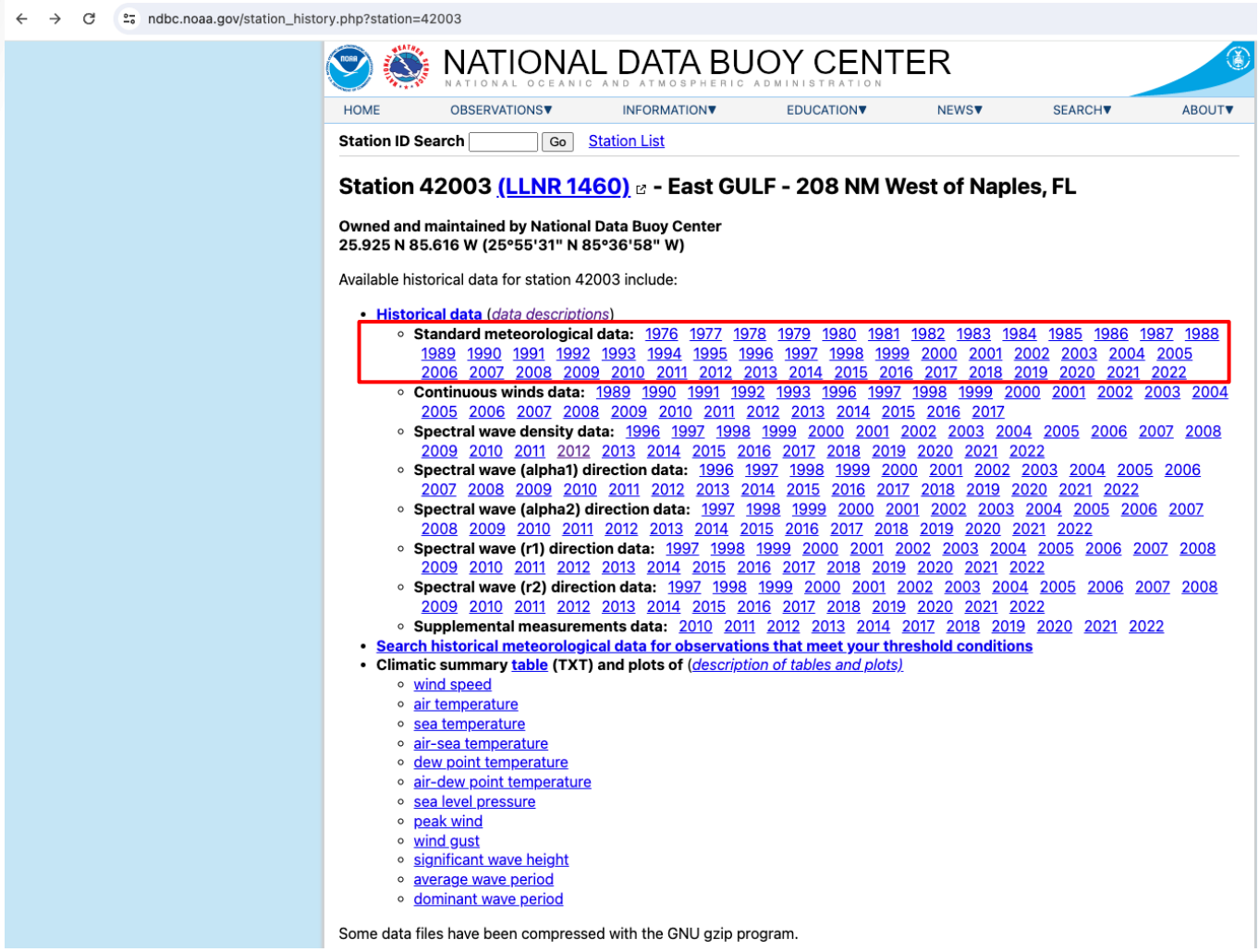
Navigate to *boundary_condtion/bounc* and use ncdisp('42003h2018.nc') command in matlab or ncdump –h 42003h2018.nc in terminal

```
Dimensions:
        time      = 44373 (UNLIMITED)
        latitude  = 1
        longitude = 1
Variables:
   time
        Size:       44373x1
        Dimensions: time
        Datatype:   int32
        Attributes:
                long_name     = 'Epoch Time'
                short_name    = 'time'
                standard_name = 'time'
                units         = 'seconds since 1970-01-01 00:00:00 UTC'
   latitude
        Size:       1x1
        Dimensions: latitude
        Datatype:   single
        Attributes:
                long_name     = 'Latitude'
                short_name    = 'latitude'
                standard_name = 'latitude'
                units         = 'degrees_north'
   longitude
        Size:       1x1
        Dimensions: longitude
        Datatype:   single
        Attributes:
                long_name     = 'Longitude'
                short_name    = 'longitude'
                standard_name = 'longitude'
                units         = 'degrees_east'
   wind_dir
        Size:       1x1x44373
        Dimensions: longitude,latitude,time
        Datatype:   int32
        Attributes:
                long_name     = 'Wind Direction'
                short_name    = 'wdir'
                standard_name = 'wind_from_direction'
                units         = 'degrees_true'
                _FillValue    = 999
   …
```

```
wave_height
        Size:       1x1x44373
        Dimensions: longitude,latitude,time
        Datatype:   single
        Attributes:
                long_name     = 'Significant Wave Height'
                short_name    = 'wvht'
                standard_name = 'significant_height_of_wave'
                units         = 'meters'
                _FillValue    = 99
   dominant_wpd
        Size:       1x1x44373
        Dimensions: longitude,latitude,time
        Datatype:   single
        Attributes:
                long_name     = 'Dominant Wave Period'
                short_name    = 'dpd'
                standard_name = 'dominant_wave_period'
                units         = 'seconds'
                _FillValue    = 99
   average_wpd
        Size:       1x1x44373
        Dimensions: longitude,latitude,time
        Datatype:   single
        Attributes:
                long_name     = 'Average Wave Period'
                short_name    = 'apd'
                standard_name = 'average_wave_period'
                units         = 'seconds'
                _FillValue    = 99
   mean_wave_dir
        Size:       1x1x44373
        Dimensions: longitude,latitude,time
        Datatype:   int32
        Attributes:
                long_name     = 'Mean Wave Direction'
                short_name    = 'mwd'
                standard_name = 'mean_wave_direction'
                units         = 'degrees_true'
                _FillValue    = 999
water_level
        Size:       1x1x44373
        Dimensions: longitude,latitude,time
        Datatype:   single
        Attributes:
                long_name     = 'Tide Water Level'
                short_name    = 'tide'
                standard_name = 'water_level'
                units         = 'feet'
                _FillValue    = 99
```

NDBC – BULK STATISTICS (ASCII)

https://www.ndbc.noaa.gov/station_history.php?station=42003



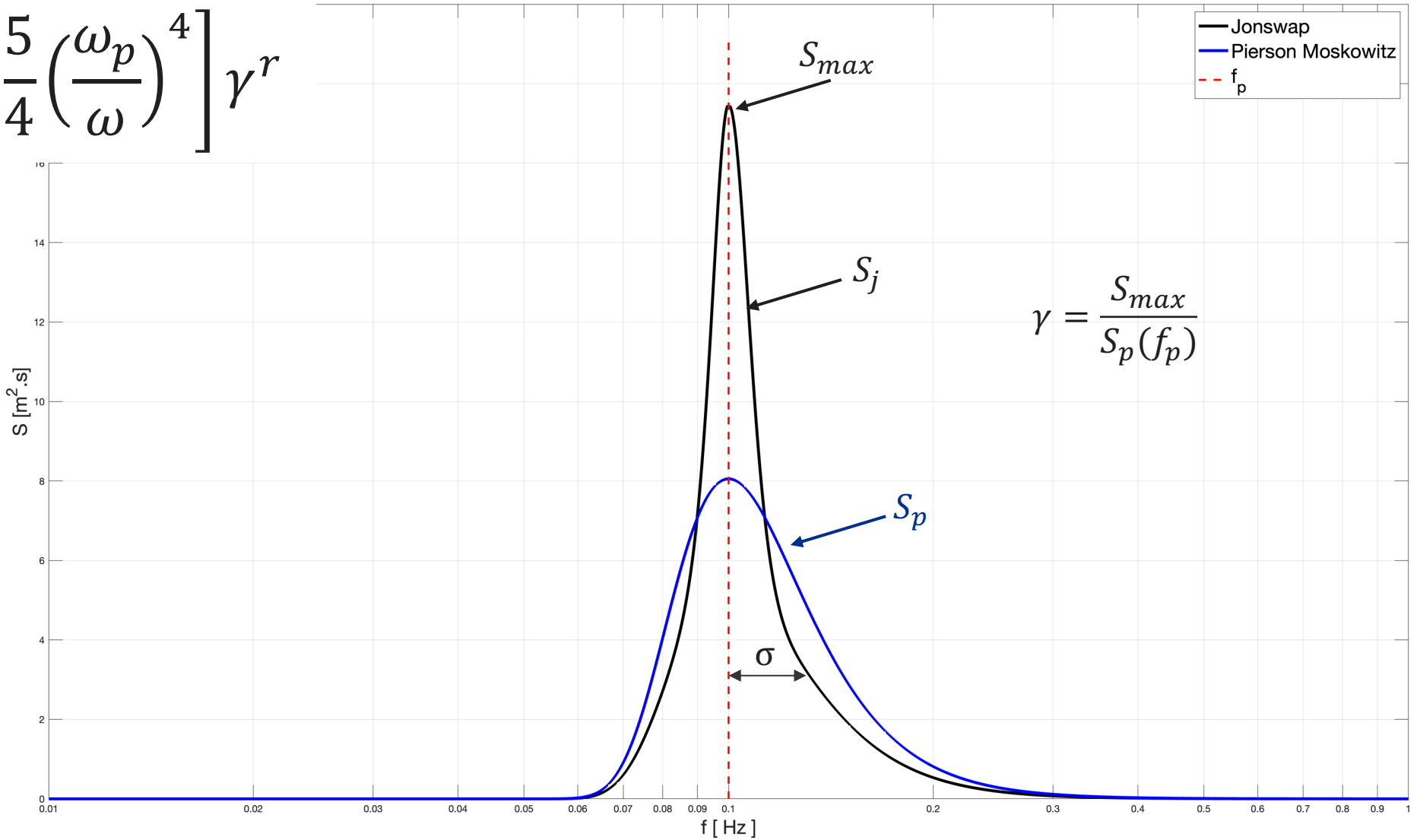
JONSWAP & PIERSON-MOSKOWITZ

$$S(\omega) = \frac{\alpha g^2}{\omega^5} \exp\left[-\frac{5}{4}\left(\frac{\omega_p}{\omega}\right)^4\right] \gamma^r$$

$$r = \exp[-\frac{(\omega - \omega_p)^2}{2\sigma^2\omega_p^2}]$$

$$\sigma = \begin{cases} 0.07 & \omega < \omega_p \\ 0.09 & \omega \geq \omega_p \end{cases}$$

$$\gamma = \begin{cases} 3.3 & JONSWAP \\ 1 & Pierson\ Moskowitz \end{cases}$$



$$\gamma = \frac{S_{max}}{S_p(f_p)}$$

INTERPOLATION

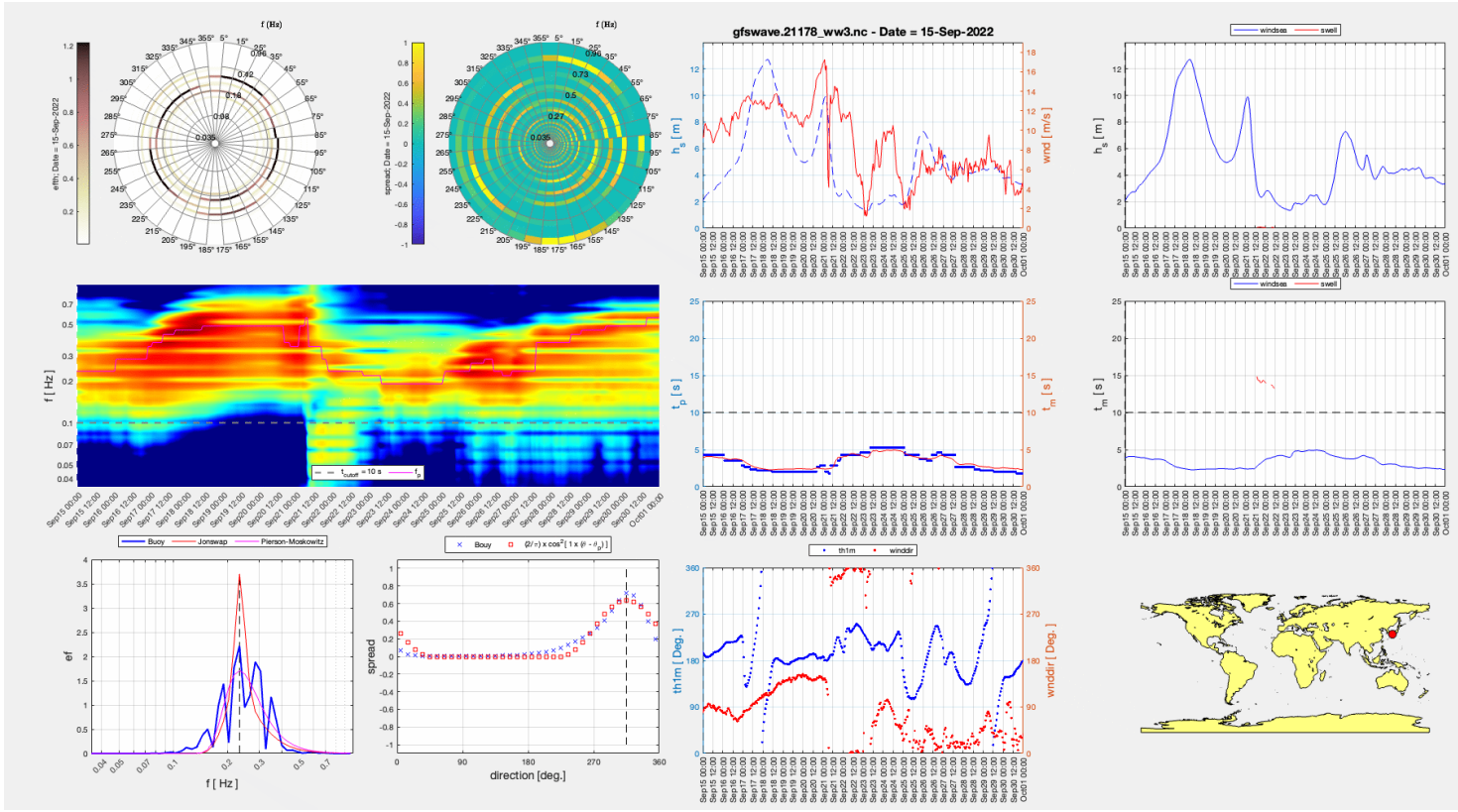
For ASCII format, there are scripts which read the ascii (either spec from WW3 or NDBC) and write them into netcdf format.
Then, interpolation scripts are used to match the spectral resolution (netcdf to netcdf).

WW3 full spec (ASCII)

```
cd boundary_condition/bound
[IN] = swden_ww3_read_ascii('gfswave.21178.spec');
testcase='boundary'; % test case
coordinate='spherical'; % coordinate : Spherical, cartesian
filenameIN='gfswave.21178_ww3.nc';% name of netcdf file (boundary)
[filename] = write_directional_spectra_nc(filenameIN,testcase,...
IN.pointID,IN.lat,IN.lon,IN.dpt,IN.wnd,IN.wnddir,IN.cur,IN.curdir,...
IN.time,IN.frequency,IN.direction,IN.efth,coordinate);
```

To visualize it, you can try the following:

```
input_nc='gfswave.21178_ww3.nc'
ID=1
START='20220915 000000'
END='20221001 000000'
factor=1
dt=1/24
dtplot=1/2
visualize='true'
```

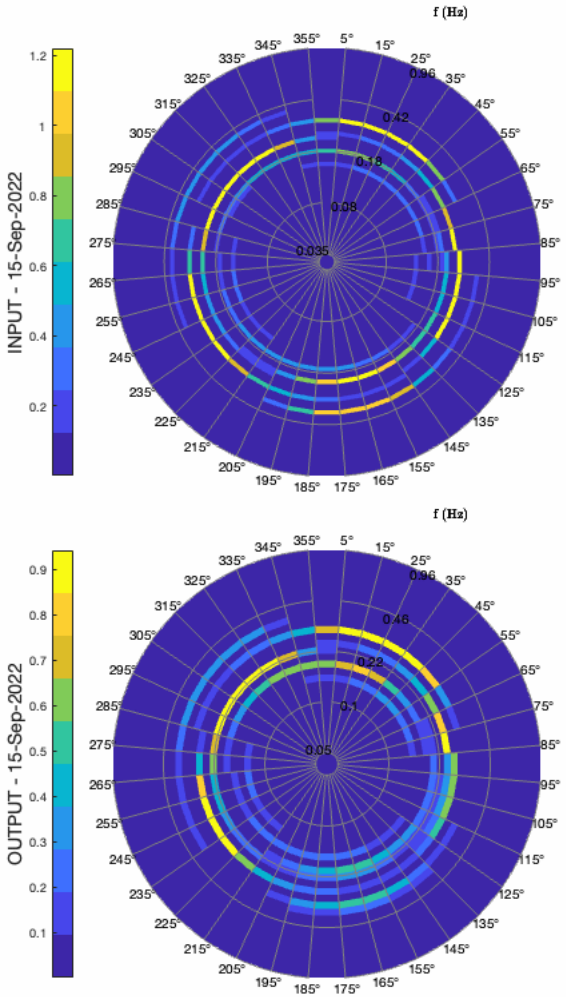



```
[SWDEN]= plot_ww3_spec(input_nc,ID,START,END,factor,dt,dtplot,visualize)
```

SPEC INTERPOLATION

To interpolate on a target spectral resolution:

```
nfreq=32; % number of frequencies
nDir=36; % number of Directions
inc=1.1; % frequency increment
f0=0.05;
filenameIN='gfswave.21178_ww3.nc';% name of netcdf file
filenameOUT='bgfswave.21178_interp_ww3.nc';% name of netcdf file
coordinate='spherical'; % coordinate : Spherical, cartesian
visualize='true';
deltatheta=360/nDir;
[IN,OUT]= spec_interp_nc(filenameIN,filenameOUT,nfreq,f0,inc,nDir,...
coordinate,visualize);
```



INTERPOLATION

NDBC full spec (NETCDF)

```
cd boundary_condition/bounc
nfreq=32; % number of frequencies
nDir=36; % number of Directions
inc=1.1; % frequency increment
f0=0.05;
deltatheta=360/nDir;
theta0=5;
freq=f0;
for i=1:nfreq-1
freq(end+1,1)=freq(end)*inc;
end
[SWDEN] = swden_ndbc_read('42003w2018.nc',deltatheta,theta0,freq);
```

ADD REQUIRED VARIABLES

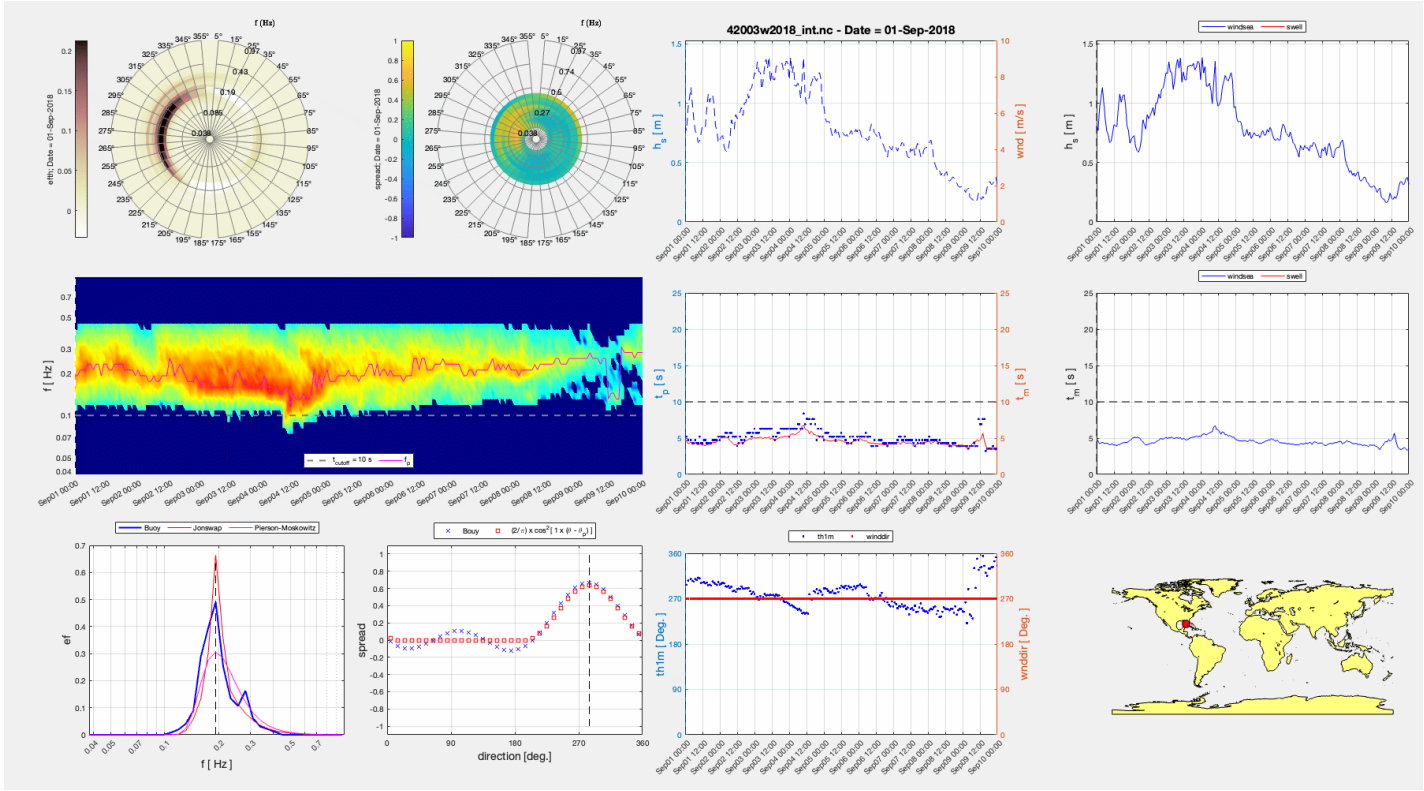
```
%% input data %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
nfreq=35;                  % number of frequencies
nDir=36;                   % number of Directions
Dir0=5;                    % first direction (deg)
inc=1.1;                   % frequency increment
f0=0.038;
filename='42003w2018_int.nc';% name of netcdf file (boundary)
testcase='42003';           % test vase
pointID='42003';           % id (length should be 16)
Lat=25.925;                 % latitude of BC (deg)
Lon=-85.616;               % longitude of BC (deg)
dpt=3273;                    % Depth of BC (m)
wndspd=0;                  % wind speed (m/s)
wnddir=270;                % wind direction (deg)
curspd=0;                  % current velocity (m/s)
curdir=270;                % current direction (deg)
coordinate='spherical';    % coordinate : Spherical, cartesian
ncf='42003w2018.nc';       % input netcdf file from https://dods.ndbc.noaa.gov/
visualize='true';          % options:true, false- this requires polarPcolor function and can be obtained from ...
                           % https://www.mathworks.com/matlabcentral/fileexchange/49040-pcolor-in-polar-coordinates
deltatheta=360/nDir;            % DeltaDir
theta0=5;                  % first Dir
%-------------------------------------------------------------%
%-------------------------------------------------------------%
% frequency (log)
Omega(1)=2*pi*f0;
for i=2:nfreq
Omega(i)=inc*Omega(i-1);
end
freq=Omega/2/pi;
```

ADD REQUIRED VARIABLES

```
pointID = [pointID,repmat(' ', [1, 16-strlength(pointID)])];
Lon=Lon*ones(1,length(SWDEN.Int.time));
Lat=Lat*ones(1,length(SWDEN.Int.time));
curspd=curspd*ones(1,length(SWDEN.Int.time));
curdir=curdir*ones(1,length(SWDEN.Int.time));
wndspd=wndspd*ones(1,length(SWDEN.Int.time));
wnddir=wnddir*ones(1,length(SWDEN.Int.time));
dpt=dpt*ones(1,length(SWDEN.Int.time));
time(1,:)=SWDEN.Int.time;
dir=pi*SWDEN.Int.Dir/180; %radian
dir0=SWDEN.Int.Dir;%degree
EFTH(:,:,1,:)=SWDEN.Int.DENS; %directional spectral density time series
%-----------------------------------------------------%
display (['Generating ', filename,' ...'])
%dump into netcdf
[filename] = write_directional_spectra_nc(filename,testcase,...
pointID,Lat,Lon,dpt,wndspd,wnddir,curspd,curdir,time,...
SWDEN.Int.f,dir0,EFTH,coordinate);
```

To visualize it, you can try the following:

```
input_nc= '42003w2018_int.nc';
ID=1
START= '20180901 000000'
END= '20180910 000000'
factor=1
dt=1/24
dtplot=1/2
visualize='true'
```

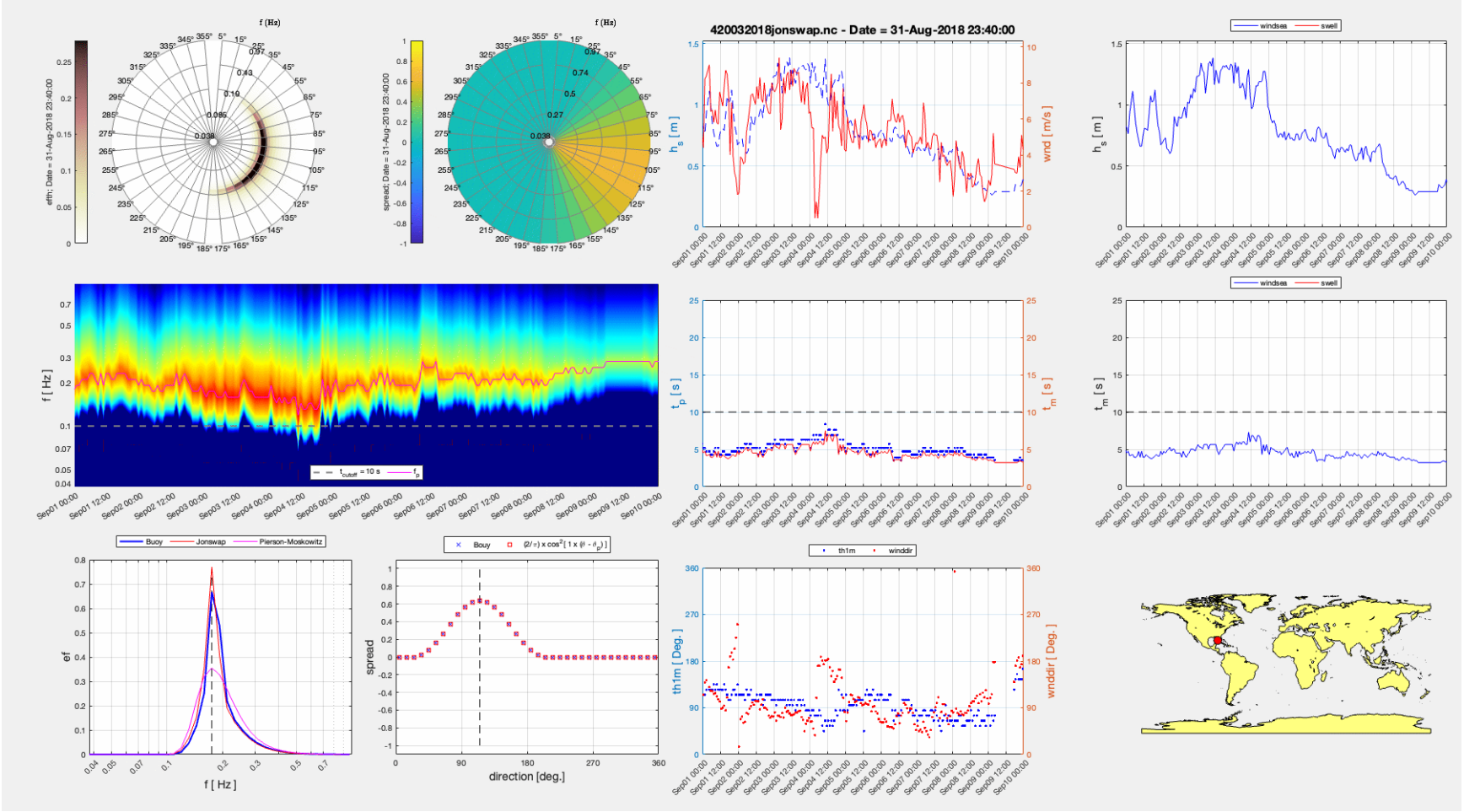



```
[SWDEN]= plot_ww3_spec(input_nc,ID,START,END,factor,dt,dtplot,visualize)
```

GENERATE BC FROM BULK STATS AT NDBC (JONSWAP)

```
input='42003h2018.nc'


factor=1
nfreq=35
nDir=36
inc=1.1
f0=0.038
method1='JONSWAP'
output1='420032018jonswap.nc'
coordinate='spherical'
START='20180901 000000'
END='20180910 000000'
dt=1/24
dtplot=1/2
visualize='true'
```

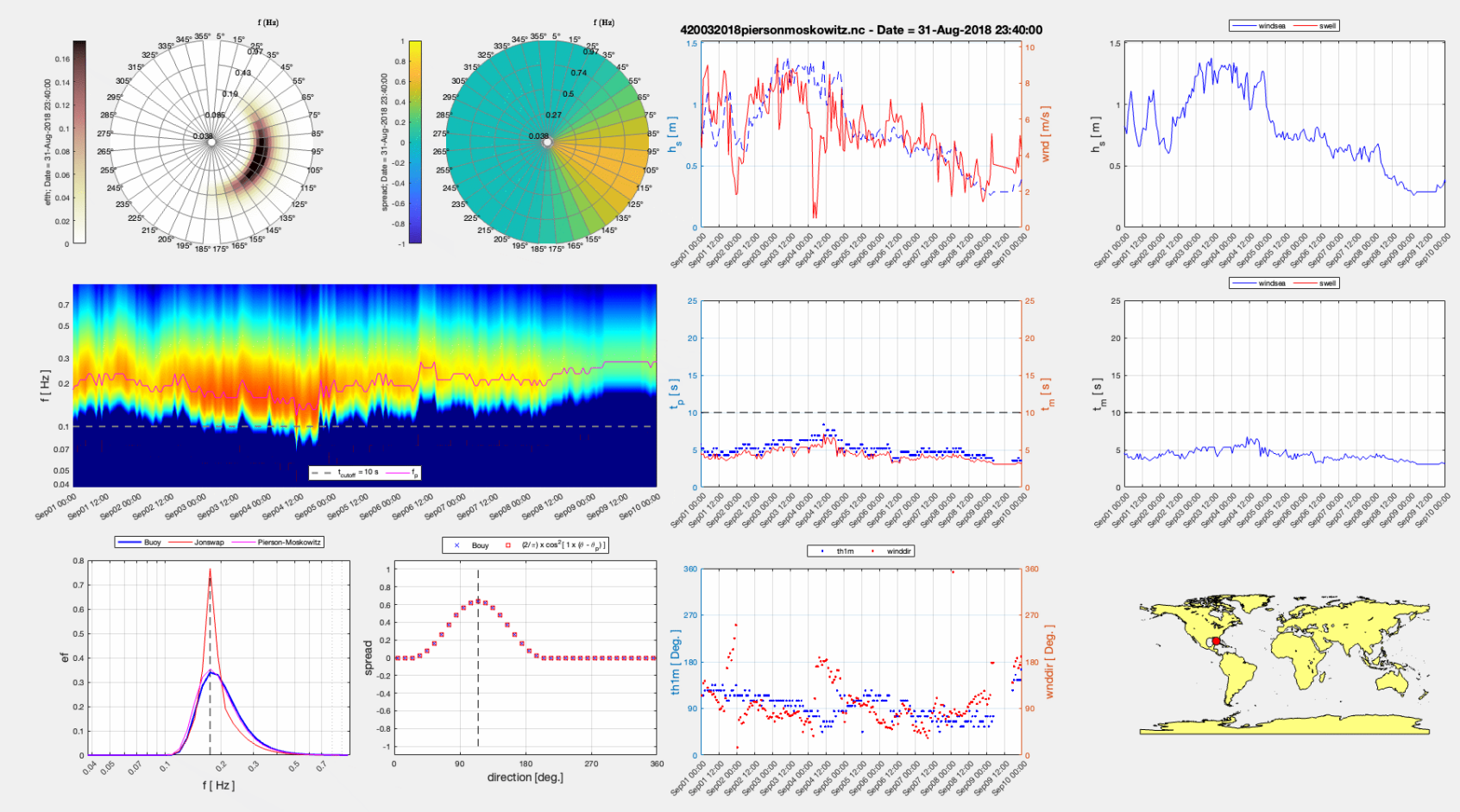



```
[SWDEN1]= stden_2_swden(input,output1,factor,nfreq,f0,inc,...
nDir,method1,coordinate,START,END,dt,dtplot,visualize);
```

GENERATE BC FROM BULK STATS AT NDBC (PIERSON MOSKOWITZ)

```
input='42003h2018.nc'


factor=1
nfreq=35
nDir=36
inc=1.1
f0=0.038
method2='PIERSONMOSKOWITZ'
output2='420032018piersonmoskowitz.nc
coordinate='spherical'
START='20180901 000000'
END='20180910 000000'
dt=1/24
dtplot=1/2
visualize='true’
```




```
[SWDEN2]= stden_2_swden(input,output2,factor,nfreq,f0,inc,...
nDir,method2,coordinate,START,END,dt,dtplot,visualize);
```