US Army Corps of Engineers®

# U.S. Army Corps of Engineers Standard Operating Procedures for Assimilation of NOAA National Data Buoy Center Archive Data

*Candice Hall*
*candice.hall@usace.army.mil*

**SYNOPSIS:**

The U.S. NOAA National Data Buoy Center (NDBC) releases duplicate meteorological and wave measurement data in multiple archive locations. Each data source has their own idiosyncrasies that cause differences within data with matching dates and times, even when collected by the same sensor at the same location. These variances need to be accounted for when using these NBDC data within U.S. Army Corps of Engineers (USACE) Engineers and Research Development Center (ERDC) products. These data issues are clearly outlined within the following USACE document: ERDC/CHL CHETN-I-100 [1].

These NDBC archive errors[1] necessitated the development of in-house USACE quality control (QA/QC) checks and metadata corrections to develop a best available measurement archive (herewith called the USACE QCC measurement archive) that includes comprehensive collection metadata. Data within this archive are sourced from the NDBC website historical station pages (https://www.ndbc.noaa.gov/) and the office NOAA archives stored on the National Center for Environmental Information (NCEI) servers (https://www.ncei.noaa.gov/access/marine-environmental-buoy-database/).

These self-described USACE QCC measurement data are stored alongside the USACE Coastal and Hydraulic Laboratory (CHL) Thredds WIS long-term hindcast, accessible to both the USACE and the public. Near-real time data updates to this USACE QCC measurement archive are completed annually.

This Standard Operating Procedure (SOP) outlines the processing steps necessary to produce these annual updates. To prevent knowledge loss, this SOP also includes instructions on how to re-process the full historical dataset. For full comprehension, review this SOP in conjunction with an associated USACE QCC NDBC Archive R scripts pdf.

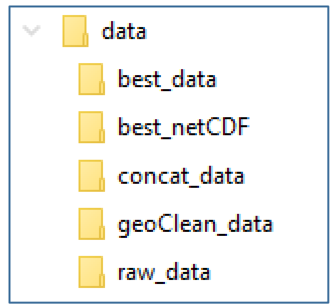
# Table of Contents

**USACE QCC Measurement Archive Procedure:**

This SOP details the steps required to combine NDBC data sources; check for geographical consistency and QA/QC the data; assign verified metadata to the individual NDBC data variables; and develop a best available dataset for storage in netCDF format for each NDBC buoy station.

The procedure is as follows using the depicted folder structure:

Step 1: Download data from NDBC website and NCEI data sources.
Step 2: Concatenate individual NDBC and NCEI datasets for each variable.
Step 3: Verify individual NCEI variable metadata with independent, manually created NDBC spreadsheets.
Step 4: Compare, geographically QA/QC the datasets by removing data with erroneous GPS positions and outliers, and assign metadata to the datasets.
Step 5: Create the best available NDBC dataset with verified NCEI metadata.
Step 6: Build year_month NDBC netCDF files for storage on the CHL Thredds server.




# These steps are automated using a product suite of scripts developed in R software[2].
# The scripts automatically build the folder structure as required.
# Individual scripts are structured to run in stand-alone or function mode.
# A master script utilizes the function mode of these scripts and enables the product suite to be run in serial on a local computer or in parallel on a Department of Defense (DoD) High Performance Computing system (HPC).
# All scripts are well described with comments to explain each section of code.
# All scripts have a functionality that allows for the export of progress text files that capture processing messages for later viewing.
# Individual script details and processing steps are outlined within the both the master script and the individual script sections below.
# R software uses a forward slash to delineate folders.
# Refer to the following NDBC documents for NDBC data collection procedures and quality control practices:
  - NDBC, 2009: Handbook of Automated Data Quality Control Checks and Procedures. NDBC Technical Document 09-02, U.S. National Oceanic and Atmospheric Administration, National Weather Service, National Data Buoy Center. https://www.ndbc.noaa.gov/NDBCHandbookofAutomatedDataQualityControl2009.pdf
  - NDBC, 2003: Nondirectional and Directional Wave Data Analysis Procedure. NDBC Technical Document 03-01, U.S. National Oceanic and Atmospheric Administration, National Weather Service, National Data Buoy Center (request from NDBC, only the 1996 version is available online: https://www.ndbc.noaa.gov/wavemeas.pdf)

**Running the product suite**

Review this SOP in conjunction with the associated USACE_QCC_NDBC_Archive_R_ scripts.pdf.

**Serial on a local computer**:
# To run the product suite via the function modes, enable the 'buoy list' code section within the master script. Follow the directions in the Master Script section below.
# To run the product suite in stand-alone mode, follow the stand-alone instructions within each script.

**Parallel on an HPC**:
# Due to limited R software[2] version on the Department of Defense (DoD) HPC in 2021, the ncdf4 package is unavailable. Therefore, only steps 2-5 can be completed on the HPC.
# To run the product suite in parallel on an HPC, enable the 'read file name' section within the master script.
# De-select the functional steps that you would like to run within your master script.
# Save this version of your master script as 'run_*xxxx*' in your data directory.
# Use the script 'copy_rename_run_scripts.R' to create an individually named master script for each NDBC buoy station (e.g. 'run_41001.R', 'run_41002.R', 'run_41003.R'). This allows the HPC to run these function codes in parallel on multiple processors without requiring copies of the individual step functions for each station.

HPC requirements
# To run these buoy stations in parallel, the HPC runs require 'submit_runs_*x*.sh' scripts (Appendix A). These scripts identify charge codes for the run; the number of cpu and processors to engage; the time limit for the runs; etc. These codes are HPC specific.
# To increase run speeds, multiple 'submit_runs_*x*.sh' scripts call 'run_buoy_*x*.sh' scripts, which in turn call the individual 'run_*xxxx*.R' master scripts that were created above. Appendix A and the 'USACE_QCC_NDBC_Archive_R_scripts.pdf' provide examples of these scripts.
# Transfer all relevant data, HPC 'submit_runs_*x*.sh', 'run_buoy_*x*.sh', and 'run_*xxxx*.R' scripts over to your HPC working folder. Submit runs into the HPC queue as normal.
# Contact candice.hall@usace.army.mil with any issues.

------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------
**Procedure Outline: Master Script (run_*xxxx*)**
------------------------------------------------------------------------------------------------------------
------------------------------------------------------------------------------------------------------------

This master script initiates the step function scripts within this product suite within the correct order. This script can run all the buoy stations in serial, or can be copied to run the function scripts on the HPC in parallel. All directories are generated automatically during script runs.

Open the master script ('run_*xxxx*') and follow along in the code as you review this section.

To allow for master script comprehension, the section outlines the individual step 1-7 functionalities. However, specific details of the procedures incorporated within each step are found within the relevant script sections of this document.

---

To run the master script:

- # install and load the relevant libraries
- # select the following data locations and start date
- # to run all of the buoy stations in serial, select the code to read in the NDBC buoy list
- # to run all of the buoy stations in serial, select the code that assigns the names of this master files to the scripts to isolate a specific buoy number

---

**Step 1: Download datasets from NDBC and NCEI websites**

This step downloads the datasets from the NDBC website historical station pages, and the official NOAA archives that are stored on the NCEI servers. The NDBC website stores data in compressed text files on an annual basis, except for the present year, which has data stored on a monthly basis[1]. The NCEI data are stored in Unidata's Network Common Data Form (netCDF) files on a monthly basis[1].

- # The code function for this step is called 'download_data_1.R', located within a user defined data directory (*data_dir*) and is loaded as follows:
  R code: source(paste0(data_dir,"download_data_1.R"))
- # To run, this function requires the buoy list, a start date, and a data directory as follows:
  R code: download_data_1(buoys, start_year = start_date, data_dir)
- # The script also needs access to the NDBC_buoy.csv spreadsheet that is found in the ./data directory. If NDBC adds new deployment stations, this file should be amended.
- # Due to HPC R ncdf4 package restrictions in 2021, this step cannot be completed on the HPC.

---

**Step 2: Concatenate individual NDBC and NCEI datasets for each variable**

This step merges each yearly and monthly data files to produce a single time series of concatenated NDBC standard meteorological (stdmet) data and time series files for each individual spectral wave variable from the two data sources. Each variable is treated uniquely to allow for

- future comparisons between NDBC and NCEI datasets,
- individual metadata assignment, and
- to facilitate month-long Thredds server files that span NDBC servicing interruptions

- # The code function for this step is called 'concat_data_2.R', and is loaded as follows:
  R code: source(paste0(data_dir,"concat_data_2.R"))
- # To run, this function requires a list of buoy, a start date and a data directory as follows:
  R code: concat_data_2(buoys, start_year = start_date, data_dir)

---

**Step 3: Verify NCEI variable metadata**

This step is applied solely to the NDBC netCDF data files to validate the netCDF metadata with NDBC-sourced, buoy specific metadata spreadsheets. These metadata were sourced from the NDBC database and original NDBC service technician log books, and represent the most accurate sensor information.

- # The code function for this step is called 'verify_netcdf_3.R', and is loaded as follows:
  R code: source(paste0(data_dir,"verify_netcdf_3.R"))
- # To run, this function requires a buoy list and a data directory as follows:
  R code: verify_netcdf_3(buoys, data_dir)

**Notes**:

a. Be sure to download the latest NDBC metadata sheets from the shared NDBC – CHL Google Drive (individual access required, contact Candice Hall, candice.hall@usace.army.mil, for details). These spreadsheets are updated upon request by Steven DiNapoli, steven.dinapoli@noaa.gov.

----------------------------------------------------------------------------------------------------------

**Step 4: Compare, geographically QC and assign metadata to the datasets**

Although these data originate from the same sensor, storage protocols resulted in different time stamps for each dataset[1]. This step compares the NDBC and NCEI sourced data by matching the datasets on 'nearest' date and time (to the minute). Each dataset is filtered to remove GPS positions and associated data that are not within a one degree radius of the NDBC station watch circles (the surface area through which a buoy can oscillate while tethered to specific location by a mooring). Once the data are geographically quality controlled, this step assigns verified metadata to the stdmet datasets.

- # The code function for this step is called 'geoClean_data_4.R', and is loaded as follows:
  R code: source(paste0(data_dir,"geoClean_data_4.R"))
- # To run, this function requires a buoy list and a data directory as follows:
  R code: geoClean_data_4(buoys, data_dir)

**Notes**:

Compare step:

a. A number of earlier datasets that were found within the NCEI datasets were not housed on the NDBC website. These data are likely from experimental buoy deployments that were intentionally never released to the public, as sensors and processing techniques were tested. If viable, these early data are included in the USACE QCC wave measurement archive with QC flags that rate them as unreliable. For more information on these earlier datasets, please reference the technical note on utilizing NDBC data, ERDC/CHL CHETN-I-100[1]. It is unlikely this will occur in the future as NDBC monitors what data are transmitted to the NDBC website, removing any non-public releasable data.
b. Manual confirmation and insertion of the multiple station locations were sourced from NDBC-developed, buoy specific metadata spreadsheets. This manual step is especially relevant for buoys that have not consistently held their stations due to high vandalism rates or heavy currents.

c. Geographical positions were filtered to remove data points that were outside a radius of one geographical degree (approximately 60 nautical miles) from the median GPS positions of these data. This generous buffer of two degrees (approximately 120 nautical miles) allows for shifting of NDBC deployment locations over the decades. This one degree radius was selected after tests exploring geographical radii of 0.25, 0.5, and 0.75 degrees removed significant portions of good data collected during the early NDBC deployment era. Users may wish to further filter their specific datasets to remove additional data points that are outside the target deployment locations, a task that is now easily achievable using the fully-described, verified metadata included within this USACE QCC wave measurement archive.

--------------------------------------------------------------------------------------------------------------

**Step 5: Create the best available NDBC dataset with verified NCEI metadata**
This step combines the geographically QA/QC datasets that were created in the step above. These best available, self-describing datasets include:

- NDBC website sourced wind direction, wind speed, wind gust, air pressure at sea level, air temperature, sea surface temperature, significant wave height, dominant and mean wave periods, mean wave direction, spectral c11, alpha1, alpha2, r1, r2, and their associated metadata.
- NCEI website sourced spectral c11m (m = uncorrected frequency spectra) and any data for the above variables that pre-date the NDBC datasets.

# The code function for this step is called ‘create_best_data_5.R’, and is loaded as follows:
R code: source(paste0(data_dir," create_best_data_5.R "))
# To run, this function requires a buoy list and a data directory as follows:
R code: create_best_data_5(buoys, data_dir)

--------------------------------------------------------------------------------------------------------------

**Step 6: Build NDBC netCDF files for storage on the CHL Thredds server**
This step creates monthly netCDF NDBC data files that collate all of the best available data and metadata variables that were selected in the step above.

# The code function for this step is called ‘build_thredds_netcdf_6.R’, and is loaded as follows:
R code: source(paste0(data_dir," build_thredds_netcdf_6.R "))
# To run, this function requires a buoy list and a data directory as follows:
R code: build_thredds_netcdf_6(buoys, data_dir)
# This function also needs access to the NDBC_buoy.csv and NDBC_buoy_descriptions.csv spreadsheets that are found in the /data directory.
# Due to USACE HPC R package restrictions in 2021, this step cannot be completed on the HPC.

--------------------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------------------

-----------------------------------------------------------------------------------------------------------

-----------------------------------------------------------------------------------------------------------

**Step Scripts**

-----------------------------------------------------------------------------------------------------------

-----------------------------------------------------------------------------------------------------------

**Step 1: Download functions**

-----------------------------------------------------------------------------------------------------------

# Due to the R software[2] ncdf4 package limitations on accessible USACE HPC in 2021, only steps 2-5 can be completed on the HPC.
# This function needs access to the NDBC_buoy.csv spreadsheet that is found in the ./data directory and is required prior to execution.

**NDBC data details[3]**:
# The NDBC website stores data (Table 1) in zipped yearly and monthly files as standard meteorological (stdmet), spectral wave density (swden), spectral wave (alpha1) direction (swdir), spectral wave (alpha2) direction (swdir2), spectral wave (r1) direction (swr1), and spectral wave (r2) direction data (swr2).
# These files require unzipping.
# Included within the NDBC stdmet datasets are collected meteorological and integral wave data in the following structure: wind direction (˚), wind speed (m/s), wind gusts (m/s), significant wave height (m), dominant wave period (seconds), average wave period (seconds), mean wave direction (˚), air pressure at sea level (hPa), air temperature (˚C), water temperature (˚C), dew point temperature (˚C), visibility (miles) and tide (ft).
# Visibility and tide are no longer collected by NDBC stations, and are disregarded.

| **Table 1. Example NDBC website download Uniform Resource Locators (URLs) for zipped text data files[1].** | |
|---|---|
| Yearly data file URL | https://www.ndbc.noaa.gov/data/historical/swden/46029w1996.txt.gz |
| Monthly data file URL | https://www.ndbc.noaa.gov/data/swden/Jan/4602912020.txt.gz |

**NCEI data details[3]**:
# The NCEI website stores monthly NDBC files per year in netCDF format.
# All available data and metadata are extracted from these netCDF files.
# These files contain the same NDBC data as listed above, but also include additional spectral parameters such as uncorrected spectral energy wave data ($c_{11}^{m}$), spectral wave co- and quad-spectra (‘$C_{12}$’, ‘$C_{13}$’, ‘$C_{22}$’, ‘$C_{33}$’, ‘$Q_{12}$’, ‘$Q_{13}$’), four wave data quality assurance parameters (‘$gamma_2$’, ‘$gamma_3$’, ‘$phi_h$’, ‘rhq’), and a legacy ‘sensor_output’ variable that indicates whether the wave spectral data are in displacement or acceleration.
# The uncorrected spectral wave energy is retained within the USACE QCC wave measurement archive to allow wave parameter re-calculations without the influences of NDBC shore-side processing protocols.
# Where available, multiple sensor data (primary and secondary) from redundant meteorological sensors deployed on the NDBC buoy stations, as well as the appropriate variable QA/QC and data release flags, are extracted.

# NDBC does not deploy dual wave systems on their buoy platforms, so the single sensor wave data, as well as the appropriate variable QA/QC and data release flags, are included in the extraction process.
# The NCEI netCDF file naming conventions (Table 2) and netCDF format (Appendix B and C) are different for files containing data from before and after January 2011.
# Each format requires format-specific code to extract the data from the file variable fields.
# Of note is that the post-2011 netCDF files are structured by ‘payloads’, then ‘sensor’ fields, and then ‘variable’ fields (Appendix B and C). This structure resembles the physical structure of the buoy stations with their multiple sensor attachments. Users have to dig into these payload and sensor subfields to discover the variable data with their associated metadata.
# As of 2022, NDBC is in the process of redesigning these netCDF file formats to be more user friendly. Release date of the new data format is unknown at this time.

**Table 2. Example NCEI download URLs for NDBC netCDF data files[1].**

| | |
|---|---|
| NCEI Download URL for Station 44014 prior to April 2020 | |
| Pre-2011 | https://data.nodc.noaa.gov/thredds/fileServer/ndbc/cmanwx/2006/12/44014_200612.nc |
| Post-2011 | https://data.nodc.noaa.gov/thredds/fileServer/ndbc/cmanwx/2011/01/NDBC_44014_201101_D1_v00.nc |
| NCEI Download URL for Station 41001 after April 2020 | |
| Pre-2011 | https://www.ncei.noaa.gov/data/oceans/ndbc/cmanwx/1980/01/41001_198001.nc |
| Post-2011 | https://www.ncei.noaa.gov/data/oceans/ndbc/cmanwx/2020/04/NDBC_41001_202004_D4_v00.nc |

**Notes**:
# The ‘download_data_1’ function requires a list of buoy stations, a start year, and a data directory. This script can be used as a function or for stand-alone use.
# To handle the various NCEI netCDF data formats, the ‘download_data_1’ script requires additional functions scripts that are located within the data folder: ‘read_NetCDF_pre_2011.R’; ‘read_NetCDF_post_2011.R’; ‘read_NetCDF_metadata.R’.

**Actions**:

*NDBC website download*:
a. The ‘download_data_1’ code initially loops through the list of buoy stations, years and months to download the NDBC website specific data files into a ‘zipped’ folder. Downloads are not initiated if the file is already present in the NDBC zipped folder.
b. Downloaded, zipped NDBC files are unzipped into an ‘unzipped’ folder. Similarly, files are not unzipped if already present in the NDBC unzipped folder.

*NCEI website download*:
c. The ‘download_data_1’ code toggles through the list of buoy stations, yearly folders and monthly files to download the NCEI stored netCDF data into a ‘netCDF’ folder. Downloads are not initiated if the file is already present in the NCEI netCDF folder.

- **Be aware that NDBC changes 'Dxx' extensions in NCEI NetCDF data. The code is currently set to test for 'D' extensions between 1 and 20**.

d. Pre-2011 netCDF file format:
   - Once downloads are complete, the 'download_data_1' function calls the 'read_NetCDF_pre_2011.R' to extracts station and variable data and metadata (variable attributes) from the netCDF files containing pre-2011 data.
   - Variable data are matched with extracted header information to remove any uncertainty in spectral frequency bands and stdmet variable units. This is in response to erroneous frequency bands that are found within the NCEI NDBC NetCDF archive files and appear to be an artifact that is introduced into the data during the netCDF creation process[1].
   - Where available, individual variable QA/QC and data release flags are included in the extraction process.

e. Post-2011 netCDF file format:
   - The 'download_data_1' function then calls the 'read_NetCDF_post_2011.R' to extract station and variable data and metadata (variable attributes) from the netCDF files containing post-2011 data.
   - Again, variable data are matched with extracted header information to remove any uncertainty in spectral frequency bands and stdmet variable units. This is in response to the erroneous frequency artifacts found within the NCEI NDBC NetCDF archive files.
   - Where available, multiple sensor data (primary and secondary) from redundant sensors deployed on the NDBC buoy stations are included in the extraction process.
   - Where available, individual variable QA/QC and data release flags are included in the extraction process.

f. Finally the 'download_data_1' calls the 'read_NetCDF_metadata.R' script to loop through the list of buoy stations to extract additional metadata from the netCDF files that were initially not captured (missing global and variable attributes).
   - This script searches all available global and variable attributes fields to extract buoy ID, depth, latitude, longitude, hull ID; payload ID; sensor descriptions; types; processing software; manufacturer; and deployment heights. These metadata are verified and augmented within Step 3 of this process.

---

**Step 2: Concatenate Datasets**

---

**Notes**:

# This step can be completed on a DoD accessible HPC.
# The 'concat_data_2.R' function requires a list of buoy stations, a start year, and a data directory. This script can be used as a function or for stand-alone use.
# To handle the various NDBC and NCEI netCDF data formats, the 'concat_data_2.R' script requires additional functions scripts that are located within the data folder: 'concat_ndbc.R', and 'concat_ncei.R'.

To handle the downloaded NDBC website data, this step allows for the management of:

# Differing yearly file formats and spectral frequencies (detailed within Appendix D);

# Concatenates multiple date and time columns into one field (detailed within Appendix D);
# Removes redundant dates, as well as visibility and tide columns in stdmet data;
# If necessary (not needed for NDBC data but code provision is in place), allocates spectral data into appropriate 38 frequencies (old wave sensors), and 47 frequencies (new wave sensors) (detailed within Appendix D); and
# Converts NDBC r1 and r2 values to their correct values (NDBC stored data are scaled by 100 to reduce storage requirements, so data are multiplied by 0.01).
# All stdmet and individual datasets are saved in an *x*.RData container file within buoy station specific folders.

To handle the NCEI data, this step:
# Selects only relevant data for concatenation (detailed within Appendix E).
# Concatenates stdmet data to match NDBC website data nomenclature;
# Applies NDBC netCDF QC flags to the extracted data;
# Converts pressure units to match NDBC data (Pa to hPa);
# Converts air, water and dew point temperatures from Kelvin to degree Celsius to match NDBC data;
# Removes zero ('0') wind gust values when no wind speed is present (data that weren't flagged in file).
# To handle the erroneous netCDF spectral frequency data that are likely introduced into the data during the netCDF creation process, this part of the code steps through each row of spectral data and attempt to match the available spectral frequency data to the appropriate 38 frequencies (old wave sensors) or 47 frequencies (new wave sensors) (detailed within Appendix F).
# Removes redundant netCDF data points that are ~5-10 seconds apart.
# All stdmet and individual datasets are saved in an *x*.RData container file within buoy station specific folders.

**Actions**:
a. The 'concat_data_2.R' function initially concatenates all available, historical NDBC website data before addressing the NCEI netCDF data concatenation.

*NDBC data concatenation*:
*NDBC Stdmet:*
b. NDBC data types and file label abbreviations: 'stdmet'/'h', 'swden'/'w', 'swdir'/'d', 'swdir2'/'i', 'swr1'/'j','swr2' /'k'. These abbreviations are used to isolate data within the concatenation process.
c. NDBC variables included within the stdmet files: 'wind_direction', 'wind_speed', 'wind_gust', 'significant_wave_height', 'dominant_period', 'average_period', 'mean_wave_direction', 'air_pressure', 'air_temperature', 'sea_surface_temperature', 'dew_point_temperature'.
d. NDBC spectral variables: 'c11' (wave energy density), 'c11m' (uncorrected wave energy density), 'alpha1' (spectral mean wave direction), 'alpha2' (spectral principal wave direction), 'r1' (first normalized polar coordinates of the Fourier coefficients), 'r2' (second normalized polar coordinates of the Fourier coefficients).

e. The ‘concat_data_2.R’ script loops through the data types and concatenates all data in sequence.
f. If the data type == ‘stdmet’/‘h’, the ‘concat_data_2.R’ script calls ‘concat_ndbc.R’ to initiate the stdmet portion of that script to concatenate the raw data.
    - The ‘concat_ndbc.R’ handles all stdmet date formats and returns a concatenated single dataset (detailed within Appendix D).
g. The ‘concat_data_2.R’ function then completes the following QA/QC checks:
    - Removes the obsolete visibility and tide columns from this stdmet dataset
    - Removes duplicated rows of data (previous year data rows frequently included within next annual file)
    - Renames variables for consistency between NDBC and NCEI data sources
    - Formats the stdmet dataset to include missing variable column (possible in older datasets)
    - Dataset columns are re-ordered for consistency
    - The code also removes redundant date/time records and order the datasets by date/time before exporting

*NDBC Spectral:*

h. If the data type is spectral (‘swden’/‘w’; ‘swdir’/‘d’; ‘swdir2’/‘i’; ‘swr1’/‘j’;’swr2’ /‘k’), the ‘concat_data_2.R’ script re-calls ‘concat_ndbc.R’ to initiate the spectral portion of that script to concatenate the raw data.
    - The ‘concat_ndbc.R’ handles all raw date/time formats and returns a concatenated single dataset for each variable (detailed within Appendix D).
    - Due to the use of differing wave sensors over the decades that collect data with different frequency ranges (38 vs 47 frequency bands), this code extracts the data from the annual or monthly files, and attempts to match the headers of the extracted data with the common old (38 frequencies) or new (47 frequency) bands (e.g. ‘spectral_dataset_old’, ‘spectral_dataset_new’). If neither match, these data are returned with file nomenclatures that include a count of their frequency bands for further absorption testing (e.g. ‘spectral_dataset_52cols’).
    - Once complete, these spectral datasets are returned to the ‘concat_data_2.R’ script for additional testing.
i. If no erroneous wave frequency data are detected within the station’s datasets, the ‘concat_data_2.R’ removes redundant date/time records and order the datasets by date/time for consistency before saving.
j. If erroneous wave frequency data are detected within the station’s datasets, the ‘concat_data_2.R’ applies additional tests in an attempt to ingest the data:
    - Reformat the odd frequencies by removing empty columns with only NA data throughout and testing new and old frequency compatibility.
    - Loops through individual rows within the odd frequencies datasets in an attempt a merge each row with new and old frequency datasets (using similar NA removal technique as above, but on a per row basis).
k. If erroneous wave frequency data are still present within the station’s datasets, they are exported using the previous ‘_*xx*cols’ suffix for manual review.
l. If the data type == ‘swden’/‘w’ OR ‘swr1’/‘j’:
    - The ‘concat_data_2.R’ script builds and exports a station specific file containing all available frequency bands within these raw data files. This spectral frequency band

extraction on multiple spectral datasets provides a comparison tool for the investigation of possibly erroneous frequency band values.

m. All stdmet and individual datasets are saved in an '*s_buoy#_ndbc_ALL.Rdata*' container file within buoy station specific folders.

*NCEI data concatenation*:
*NCEI Stdmet:*

n. After completing the NDBC website data concatenation for the entire station history, the 'concat_data_2.R' function initiates the NCEI netCDF extracted data concatenation process.
o. The 'concat_data_2.R' script calls 'concat_ncei.R' to initiate the stdmet portion of that script to concatenate the raw data.
p. For 'stdmet'/'h' data types, the 'concat_ ncei.R' function loads the previously extracted, station specific metadata spreadsheet from the download step.
    - The 'concat_ ncei.R' function filters through all of the unique extracted netCDF station data to select and concatenate the single or multiple sensor (primary and secondary) NDBC netCDF stdmet variables: 'lat', 'lon', 'wind_direction', 'wind_speed', 'wind_gust', 'significant_wave_height', 'dominant_period', 'average_period', 'mean_wave_direction', 'air_pressure', 'air_temperature', 'sea_surface_temperature', 'dew_point_temperature'.
    - After each individual single or multiple sensor (primary and secondary) variable dataset is concatenated, relevant metadata from the extracted netCDF spreadsheets are assigned.
    - Individual variable QA/QC and data release flags are assigned to the single or multiple sensor (primary and secondary) variable dataset.
    - The code then applies the following QA/QC procedures to the concatenated single or multiple sensor (primary and secondary) variable dataset:
        - Removes redundant date/time records and orders the datasets by date/time
        - Applies the NDBC netCDF QC flags to the extracted data; before removing the additional flag data from the individual variable dataset
        - Converts pressure units to match NDBC data (Pa to hPa)
        - Converts air, water and dew point temperatures from Kelvin to degree Celsius to match NDBC data
        - Removes zero ('0') wind gust values when no wind speed is present (data that weren't flagged in file)
        - Removes redundant netCDF data points that are ~5-10 seconds apart
    - Once each single or multiple sensor (primary and secondary) variable dataset has been concatenated and quality controlled, the concat_ ncei.R' function builds a dataset that resembles the NDBC website stdmet dataset, which is returned to the 'concat_data_2.R' script.
q. The 'concat_data_2.R' function performs the following additional QA/QC checks:
    - Renames variables for consistency between NDBC and NCEI data sources
    - Formats the stdmet dataset to include missing variable column (missing within netCDF files if sensors were offline)
    - Columns are re-ordered for consistency
    - The code also removes redundant date/time records and orders the datasets by date/time before saving

*NCEI spectral:*

r. If the data type is spectral, the 'concat_data_2.R' script re-calls 'concat_ncei.R' to initiate the spectral portion of that script to concatenate the raw data.
s. The 'concat_ ncei.R' function filters through all of the extracted unique netCDF station data to select and concatenate each spectral variable: ('c11', 'c11m', 'alpha1', 'alpha2', 'r1', 'r2', 'C12', 'C13', 'C22', 'C33', 'Q12', 'Q13', 'gamma2', 'gamma3', 'phih', 'rhq', or sensor_output).
t. If the data type == 'sensor_output', the 'concat_ncei.R' handles all raw date/time formats during concatenation and removes redundant date/time records and orders the datasets by date/time before returning to the 'concat_data_2.R' function.
u. For all other spectral variables:
   - The 'concat_ ncei.R' handles all raw date/time formats for concatenation
   - Due to the erroneous wave frequency bands that appear to be an artifact of the NetCDF creation process[1] (Appendix F), the 'concat_ ncei.R' attempts to match the headers of the extracted data with the common old (38 frequencies) or new (47 frequency) bands (e.g. 'spectral_dataset_old', 'spectral_dataset_new'). If neither match, these data are returned with file nomenclatures that include a count of their frequency bands for further absorption testing (e.g. 'spectral_dataset_52cols').
   - Once complete, these individual spectral datasets are returned to the 'concat_data_2.R' script for additional testing.
v. If no erroneous wave frequency data are detected within the station's spectral datasets, the 'concat_data_2.R' removes redundant date/time records and orders the datasets by date/time for consistency before saving.
w. If erroneous wave frequency data are detected within the station's datasets, the 'concat_data_2.R' applies additional tests in attempt to ingest the data:
   - Reformat the odd frequencies by removing empty columns with only NA data throughout and testing new and old frequency compatibility.
   - Loops through individual rows within the odd frequencies datasets in an attempt to merge each row with new and old frequency datasets (using similar NA removal technique as above, but on a per row basis).
x. If erroneous wave frequency data are still present within the station's datasets, they are exported using the previous '_*xx*cols' suffix for manual review.
y. All stdmet and individual datasets are saved in an '*s_buoy#_ncei_ALL.Rdata*' container file within buoy station specific folders.

---

**Step 3: Verify NCEI variable metadata**

---

**Notes**:

# This step can be completed on a USACE accessible HPC.
# This step is applied solely to the NDBC netCDF data files to validate the netCDF metadata with NDBC-sourced, buoy specific metadata spreadsheets.
# These metadata were sourced from the NDBC database and original NDBC service technician log books, and represent the most accurate sensor information.

# Be sure to download the latest NDBC metadata sheets from the shared NDBC – CHL Google Drive (individual access required, contact Candice Hall, candice.hall@usace.army.mil, for details).
# The 'verify_netcdf_3.R' function requires a buoy list and a data directory

**Actions**:

a. The 'verify_netcdf_3.R' function initially loads pre-set sensor heights that are sourced from available literature and verified with NDBC Engineer, Rodney Riley (Rodney.riley@noaa.gov).

b. Preset information, with references:
NDBC published sensor heights (August 3, 2016)[4] – text included here in case of removal from NDBC website:
*"At what heights are the sensors located on moored buoys?*
*Meteorological sensors are normally located at the ten meter level for the 10-meter and the 12-meter discus buoys and are at a nominal height of five meters for the 3-meter discus buoys and the 6-meter NOMAD buoys. However, barometers are located inside the hull at the water level. Sea surface temperature sensors are located at a depth of 1.5 meters for 10-m and 12-m buoys and at 1 meter for all 3-m and 6-m moored buoys. Sea surface temperature sensors on NDBC buoys are located near one meter below the water line but they vary by hull type. Current hull configurations for water temperature sensors are: for 2.4- and 3-meter hulls at 0.7 meters; for 6-meter hulls at 0.8 meters; for 10-meter hulls at 1.1 meters; and for 12-meter hulls at 0.9 meters. Historically, water temperature sensors for 3-meter "foam" hulls were at 0.75 meters and 1.8-meter hulls were at 0.35 meters."*

'verify_netcdf_3.R' presets (2021):

| Hull | Wind sensor height | Sea surface temperature (sst) sensor height | Reference |
|---|---|---|---|
| 12-m | 10 | 0.9 | NDBC, 2016[4] |
| 10-m | 10 | 1.1 | NDBC, 2016[4] |
| 6-m | 5 | 0.8 | NDBC, 2016[4] |
| 5-m | 5 | 0.7 | No information, Rodney Riley, NDBC Engineer (pers. comms. 03/02/2021) |
| 3-m | 5 | 0.7 | NDBC, 2016[4] |
| ELB* | 5.2 | 1.8 | Smith, 1985[5] |
| LNB** | 10 | 0.9 | NDBC, 1992[6] |
| 2.8-m | NA | NA | No information, Rodney Riley, NDBC Engineer (pers. comms. 03/02/2021) |
| 2.6-m | NA | NA | No information, Rodney Riley, NDBC Engineer (pers. comms. 03/02/2021) |
| 2.4-m | 3.3 | 0.7 | Gay, E., 2018'[7] |
| 2.3-m | 3.2 | 1.3 | Bouchard et al., 2017[8]; Gay, E., 2018'[7] |
| 2.1-m | 3.2 | 1.1 | Hall et al., 2018[9]; NDBC, 2021[10] |
| 1.8-m | 2.1 | 0.4 | Crout et al., 2008[11] |

* ELB: Exposed Location Buoy   ** LNB: Large Navigational Buoy

The 'verify_netcdf_3.R' function

c. Loads and formats the station specific metadata spreadsheet that was concatenated in step 1a above.
d. Selects the station specific NCEI stdmet data and removes NA and duplicate data
e. Concatenates and verifies wave metadata with NDBC metadata spreadsheets
f. Concatenates and verifies other dual meteorological sensor metadata with the NDBC metadata spreadsheets
g. Performs housekeeping tasks to remove excess sensor information labels that were included within the metadata extraction process.
h. Removes excess hull info and verifies sensor height information
i. Double-checks that NCEI netCDF extracted metadata fields no longer contain 'no available information'.
j. Filters for unique date/time data and reorders the datasets by date/time, before reordering the datasets to columns structures that match NDBC stdmet datasets.
k. The verified stdmet with newly verified metadata, as well as all of the individual spectral wave variable datasets, are saved in an '*s_buoy#_ncei_ALL_verified.Rdata*' container file within buoy station specific folders.

------------------------------------------------------------------------------------------------------------

**Step 4: Compare, geographically QA/QC and assign metadata to the datasets**

------------------------------------------------------------------------------------------------------------

**Notes**:

\# This step can be completed on a USACE accessible HPC.

\# Same sensor, storage protocols resulted in different time stamps for each dataset[1], so this script compares the NDBC and NCEI sourced data by matching the datasets on 'nearest' date and time.

\# This step geographically quality controls each dataset by removing GPS positions and associated data that are not within a pre-selected one degree radius (~120 nautical miles) of the NDBC station watch circles.

\# This step assigns verified metadata to the NDBC stdmet datasets.

\# The 'geoClean_data_4.R' requires a buoy list and a data directory.

\# If desired, the final section of code within the 'geoClean_data_4.R' script produces statistical comparisons and plots of the NDBC and NCEI datasets. A 'switch_set' turns this functionality on and off during stand-alone use of this script. To activate this functionality in HPC runs, the 'geoClean_data_4.R' will require and additional 'switch-set' input value to function.

\# As mentioned within the master script section, a number of earlier data points were identified within the NCEI data source than the NDBC website source; likely from sensors and processing technique test datasets that were intentionally never released to the public. These early data are included in the USACE QCC wave measurement archive with QC flags that rate them as unreliable. For more information on these earlier datasets, please reference the technical note on utilizing NDBC data, ERDC/CHL CHETN-I-1007.

\# Buoy stations that have historically not held their mooring positions were manually reviewed for accurate geographic location information.

**Actions**:

The 'geoClean_data_4.R' script **geographically quality controls** the datasets by removing data with erroneous GPS positions using the following steps:

a. Sets the pre-selected GPS_buffer (radius of one degree) and indicates whether the script to run the plots/stats block.
b. Loads the concatenated NDBC and verified NCEI datasets
a. Isolates the common station metadata from the verified NCEI variable data
b. Creates a unique reference position dataset containing date/time and NCEI GPS latitude and longitude fields
c. Filters position dataset to remove GPS positions that are not within a one degree radius (~120 nautical miles) of the NDBC station watch
d. Two methods were used to geographically QA/QC these data:
    - A sorted table of value occurrences to find the most common latitude and longitude positions (using the assumption that the buoy held its correct station for the majority of its life cycle);
    - Manual confirmation and insertion of the documented station location that were sourced from NDBC-sourced, buoy specific metadata spreadsheets. This manual step was especially relevant for buoys that historically did not consistently hold their stations due to high vandalism rates or heavy currents.

The 'geoClean_data_4.R' script **compares** the NDBC and NCEI datasets using the following steps:

e. Matches complementary datasets on 'nearest' date and time, handling instances of NDBC data that has no matching NCEI data, and NCEI data that pre-dates the NDBC datasets.
f. Geographically quality controls the data using the methods above, with manual verification from the reference position dataset where necessary.
g. Applies additional outlier QA/QC to the stdmet datasets.

The 'geoClean_data_4.R' script **assigns** the following verified NCEI stdmet **metadata** to the geographically quality controlled NDBC stdmet dataset:

- From NDBC-sourced, buoy specific metadata spreadsheets, specific station hull type, water depth, payload, mooring type are appended to the NDBC and NCEI stdmet datasets.
- From these NDBC spreadsheets and the verified NCEI netCDF metadata, the correct single or multiple sensor (primary and secondary) designation, including instrument processing systems (for waves) and instrumentation information (names, deployment heights etc.) are inserted into the NDBC stdmet datasets by matching the time paired NDBC variable values with the exact NCEI values (value-by-value comparison method).
- For computation efficiency, only a single variable for each sensor is tested for metadata assignment, as the metadata will be common for all of the same sensor variables.

h. The 'geoClean_data_4.R' script assigns these metadata by matching individual NDBC variable values with NCEI variable values, and then assigning the appropriate metadata to the NDBC metadata fields. This value-by-value comparison method is required due to the uncertainty of the data release flags in the NCEI netCDF files, and the need to match the appropriate single or multiple sensor (primary and secondary) metadata to the unknown sensor NDBC stdmet variable data.

i. This value-by-value comparison method handles single or multiple dual sensors (primary and secondary) by assigning metadata to the NDBC datasets with respect to the following scenarios. When scenarios match, the NDBC metadata field is populated and the assigned metadata field is cleared. These scenarios are intentionally ordered for computational efficiency:
    - matching NDBC values and NCEI primary sensor values where secondary data is NA
    - matching NDBC values and NCEI primary and secondary sensor values - accounts for NDBC's NCEI netCDF file creation practice of publishing duplicate data in multiple payload subsets
    - NDBC data that does not match NCEI primary and secondary sensor values - accounts for possible NDBC's NCEI netCDF file creation build errors
    - NDBC data that matches NCEI secondary values but does not match primary sensor - logical secondary sensor usage
    - NDBC data that matches NCEI primary but does not match secondary sensor values - logical primary sensor usage
j. The value-by-value comparison method then tests empty variable and metadata fields that possibly fall into the following categories:
    - NDBC data that does not match NCEI primary or secondary values but is still present - possible NDBC data QC
    - NDBC data that does not match NCEI primary values but is still present - possible NDBC rounding issues or data QC
    - NDBC data that does not match NCEI secondary values but is still present - possible NDBC rounding issues or data QC
    - Remaining NCEI primary sensor metadata with no matching NDBC values
    - Remaining NCEI secondary sensor metadata with no matching NDBC values
k. Finally, the value-by-value comparison method tests for possible post-real-time shoreside corrections to the NDBC sourced data that would not be reflected in the real-time NCEI netCDF files (aka, no matching NDBC and NCEI variable data):
    - Remaining NDBC values with no assigned metadata after the above assignments
    - Remaining NCEI metadata fields with no NDBC variable data matches
l. Once complete, the code checks for matching NDBC and NCEI values that inadvertently resulted in the incorrect metadata assignment, i.e. sensor metadata information that is not consistent with the metadata blocks above and below, as data release / sensors switches for a single hour of data are improbable. These tests are only applicable to the post 2011 data that contains dual sensor deployment.
m. The 'geoClean_data_4.R' script then assigns common metadata to the matching sensor variables e.g. wind speed metadata as wind direction and wind gust metadata, as well as significant wave height metadata as mean wave direction, dominant and average wave period metadata.
n. The code performs some housekeeping tasks: re-orders all stdmet dataset columns to be comparable; corrects NCEI datasets that contained primary sensor variable data that was overwritten by secondary sensor variable data in the netCDF file creation process; and saves all unique stdmet datasets.

o. The 'geoClean_data_4.R' script then applies the same geographical quality control techniques as above to the remaining NCEI unique datasets where there are no matching NDBC data, e.g. c11m, 'C12', 'C13', 'C22', 'C33', 'Q12', 'Q13', 'gamma2', 'gamma3', 'phih', 'rhq', or sensor_output.
p. Once complete, the code exports a table that details the historical, common metadata usage at each NDBC buoy station (mooring depth; type; buoy hull; and payload type). This table allows for a quick look at the history of individual NDBC buoy stations.
q. The compared, geographically quality controlled datasets with self-describing metadata are saved in '*s_buoy#_ndbc/ncei_comp_geoClean.Rdata*' container files within buoy station specific folders.

Plots and statistical outputs:
r. If desired, the final section of code within the 'geoClean_data_4.R' script produces statistical comparisons and plots of the NDBC and NCEI datasets.
s. A 'switch_set' turns this functionality on and off during stand-alone use of this script. To activate this functionality in HPC runs, the 'geoClean_data_4.R' will require an additional 'switch-set' input value to function.
t. The code produces location map plots of before and after the data that are geographically quality controlled.
u. The code also produces time series comparison plots and statistical results of the stdmet variables data. These steps aggregate the data for plotting on their specific length of records: annually for long time series data, monthly for shorter time series data, and daily for very short periods of time series data.
v. These plots and statistics are invaluable for manual verification of the geographical QA/QC steps above.

------------------------------------------------------------------------------------------------------------

**Step 5: Create the best available NDBC dataset with verified NCEI metadata**

------------------------------------------------------------------------------------------------------------

\# This step can be completed on a USACE accessible HPC.
\# This step selects the best available data from the compared, geographically quality controlled and self-describing datasets created in the previous steps.
\# The 'create_best_data_5.R' requires a buoy list and a data directory.

**Actions**:
a. The 'create_best_data_5.R' script loads the '*s_buoy#_ndbc/ncei_comp_geoClean.Rdata*' container files created in the previous step.
b. The script tests for any pre-NDBC data that may have been included within the NCEI data sources and reduces any multiple sensor data to a single, best value for each variable.
c. The script then selects all available NDBC data files, and any non-common NCEI data files for inclusion in the best available dataset.
d. The selected datasets with self-describing metadata are saved in 's*_buoy#_best_data.Rdata*' container file within buoy station specific folders.

---

**Step 6: Build NDBC netCDF files for storage on the CHL Thredds server**

---

# Due to the R software[2] ncdf4 package limitations on accessible DoD HPC in 2021, only steps 2-5 can be completed on the HPC.
# This step creates monthly netCDF NDBC data files that contains the best available data and metadata variables that were selected in the step above.
# The 'build_thredds_netcdf_6.R' function requires a buoy list and a data directory. The script also needs access to the 'NDBC_buoy.csv' and 'NDBC_buoy_descriptions.csv' spreadsheets that are found in the ./data directory.
# NetCDF files are subset to create individual station files for each month and year (year_month)
# NetCDF structure is cf compliant as per netCDF NCEI point reference: https://www.nodc.noaa.gov/data/formats/netcdf/v2.0/point.cdl
# NetCDF standard names are cf compliant as per: https://cfconventions.org/Data/cf-standard-names/78/build/cf-standard-name-table.html
# Flag conventions are consistent with: Paris. Intergovernmental Oceanographic Commission of UNESCO. 2013. Ocean Data Standards, Vol.3: Recommendation for a Quality Flag Scheme for the Exchange of Oceanographic and Marine Meteorological Data. (IOC Manuals and Guides, 54, Vol. 3.) 12 pp. (English.)(IOC/2013/MG/54-3)

**Actions**:
a. The 'build_thredds_netcdf_6.R' script starts by setting flag values and data precision.
b. Flag descriptions are based on the Paris. Intergovernmental Oceanographic Commission of UNESCO. 2013. Ocean Data Standards, Vol.3: Recommendation for a Quality Flag Scheme for the Exchange of Oceanographic and Marine Meteorological Data. (IOC Manuals and Guides, 54, Vol. 3.) 12 pp. (English.)(IOC/2013/MG/54-3).
c. The 'build_thredds_netcdf_6.R' script then loads the full suite of NDBC buoy station numbers, and a previously created 'NDBC_buoy_descriptions.csv' spreadsheets that are found in the /data directory.
d. The script then loops through the NDBC buoy station number to complete the netCDF build process.
e. For each buoy station, the 'build_thredds_netcdf_6.R' script filters the 'NDBC_buoy_descriptions.csv' data to isolate the specific station name and information.
f. The 'build_thredds_netcdf_6.R' script then loads the station specific '*s_buoy#_best_data.Rdata*' container files created in the previous step.
g. After a few housekeeping steps, the 'build_thredds_netcdf_6.R' script determines the full historical yearly date range.
h. The data are subset per year, and per month (year-month period) to build the final netCDF files.
i. The output directory and output file name is set: '*s_buoy#_best_ndbc_ncei_year#_month#.nc*'
j. The 'build_thredds_netcdf_6.R' script initially checks for the presence of data for that year-month period.
k. The script then re-confirms that any multiple sensor data are reduced to a single, best value for each variable.
l. The function loads buoy station specific metadata for concatenation with the stdmet data, and checks the datasets for wave sensor output data (1 = acceleration, 2 = displacement), which is

common in the older NDBC datasets when different wave sensors were deployed. No data are transformed to account for wave sensor output differences.

m. The 'build_thredds_netcdf_6.R' script subsets the spectral time variables, and loads spectral data for inclusion.
n. The 'build_thredds_netcdf_6.R' script then preps the data for netCDF creation by pre-defining and creating a time and character variables, before selecting variable names to include the netCDF file.
o. The 'build_thredds_netcdf_6.R' script preps the data for inclusion in the netCDF files by collating time for any old and new spectral frequency data included within the specific month of interest.
p. The script then creates variable dimensions for the netCDF file and sets constants for the variable data, such as variable names, variable units, variable long names, and variable precisions.
q. The 'build_thredds_netcdf_6.R' script then subsets each individual variable as per data frame or matrix structure and assign the variable name to the data.
r. The script then creates the netCDF variable dimensions for each individual variable as per the dimensions set above.
s. The 'build_thredds_netcdf_6.R' continues by prepping the netCDF variables to handle individual variable fields.
t. The 'build_thredds_netcdf_6.R' now builds the netCDF file by creating a list of the pre-defined individual, netCDF formatted variables, and adding them to a netCDF configuration ('con') file.
u. The time attribute is added to the 'con' file.
v. Additional standard name and description name fields are added to the 'con' file, taking into account numeric, character, Flag, and metadata field parameters.
w. Finally the 'build_thredds_netcdf_6.R' script adds global attributes to the year_month netCDF file, before closing the netCDF 'con' file connection.

------------------------------------------------------------------------------------------------------------

**References**

------------------------------------------------------------------------------------------------------------

1. Hall, C. and R.E. Jensen. 2021. "Utilizing Data from the NOAA National Data Buoy Center." ERDC/CHL CHETN-I-100, http://dx.doi.org/10.25607/OBP-1087

2. R Core Team. 2017. "R: A language and environment for statistical computing". R Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org/

3. Hall, C. and R.E. Jensen. (*in review*). "United States Army Corps of Engineers Coastal and Hydraulics Laboratory Quality-controlled, Consistent Measurement Archive." Nature Scientific Data Descriptor

4. NDBC. 2016. "At what heights are the sensors located on moored buoys?" [Online], Accessed: February 03 2021. https://www.ndbc.noaa.gov/bht.shtml

5. Smith, D.W. 1985. Meteorological data from a United States Coast Guard (USCG) Exposed Location Buoy (ELB). National Data Buoy Center. OCEANS '85 - Ocean Engineering and the Environment, https://doi.org/10.1109/OCEANS.1985.1160137

6. NDBC. 1992. Environmental measurements from Coast Guard Buoys - The End of an Experiment. NOAA Technical Bulletin, National Data Buoy Center, Vol. 18 (1). ftp://ftp.library.noaa.gov/noaa_documents.lib/NWS/National_Data_Buoy_Center/technical_bulletin/June-1992_vol-18.pdf

7. Gay, E., 2018. NDBC BUOY COMPARISONS Final - with sensor heights. NDBC Buoy Schematic obtained from NDBC Engineer, Eric Gay, 2018.

8. Bouchard, R.H., Riley, R.R, LeBlanc, L.A., Vasquez, M., Robbie, M., Jensen, R.E., Bryant, M.A. and L.A. Fiorentino. 2017. Field Evaluation of the Wave Module for NDBC's New Self-Contained Ocean Observing Payload (SCOOP) on Modified NDBC Hulls. 1st International Workshop on Waves, Storm Surges, and Coastal Hazards, Liverpool, UK, 11 September 2017. http://www.waveworkshop.org/15thWaves/Presentations/A2.pdf

9. Hall, C., Bouchard, R.H., Riley, R., Stewart, R., Wang. D. and S. DiNapoli. 2018. Emerging National Data Buoy Center (NDBC) Wave Systems. Conference: JCOMM DBCP-34 Meeting, Cape Town, South Africa. https://www.researchgate.net/publication/328748840_Emerging_National_Data_Buoy_Center_NDBC_Wave_Systems

10. NDBC, 2021, 'Station 45001 (LLNR 16880) - MID SUPERIOR- 60 NM North Northeast Hancock, MI'. [Online], Accessed: February 03 2021. https://www.ndbc.noaa.gov/station_page.php?station=45001

11. Crout, R.L., Hervey, R.V. and R.H. Bouchard. 2008. Operational field test and evaluation of NDBC's compact ocean observing system configured for ocean wave, meteorological, and ocean current profiling measurements. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.486.6445&rep=rep1&type=pdf

---

## Appendices

---

### Appendix A: HPC Run Scripts

\# To run these buoy stations in parallel, the HPC runs require 'submit_runs.sh' scripts. These scripts identify charge codes for the run; the number of cpu and processors to engage; the time limit for the runs; etc. These codes are computational platform specific.

\# These 'submit_runs.sh' scripts call scripts named 'run_buoy_x.sh', which in turn call the individual 'run_xxxx.R' master scripts (see USACE_QCC_NDBC_Archive_R_scripts.pdf for a 'run_xxxx.R' script example)

Figure A1: Submit_runs_A.sh

```
#!/bin/bash

#PBS -A [redacted]
#PBS -l select=21:ncpus=44:mpiprocs=44
#PBS -l walltime=168:00:00
#PBS -q standard
#PBS -j oe
#PBS -M candice.hall@usace.army.mil
#PBS -m abe

BASE=/p/work/candice
PROJ=${BASE}/projects/WaveTrends/data
WKDIR=${PROJ}/

cd $WKDIR

module swap PrgEnv-cray PrgEnv-gnu
module load cray-R/4.0.3.0
module unload craype-hugepages2M

#chmod -R 770 ./*

# Compress the output files in parallel
( aprun -n 1 -d 3 ./run_buoys_1.sh ) &
( aprun -n 1 -d 3 ./run_buoys_2.sh ) &
( aprun -n 1 -d 3 ./run_buoys_3.sh ) &
( aprun -n 1 -d 3 ./run_buoys_4.sh ) &
( aprun -n 1 -d 3 ./run_buoys_5.sh ) &
( aprun -n 1 -d 3 ./run_buoys_6.sh ) &
( aprun -n 1 -d 3 ./run_buoys_7.sh ) &
( aprun -n 1 -d 3 ./run_buoys_8.sh ) &
( aprun -n 1 -d 3 ./run_buoys_9.sh ) &
( aprun -n 1 -d 3 ./run_buoys_10.sh ) &
( aprun -n 1 -d 3 ./run_buoys_11.sh ) &
( aprun -n 1 -d 3 ./run_buoys_12.sh ) &
( aprun -n 1 -d 3 ./run_buoys_13.sh ) &
( aprun -n 1 -d 3 ./run_buoys_14.sh ) &
( aprun -n 1 -d 3 ./run_buoys_15.sh ) &
( aprun -n 1 -d 2 ./run_buoys_16.sh ) &
( aprun -n 1 -d 2 ./run_buoys_17.sh ) &
( aprun -n 1 -d 2 ./run_buoys_18.sh ) &
( aprun -n 1 -d 2 ./run_buoys_19.sh ) &
( aprun -n 1 -d 2 ./run_buoys_20.sh ) &
( aprun -n 1 -d 2 ./run_buoys_21.sh ) &
#
# must wait for all to finish
wait

exit
```

Figure A2: Submit_runs_B.sh

```
#!/bin/bash

#PBS -A [redacted]
#PBS -l select=21:ncpus=44:mpiprocs=44
#PBS -l walltime=168:00:00
#PBS -q standard
#PBS -j oe
#PBS -M candice.hall@usace.army.mil
#PBS -m abe

BASE=/p/work/candice
PROJ=${BASE}/projects/WaveTrends/data
WKDIR=${PROJ}/

cd $WKDIR

module swap PrgEnv-cray PrgEnv-gnu
module load cray-R/4.0.3.0
module unload craype-hugepages2M

#chmod -R 770 ./*

# Compress the output files in parallel
( aprun -n 1 -d 2 ./run_buoys_22.sh ) &
( aprun -n 1 -d 2 ./run_buoys_23.sh ) &
( aprun -n 1 -d 2 ./run_buoys_24.sh ) &
( aprun -n 1 -d 2 ./run_buoys_25.sh ) &
( aprun -n 1 -d 2 ./run_buoys_26.sh ) &
( aprun -n 1 -d 2 ./run_buoys_27.sh ) &
( aprun -n 1 -d 2 ./run_buoys_28.sh ) &
( aprun -n 1 -d 2 ./run_buoys_29.sh ) &
( aprun -n 1 -d 2 ./run_buoys_30.sh ) &
( aprun -n 1 -d 2 ./run_buoys_31.sh ) &
( aprun -n 1 -d 2 ./run_buoys_32.sh ) &
( aprun -n 1 -d 2 ./run_buoys_33.sh ) &
( aprun -n 1 -d 2 ./run_buoys_34.sh ) &
( aprun -n 1 -d 2 ./run_buoys_35.sh ) &
( aprun -n 1 -d 2 ./run_buoys_36.sh ) &
( aprun -n 1 -d 2 ./run_buoys_37.sh ) &
( aprun -n 1 -d 2 ./run_buoys_38.sh ) &
( aprun -n 1 -d 2 ./run_buoys_39.sh ) &
( aprun -n 1 -d 2 ./run_buoys_40.sh ) &
( aprun -n 1 -d 2 ./run_buoys_41.sh ) &
( aprun -n 1 -d 2 ./run_buoys_42.sh ) &
#
# must wait for all to finish
wait

exit
```

Figure A3: Submit_runs_C.sh

```
#!/bin/bash

#PBS -A [redacted]
#PBS -l select=21:ncpus=44:mpiprocs=44
#PBS -l walltime=168:00:00
#PBS -q standard
#PBS -j oe
#PBS -M candice.hall@usace.army.mil
#PBS -m abe

BASE=/p/work/candice
PROJ=${BASE}/projects/WaveTrends/data
WKDIR=${PROJ}/

cd $WKDIR

module swap PrgEnv-cray PrgEnv-gnu
module load cray-R/4.0.3.0
module unload craype-hugepages2M

#chmod -R 770 ./*

# Compress the output files in parallel
( aprun -n 1 -d 2 ./run_buoys_43.sh ) &
( aprun -n 1 -d 2 ./run_buoys_44.sh ) &
( aprun -n 1 -d 2 ./run_buoys_45.sh ) &
( aprun -n 1 -d 2 ./run_buoys_46.sh ) &
( aprun -n 1 -d 2 ./run_buoys_47.sh ) &
( aprun -n 1 -d 2 ./run_buoys_48.sh ) &
( aprun -n 1 -d 2 ./run_buoys_49.sh ) &
( aprun -n 1 -d 2 ./run_buoys_50.sh ) &
( aprun -n 1 -d 2 ./run_buoys_51.sh ) &
( aprun -n 1 -d 2 ./run_buoys_52.sh ) &
( aprun -n 1 -d 2 ./run_buoys_53.sh ) &
( aprun -n 1 -d 2 ./run_buoys_54.sh ) &
( aprun -n 1 -d 2 ./run_buoys_55.sh ) &
( aprun -n 1 -d 2 ./run_buoys_56.sh ) &
( aprun -n 1 -d 2 ./run_buoys_57.sh ) &
( aprun -n 1 -d 2 ./run_buoys_58.sh ) &
( aprun -n 1 -d 2 ./run_buoys_59.sh ) &
( aprun -n 1 -d 2 ./run_buoys_60.sh ) &
( aprun -n 1 -d 2 ./run_buoys_61.sh ) &
( aprun -n 1 -d 2 ./run_buoys_62.sh ) &
( aprun -n 1 -d 2 ./run_buoys_63.sh ) &
#
# must wait for all to finish
wait

exit
```

Figure A4: run_buoys_1.sh

```
#!/bin/bash

( Rscript run__46005.R ) &
( Rscript run__45011.R ) &
( Rscript run__42053.R ) &

#
# Must wait for all jobs to finish
wait
```

## Appendix B: NCEI-sourced NDBC netCDF data format – pre 2011[1]

| Name | Long Name | Type |
|---|---|---|
| 46029_201012.nc | Meteorological and oceanographic data collected from a Nati... | Local File |
| air_pressure_at_sea_level | air pressure at sea level | 1D |
| air_temperature | air temperature | 1D |
| alpha1 | mean wave direction at specified frequency | 2D |
| alpha2 | principal wave direction at specified frequency | 2D |
| anemometer_height | anemometer height | 1D |
| average_wave_period | sea surface wave mean period from variance spectral densit... | 1D |
| bottom_depth | sea floor depth below sea level | 1D |
| c11_k | sea surface wave variance spectral density | 2D |
| C11_l | sea surface wave directional variance spectral density c11 u... | 2D |
| C12_l | sea surface wave directional variance spectral density c12 u... | 2D |
| C13_l | sea surface wave directional variance spectral density c13 u... | 2D |
| C22_l | sea surface wave directional variance spectral density c22 u... | 2D |
| C23_l | sea surface wave directional variance spectral density c23 u... | 2D |
| C33_l | sea surface wave directional variance spectral density c33 u... | 2D |
| continuous_wind_direction | continuous wind direction | 1D |
| continuous_wind_speed | continuous wind speed | 1D |
| direction_of_hourly_max_gust | direction of hourly max gust | 1D |
| dominant_wave_period | sea surface wave period at variance spectral density maximum | 1D |
| end_of_wave_data_acquisition_k | end of wave data acquisition time | 1D |
| hourly_max_gust | hourly max gust | 1D |
| lat | latitude | — |
| lon | longitude | — |
| magnetic_variation | magnetic variation from north | 1D |
| mean_wave_direction | sea surface wave from direction | 1D |
| Q12_l | sea surface wave directional variance spectral density q12 u... | 2D |
| Q13_l | sea surface wave directional variance spectral density q13 u... | 2D |
| r1 | first normalized polar coordinate of the Fourier coefficients | 2D |
| r2 | second normalized polar coordinate of the Fourier coefficients | 2D |
| sampling_duration_waves | wave sampling duration in minute | 1D |
| sampling_rates_waves | wave sampling rate per minute | 1D |
| sea_surface_temperature | sea surface temperature | 1D |
| sensor_output | sensor output type | 1D |
| significant_wave_height | sea surface wave significant height | 1D |
| speed_averaging_method | speed averaging method | 1D |
| standard_deviation_of_hourly_speed | standard deviation of hourly speed | 1D |
| station | Unique identifier for each feature instance | — |
| time | time | 1D |
| time10 | ten minute time | 1D |
| time10_bnds | ten minute time bounds | 2D |
| time_bnds | time bounds | 2D |
| time_wpm_20 | twenty minute time | 1D |
| time_wpm_20_bnds | twenty minute time bounds | 2D |
| timem | max hourly measured time | 1D |
| timem_bnds | max hourly measured time bounds | 2D |
| total_intervals_waves | frequency interval count | 1D |
| wave_wpm | sea surface wave frequency | 1D |
| wave_wpm_bnds | sea surface wave frequency bounds | 2D |
| wind_direction | average wind direction | 1D |
| wind_gust | wind speed of gust | 1D |
| wind_gust_averaging_period | wind gust averaging period in seconds | 1D |
| wind_sampling_duration | wind sampling duration in minute | 1D |
| wind_speed | average wind speed | 1D |

**Appendix C: NCEI-sourced NDBC netCDF data format – post 2011[1]**
**Note: each payload package expands as demonstrated with air_temperature_sensor_1, anemometer_1 and wave_sensor_1**

| Name | Long Name | Type |
|---|---|---|
| 46029_202001_D7.nc | Meteorological and Oceanographic Data Collected from the… | Local File |
| payload_1 | payload_1 | — |
| air_temperature_sensor_1 | payload_1/air_temperature_sensor_1 | — |
| air_temperature | air temperature | GeoTraj |
| air_temperature_detail_qc | air temperature detail qc | — |
| air_temperature_qc | air temperature qc | GeoTraj |
| air_temperature_release | air temperature release | GeoTraj |
| dew_point_temperature | dew point temperature | GeoTraj |
| dew_point_temperature_detail_qc | dew point temperature detail qc | — |
| dew_point_temperature_qc | dew point temperature qc | GeoTraj |
| dew_point_temperature_release | dew point temperature release | GeoTraj |
| air_temperature_sensor_2 | payload_1/air_temperature_sensor_2 | — |
| anemometer_1 | payload_1/anemometer_1 | — |
| continuous_wind_direction | wind from direction | 1D |
| continuous_wind_direction_detail_qc | continuous wind direction detail qc | — |
| continuous_wind_direction_qc | continuous wind direction qc | 1D |
| continuous_wind_direction_release | continuous wind direction release | GeoTraj |
| continuous_wind_speed | wind speed | 1D |
| continuous_wind_speed_detail_qc | continuous wind speed detail qc | — |
| continuous_wind_speed_qc | continuous wind speed qc | 1D |
| continuous_wind_speed_release | continuous wind speed release | GeoTraj |
| direction_of_hourly_max_gust | wind from direction | 1D |
| direction_of_hourly_max_gust_detail_qc | wind from direction detail qc | — |
| direction_of_hourly_max_gust_qc | wind from direction qc | 1D |
| direction_of_hourly_max_gust_release | direction of hourly max gust release | GeoTraj |
| direction_of_max_1_minute_wind_speed | wind from direction | 1D |
| direction_of_max_1_minute_wind_spee… | wind from direction detail qc | — |
| direction_of_max_1_minute_wind_spee… | wind from direction qc | 1D |
| direction_of_max_1_minute_wind_spee… | direction of max 1 minute wind speed release | GeoTraj |
| hourly_max_gust | wind speed of gust | 1D |
| hourly_max_gust_detail_qc | wind speed of gust detail qc | — |
| hourly_max_gust_qc | wind speed of gust qc | 1D |
| hourly_max_gust_release | hourly max gust release | GeoTraj |
| max_1_minute_wind_speed | wind speed | 1D |
| max_1_minute_wind_speed_detail_qc | wind speed detail qc | — |
| max_1_minute_wind_speed_qc | wind speed qc | 1D |
| max_1_minute_wind_speed_release | max 1 minute wind speed release | GeoTraj |
| wind_direction | wind from direction | GeoTraj |
| wind_direction_detail_qc | wind direction detail qc | — |
| wind_direction_qc | wind direction qc | GeoTraj |
| wind_direction_release | wind direction release | GeoTraj |
| wind_gust | wind speed of gust | GeoTraj |
| wind_gust_detail_qc | wind gust detail qc | — |
| wind_speed_qc | wind speed qc | GeoTraj |
| wind_speed_release | wind speed release | GeoTraj |
| anemometer_2 | payload_1/anemometer_2 | — |
| barometer_1 | payload_1/barometer_1 | — |
| air_pressure | air pressure | GeoTraj |
| air_pressure_at_sea_level | air pressure at sea level | 1D |
| air_pressure_at_sea_level_detail_qc | air pressure at sea level detail qc | — |
| air_pressure_at_sea_level_qc | air pressure at sea level qc | GeoTraj |

| Name | Long Name | Type |
|---|---|---|
| Name | Long Name | Type |
| air_pressure_at_sea_level_release | air pressure at sea level release | GeoTraj |
| air_pressure_detail_qc | air pressure detail qc | — |
| air_pressure_qc | air pressure qc | GeoTraj |
| air_pressure_release | air pressure release | GeoTraj |
| barometer_2 | payload_1/barometer_2 | — |
| gps_1 | payload_1/gps_1 | — |
| latitude | latitude | 1D |
| latitude_detail_qc | latitude detail qc | — |
| latitude_qc | latitude qc | GeoTraj |
| latitude_release | latitude release | GeoTraj |
| longitude | longitude | 1D |
| longitude_detail_qc | longitude detail qc | — |
| longitude_qc | longitude qc | GeoTraj |
| longitude_release | longitude release | GeoTraj |
| humidity_sensor_1 | payload_1/humidity_sensor_1 | — |
| relative_humidity | relative humidity | GeoTraj |
| relative_humidity_detail_qc | relative humidity detail qc | — |
| relative_humidity_qc | relative humidity qc | GeoTraj |
| relative_humidity_release | relative humidity release | GeoTraj |
| ocean_temperature_sensor_1 | payload_1/ocean_temperature_sensor_1 | — |
| sea_surface_temperature | sea surface temperature | GeoTraj |
| sea_surface_temperature_detail_qc | sea surface temperature detail qc | — |
| sea_surface_temperature_qc | sea surface temperature qc | GeoTraj |
| sea_surface_temperature_release | sea surface temperature release | GeoTraj |
| wave_sensor_1 | payload_1/wave_sensor_1 | — |
| alpha1 | mean wave direction at specified frequency | 2D |
| alpha2 | principal wave direction at specified frequency | 2D |
| average_period | sea surface wave mean period from variance spectral dens... | 1D |
| average_period_detail_qc | sea surface wave mean period from variance spectral dens... | — |
| average_period_qc | sea surface wave mean period from variance spectral dens... | 1D |
| average_period_release | average period release | GeoTraj |
| c11 | sea surface wave variance spectral density | 2D |
| c11m | sea surface wave variance spectral density uncorrected | 2D |
| dominant_period | sea surface wave period at variance spectral density maxi... | 1D |
| dominant_period_detail_qc | sea surface wave period at variance spectral density maxi... | — |
| dominant_period_qc | sea surface wave period at variance spectral density maxi... | 1D |
| dominant_period_release | dominant period release | GeoTraj |
| gamma2 | gamma2 coefficient for quad spectra | 2D |
| gamma3 | gamma3 coefficient for quad spectra | 2D |
| mean_wave_direction | sea surface wave from direction | 1D |
| mean_wave_direction_detail_qc | sea surface wave from direction detail qc | — |
| mean_wave_direction_qc | sea surface wave from direction qc | 1D |
| mean_wave_direction_release | mean wave direction release | GeoTraj |
| phih | phih coefficient for quad spectra | 2D |
| r1 | first normalized polar coordinate of the Fourier coefficients | 2D |
| r2 | second normalized polar coordinate of the Fourier coefficients | 2D |
| rhq | rhq coefficient for quad spectra | 2D |
| significant_wave_height | sea surface wave significant height | 1D |
| significant_wave_height_detail_qc | sea surface wave significant height detail qc | — |
| significant_wave_height_qc | sea surface wave significant height qc | 1D |
| significant_wave_height_release | significant wave height release | GeoTraj |
| payload_2 | payload_2 | — |
| air_temperature_sensor_1 | payload_2/air_temperature_sensor_1 | — |
| air_temperature_sensor_2 | payload_2/air_temperature_sensor_2 | — |

| Name | Long Name | Type |
|---|---|---|
| anemometer_1 | payload_2/anemometer_1 | — |
| anemometer_2 | payload_2/anemometer_2 | — |
| barometer_1 | payload_2/barometer_1 | — |
| barometer_2 | payload_2/barometer_2 | — |
| gps_1 | payload_2/gps_1 | — |
| humidity_sensor_1 | payload_2/humidity_sensor_1 | — |
| ocean_temperature_sensor_1 | payload_2/ocean_temperature_sensor_1 | — |
| wave_sensor_1 | payload_2/wave_sensor_1 | — |
| time | time | 1D |
| time10 | ten minute time | 1D |
| time_wpm_20 | time | 1D |
| timem | max hourly measured time | 1D |
| wave_wpm | sea surface wave frequency | 1D |
| wave_wpm_bnds | wave wpm bnds | 2D |

## Appendix D: NDBC standard meteorological and wave spectra text file formats: 1970 – 2020[1]

**2007 to present:**
stdmet format: ##YY MM DD hh mm WDIR WSPD GST WVHT DPD APD MWD PRES ATMP WTMP DEWP VIS TIDE
units: ##yr mo dy hr mn degT m/s m/s m sec sec deg hPa degC degC degC nmi ft

**2006/2007 to present:**
spectral format: #YY MM DD hh mm 0.0200 0.0325 0.0375 0.0425 0.0475 0.0525 0.0575 0.0625 0.0675 0.0725 0.0775 0.0825 0.0875 0.0925 0.1000 0.1100 0.1200 0.1300 0.1400 0.1500 0.1600 0.1700 0.1800 0.1900 0.2000 0.2100 0.2200 0.2300 0.2400 0.2500 0.2600 0.2700 0.2800 0.2900 0.3000 0.3100 0.3200 0.3300 0.3400 0.3500 0.3650 0.3850 0.4050 0.4250 0.4450 0.4650 0.4850

---

**2005 and 2006:** - skip first line, has no skip (#) flag
stdmet format: YYYY MM DD hh mm WD WSPD GST WVHT DPD APD MWD BAR ATMP WTMP DEWP VIS TIDE
units: no units in files

**2005/2006:**
spectral format: YYYY MM DD hh mm 0.030 0.040 0.050 0.060 0.070 0.080 0.090 0.100 0.110 0.120 0.130 0.140 0.150 0.160 0.170 0.180 0.190 0.200 0.210 0.220 0.230 0.240 0.250 0.260 0.270 0.280 0.290 0.300 0.310 0.320 0.330 0.340 0.350 0.360 0.370 0.380 0.390 0.400

---

**2000 to 2004:** - no minute column - don't skip lines, missing tide data in some sets
stdmet data format: YYYY MM DD hh WD WSPD GST WVHT DPD APD MWD BAR ATMP WTMP DEWP VIS TIDE
units: no units in files

spectral data format: YYYY MM DD hh 0.030 0.040 0.050 0.060 0.070 0.080 0.090 0.100 0.110 0.120 0.130 0.140 0.150 0.160 0.170 0.180 0.190 0.200 0.210 0.220 0.230 0.240 0.250 0.260 0.270 0.280 0.290 0.300 0.310 0.320 0.330 0.340 0.350 0.360 0.370 0.380 0.390 0.400

---

**1999:** - no TIDE or minute column
stdmet data format: YYYY MM DD hh WD WSPD GST WVHT DPD APD MWD BAR ATMP WTMP DEWP VIS
units: no units in files

spectral data format: YYYY MM DD hh 0.030 0.040 0.050 0.060 0.070 0.080 0.090 0.100 0.110 0.120 0.130 0.140 0.150 0.160 0.170 0.180 0.190 0.200 0.210 0.220 0.230 0.240 0.250 0.260 0.270 0.280 0.290 0.300 0.310 0.320 0.330 0.340 0.350 0.360 0.370 0.380 0.390 0.400

---

1**970s to 1998:** - no TIDE or minute column, note year and spectral frequency change
stdmet data format: YY MM DD hh WD WSPD GST WVHT DPD APD MWD BAR ATMP WTMP DEWP VIS
units: no units in files

**1970s to 1997/1998:**
spectral data format: YY MM DD hh 0.0200 0.0325 0.0375 0.0425 0.0475 0.0525 0.0575 0.0625 0.0675 0.0725 0.0775 0.0825 0.0875 0.0925 0.1000 0.1100 0.1200 0.1300 0.1400 0.1500 0.1600 0.1700 0.1800 0.1900 0.2000 0.2100 0.2200 0.2300 0.2400 0.2500 0.2600 0.2700 0.2800 0.2900 0.3000 0.3100 0.3200 0.3300 0.3400 0.3500 0.3650 0.3850 0.4050 0.4250 0.4450 0.4650 0.4850

## Appendix E: NCEI NDBC netCDF File Nomenclature Exceptions[1]

| |
|---|
| **Lon** |
| • solar_radiation_sensor_1_longwave_radiation |
| **Wind Direction** |
| • continuous_wind_direction |
| • wind_dir_58 |
| • multiple payloads (duplicates of primary sensor payload) |
| **Wind Speed** |
| • continuous_wind_speed |
| • wind_speed_max_1 |
| • peak_wind_speed |
| • wind_speed_58 |
| • multiple payloads (duplicates of primary sensor payload) |
| **Wind Gust** |
| • wind_gust_averaging_period |
| • wind_gust_2 |
| • wind_gust_58 |
| • multiple payloads (duplicates of primary sensor payload) |
| **Significant Wave Height, Mean Wave Direction, Dominant Wave Period, Average Wave Period** |
| • multiple payloads (duplicates of primary sensor payload) |
| • secondary files (duplicate of primary sensor files) |
| **Air Pressure, Air Temperature** |
| • multiple payloads (duplicates of primary sensor payload) |
| **Sea Surface Temperature, Dew Point Temperature** |
| • multiple payloads (duplicates of primary sensor payload) |
| • secondary files (duplicate of primary sensor files) |
| **$C_{11}$** |
| • $C_{11\_i}$ |
| • multiple payloads (duplicates of primary sensor payload) |
| **$C_{11m}$** |
| • $C_{11\_i}$ |
| • $C_{11\_k}$ |
| • multiple payloads (duplicates of primary sensor payload) |
| **alpha$_1$, alpha$_2$, $r_1$, $r_2$, $C_{12}$, $C_{13}$, $C_{22}$, $C_{33}$, $Q_{12}$, $Q_{13}$, gamma$_2$, gamma$_3$; phi$_h$, $R^h$/q, sensor_output** |
| • multiple payloads (duplicates of primary sensor payload) |

## Appendix F: NDBC website versus NCEI official archive spectral frequency discrepancies[1]

NDBC central spectral frequencies vary over time with the use of different wave sensors:

1996/01–1997/03 == 47 frequencies

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0.0200 | 0.0325 | 0.0375 | 0.0425 | 0.0475 | 0.0525 | 0.0575 | 0.0625 | 0.0675 | 0.0725 | 0.0775 | 0.0825 |
| 0.0875 | 0.0925 | 0.1000 | 0.1100 | 0.1200 | 0.1300 | 0.1400 | 0.1500 | 0.1600 | 0.1700 | 0.1800 | 0.1900 |
| 0.2000 | 0.2100 | 0.2200 | 0.2300 | 0.2400 | 0.2500 | 0.2600 | 0.2700 | 0.2800 | 0.2900 | 0.3000 | 0.3100 |
| 0.3200 | 0.3300 | 0.3400 | 0.3500 | 0.3650 | 0.3850 | 0.4050 | 0.4250 | 0.4450 | 0.4650 | 0.4850 | |

1997/03 - 2006/03 == 38 frequencies

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0.03 | 0.04 | 0.05 | 0.06 | 0.07 | 0.08 | 0.09 | 0.10 | 0.11 | 0.12 | 0.13 | 0.14 |
| 0.15 | 0.16 | 0.17 | 0.18 | 0.19 | 0.20 | 0.21 | 0.22 | 0.23 | 0.24 | 0.25 | 0.26 |
| 0.27 | 0.28 | 0.29 | 0.30 | 0.31 | 0.32 | 0.33 | 0.34 | 0.35 | 0.36 | 0.37 | 0.38 |
| 0.39 | 0.40 | | | | | | | | | | |

2006/06 - 2019/11 == 47 frequencies

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0.0200 | 0.0325 | 0.0375 | 0.0425 | 0.0475 | 0.0525 | 0.0575 | 0.0625 | 0.0675 | 0.0725 | 0.0775 | 0.0825 |
| 0.0875 | 0.0925 | 0.1000 | 0.1100 | 0.1200 | 0.1300 | 0.1400 | 0.1500 | 0.1600 | 0.1700 | 0.1800 | 0.1900 |
| 0.2000 | 0.2100 | 0.2200 | 0.2300 | 0.2400 | 0.2500 | 0.2600 | 0.2700 | 0.2800 | 0.2900 | 0.3000 | 0.3100 |
| 0.3200 | 0.3300 | 0.3400 | 0.3500 | 0.3650 | 0.3850 | 0.4050 | 0.4250 | 0.4450 | 0.4650 | 0.4850 | |

NOAA NCEI official archives show different frequencies bands for the same data.

1990/10 - 1995/05 == 40 frequencies

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0.01 | 0.02 | 0.03 | 0.04 | 0.05 | 0.06 | 0.07 | 0.08 | 0.09 | 0.10 | 0.11 | 0.12 |
| 0.13 | 0.14 | 0.15 | 0.16 | 0.17 | 0.18 | 0.19 | 0.20 | 0.21 | 0.22 | 0.23 | 0.24 |
| 0.25 | 0.26 | 0.27 | 0.28 | 0.29 | 0.30 | 0.31 | 0.32 | 0.33 | 0.34 | 0.35 | 0.36 |
| 0.37 | 0.38 | 0.39 | 0.40 | | | | | | | | |

1995/06 - 1997/02 == 47 frequencies

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0.0200 | 0.0325 | 0.0375 | 0.0425 | 0.0475 | 0.0525 | 0.0575 | 0.0625 | 0.0675 | 0.0725 | 0.0775 | 0.0825 |
| 0.0875 | 0.0925 | 0.1000 | 0.1100 | 0.1200 | 0.1300 | 0.1400 | 0.1500 | 0.1600 | 0.1700 | 0.1800 | 0.1900 |
| 0.2000 | 0.2100 | 0.2200 | 0.2300 | 0.2400 | 0.2500 | 0.2600 | 0.2700 | 0.2800 | 0.2900 | 0.3000 | 0.3100 |
| 0.3200 | 0.3300 | 0.3400 | 0.3500 | 0.3650 | 0.3850 | 0.4050 | 0.4250 | 0.4450 | 0.4650 | 0.4850 | |

1997/03 - 1997/03 == 60 frequencies

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0.0100 | 0.0200 | 0.0300 | 0.0325 | 0.0375 | 0.0400 | 0.0425 | 0.0475 | 0.0500 | 0.0525 | 0.0575 | 0.0600 |
| 0.0625 | 0.0675 | 0.0700 | 0.0725 | 0.0775 | 0.0800 | 0.0825 | 0.0875 | 0.0900 | 0.0925 | 0.1000 | 0.1100 |
| 0.1200 | 0.1300 | 0.1400 | 0.1500 | 0.1600 | 0.1700 | 0.1800 | 0.1900 | 0.2000 | 0.2100 | 0.2200 | 0.2300 |
| 0.2400 | 0.2500 | 0.2600 | 0.2700 | 0.2800 | 0.2900 | 0.3000 | 0.3100 | 0.3200 | 0.3300 | 0.3400 | 0.3500 |
| 0.3600 | 0.3650 | 0.3700 | 0.3800 | 0.3850 | 0.3900 | 0.4000 | 0.4050 | 0.4250 | 0.4450 | 0.4650 | 0.4850 |

1997/04 - 2006/05 == 40 frequencies

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0.01 | 0.02 | 0.03 | 0.04 | 0.05 | 0.06 | 0.07 | 0.08 | 0.09 | 0.10 | 0.11 | 0.12 |
| 0.13 | 0.14 | 0.15 | 0.16 | 0.17 | 0.18 | 0.19 | 0.20 | 0.21 | 0.22 | 0.23 | 0.24 |
| 0.25 | 0.26 | 0.27 | 0.28 | 0.29 | 0.30 | 0.31 | 0.32 | 0.33 | 0.34 | 0.35 | 0.36 |
| 0.37 | 0.38 | 0.39 | 0.40 | | | | | | | | |

2006/06 - 2006/06 == 60 frequencies

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0.0100 | 0.020 | 0.0300 | 0.0325 | 0.0375 | 0.0400 | 0.0425 | 0.0475 | 0.0500 | 0.0525 | 0.0575 | 0.0600 |
| 0.0625 | 0.0675 | 0.0700 | 0.0725 | 0.0775 | 0.0800 | 0.0825 | 0.0875 | 0.0900 | 0.0925 | 0.1000 | 0.1100 |
| 0.1200 | 0.1300 | 0.1400 | 0.1500 | 0.1600 | 0.1700 | 0.1800 | 0.1900 | 0.2000 | 0.2100 | 0.2200 | 0.2300 |
| 0.2400 | 0.2500 | 0.2600 | 0.2700 | 0.2800 | 0.2900 | 0.3000 | 0.3100 | 0.3200 | 0.3300 | 0.3400 | 0.3500 |
| 0.3600 | 0.3650 | 0.3700 | 0.3800 | 0.3850 | 0.3900 | 0.4000 | 0.4050 | 0.4250 | 0.4450 | 0.4650 | 0.4850 |

2006/07 - 2019/11 == 47 frequencies

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0.0200 | 0.0325 | 0.0375 | 0.0425 | 0.0475 | 0.0525 | 0.0575 | 0.0625 | 0.0675 | 0.0725 | 0.0775 | 0.0825 |
| 0.0875 | 0.0925 | 0.1000 | 0.1100 | 0.1200 | 0.1300 | 0.1400 | 0.1500 | 0.1600 | 0.1700 | 0.1800 | 0.1900 |
| 0.2000 | 0.2100 | 0.2200 | 0.2300 | 0.2400 | 0.2500 | 0.2600 | 0.2700 | 0.2800 | 0.2900 | 0.3000 | 0.3100 |
| 0.3200 | 0.3300 | 0.3400 | 0.3500 | 0.3650 | 0.3850 | 0.4050 | 0.4250 | 0.4450 | 0.4650 | 0.4850 | |