# Cyclone Tracking Description – Cyclone Module Version 13.3

**Citation for Most Recent Version of Algorithm**: Crawford, A. D., E. A. P. Schreiber, N. Sommer, M. C. Serreze, J. C. Stroeve, and D. G. Barber, 2021: Sensitivity of Northern Hemisphere Cyclone Detection and Tracking Results to Fine Spatial and Temporal Resolution using ERA5. *Monthly Weather Review*, **149,** 2581-2598. https://doi.org/10.1175/mwr-d-20-0417.1.

**Citation for Original Algorithm**: Crawford, A. D., and M. C. Serreze, 2016: Does the summer Arctic Frontal Zone influence Arctic Ocean cyclone activity? *Journal of Climate*, **29**, 4977–4993, doi:10.1175/JCLI-D-15-0755.s1.

# 1. Input Parameters

The algorithm described here is designed to work with any of several gridded atmospheric datasets (e.g., reanalysis or climate model) with spatial resolution of 25 km to 250 km and a temporal resolution of 1 h to 12 h. To achieve the needed flexibility, it incorporates nine parameters that can be used to tune the algorithm for different data sources or different research questions (**Table 1**).

|  |  |  |
| --- | --- | --- |
| **Table 1**. Cyclone detection and tracking parameters that can be used to tune the algorithm and the default values used with ERA5. The first set of variables define cyclone center detection, the second set defines cyclone area and multi-center cyclone (MCC) detection, and the third define cyclone tracking. (Note, units used in the code are m and Pa, not km and hPa). | | |
| **Parameter** | **Default Value** | **Explanation** |
| kSizekm | 200 km | Half-length of the kernel used to identify whether a grid cell is a local minimum. This is measured from cell center to cell center, so “kSizekm = 200” means the kernel is 5 cells by 5 cells for spatial resolution of 100 km. |
| nanThresh | 0.40 | Maximum fraction of neighboring grid cells with no data allowed for a grid cell to be considered a candidate minimum |
| d\_slp / d\_dist | 7.5 hPa  (1000 km)-1 | Minimum average pressure difference at a certain distance measured from a SLP minimum required for that SLP minimum to be considered a center |
| max\_elev | 1500 m | All elevations above this threshold are masked before analysis; no centers can be detected at elevations above this threshold |
| contint | 2 hPa | Contour interval used when searching for the last closed contour around a cyclone center; used to define cyclone areas and MCCs |
| mcctol | 0.5 | For multiple centers to be grouped as a MCC, this is the maximum allowed ratio of unshared area (defined by closed contours) around the lowest pressure center to shared area for all centers |
| mccdist | 1200 km | Maximum distance allowed between the primary center of a MCC and any secondary center |
| maxSpeed | 150 km hr-1 | Defines the search radius for extending cyclone tracks |
| red | 0.75 | Modifies the projection of a cyclone center’s propagation between two time steps, accounting for the tendency for cyclone propagation to slow with age |

**Additional Notes:**

a. The algorithm will run on any gridded SLP data, but it’s advised that you use a polar-centered equal-area projection like EASE2.

b. Any format that can be read as a numpy array will do, but the code shared here is set up for netcdf files.

c. You also need fields for elevation, latitude, longitude, x distance, and y distance of the same projection, grid cell size, and extent as the SLP inputs. Any format that can be read as a numpy array will do. On [Google Drive](https://drive.google.com/drive/folders/1ejZ82e59XH2WmfLbztt_h0MwUpGphLF5?usp=sharing), I’ve loaded netcdf files with projection criteria for three resolutions: 100 km (180 by 180), 50 km (360 by 360), and 25 km (720 by 720).

d. All scripts related to this algorithm use a custom Python module and custom Python objects. For ease of use, always store the module script in the appropriate site-packages folder (e.g. ~/opt/anaconda3/envs/myenv/lib/python3.8/site-packages). For version 13\_3, the module script is *CycloneModule\_13\_3.py*.

e. The python modules I am currently using for version 13.3. are as follows:

xesmf 0.8.8 (reprojection only – install this first)

netcdf4 1.6.5 (input/output – note: *must* be installed with xesmf)\*

python 3.12.3

numpy 1.26.4

scipy 1.13.0

pandas 2.2.2

I suggest starting with this: <https://xesmf.readthedocs.io/en/latest/installation.html>.

\*You can use xarray instead

# 2. Updates from past version(s)

The main script for running the algorithm is *C03\_CycloneDetection\_13\_3.py*. In my workflow, this is often the third step (after downloading and reprojection). That’s where the prefix “C03” comes from. The suffix: “13\_3” means that the code is version 13.3. This version should yield identical cyclone frequency, intensity, etc. as version 12.4 (Crawford et al., 2021), since none of the core functions have changed.

The main updates from version 13.2 to 13.3 involve the outputs up the main data frame in the cyclonetrack objects. Namely,

1. Updating some syntax to be compatible with pandas≥2.2.2 and numpy≥1.20
2. Changing the top-level detection/tracking script (C03) so that if you set prior== 1, the algorithm will assume you want to use the same input parameters as before and look for a cycloneparams.pkl file. This will override any new user inputs.

# 3. Outputs

Output from this algorithm is two-fold. First, synoptic information for each SLP field is stored in a customized cyclone field object, which is readable in Python. These objects contain information regarding cyclone location, area, and intensity. **Table 2** lists the full set of recorded characteristics. Second, a list of cyclone track objects is saved for each month. Cyclones that exist during multiple months are grouped with the month in which they experience cyclolysis.

|  |  |  |
| --- | --- | --- |
| **Table 2.** Characteristics recorded for each cyclone observation time. | | |
| **Variable** | **Units** | **Description** |
| **Location & Propagation** | | |
| x, y | grid cells | Column and row in EASE2 grid (from upper-left) of cyclone center |
| lon, lat | -180 to +180°E  -90 to 90°N | Longitude and latitude of cyclone center  (warning: “long” was used prior to version 13.0) |
| u, v | km hr-1 | Zonal and meridional propagation velocity since last observation (warning: this was speed, not velocity prior to version 13.0) |
| uv | km hr-1 | Propagation speed since last observation |
| **Cyclone Identification** | | |
| id | -- | Unique ID for the cyclone center in the instantaneous cyclone field |
| pid | -- | Unique ID of the lowest pressure cyclone center in a MCC in the instantaneous cyclone field |
| tid | -- | Unique ID of the cyclone center track for the given month |
| ftid | -- | Former track ID of the cyclone center track, only relevant if it existed in the prior month |
| ptid | -- | Track ID of the primary center in a MCC |
| sid | -- | Unique ID for the track of a cyclone system for the given month |
| **Cyclone Size & Intensity** | | |
| p\_cent | Pa | SLP at cyclone center |
| p\_edge | Pa | SLP at cyclone edge (last closed isobar) |
| area | 104 km2 | \* Area enclosed by last closed isobar |
| radius | 102 km | \* Radius of a circle with the same area as cyclone (“effective radius”) |
| depth | Pa | Edge pressure – central pressure |
| p\_grad | Pa / m | Average pressure gradient between cyclone center and encircling grid cells at approximately 1000-km radius |
| Dp | Pa | Pressure change (Pa) since previous observation |
| DpDt | Pa / day | Deepening rate (scaled by latitude) |
| DsqP | Pa / 104 km2 | \* Laplacian of central pressure (∇2p) |
| **Other** | | |
| type | 0, 1, 2 | 1 = primary center, 2 = secondary center (in a MCC), 0 = this row is only present for calculating propagation (used during splits, merges, and lysis events) |
| centers | # | Number of centers in the cyclone system; if a secondary center of a MCC, set to 0 |
| time | days | Days since 1 Jan 1900 0000 UTC |
| Ege, Ely, Emg, Esp, Erg | 0, 1, 2, or 3 | Records whether the cyclone experienced genesis (ge), lysis (ly), merging (mg), splitting (sp), or regenesis (rg); 0 = no event, 1 = center-only, 2 = area-only, 3 = both center and area involved in event |

\*If the grid cell size is 100 km. Adjust as needed if using a different resolution.

**Additional Notes:**

**Type:** Whenever I calculate track-wide statistics (e.g., maximum deepening rate, average intensity, etc.), I do not use any row with type == 0. These rows are used to help identify splitting and merging events and indicate a projection of the cyclone track to before (after) it first (last) appears. Most cyclone data tables should end with one such row with type == 0 and Ely > 0 (a lysis event).

**Depth**: For single-center cyclones (SCC) and primary centers in multi-center cyclones (MCC), the depth will always be a multiple of *contint* (the contour interval used for area detection). However, for secondary centers of a MCC (centers which are not the lowest pressure in the system), the depth will probably not be a multiple of *contint* because the edge is defined based on the pressure of a different center.

**Area:** The minimum area for a cyclone system is 1 grid cell (e.g., 100 km \* 100 km = 10,000 km2). If a cyclone has the minimum area, then the central pressure and edge pressure are identical, so the depth is 0 Pa.

**p\_grad:** This is **not** the same as taking depth / radius. Rather, it calculated using a common radius for all cyclones (*d\_dist* in the inputs). The value should never be lower than *d\_slp / d\_dist* because those set the minimum intensity for a SLP minimum to be considered a cyclone center.

**Empty Cells**: Empty cells may occur in the first and last rows. Nothing is wrong; there’s just no data for that variable at that time.

**Time:** Whether leap days are included or not depends on the inputs – so be careful with climate models. MERRA2 and ERA5 both have leap days included. The time functions I use are customized to handle the Gregorian calendar, 365-day calendars, or 360-day calendars, so the code can be modified to work with them. You just may need to specify lys=0 and/or dpy=360 in the time functions.

**Directory Structure:** I tend to use many directories to keep data organized. If you want to use a similar system to what I use for storing output, look at the structure in the test dataset. The C03 code will build this structure in your output directory. You may prefer to change directory names or styles, in which case all editing of filepaths can be done on the top-level scripts.

**BBox:** I use the “BBox##” subdirectories for storing any subset of the data I might want. (I use the term “BBox” for “bounding box”, even though sometimes “subset” would be more appropriate. There’s no reason why you can’t change the name, but if I share any code with “BBox” in it, you might need to change that code, too.)

**Unique Track Identifiers**: there are several id numbers used in this algorithm, and they can be confusing to look at. For most purposes, if you’re interested in system tracks only, you’re going to be interested in the “sid” (system id). If you want to ignore the multi-center cyclone aspect and just look at cyclone center tracks, the “tid” (track id) is what you need. The “ftid” (former track id) and “ptid” (parent track id in a multi-center cyclone) are only needed if you are looking at cyclone interactions. In all cases, the id number is unique to the month under consideration. For a universally unique identifier, you need to combine the year, month, and id number. Note that cyclones are always stored with the month in which they experience lysis.

# 4. Common Post-Processing Tasks

A. Output from an algorithm like this invariably includes features that are not of interest. To limit the output, it is customary to only look at storms satisfying certain criteria. Common choices are listed below, but note that applying these must be done in post-processing. The main script will not do it.

1) Lifespan >= 24 hr (5 rows in the data frame if the temporal resolution is 6 h)

2) Track Length >= 1000 km (roughly 10 grid cells if the spatial resolution is 100 km)

3) Maximum displacement >= 500 km\*

4) Minimum Elevation <= 500 m\*\*

\* This means that at some point in its track, a storm is at least 500 km from its genesis point. This is helpful for removing spurious results common in areas of complex topography (i.e., high frequency elevation variability in the horizontal dimensions).

\*\* This means the cyclone can spend most of its lifetime over high elevations, but there must be at least one observation of the system over elevations lower than 500 m. This differs from the max\_elev variable used as an input. Whereas the input parameter determines whether a grid cell can be considered a cyclone center, the post-processing variable determines whether an entire track should be kept. However, the two are related. If, for example, max\_elev = 500 m, having a minimum elevation <= 500 m (or 600 m or 700 m, etc.) criterion is redundant and pointless.

B. The output from this algorithm has a separate track for each cyclone center, which means some storms will be represented multiple times because they are multi-center cyclones. To limit the dataset to only one track per system, you must run the *C03\_SystemDetection\_13.*py script *after* running the *C03\_CycloneDetection\_13\_3.py* script.

C. I use a suite of scripts for subsetting, aggregating, and analyzing results from the algorithm. The following are my main workhorses, so those are the ones I share on github:

* C02 = Reprojection script designed for ERA5 data as the input.
* C04 = Subsetting of cyclone output by location and other characteristics
* C05 = Summarize basic cyclone track statistics; each storm stored as a row in a CSV file
* C06 = Aggregation of cyclone characteristics by month/season, including climatologies; the output is netcdf files.

# 5. Test Dataset

To test one month of data, use the ERA5 SLP from January 1979, accessible on Google Drive at this link: <https://drive.google.com/drive/folders/1ejZ82e59XH2WmfLbztt_h0MwUpGphLF5?usp=sharingfolder>

I’ve already re-projected these to a 100 km by 100 km EASE2 grid (Northern Hemisphere). Compare your results to those I’ve included. You should end up with 523 raw tracks (i.e., “cyclone tracks”) for that month total, but only 201 of those have a lifespan of at least 24 hours (i.e., lifespan >= 1), and only 157 have both a lifespan of at least 24 hours and a track length of at least 1000 km (trackLength >= 1000). The projection file is also included as EASE2\_N0\_100km\_Projection.nc.

That said, note that because I’m always tweaking the algorithm, it’s always possible I update the code and forget I need to update the “CompareToTheseResults” folder on Google Drive. If that’s the case, you might see discrepancies around 1% or 5%. If you’re seeing differences of 10% or more, though, that means something is wrong.

# Appendix: All Updates since first public version

**The main updates from version 10.3 to 11.1 were:**

1. More sophisticated distance measurements
2. A more flexible multi-center cyclone scheme that a) allows for a greater variety of situations and b) has a *slightly* more accurate area calculation
3. More refinement to the identification of splitting and merging events that decreases that number of false positives
4. A fix to a bug in the “regenesis” code that mixed up cyclone IDs for a small percentage of cases
5. The addition of a track-matching function for assessing the impact of using different parameters or inputs.
6. Everything is now written for Python 3 and pandas 0.24.

**The main updates from version 11.1 to 12.4 were:**

1. All inputs and outputs are now netcdf or custom Python objects that have been pickled (.pkl). Pickled objects can be opened with the pandas module. This means the algorithm no longer uses GDAL.
2. I’ve switched from grids that have the north pole at the center of a grid cell to grids that have the north pole at the vertex, so the example data is slightly different than in the past.
3. The kernel size is now defined as a distance in km instead of grid cells. Doing so removes the sensitivity of results to spatial resolution.
4. For cases where two tracks merge, only one of those tracks is continued. The nearest-neighbor track is continued by default, but if both cyclones are the same distance from the merge point in the prior time step, the deeper center is now continued instead of the older center. This results in a reduction in lifespan for the longest-lived storms and makes more intense storms more likely to be continued. However, it effects only a very small percentage of tracks.
5. The algorithm is faster thanks to some optimization of the kernelgradient and Laplacian functions.
6. The detection output has reduced file size by removal of a redundant output field and by saving the data as 8-bit unsigned integers.
7. Using xesmf instead of basemap for reprojection.
8. Separating the cyclone-associated precipitation (CAP) detection from the cyclone detection. All CAP is now calculated after detecting cyclones. This was done because I always performed the CAP analysis on a subset of cyclones anyway.
9. Switched when NaNs are counted during minimum detection, allowing minima over masked areas to interfere with nearby minima. This only impacts areas around the edges of masked topography and was done mostly for improved performance in the Canadian Arctic Archipelago.

**The main updates from version 12.4 to version 13.2 were:**

1. Changed u and v variables in standard cyclone dataframe to be velocity instead of speed
2. Removed Dx and Dy columns from standard cyclone dataframe output. (If desired, these can be calculated from the x and y columns.)
3. Replaced all "longs" with "lons" when referring to longitude, making it the same number of characters as “lats” for latitude.
4. Changed sorting of cyclone main data frame to no longer always be alphabetical
5. Added the SLP gradient measure used to limit cyclone detection (p\_grad) as a variable in the main cyclone data frame.
6. The dist2lon function was changed so that if the input is negative, the output will be negative. (It previously gave an absolute value).
7. Added a try/except statement to cTrack2sTrack function so that it will ignore cases where tids between two months don’t align instead of breaking. Such cases lead to there being two separate system tracks despite a regenesis event being detected. This is a fringe case that probably effects fewer than one in 100,000 tracks.

**The main updates from version 13.2 to 13.3 were:**

1. Updating some syntax to be compatible with pandas≥2.2.2 and numpy≥1.20
2. Changing the top-level detection/tracking script (C03) so that if you set prior== 1, the algorithm will assume you want to use the same input parameters as before and look for a cycloneparams.pkl file. This will override any new user inputs.