

TempestExtremes: A Framework for Scale-Insensitive Pointwise Feature Tracking on Unstructured Grids

Paul A. Ullrich and Colin Zarzycki

Paul A. Ullrich, Department of Land, Air and Water Resources, University of California, Davis,
One Shields Ave., Davis, CA 95616. Email: paulullrich@ucdavis.edu

Correspondence to: Paul A. Ullrich
(paulullrich@ucdavis.edu)

Abstract

This paper describes the new pointwise feature tracking module in the TempestExtremes package.

1 Introduction

- 5 The need for pointwise feature tracking has emerged as a major data processing challenge in climate science. These “trackers” have been employed throughout the literature to answer scientific questions on the expected changes in important atmospheric features under climate change.

This paper presents a general software tool for pointwise feature tracking that...

10 ...uses great-circle arcs for all distance calculations. This avoids issues associated with latitude-longitude distance that emerges near the poles.

...supports structured and unstructured grids. This eliminates the need for post-processing of large native-grid output files and enables detection and characterization simultaneous with the model execution.

15 ...does not contain hard-coded variable names, so as to ensure robust applicability across reanalysis datasets.

...allows for easy intercomparison of detection schemes by enabling all detection criteria to be specified on the command line.

20 As observed by Neu et al. and others, feature tracking schemes can produce wildly varying results depending on the specific choice of threshold variables and values. Consequently, we argue that conclusions drawn from these tracking schemes should use an ensemble of detection schemes.

1.1 Extratropical cyclones

Manual counts of cyclones were performed by ? in the Northern hemisphere from 1899-1939, and latter binned by ? to determine the spatial distribution of such storms. These techniques
25 were later refined by ? by accounting for cyclone trajectories. A similar survey in the Southern hemisphere was performed by ? for July 1957 - December 1958. Manual tracking and characterization of cyclones was also performed by ? using ECMWF forecast data for the 1981/82 winter.

One of the first automated detection and tracking for extratropical cyclones was developed by
30 ? using nonlinear optimization to fit cyclonic profiles to anomalies in the 500-mb geopotential height field. Storms were then tracked over a short forecast period using the best fit to the cyclone's centerpoint.

Counts of cyclones neglecting the cyclone trajectory were automatically generated from climate model output for both hemispheres by ? using local minima in 1000-hPa geopotential
35 height. This method had some shortcomings, including mischaracterization of local lows due to Gibbs' ringing and topographically-driven lows. To overcome these problems, ? proposed an additional minimum threshold on the local pressure gradient. Similarly, ? detected cyclones in ECMWF pressure data using a local minima in the sea-level pressure that must also be 4 mb below the average sea-level pressure of neighboring grid points, and must persist for three
40 successive 6- or 12- hour intervals.

? extracted low pressure centers from interpolated GCM data using local optimization, based on earlier work in ?.

Feature tracking on the sphere was revisited by ?, which extended tracking algorithms designed for Cartesian geometry ? that were built from image processing techniques.

1.2 Tropical cyclones

Extratropical cyclone tracking techniques have been modified in order to support tropical cyclone tracking. To eliminate "false positives" associated with extratropical cyclones and weak
45 cyclonic depressions, many schemes require that the candidate be associated with a nearby

warm core and be associated with a minimum threshold on surface winds for at least 1-3 days. The definition of a “warm core” varies between modeling centers, including such options as air temperature anomaly on pressure surfaces (???), geopotential thickness (?) and decay of vorticity with height (??). Additional filtering of candidate storms over topography or within a specified latitudinal range may be required. To better match observations, additional geographical, model or feature-dependent criteria may be applied (????). In ?, tropical cyclone-like features were detected by seeking local minima (occupying one grid point due to the coarse model resolution) in the surface pressure field that accompanied a minimum surface wind speed of 17 m/s. A tracking procedure was also developed using a minimum distance between detections that represented a maximum velocity of 50 km/hr for TCs. Tracking in ? instead detected maxima in the 850mb relative vorticity (greater than $3.5 \times 10^{-5} \text{s}^{-1}$ accompanied by a weak warm core (using local temperature anomaly at 250mb, 500mb and 850mb) and required TCs to exist for a minimum of 3 days. ? made use of thresholds on 850mb relative vorticity, maximum local velocity, temperature anomaly at 700, 500 and 300 hPa and required systems exist for at least 1.5 days. ? detects geopotential height minima at 1000mb and then imposes additional thresholds on 900mb relative vorticity, divergence, vertical pressure velocity, geopotential thickness and zonal wind velocity. ? use the absolute value of relative vorticity for detection, then further employ thresholds on maximum wind speed, sea level pressure, temperature anomaly, mean wind speed and track length.

A tabulated overview of the thresholds utilized by many of these schemes can be found in ?, along with several proposed guidelines on detection schemes. Nonetheless, the definition of an optimal objective criteria for atmospheric features has eluded development, suggesting that there may be no singular criteria capable of both perfect detection and zero false positive rate. It is widely acknowledged that weaker tropical storms are difficult to track, and the observational record of these less-intense, short-lived storms is questionable (?).

1.3 African Easterly Waves

A similar manual study to ? was performed by ? to detect and track African easterly waves. Storm tracks and African easterly waves were tracked by ?.

? uses minima in the 1000 hPa geopotential height to identify cyclones with associated maxima in the 850 hPa vorticity.

1.4 Building blocks of detection schemes

Most of the detection and characterization algorithm that are present in the literature share a common detection procedure:

1. Identify an initial set of candidate points by finding local extrema. Local extrema can be further isolated, for instance by requiring that the local extrema be sufficiently anomalous when contrasted with their neighbors. For most cyclonic structures, either minima in the sea level pressure field or maxima in the absolute value of the relative vorticity are used.
2. Eliminate candidate points that do not satisfy a prescribed set of thresholds. For instance, tropical cyclones typically require the presence of an upper-level warm core that is sufficiently near the sea level pressure minima.
3. Connect candidate points together in time to generate feature paths, eliminating paths that are of insufficient length or do not meet additional criteria.

At high horizontal resolutions, climate datasets become increasingly unwieldy to store and analyze. Simultaneously, single-core computing systems have effectively reached peak performance, and so supercomputing systems have been forced to adopt massively parallel infrastructure. To address the need for feature tracking in multi-decadal high-resolution ensemble datasets, algorithms based on MapReduce have been developed (??). In this case, candidates are first isolated on individual time slices (in an embarrassingly parallel manner), and then stitched across time to build trajectories. For detection codes based on MapReduce, parallel performance is usually bottlenecked by file system access times (I/O).

2 Algorithms

100 This section describes the building blocks that have been utilized in constructing our detection and characterization framework. As mentioned earlier, in order to avoid sensitivity of the detection scheme to grid resolution, great circle distance has been employed throughout. In terms of regular latitude-longitude coordinates, great circle distance between points (λ_1, φ_1) and (λ_2, φ_2) is defined via the symmetric operation

$$105 \quad r(\lambda_1, \varphi_1; \lambda_2, \varphi_2) = a \arccos(\sin \varphi_1 \sin \varphi_2 + \cos \varphi_1 \cos \varphi_2 \cos(\lambda_1 - \lambda_2)). \quad (1)$$

Algorithmically, this calculation is implemented as `gcdist(p, q)` for given graph nodes p and q .

2.1 Efficient neighbor search using k -d Trees

110 Three-dimensional ($k = 3$) k -d trees (?) are used throughout our detection code using the implementation of ?. Although k -d trees use straight-line instance instead of great circle distance, we utilize the observation that straight-line and great-circle distance maintain the same ordering for points confined to the surface of the sphere. In particular, we utilize three key functions made available by the k -d tree implementation:

`K = build_kdtree(P)` constructs a k -d tree K from a point set P .

115 `q = kdtree_nearest_neighbor(K, p)` locates the nearest neighbor q to point p using the k -d tree K .

`S = kdtree_all_neighbors(K, p, dist)` locates all points that are within a distance `dist` of a point p within the k -d tree K .

120 A key advantage of KD trees is their relatively efficient $O(n \log n)$ construction time and $O(\log n)$ average time nearest neighbor search.

2.2 Unstructured grid specification

For purposes of determining connectivity of the unstructured grid, we require the specification of a graph such as the one depicted in Figure ?? . The connectivity information is stored textually as an adjacency list via a variable-length comma-separated variable file. The total number of nodes (N) is specified at the top of the file, followed by N lines containing the longitude (lon), latitude (lat), number of adjacent nodes, and finally a 1-indexed list of all adjacent nodes, such as depicted below:

```
<total number of nodes>
<lon>,<lat>,<# adj. nodes>,<first adj. node>,...,<last adj. node>
...
```

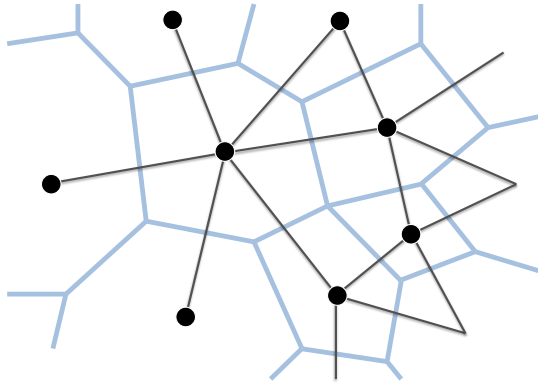


Fig. 1. An example adjacency graph describing an unstructured grid (blue lines), where nodes are co-located with volume centerpoint locations (solid circles) and edges connect adjacent volumes.

2.3 Extrema detection

For purposes of computational efficiency, candidate points are initially located by identifying local extrema in a given field (for instance SLP) via `find_all_minima` (Algorithm ??). Candidates are then eliminated if they are “too close” to stronger extrema (Algorithm ??). The initial search field is specified to `TempestExtremes` either via the `--searchbymin` or `--searchbymax` command line argument. The merge distance used in `merge_candidates_min` is specified via the `--mergedist` command line argument.

Algorithm 1. Locate the set of all nodes P that are local minima for a field G (for instance, SLP) defined on an unstructured grid. The procedure for locating maxima is analogous.

```
set  $P = \text{find\_all\_minima}(\text{field } G)$ 
for each node  $f$ 
   $\text{is\_minima}[f] = \text{true}$ 
  for each neighbor node  $v$  of  $f$ 
    if  $G[v] < G[f]$  then
       $\text{is\_minima}[f] = \text{false}$ 
  if  $\text{is\_minima}[f]$  then
    insert  $f$  into  $P$ 
```

Algorithm 2. Given a field G defined on an unstructured grid and a set of candidate points P , remove candidate minima that are within a distance $dist$ of a more extreme minimum, and return the new candidate set Q .

```
set  $Q$  = merge_candidates_minima(field  $G$ , set  $P$ ,  $dist$ )
 $K$  = build_kd_tree( $P$ )
for each candidate  $p$  in  $P$ 
    retain_p = true
     $N$  = kd_tree_all_neighbors( $K$ ,  $p$ ,  $dist$ )
    for all  $q$  in  $N$ 
        if ( $G[q] < G[p]$ ) then retain_p = false
    if retain_p then insert  $p$  into  $Q$ 
```

2.4 Closed contour criteria

Although a first pass at candidate points may be made by looking for local extrema (comparing against all neighboring nodes), this criteria is not robust across model resolution. That is, the distance between a node and its neighbors decreases proportional to the local grid spacing, and so does not define a “physical” criterion. Consequently, we instead advocate for a *closed contour criteria* to define candidate nodes. Closed contours were first employed by ?, who used a 30m 500 mb geopotential height contour to identify closed circulation centers. Their approach used radial arms generated at 15° intervals over a great circle distance of 2° and required that geopotential heights rise by at least 30m along each arm. Unfortunately, the use of radial arms to define the closed contour is again sensitive to model resolution, since it has the potential to only sample as many neighbors as radial arms employed.

Here we propose an alternative closed contour criteria that is largely insensitive to model resolution that uses graph search to ensure that all paths along the unstructured grid from an initial location p_0 lead to a sufficiently large decrease (or increase) in a given field G . This criteria is illustrated in Figure ??, and is implemented in Algorithm ?? and ?? (for closed contours around local maxima). An analogous algorithm `no_closed_contour` is also provided, which has similar functionality but discards candidates that satisfy the closed contour criteria (this may be desirable, for instance, to identify cyclonic structures that do not have a warm core).

Algorithm 3. Find the node p_{\max} containing the maximal value of the field G within a distance maxdist of the node p . An analogous procedure find_min_near is provided for locating nodes containing minimal values of the field.

```
node pmax = find_max_near(node p, field G, maxdist)
  set visited = {}
  set tovisit = {p}
  pmax = p
  while tovisit is not empty
    q = remove node from tovisit
    if (q in visited) then continue
    add q to visited
    if (gcdist(p,q) > maxdist) then continue
    if (G[q] > G[pmax]) then pmax = q
```

Algorithm 4. Determine if there is a closed contour in field G of magnitude `thresh` around the point p_0 , defined by $p_0 = \text{find_max_near}(p, G, \text{maxdist})$, within distance `dist`. That is, along all paths away from p_0 , the field G must drop by at least `thresh` within distance `dist`. The closed contour criteria is depicted in Figure ?? . An analogous procedure is defined for closed contours around minima.

```
closed_contour_max(point p, field G, dist, maxdist, thresh)
  p0 = find_max_near(p, G, maxdist)
  set visited = {}
  set tovisit = {p0}
  while tovisit is not empty
    q = remove point from tovisit
    if (q in visited) then continue
    add q to visited
    if (gcdist(p0,q) > dist) then return false
    if (G[p0] - G[q] < thresh) then
      add all neighbors of q to tovisit
  return true
```

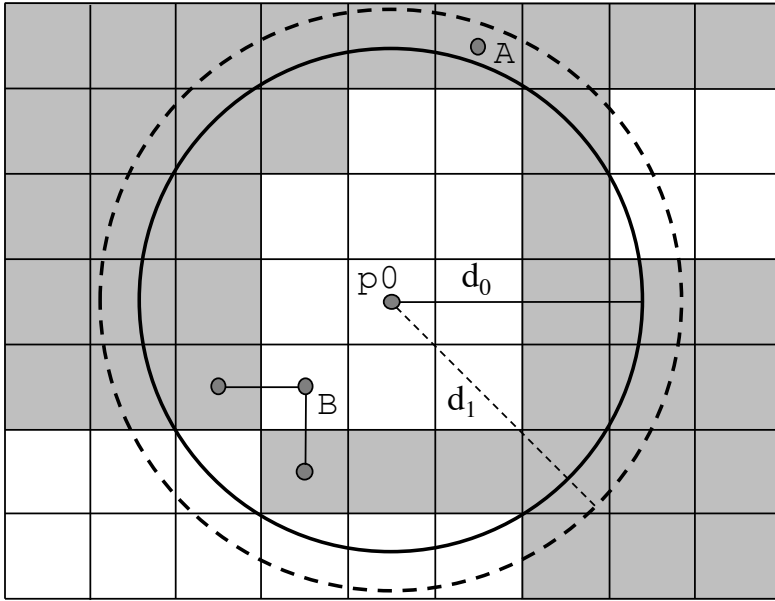


Fig. 2. An illustration of the closed contour criteria. Nodes shaded in white (gray) satisfy (do not satisfy) the threshold of the field value at p_0 . Since only edge-neighbors are included, B constitutes a boundary to the interior of the closed contour. Because A lays outside the solid circle, the contour with distance d_0 is not a closed contour, whereas the dashed contour with distance d_1 does satisfy the closed contour criteria.

2.5 Thresholding

Additional threshold criteria may be applied at the Map() stage in order to further eliminate undesirable candidates. For example, a common threshold criteria requires that a field G satisfy

some minimum value within a distance `dist` of the candidate, as implemented in Algorithm ??.

Algorithm 5. Determine if a candidate node p satisfies the requirement that there exists another node p_0 within distance dist of p with $G[p] > \text{thresh}$.

```
threshold_max(node p, field G, dist, thresh)
  p0 = find_max_near(p, G, dist)
  if (G[p0] < thresh) then
    return false
  else
    return true
```

2.6 Stitching

The basic stitching procedure is implemented in Algorithm ??.

Algorithm 6. Determine all feature paths S , given array of nodes $P[1..T]$ and maximum great-circle distance between nodes at subsequent time levels $dist$.

```

path set  $S$  = stitch_nodes(set array  $P[1..T]$ ,  $dist$ )
  for each time level  $t = 1..T$ 
     $K[t] = build\_kd\_tree(P[t])$ 
  for each time level  $t = 1..T$ 
    while  $P[t]$  is not empty
      initialize empty path  $s$ 
       $p = remove\ next\ candidate\ from\ P[t]$ 
      add  $p$  into  $s$ 
      for time level  $u = t+1..T$ 
         $q = kd\_tree\_nearest\_neighbor(K[u], p)$ 
        if ( $q$  in  $P[u]$ ) and ( $gcdist(p, q) < dist$ ) then
          add  $q$  into  $s$ 
          remove  $q$  from  $P[u]$ 
           $p = q$ 
        else break
      add  $s$  into  $S$ 

```

3 Selected examples

A few selected examples of the feature detection tool are now provided. The first three examples use data from the NCEP Climate Forecast System Reanalysis (CFSR), available at 0.5 degree global resolution with 6-hourly output from 1979-present (?).

3.1 Tropical cyclones in CFSR

The command line we use to detect tropical cyclone-like features in CFSR is provided below. Three-dimensional (time + 2D space) hyperslabs of CFSR data have been extracted, with TMP_L100 corresponding to 400hPa air temperature, and U_GRD_L100 and V_GRD_L100 corresponding to 850hPa zonal and meridional velocities. Candidates are initially identified by minima in the sea level pressure (PRMSL_L101), and then eliminated if a smaller minimum exists within a great circle distance of 2.0 degrees. The closed contour criteria is then applied, requiring an increase in SLP of at least 200Pa over a distance of 4 degrees away from the candidate node, and a decrease in 400hPa air temperature of 0.4K within 8 degrees of the node within 1.1 degrees of the candidate with maximum air temperature. Since CFSR is on a structured latitude-longitude grid, the output format is `i, j, lon, lat, psl, maxu, zs`, where `i, j` are the longitude and latitude coordinates within the dataset, `lon, lat` are the actual longitude and latitude of the candidate, `psl` is the SLP at the candidate point (equal to the maximum SLP within 0 degrees of the candidate), `maxu` is the vector magnitude of the maximum 850 hPa wind within 4 degrees of the candidate, and `zs` is the topographic height at the candidate point.

```
./DetectCyclonesUnstructured
--in_data "$uvfile;$stpfile;$hfile" --out $outf
--searchbymin PRMSL_L101
--mergedist 2.0
--closedcontourcmd "PRMSL_L101,200.,4,0;
    TMP_L100,-0.4,8.0,1.1"
--outputcmd "PRMSL_L101,max,0;
    _VECMAG(U_GRD_L100,V_GRD_L100),max,4;
    HGT_L1,max,0"
```

All outputs from DetectCyclonesUnstructured are then concatenated into a single file containing candidates at all times (`pgbhn1.dcu.tc.all.dat`). Candidates are then stitched in time to form paths, with a maximum distance between candidates of 8.0 degrees (great circle distance), consisting of at least 8 candidates per path, and with a maximum gap size of 2 (most consecutive timesteps with no associated candidate). Because localized low-pressure regions

195 that are unrelated to tropical cyclones can form as a consequence of topographic forcing, we also require that for at least 8 time steps the underlying topographic height (z_s) be at most 100 meters. The associated command line for StitchNodes is:

```
./StitchNodes
--in pgbhnl.dcu_tc_all.dat
200 --out pgbhnl.dcu_tc_stitch.dat
--format "i,j,lon,lat,psl,maxu,zs"
--range 8.0 --minlength 8 --maxgap 2
--threshold "zs,<=,100.0,8"
```

205 Once the complete set of tropical cyclone paths have been computed, total tropical cyclone density over each 0.5 degree grid cell is plotted in Figure ???. Overall the results show very good agreement with reference fields (Citation?).

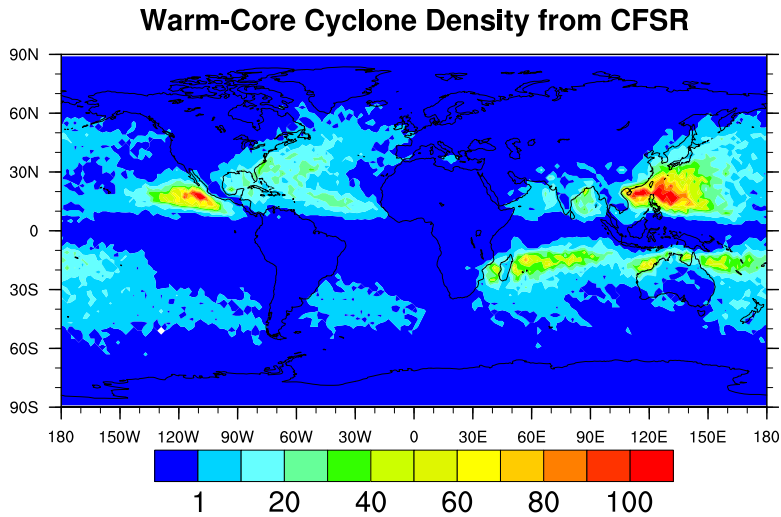


Fig. 3. Tropical cyclone count over the period 1979-2010 obtained using the procedure described in section ??.

3.2 Extratropical cyclones in CFSR

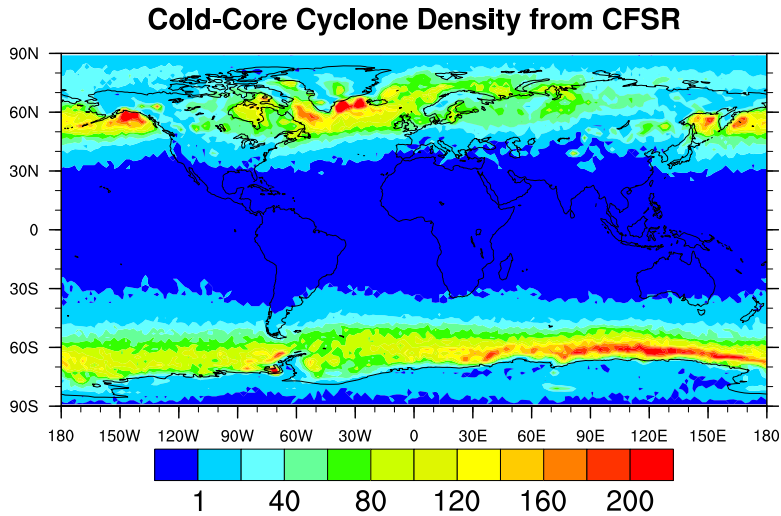


Fig. 4. Extratropical cyclone count over the period 1979-2010 obtained using the procedure described in section ??.

3.3 Tropical easterly waves in CFSR

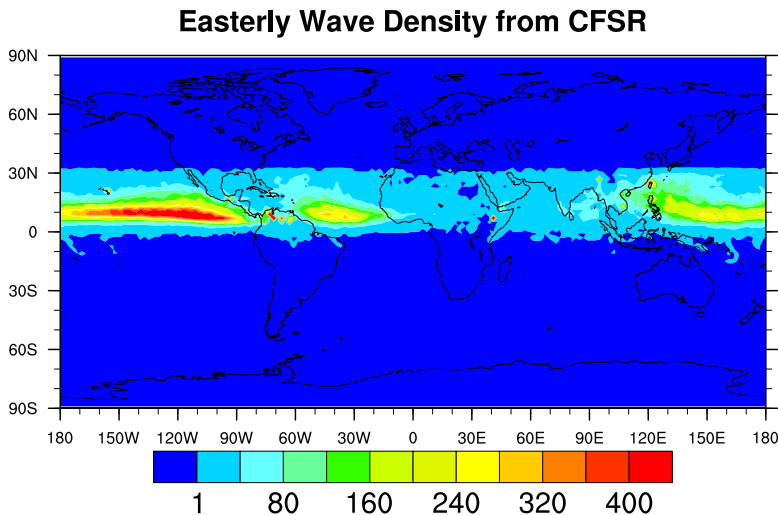


Fig. 5. Tropical easterly wave density over the period 1979-2010 obtained using the procedure described in section ??.

3.4 Tropical cyclones in a simulation with variable-resolution CESM

4 Conclusions

Acknowledgements.

A1 Software Documentation: DetectCyclonesUnstructured

Usage: DetectCyclonesUnstructured <parameter list>

Parameters:

```

215  --in_data <string> [""]
    --in_connect <string> [""]
    --out <string> [""]
    --searchbymin <string> [""] (default PSL)
    --searchbymax <string> [""]
220  --maxlat <double> [0.000000] (degrees)
    --minlat <double> [0.000000] (degrees)
    --topofile <string> [""]
    --maxtopoht <double> [0.000000] (m)
    --mergedist <double> [0.000000] (degrees)
225  --closedcontourcmd <string> [""] [var,dist,delta,minmaxdist;...]
    --noclosedcontourcmd <string> [""] [var,dist,delta,minmaxdist;...]
    --thresholdcmd <string> [""] [var,op,value,dist;...]
    --outputcmd <string> [""] [var,op,dist;...]
    --timestride <integer> [1]
230  --regional <bool> [false]
    --out_header <bool> [false]
    --verbosity <integer> [0]

```

--in_data <string>

The input datafile in NetCDF format.

```

235  --in_connect <string>

```

A connectivity file, which uses a vertex list to describe the graph structure of the input grid. This parameter is not required if the data is on a latitude-longitude grid.

`--out <string>`

The output file containing the filtered list of candidates in plain text format.

240 `--searchbymin <string>`

The input variable to use for initially selecting candidate points (defined as local minima). By default this is “PSL”, representing detection of surface pressure minima. Only one of `searchbymin` and `searchbymax` may be set.

`--searchbymax <string>`

245 The input variable to use for initially selecting candidate points (defined as local maxima). Only one of `searchbymin` and `searchbymax` may be set.

`--maxlat <double>`

The maximum absolute latitude for candidate points. Candidates at higher latitudes are discarded.

250 `--minlat <double>`

The minimum absolute latitude for candidate points. Candidates at lower latitudes are discarded.

`--topofile <string>`

An auxiliary file containing topographic information.

255 `--maxtopoht <double>`

If `--topofile` is specified, discard candidates over topography higher than specified by this parameter.

`--mergedist <double>`

260 Merge candidate points with distance (in degrees) shorter than the specified value. Among two candidates within the merge distance, only the candidate with lowest `searchbymin` or highest `searchbymax` value will be retained.

265 `--closedcontourcmd <cmd1>;<cmd2>;...` Eliminate candidates if they do not have a closed contour. Closed contour commands are separated by a semi-colon. Each closed contour command takes the form `var,dist,delta,pivotdist`. These arguments are as follows.

`var <variable>` The variable used for the contour search.

`dist <double>` The great-circle distance (in degrees) from the pivot within which the closed contour criteria must be satisfied.

270 `delta <double>` The amount by which the field must change from the pivot value. If positive (negative) the field must increase (decrease) by this value along the contour.

`pivotdist <double>` The distance away from the candidate to search for the pivot. If delta is positive (negative), the pivot is a local minimum (maximum).

`--noclosedcontourcmd <cmd1>;<cmd2>;...`

275 As `closedcontourcmd`, except eliminates candidates if a closed contour is present.

`--thresholdcmd <cmd1>;<cmd2>;...` Eliminate candidates that do not satisfy a threshold criteria (there must exist a point within a given distance of the candidate that satisfies a given equality or inequality). Threshold commands are separated by a semi-colon. Each threshold command takes the form `var,op,value,dist`. These arguments are as follows.

`var <variable>` The variable used for the contour search.

`op <string>` Operator that must be satisfied for threshold (options include `>`, `>=`, `<`, `<=`, `=`, `!=`).

`value <double>` The value on the RHS of the comparison.

285 `dist <double>` The great circle distance away from the candidate to search for a point that satisfies the threshold (in degrees).

--outputcmd <cmd1>;<cmd2>;... Include additional columns in the output file.
Output commands take the form *var*, *op*, *dist*. These arguments are as follows.

var <variable> The variable used for the contour search.

op <string> Operator that is applied over all points within the specified distance of the candidate (options include *max*, *min*, *avg*, *maxdist*, *mindist*).

dist <double> The great circle distance away from the candidate wherein the operator is applied (in degrees).

--timestep <integer>

Only examine discrete times at the given stride (by default 1).

--regional

When a latitude-longitude grid is employed, do not consider longitudinal boundaries to be periodic.

--out_header

Output a header describing the columns of the data file.

--verbosity <integer>

Set the verbosity level (default 0).

A1.1 Variable Specification

Quantities of type <variable> include both NetCDF variables in the input file (for example, “Z850”) and simple operations performed on those variables. By default it is assumed that NetCDF variables are specified in the *.nc* file as

```
float Z850(time, lat, lon) or float Z850(time, ncol)
```

for structured latitude-longitude grids and unstructured grids, respectively. If variables have no time variable, they have the related specification

310 float Z850(lat, lon) or float Z850(ncol)

If variables include an additional dimension, for instance,

 float Z(time, lev, lat, lon) or float Z(time, lev, ncol)

they may be specified on the command-line as `Z(<lev>)`, where the integer index `<lev>` corresponds to the first dimension (or the dimension after `time`, if present).

315 Simple operations on variables are also supported, including

`_VECMAG(<variable>, <variable>)` 2-component vector magnitude,

`_PLUS(<variable>, <variable>)` Pointwise sum of variables,

`_DIFF(<variable>, <variable>)` Pointwise difference of variables.

The following are valid examples of `<variable>` type,

320 `_VECMAG(U850, V850)` and `_DIFF(U(3),U(5))`.

A2 Software Documentation: StitchNodes

Usage: `StitchNodes <parameter list>`

Parameters:

325 `--in <string> [""]`
 `--out <string> [""]`
 `--format <string> ["no,i,j,lon,lat"]`
 `--range <double> [5.000000] (degrees)`
 `--minlength <integer> [3]`
 `--min_endpoint_dist <double> [0.000000] (degrees)`
 330 `--min_path_dist <double> [0.000000] (degrees)`
 `--maxgap <integer> [0]`
 `--threshold <string> ["" [col,op,value,count;...]`
 `--timestride <integer> [1]`
 `--out_format <string> ["std"] (std|visit)`

335 `--in <string>`

The input file (a list of candidates from DetectCyclonesUnstructured).

`--out <string>`

The output file containing the filtered list of candidates in plain text format.

`--format <string>`

340 The structure of the columns of the input file.

`--range <double>`

The maximum distance between candidates along a path.

`--minlength <integer>`

The minimum length of a path (in terms of number of discrete times).

345 `--min_endpoint_dist <double>`

The minimum great-circle distance between the first candidate on a path and the last candidate (in degrees).

`--min_path_dist <double>`

The minimum path length, defined as the sum of all great-circle distances between candidate nodes (in degrees).

350 `--maxgap <integer>`

The largest gap (missing candidate nodes) along the path (in discrete time points).

`--threshold <cmd1>;<cmd2>;...`

355 Eliminate paths that do not satisfy a threshold criteria (a specified number of candidates along path must satisfy an equality or inequality). Threshold commands are separated by a semi-colon. Each threshold command takes the form `col,op,value,count`. These arguments are as follows.

`col <integer>` The column in the input file to use in the threshold criteria.

op <string> Operator used for comparison of column value (options include >, >=, <, <=, =, !=).

value <double> The value on the right-hand-side of the operator.

count <integer> The minimum number of candidates along the path that must satisfy this criteria.

--timestep <integer>

Only examine discrete times at the given stride (by default 1).

References

Akyildiz, V.: Systematic errors in the behaviour of cyclones in the ECMWF operational models, *Tellus A*, 37, 297–308, 1985.

Alpert, P., Neeman, B., and Shay-El, Y.: Climatological analysis of Mediterranean cyclones using ECMWF data, *Tellus A*, 42, 65–77, 1990.

Bell, G. D. and Bosart, L. F.: A 15-year climatology of Northern Hemisphere 500 mb closed cyclone and anticyclone centers, *Monthly Weather Review*, 117, 2142–2164, 1989.

Bengtsson, L., Botzet, M., and Esch, M.: Hurricane-type vortices in a general circulation model, *Tellus A*, 47, 175–196, 1995.

Bengtsson, L., Hodges, K. I., and Esch, M.: Tropical cyclones in a T159 resolution global climate model: comparison with observations and re-analyses, *Tellus A*, 59, 396–416, doi:10.1111/j.1600-0870.2007.00236.x, 2007.

Bentley, J. L.: Multidimensional binary search trees used for associative searching, *Communications of the ACM*, 18, 509–517, 1975.

Broccoli, A. and Manabe, S.: Can existing climate models be used to study anthropogenic changes in tropical cyclone climate?, *Journal of Geophysical Research Letters*, volume=17, number=11, pages=1917–1920, year=1990.

Camargo, S. J. and Zebiak, S. E.: Improving the Detection and Tracking of Tropical Cyclones in Atmospheric General Circulation Models, *Weather and Forecasting*, 17, 1152–1162, 2002a.

Camargo, S. J. and Zebiak, S. E.: Improving the detection and tracking of tropical cyclones in atmospheric general circulation models, *Weather and forecasting*, 17, 1152–1162, 2002b.

- Haarsma, R. J., Mitchell, J. F., and Senior, C.: Tropical disturbances in a GCM, *Climate Dynamics*, 8, 247–257, 1993.
- Hodges, K.: Feature tracking on the unit sphere, *Monthly Weather Review*, 123, 3458–3465, 1995.
- 390 Hodges, K. et al.: A general-method for tracking analysis and its application to meteorological data, *Monthly Weather Review*, 122, 2573–2586, 1994.
- Hodges, K. I., Hoskins, B. J., Boyle, J., and Thorncroft, C.: A comparison of recent reanalysis datasets using objective feature tracking: Storm tracks and tropical easterly waves, *Monthly Weather Review*, 131, 2012–2037, 2003.
- 395 Klein, W. H.: Principle tracks and mean frequencies of cyclones and anticyclones in the Northern Hemisphere, US Weather Bureau, 1957.
- König, W., Sausen, R., and Sielmann, F.: Objective identification of cyclones in GCM simulations, *Journal of Climate*, 6, 2217–2231, 1993.
- Lambert, S. J.: A cyclone climatology of the Canadian Climate Centre general circulation model, *Journal of Climate*, 1, 109–115, 1988.
- 400 Landsea, C. W., Vecchi, G. A., Bengtsson, L., and Knutson, T. R.: Impact of Duration Thresholds on Atlantic Tropical Cyclone Counts, *Journal of Climate*, 23, 2508–2519, doi:10.1175/2009JCLI3034.1, 2010.
- Le Treut, H. and Kalnay, E.: Comparison of observed and simulated cyclone frequency distribution as determined by an objective method, *Atmosfera*, 3, 1990.
- 405 Murakami, H. and Sugi, M.: Effect of Model Resolution on Tropical Cyclone Climate Projections, *SOLA*, 6, 73–76, doi:10.2151/sola.2010-019, 2010.
- Murakami, H., Wang, Y., Yoshimura, H., Mizuta, R., Sugi, M., Shindo, E., Adachi, Y., Yukimoto, S., Hosaka, M., Kusunoki, S., Ose, T., and Kitoh, A.: Future Changes in Tropical Cyclone Activity Projected by the New High-Resolution MRI-AGCM*, *J. Clim.*, 25, 3237–3260, doi:10.1175/JCLI-D-11-00415.1, 2012.
- 410 Murray, R. J. and Simmonds, I.: A numerical scheme for tracking cyclone centres from digital data. Part I: Development and operation of the scheme, *Australian Meteorological Magazine*, 39, 155–166, 1991.
- Petterssen, S.: *Weather analysis and forecasting. 2. Weather and weather systems*, McGraw-Hill, 1956.
- Prabhat, Rübel, O., Byna, S., Wu, K., Li, F., Wehner, M., and Bethel, W.: TECA: A Parallel Toolkit for Extreme Climate Analysis, *Procedia Computer Science*, 9, 866 – 876, doi:10.1016/j.procs.2012.04.093, proceedings of the International Conference on Computational Science, 2012, 2012.

- 420 Reed, R., Hollingsworth, A., Heckley, W., and Delsol, F.: An evaluation of the performance of the ECMWF operational system in analyzing and forecasting easterly wave disturbances over Africa and the tropical Atlantic, *Monthly weather review*, 116, 824–865, 1988.
- Rice, J.: The Durivation of Computer-based Synoptic Climatology of Southern Hemisphere Extratropical Cyclones, Unpublished B. Sc. Honours thesis. University of Melbourne, 1982.
- 425 Saha, S., Moorthi, S., Pan, H.-L., Wu, X., Wang, J., Nadiga, S., Tripp, P., Kistler, R., Woollen, J., Behringer, D., Liu, H., Stokes, D., Grumbine, R., Gayno, G., Wang, J., Hou, Y.-T., Chuang, H.-Y., Juang, H.-M. H., Sela, J., Iredell, M., Treadon, R., Kleist, D., Delst, P. V., Keyser, D., Derber, J., Ek, M., Meng, J., Wei, H., Yang, R., Lord, S., Dool, H. V. D., Kumar, A., Wang, W., Long, C., Chelliah, M., Xue, Y., Huang, B., Schemm, J.-K., Ebisuzaki, W., Lin, R., Xie, P., Chen, M., Zhou, S., Higgins, W., Zou, C.-Z., Liu, Q., Chen, Y., Han, Y., Cucurull, L., Reynolds, R. W., Rutledge, G., and Goldberg, M.: The NCEP climate forecast system reanalysis, *Bulletin of the American Meteorological Society*, 91, 1015–1057, 2010.
- 430 Strachan, J., Vidale, P. L., Hodges, K., Roberts, M., and Demory, M.-E.: Investigating Global Tropical Cyclone Activity with a Hierarchy of AGCMs: The Role of Model Resolution, *J. Clim.*, 26, 133–152, doi:10.1175/JCLI-D-12-00012.1, 2013.
- 435 Taljaard, J.: Development, distribution and movement of cyclones and anticyclones in the Southern Hemisphere during the IGY, *Journal of Applied Meteorology*, 6, 973–987, 1967.
- Tsiombikas, J.: kdtree: A simple C library for working with KD-Trees, <https://github.com/jtsiomb/kdtree>, accessed: 2015-09-18, 2015.
- 440 Tsutsui, J.-i. and Kasahara, A.: Simulated tropical cyclones using the National Center for Atmospheric Research community climate model, *Journal of Geophysical Research: Atmospheres*, 101, 15 013–15 032, doi:10.1029/95JD03774, 1996a.
- Tsutsui, J.-i. and Kasahara, A.: Simulated tropical cyclones using the National Center for Atmospheric Research community climate model, *Journal of Geophysical Research: Atmospheres*, 101, 15 013–15 032, 1996b.
- 445 Vitart, F., Anderson, J. L., and Stern, W. F.: Simulation of interannual variability of tropical storm frequency in an ensemble of GCM integrations, *Journal of Climate*, 10, 745–760, 1997.
- Walsh, K., Fiorino, M., Landsea, C., and McInnes, K.: Objectively determined resolution-dependent threshold criteria for the detection of tropical cyclones in climate models and reanalyses, *Journal of climate*, 20, 2307–2314, 2007.
- 450

Walsh, K. J. E., Fiorino, M., Landsea, C. W., and McInnes, K. L.: Objectively Determined Resolution-Dependent Threshold Criteria for the Detection of Tropical Cyclones in Climate Models and Reanalyses, *J. Clim.*, 20, 2307, doi:10.1175/JCLI4074.1, 2007.

Whittaker, L. M. and Horn, L.: Atlas of Northern Hemisphere extratropical cyclone activity, 1958- 1977, 1982.

Williamson, D. L.: Storm track representation and verification, *Tellus*, 33, 513–530, 1981.

Zarzycki, C. M. and Jablonowski, C.: A multidecadal simulation of Atlantic tropical cyclones using a variable-resolution global atmospheric general circulation model, *J. Adv. Model. Earth Syst.*, 6, 805–828, doi:10.1002/2014MS000352, 2014.

Zhao, M., Held, I. M., Lin, S.-J., and Vecchi, G. A.: Simulations of Global Hurricane Climatology, Interannual Variability, and Response to Global Warming Using a 50-km Resolution GCM, *J. Clim.*, 22, 6653–6678, 2009.