Manuscript prepared for Geosci. Model Dev. Discuss.
with version 4.1 of the LaTeX class copernicus_discussions.cls.
Date: 11 July 2016

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: paullrich@ucdavis.edu

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

Abstract

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

1 Introduction

The need for automated 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...

...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.

...does not contain hard-coded variable names, so as to ensure robust applicability across reanalysis datasets and applicability to a variety of problems.

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

Well-known automated software trackers include Kevin Hodges' TRACK code (Hodges, 2015) and the Geophysical Fluid Dynamics Laboratory (GFDL) TSTORMS package (**?**) Others?. Both of these software packages have been used extensively throughout the literature for studies examining pointwise features in the atmosphere. However, there is a growing need for scale-insensitive detection techniques and unstructured grid support that is not well met with existing

schemes. Consequently, it is the goal of this paper is to describe a tracking technique that can be applied on essentially any grid at any resolution, and provide insight into currently available algorithmic techniques underlying modern tracking algorithms.

As observed by Neu et al. (2013) 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 thresholds and variables.

The software described in this manuscript has been released as part of the TempestExtremes software package, and is available for use under the Lesser GNU Public License (LGPL). All software can be obtained from GitHub via the following clone URL:

```
https://github.com/ClimateGlobalChange/tempestextremes
```

2 Automated tracking

2.1 Extratropical cyclones

Manual counts of cyclones were performed by Petterssen (1956) in the Northern hemisphere from 1899-1939, and latter binned by Klein (1957) to determine the spatial distribution of such storms. These techniques were later refined by Whittaker and Horn (1982) by accounting for cyclone trajectories. A similar survey in the Southern hemisphere was performed by Taljaard (1967) for July 1957 - December 1958. Manual tracking and characterization of cyclones was also performed by Akyildiz (1985) using ECMWF forecast data for the 1981/82 winter.

One of the first automated detection and tracking for extratropical cyclones was developed by Williamson (1981) 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 Lambert (1988) using local minima in 1000-hPa geopotential height. This method had some shortcomings, including mischaracterization of lo-

cal lows due to Gibbs' ringing and topographically-driven lows. To overcome these problems, Alpert et al. (1990) proposed an additional minimum threshold on the local pressure gradient. Similarly, Le Treut and Kalnay (1990) 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 successive 6- or 12- hour intervals.

Murray and Simmonds (1991) extracted low pressure centers from interpolated GCM data using local optimization, based on earlier work in Rice (1982).

Feature tracking on the sphere was revisited by Hodges (1995), which extended tracking algorithms designed for Cartesian geometry Hodges et al. (1994) that were built from image processing techniques.

2.2 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 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 (Vitart et al., 1997; Zhao et al., 2009; Murakami et al., 2012), geopotential thickness (Tsutsui and Kasahara, 1996) and decay of vorticity with height (Bengtsson et al., 2007; Strachan et al., 2013). 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 (Camargo and Zebiak, 2002; Walsh et al., 2007; Murakami and Sugi, 2010; Murakami et al., 2012). In Broccoli and Manabe (1990), 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 Haarsma et al. (1993) instead detected maxima in the 850mb relative vorticity (greater than $3.5 \times 10^{-5}\mathrm{s}^{-1}$ 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. Bengtsson et al. (1995) 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. Tsutsui and Kasahara (1996) 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. Camargo and Zebiak (2002) 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 Walsh et al. (2007), 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 (Landsea et al., 2010).

2.3 African Easterly Waves

A similar manual study to Akyildiz (1985) was performed by Reed et al. (1988) to detect and track African easterly waves.

Storm tracks and African easterly waves were tracked by Hodges et al. (2003).

König et al. (1993) uses minima in the 1000 hPa geopotential height to identify cyclones with associated maxima in the 850 hPa vorticity.

2.4 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 (Prabhat et al., 2012; Zarzycki and Jablonowski, 2014). 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). This is because the Map() functionality, which is typically the most computationally intensive, is also embarrassingly parallel.

3 Algorithms

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

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

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

3.1 Efficient neighbor search using k-d Trees

Three-dimensional ($k = 3$) k-d trees (Bentley, 1975) are used throughout our detection code using the implementation of Tsiombikas (2015). 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_kd_tree(P)` constructs a k-d tree `K` from a point set `P`.

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

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

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.

3.2 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 1. 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>
...
```

3.3 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 1). Candidates are then eliminated if they are "too close" to stronger extrema (Algorithm 2). The command line specification for the search procedure in TempestExtremes can be found in Appendix A1.

3.4 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 definite "physical" criterion. Consequently, we instead advocate for a *closed contour criteria* to define candidate nodes. Closed contours were first employed by Bell and Bosart (1989), 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 `p0` lead to a sufficiently large decrease (or increase) in a given field `G`. This criteria is illustrated in Figure 2, and is implemented in Algorithm 3 and 4 (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).

3.5 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 5.

3.6 Stitching

The basic track stitching procedure (which represents the Reduce() stage in MapReduce) is implemented in Algorithm 6 using the output from the Map() procedure at each time level (stored in set array `P[1..T]`), a maximum great-circle-distance between nodes, `dist`, and a maximum gap size, `maxgap`. Here, gap size refers to the maximum number of sequential non-detections that can occur before a path is considered terminated. This argument is useful, for instance, for tracking tropical storms that temporarily weaken before strengthening into tropical cyclones.

For simplicity, k-d trees are constructed at each time level in order to maximize the efficiency of the search. Each candidate pair (time, node) can only be used in one path, and so construction simply requires exhausting the list of available candidates. Once paths have been constructed, additional criteria can be applied – for instance, minimum path length or additional criteria based on minimum path length or minimum distance between the start and endpoints of the path (see Appendix A2). Thresholds based on field values may also be applied, *e.g.* sea level pressure must be below a particular value for at least 8 time steps of each track.

4 Selected examples

Several 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 (Saha et al., 2010).

4.1 Tropical cyclones in CFSR

Our first example is to employ the tracker for tropical cyclones. 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 "$uvfile;$tpfile;$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 (`pgbhnl.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 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 (`zs`) be at most 100 meters. The associated command line for StitchNodes is:

```
./StitchNodes
  --in pgbhnl.dcu_tc_all.dat
  --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"
```

Once the complete set of tropical cyclone paths has been computed, total tropical cyclone counts over each 0.5 degree grid cell is plotted in Figure 3. Overall the results show very good agreement with reference fields (Citation?).

4.2 Extratropical cyclones in CFSR

For our second example, we are interested in tracking extratropical cyclone features. The command line we have used to detect cyclonic features without the characteristic warm-core of tropical cyclones (here referred to as extratropical cyclones) is given below. The command is identical to the TC detection configuration specified in section 4.1, except requiring that the feature does not possess a closed contour in the 400hPa temperature field (no warm core).

```
./DetectCyclonesUnstructured
  --in_data "$uvfile;$tpfile;$hfile" --out $outf
```

```
  --searchbymin PRMSL_L101 --mergedist 2.0
  --closedcontourcmd "PRMSL_L101,200.,4,0?
  --noclosedcontourcmd "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"
```

Stitching is similarly analogous to section 4.1, except using a slightly more strict criteria on the underlying topographic height. The topographic filtering proved necessary in order to adequately filter out an abundance of topographically-driven low pressure systems, particularly in the Himalayas region. The command line used for stitching is given below:

```
./StitchNodes
  --in pgbhnl.dcu_tc_all.dat
  --out pgbhnl.dcu_tc_stitch.dat
  --format "i,j,lon,lat,psl,maxu,zs"
  --range 8.0 --minlength 8 --maxgap 2
  --threshold "zs,<=,70.0,8"
```

Once the complete set of extratropical cyclone paths has been computed, total extratropical cyclone density over each 0.5 degree grid cell is plotted in Figure 4. Although not extensively verified, the qualitative density of extratropical cyclones is well within the range of results from different trackers, as given by Neu et al. (2013).

4.3 Northern hemisphere tropical easterly waves in CFSR

Tropical easterly waves are our third example of a pointwise feature that has been assessed in the tracking literature. In this example, tracking is performed using maxima in the 850hPa relative vorticity field. Since CFSR only provides absolute vorticity, relative vorticity must first be extracted by taking the difference between absolute vorticity and the planetary vorticity (the Coriolis parameter). This is done on the command line via `_DIFF(ABS_V_L100,_F())`, where `ABS_V_L100` is the CFSR absolute vorticity variable and `_F()` is a built-in function

for computing the Coriolis parameter (equal to $f = 2\Omega \sin\phi$). Here, we have isolated tropical easterly wave features by requiring a drop of relative vorticity equal to 5×10^{-5} s^{-1}, and have discarded detections outside of the latitudinal range $[30S, 30N]$. The command line used is as follows:

```
./DetectCyclonesUnstructured
  --in_data "$uvfile;$hfile" --out $outf
  --searchbymax "_DIFF(ABS_V_L100,_F())" --mergedist 2.0
  --closedcontourcmd "_DIFF(ABS_V_L100,_F()),-5.e-5,4,0"
  --outputcmd "_DIFF(ABS_V_L100,_F()),max,0;
      HGT_L1,max,0"
  --maxlat 30.0
```

Tropical easterly wave paths are constructed using a maximum distance of 4° great-circle-distance between subsequent detections, a minimum path length equal to 8 sequential detections and no allowed gaps. We further require that the easterly waves have a distance of at least 16° between start and endpoints and are present in the northern hemisphere ($\phi \geq 0°$) for at least 8 timesteps. The command line is as follows.

```
./StitchNodes
  --in pgbhnl.dcu_aew_all.dat --out pgbhnl.dcu_aew_stitch.dat
  --format "i,j,lon,lat,relv,zs"
  --range 4.0 --minlength 8 --maxgap 0
  --min_endpoint_dist 16.0
  --threshold "lat,>=,0,8"
```

Counts of tropical easterly waves within each CFSR grid volume are given in Figure 5, showing heavy wave activity throughout the Atlantic and Pacific basins. Examining the Figure 3, it is clear that regions with a northward bulge in tropical Easterly wave activity are also key regions for tropical cyclone development.

4.4 Tropical cyclones in a simulation with CESM

For our final example, we assess the differences in tropical cyclone character obtained from native and regridded datasets. Using the variable-resolution option in the Community Earth System Model (VR-CESM) to refine the northern hemisphere to 0.25° resolution, a simulation of a complete hurricane season (September - January) has been performed. With the high-order spectral element dynamical core used to solve the fluid equations in the atmosphere, VR-CESM has been demonstrated to be effective in simulating tropical cyclone-like features (Zarzycki and Jablonowski, 2014; Zarzycki et al., 2014). However, even at the relatively fine global resolution of 0.25°, the eye of the tropical cyclone is only partially resolved. Since VR-CESM uses an unstructured mesh with degrees of freedom stored at spectral element Gauss-Lobatto (GL) nodes, data is typically analyzed only after being regridded to a regular latitude-longitude mesh of approximately equal resolution. The regridding procedure has the potential to smear out grid-scale features.

For this example, we use the high-order regridding package TempestRemap (Ullrich and Taylor, 2015; Ullrich et al., 2016) for remapping the native spectral element output to a regular latitude-longitude grid with 0.25° grid spacing. For purposes of determining connectivity on the variable-resolution spectral element mesh, we connect GL nodes along the coordinate axis of each quadrilateral element. DetectCyclonesUnstructured is then applied to both the native grid data and the regridded data on the regular latitude-longitude mesh (using the configuration specified in section 4.1) and tropical cyclones categorized by maximum wind speed (Colin: more here). The results of this analysis are depicted in Figure 6. As expected, the native grid output produces almost identical tracks, but more powerful tropical cyclones overall (with many tropical cyclones dropping down by a full category as a consequence of the remapping procedure).

5 Conclusions

Acknowledgements.

A1 Software Documentation: DetectCyclonesUnstructured

```
Usage: DetectCyclonesUnstructured <parameter list>
Parameters:
  --in_data <string> [""]
  --in_connect <string> [""]
  --out <string> [""]
  --searchbymin <string> [""] (default PSL)
  --searchbymax <string> [""]
  --maxlat <double> [0.000000] (degrees)
  --minlat <double> [0.000000] (degrees)
  --topofile <string> [""]
  --maxtopoht <double> [0.000000] (m)
  --mergedist <double> [0.000000] (degrees)
  --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]
  --regional <bool> [false]
  --out_header <bool> [false]
  --verbosity <integer> [0]
```

`--in_data <string>`
A list of input datafiles in NetCDF format, separated by semi-colons.

`--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.

`--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>`
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.

`--minlat <double>`
The minimum absolute latitude for candidate points. Candidates at lower latitudes are discarded.

`--mergedist <double>`
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.

`--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.

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>;...
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.

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).

`--timestride <integer>`
Only examine discrete times at the given stride (by default 1).

`--regional`
When a latitude-longitude grid is employed, do not assume 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

`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).

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,

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

A2 Software Documentation: StitchNodes

```
Usage: StitchNodes <parameter list>
Parameters:
  --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)
  --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)
```

--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>
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).

`--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).

`--maxgap <integer>`
The largest gap (missing candidate nodes) along the path (in discrete time points).

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

`--timestride <integer>`
Only examine discrete times at the given stride (by default 1).

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 = find_all_minima(field G)
  for each node f
    is_minima[f] = true
    for each neighbor node v of f
      if G[v] < G[f] then
        is_minima[f] = false
    if 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
```

Algorithm 3. Find the node `pmax` 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 `p0`, defined by `p0 = find_max_near(p, G, maxdist)`, within distance `dist`. That is, along all paths away from `p0`, the field `G` must drop by at least `thresh` within distance `dist`. The closed contour criteria is depicted in Figure 2. 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
```

Algorithm 5. Determine if a candidate node `p` satisfies the requirement that there exists another node `p0` within distance `dist` of `p` with `G[p] > 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
```

Algorithm 6. Determine all feature paths `S`, given array of candidate 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, maxgap)
  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
      gap = 0
      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 if (gap < maxgap) then
          gap = gap + 1
        else
          break
      add s into S
```

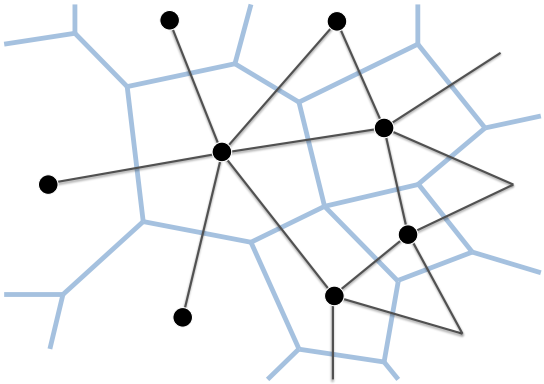

Fig. 1. An example adjacency graph describing an unstructured grid (blue lines), where nodes are co-located with volume centerpoint locations (solid circles) and edges connect adjacent volumes.

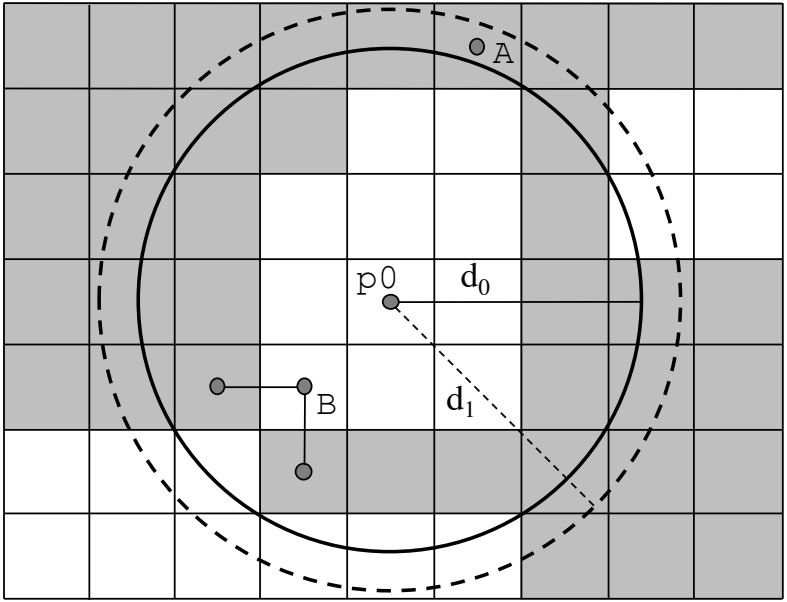

Fig. 2. An illustration of the closed contour criteria. Nodes shaded in white (gray) satisfy (do not satisfy) the threshold of the field value at `p0`. Since only edge-neighbors are included, `B` constitutes a boundary to the interior of the closed contour. Because `A` lays outside the solid circle, 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.

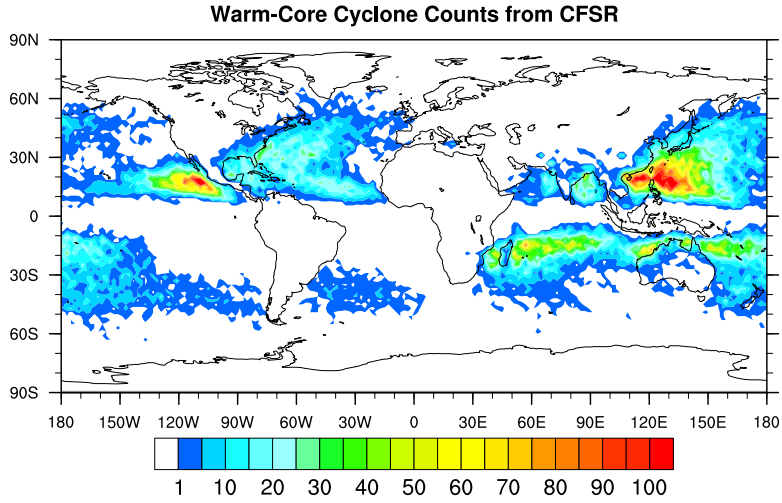

Fig. 3. Tropical cyclone counts over the period 1979-2010 obtained using the procedure described in section 4.1.

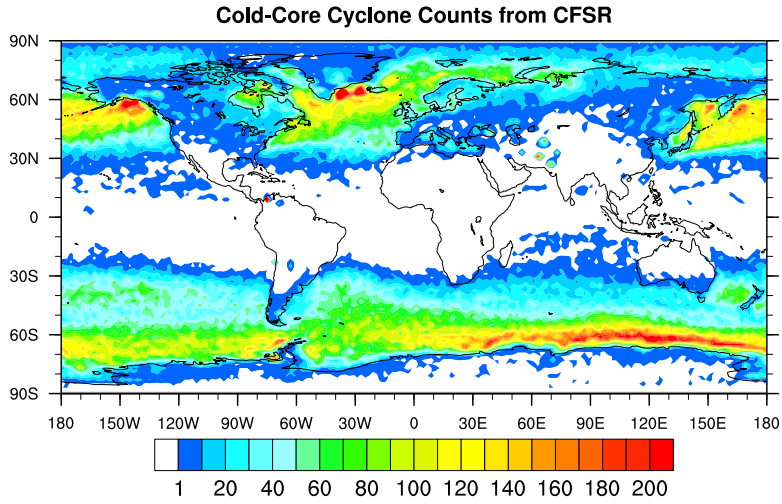

Fig. 4. Extratropical cyclone counts over the period 1979-2010 obtained using the procedure described in section 4.2.

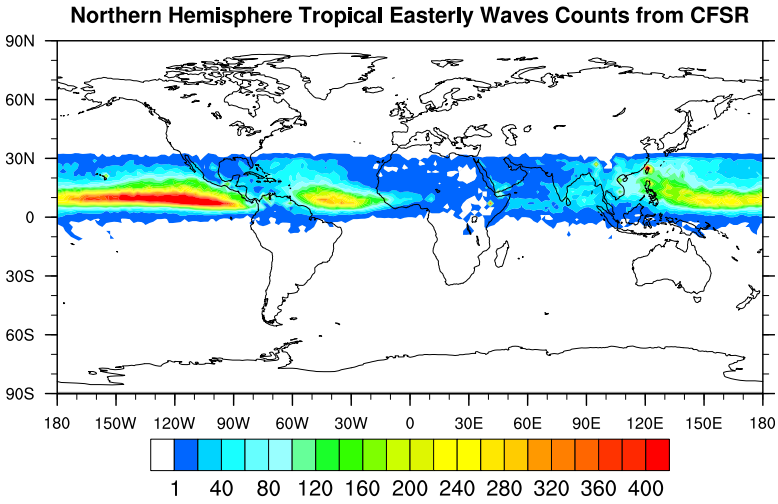

Fig. 5. Tropical easterly wave counts over the period 1979-2010 obtained using the procedure described in section 4.3.

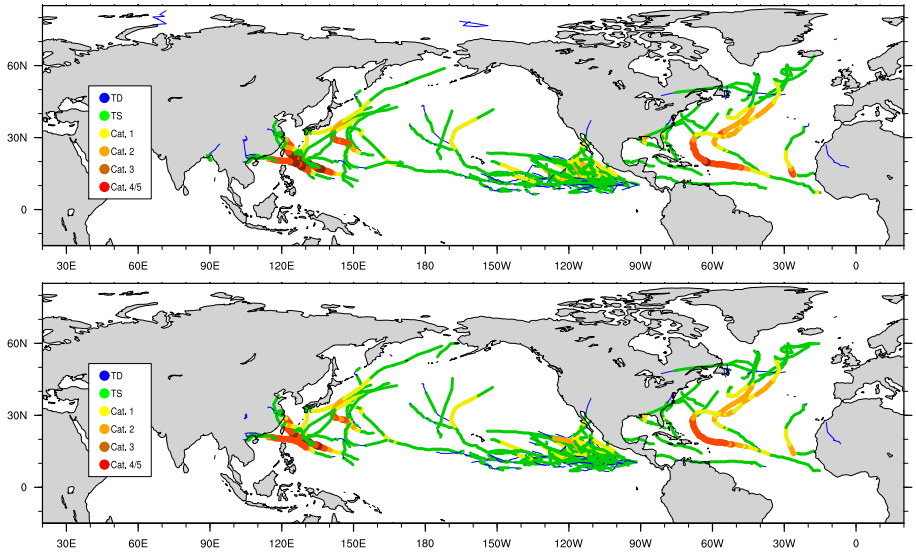

Fig. 6. Tropical cyclone trajectories and associated intensities as obtained from the simulation of a single hurricane season in CESM 4.4 using (top) native spectral-element grid data and (bottom) data regridded to a regular latitude-longitude grid with 0.25° grid spacing.

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?, Geophys. Res. Lett., 17, 1917–1920, 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, 2002.

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.

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.: TRACK, http://www.nerc-essc.ac.uk/~kih/TRACK/Track.html, accessed July 8, 2016, 2015.

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.

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.

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.

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.

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.

Neu, U., Akperov, M. G., Bellenbaum, N., Benestad, R., Blender, R., Caballero, R., Cocozza, A., Dacre, H. F., Feng, Y., Fraedrich, K., et al.: IMILAST: a community effort to intercompare extratropical cyclone detection and tracking algorithms, Bulletin of the American Meteorological Society, 94, 529–547, 2013.

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.

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 Synophic Climatology of Southern Hemisphere Extratropical Cyclones, Unpublished B. Sc. Honours thesis. University of Melbourne, 1982.

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.

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.

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.

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, 1996.

Ullrich, P. A. and Taylor, M. A.: Arbitrary-Order Conservative and Consistent Remapping and a Theory of Linear Maps: Part I, Monthly Weather Review, 143, 2419–2440, doi:10.1175/MWR-D-14-00343.1, http://journals.ametsoc.org/doi/abs/10.1175/MWR-D-14-00343.1, 2015.

Ullrich, P. A., Devendran, D., and Johansen, H.: Arbitrary-Order Conservative and Consistent Remapping and a Theory of Linear Maps, Part 2, Monthly Weather Review, In Press, doi:10.1175/MWR-D-15-0301.1, http://journals.ametsoc.org/doi/abs/10.1175/MWR-D-15-0301.1, 2016.

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.

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, Journal of Advances in Modeling Earth Systems, 6, 805–828, doi:10.1002/2014MS000352, 2014.

Zarzycki, C. M., Jablonowski, C., and Taylor, M. A.: Using Variable-Resolution Meshes to Model Tropical Cyclones in the Community Atmosphere Model, Monthly Weather Review, 142, 1221–1239, 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.