TempestExtremes

1 Files contained in `src/blocking`

1.1 Explanation of blocking methods

There are two main methods for block detection: the Z500 gradient method of Tibaldi and Molteni 1990 (hereafter referred to as TM90), and anomaly-based methods such as that of Dole and Gordon 1983 (Z500 anomaly, DG83) or Schwierz et al 2004 (potential vorticity anomaly, S04). Because TM90 is based on an instantaneous field, the calculations are done in a single step with the `BlockingGHG` program, as explained in Section 1.2. The anomaly methods involve a multi-step process that will be outlined in section 1.3.

1.2 BlockingGHG

```
Usage: BlockingGHG <parameter list>
Parameters:
  --in <string> [""]
  --inlist <string>  [""]
  --varname <string> [""]
  --tname <string> ["time"]
  --latname <string> ["lat"]
  --lonname <string> ["lon"]
```

`--in <string>`
input file in NetCDF format, with axes `[time, latitude, longitude]`.

`--inlist <string>`
list of input datafiles (one file per line) in NetCDF format, with axes `[time, latitude, longitude]`.

`--varname <string>`
name of Z500 variable

`--tname <string>`
Name of the time axis (default `time`).

`--latname <string>`
Name of the latitude axis (default `lat`).

`--lonname <string>`
Name of the longitude axis (default `lon`).

1.2.1 File input and output

This program takes each input file containing the Z500 variable (specified by the `--varname` flag) and performs calculations for all gridpoints in the 35-75° latitude range (both hemispheres). The output file contains the variable `GHGrad`, which has a value of either 1 (gridpoint is instantaneously blocked) or 0 (not blocked).

1.3 Anomaly Method

1.4 AvgVar

```
Usage: AvgVar <parameter list>
Parameters:
  --inlist <string>  [""]
  --out <string> [""]
  --varlist <string> [""]
  --tname <string> ["time"]
  --latname <string> ["lat"]
  --lonname <string> ["lon"]
```

`--inlist <string>`
A list of input datafiles (one file per line) in NetCDF format, with axes `[time, latitude, longitude]`.

`--out <string>`
The output filename.

`--varlist <string>`
List of variable names, separated by commas (with no spaces). Example: `T,U,V`

`--tname <string>`
Name of the time axis (default `time`).

`--latname <string>`
Name of the latitude axis (default `lat`).

`--lonname <string>`
Name of the longitude axis (default `lon`).

1.4.1 File input and output

This program takes a list of files and, for the specified list of variables, averages quantities from all files together along the time axis. The program assumes that input variables are on a single vertical level and therefore have the axes `[time, latitude, longitude]`.

The output file contains time-averaged variables with axes `[latitude, longitude]`, and have the same names as the original inputs .

1.5 BlockingAvg

```
Usage: BlockingAvg <parameter list>
Parameters:
  --inlist <string>  [""]
  --out <string> [""]
  --varname <string> [""]
  --avgname <string> [""]
  --missing
  --tname <string> ["time"]
  --latname <string> ["lat"]
  --lonname <string> ["lon"]
```

`--inlist <string>`
A list of input datafiles (one file per line) in NetCDF format. All files contained within the list will be averaged together along the time axis.

`--out <string>`
The output filename.

`--varname <string>`
Name of input variable which will be averaged.

`--avgname <string>`
Name of output long term daily average variable.

`--missing`
Boolean in the case of files that are missing from the sequence (an exception will be raised if this flag is not included and files are not consecutive in time).

`--tname <string>`
Name of the time axis (default `time`).

`--latname <string>`
Name of the latitude axis (default `lat`).

`--lonname <string>`
Name of the longitude axis (default `lon`).

1.5.1 Averaging method

The averaging method is essentially a smoothed long term daily average. Per year, each day is averaged with the 15 days preceding and succeeding that day; each smoothed day is averaged with the corresponding smoothed days from other years.

1.5.2 File input and output

This program takes a list of files and produces a variable that is the long term daily average of variable specified by `--varname`. The program assumes that input variables are on a single vertical level and therefore have the axes `[time, latitude, longitude]`.

The input list must have the files in chronological order for this program to work. Because the program checks the length of time in between one file and the next, the `--missing` flag must be included if there are files that are missing from the sequence, otherwise an exception will be raised and the program will terminate.

The output file contains the long term daily average variable, with the name specified by `--avgname`. The averaged variable is 365 days (leap days are omitted) along the time axis and has time units of "days since 0001-12-31".

1.6 BlockingDevs

```
Usage: BlockingDevs <parameter list>
Parameters:
  --inlist <string>  [""]
```

```
  --avg <string> [""]
  --varname <string> [""]
  --avgname <string> [""]
  --pv
  --gh
  --tname <string> ["time"]
  --latname <string> ["lat"]
  --lonname <string> ["lon"]
```

`--inlist <string>`
A list of input datafiles in NetCDF format.

`--avg <string>`
The name of the long term daily average file

`--varname <string>`
Name of input variable for which anomaly will be calculated.

`--avgname <string>`
Name of input long term daily average variable.

`--pv` or `--gh`
Boolean which specifies whether the vertically averaged potential anomaly (`--pv`) or 500 mb geopotential height (`--gh`) anomaly will be calculated. One or the other must be specified, not both.

`--tname <string>`
Name of the time axis (default `time`).

`--latname <string>`
Name of the latitude axis (default `lat`).

`--lonname <string>`
Name of the longitude axis (default `lon`).

1.6.1 File input and output

This program takes a list of files and calculates anomalies from the long term daily mean (the file which was calculated in Section 1.5). There are two options for outputs, specified by the Boolean `--pv` (potential vorticity) and `--gh` (500 mb geopotential height) flags.

For each input file, the program produces a corresponding output file containing the calculated anomalies with the suffix `_devs` appended to the original file name. There are 3 output variables: the unsmoothed anomaly (`DIPV/DGH`), the anomaly with 2-day smoothing (`ADIPV/ADGH`), and the smoothed anomalies divided by the threshold value (`INT_ADIPV/INT_ADGH`).

1.7 BlockingGHG

```
Usage: BlockingGHG <parameter list>
Parameters:
  --in<string>  [""]
  --out <string> [""]
  --varname <string> [""]
  --tname <string> ["time"]
  --latname <string> ["lat"]
  --lonname <string> ["lon"]
```

`--in <string>`
The name of the input NetCDF file (must contain variable with 500 mb geopotential height).

`--out <string>`
The name of the output NetCDF file (if omitted, a file with the suffix `_GHG` appended to the file name will be produced).

`--varname <string>`
Name of input 500 mb height variable.

`--tname <string>`
Name of the time axis (default `time`).

`--latname <string>`
Name of the latitude axis (default `lat`).

`--lonname <string>`
Name of the longitude axis (default `lon`).

1.7.1 Brief algorithm description

2 DetectCyclonesUnstructured

```
Usage: DetectCyclonesUnstructured <parameter list>
Parameters:
  --in_data <string> [""]
  --in_data_list <string> [""]
  --in_connect <string> [""]
  --out <string> [""]
  --out_file_list <string> [""]
  --searchbymin <string> [""] (default PSL)
  --searchbymax <string> [""]
  --minlon <double> [0.000000] (degrees)
  --maxlon <double> [0.000000] (degrees)
  --minlat <double> [0.000000] (degrees)
  --maxlat <double> [0.000000] (degrees)
  --topofile <string> [""]
  --maxtopoht <double> [0.000000] (m)
  --mergedist <double> [0.000000] (degrees)
  --closedcontourcmd <string> [""] [var,delta,dist,minmaxdist;...]
  --noclosedcontourcmd <string> [""] [var,delta,dist,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_data_list <string>`
A file containing the `--in_data` argument for a sequence of processing operations (one per line).

`--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.

`--out_file_list <string>`
A file containing the `--out` argument for a sequence of processing operations (one per line).

`--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.

`--minlon <double>`
The minimum longitude for candidate points.

`--maxlon <double>`
The maximum longitude for candidate points.

`--minlat <double>`
The minimum latitude for candidate points.

`--maxlat <double>`
The maximum latitude for candidate points.

`--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,delta,dist,minmaxdist`. 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.
>
> `minmaxdist <double>` The distance away from the candidate to search for the minima/maxima. 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).

2.1 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 operators are also supported, including

`_ABS(<variable>)` Absolute value of a variable,

`_AVG(<variable>, <variable>)` Pointwise average of variables,

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

_F() Coriolis parameter,

_MEAN(<variable>, <distance>) Spatial mean over a given radius,

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

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

For instance, the following are valid examples of <variable> type,

_MEAN(PSL,2.0), _VECMAG(U850, V850) and _DIFF(U(3),U(5)).

2.2 MPI Support

The DetectCyclonesUnstructured executable supports parallelization via MPI when the --in_data_list argument is specified. When enabled, the parallelization procedure simply distributes the processing operations evenly among available MPI threads.

3 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).