TempestExtremes

Contents

1 Objective blocking detection 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).

Geopotential height gradient (TM90) Two slopes are calculated around a single gridpoint:

$$GHGN = \frac{Z(\phi_n) - Z(\phi_0)}{\phi_n - \phi_0} \tag{1}$$

$$GHGS = \frac{Z(\phi_0) - Z(\phi_s)}{\phi_0 - \phi_s} \tag{2}$$

where ϕ_0, ϕ_n, and ϕ_s are the reference latitude and the latitudes 15° poleward and equatorward of ϕ_0, respectively, and $GHGN$ and $GHGS$ are the height gradients. For $GHGN < -10$ m/deg lat and $GHGS > 0$ m/deg lat, the point is considered instantaneously blocked.

This is a modification of the original TM90 algorithm, which was calculated for all longitude values about a single predetermined latitude. Scherrer et al 2006 and others expanded the definition to encompass all latitudes 35-75 degrees in both hemispheres.

Anomaly methods (S04 and DG83) Each anomaly method has a general workflow that can be described as follows:

1. Generate the input files, using either `Var4Dto3D` for vertically averaged potential vorticity (Section 1.3) or Z500 (Section (STILL TO COME], optional if already have files with Z500) or `BlockingPV`.
2. Calculate the long term daily average, using `BlockingDFT` (Section)
3. Calculate the anomalies from the long term daily average, using `BlockingDevs` (Section 1.5)
4. (Optional) Calculate the long term daily average of the anomalies using `BlockingDFT` in order to use in the threshold calculations
5. (Optional) Calculate a spatiotemporally varying threshold variable using `BlockingThresh`
6. Calculate the normalized anomalies using `BlockingNormDevs`

Potential Vorticity anomaly: S04 uses vertically averaged (150-500 hPa) potential vorticity as the variable of choice for block detection. The presence of negative (positive) anomalies in the Northern (Southern) Hemisphere denotes either a slowing or reversal of the flow with respect to the climatological mean.

Z500 anomaly: DG83 uses 500 hPa geopotential height (Z500) anomaly as the variable of choice for block detection. The anomaly signifies heights that exceed a threshold above a climatological mean,

1.1 Optional flags

Some of the flags, such as `--latname`, are optional; if not provided, they will default to the value specified in the code. These flags are not all available for each program, so check the documentation to see which ones

can be utilized (you can also run the desired program without any flags, which will print out the command list, raise an exception, and terminate).

1.1.1 File and variable names

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

`--levname <string>`
Name of the vertical level axis (default `lev`).

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

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

`--insuff <string>`
Input file extension string (default `.nc`).

`--outsuff <string>`
Output file naming convention (default varies between programs).

`--insuff2d <string>`
Input surface pressure file extension string (default `.nc`).

`--outsuffipl<string>`
Output interpolated variable file naming convention (default varies between programs).

1.2 BlockingGHG

```
Usage: BlockingGHG <parameter list>
Parameters:
  --in <string> [""]
  --inlist <string>  [""]
  --varname <string> ["Z"]
  --tname <string> ["time"]
  --latname <string> ["lat"]
  --lonname <string> ["lon"]
  --insuff <string> [".nc"]
  --outsuff <string> ["_GHG.nc"]
```

`--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 (default `Z`).

Note: do not use both `--in` and `--inlist` flags simultaneously.

1.2.1 File input and output

This program takes each input file containing the Z500 variable (specified by the `--varname` flag if name is not default `Z`) and outputs a file (original file name plus output suffix), containing the variable `ZG`, which has a value of either 1 (gridpoint is instantaneously blocked) or 0 (not blocked).

1.3 Anomalies, Step 1a: Input files: BlockingPV

```
Usage: BlockingPV <parameter list>
Parameters:
  --in <string> [""]
  --in2d <string> [""]
  --inlist <string>  [""]
  --inlist2d <string> [""]
  --ipl
  --hpa
  --hasPV
  --PVname <string> ["PV"]
  --VPVname <string> ["VPV"]
  --uname <string> ["U"]
  --vname <string> ["V"]
  --tempname <string> ["T"]
  --tname <string> ["time"]
  --levname <string> ["lev"]
  --latname <string> ["lat"]
  --lonname <string> ["lon"]
  --insuff <string> [".nc"]
  --outsuff <string> ["_integ.nc"]
  --insuff2d <string> [".nc"]
  --outsuff2d <string> ["_ipl.nc"]
```

`--in <string>`
Input file in NetCDF format containing temperature and wind variables, with axes `[time, lev, lat, lon]`

`--in2d <string>`
Optional input file in NetCDF format containing surface pressure, with axes `[time, lat, lon]`. Only necessary if input file's vertical axis is on model levels.

`--inlist <string>`
A list of input datafiles in NetCDF format containing temperature and wind variables, with axes `[time, lev, lat, lon]`

`--inlist2d <string>`
Optional list of input datafiles in NetCDF format containing surface pressure, with axes `[time, lat, lon]`. Only necessary if input files' vertical axis is on model levels.

`--ipl`
Optional bool directing program to create new input file with vertical pressure axis (using input file with vertical model level axis and surface pressure file)

`--hpa`
Optional bool that tells program that pressure levels have units of hPa rather than Pa.

`--hasPV`
Optional bool that tells program that PV variable is already present, and only vertically averaged PV needs to be calculated.

`--PVname <string>`
Name of the potential vorticity variable (default `PV`).

`--VPVname <string>`
Name of the output vertically averaged potential vorticity variable (default `VPV`)

`--uname <string>`
Name of the zonal wind variable (default `U`).

`--vname <string>`
Name of the meridional wind variable (default `V`).

`--tempname <string>`
Name of the temperature variable (default `T`).

Note: do not use both `--in` and `--inlist` flags simultaneously. Also, do not use both `--in2d` and `--inlist2d` flags simultaneously.

1.3.1 File input and output

Without the `--hasPV flag`, this program takes input files containing temperature (`T`) and zonal (`U`) and meridional (`V`) winds and outputs files (original file name plus suffix `_integ.nc`) with both potential vorticity (`PV`) and vertically averaged potential vorticity (`VPV`).

If potential vorticity variable is already present, use the `--hasPV` flag (with optional `--PVname` flag if name of variable differs from the default) and the program will output new files (original file name plus suffix `_integ.nc`) containing the vertically averaged `IPV` variable.

Input files should have a vertical axis on pressure levels with units of Pa (must contain pressure levels from 15000-50000 Pa for the code to work). If the vertical axis is on model levels, use the `--ipl` flag, as well as input surface pressure files (`--in2d` or `--inlist2d`) to first interpolate variables to pressure levels. If the pressure levels are in units of hPa (or mb), use the `--hpa` flag to inform the program that the units need to be converted.

1.4 Anomalies, Step 1b: Input Files: Var4Dto3D

```
Usage: Var4Dto3D <parameter list>
Parameters:
  --in <string>  [""]
  --in2d <string> [""]
  --out <string> [""]
  --hpa
  --ipl
  --gh
  --varlist <string> [""]
  --tname <string> ["time"]
  --levname <string> ["lev"]
  --latname <string> ["lat"]
  --lonname <string> ["lon"]
```

```
  --zname <string> ["Z"]
  --insuff <string> [".nc"]
  --outsuff <string> ["_3D.nc"]
  --insuff2d <string> [".nc"]
  --outsuffipl<string> ["_ipl_3D.nc"]
```

`--in <string>`
Input datafile in NetCDF format, with axes `[time, pressure, latitude, longitude]`.

`--in2d <string>`
Optional input file in NetCDF format containing surface pressure, with axes `[time, lat, lon]`. Only necessary if input file's vertical axis is on model levels.

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

`--hpa`
Optional bool that tells program that pressure levels have units of hPa rather than Pa.

`--ipl`
Optional bool directing program to create new input file with vertical pressure axis (using input file with vertical model level axis and surface pressure file)

`--gh`
Optional bool if geopotential needs to be converted to geopotential height

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

`--zname <string>`
Name of the geopotential/geopotential height variable.

1.4.1 File input and output

This program takes a file with axes `[time, level, latitude, longitude]` and extracts the 500 mb data. If the data is on model levels, there is an option to first interpolate to pressure levels (which will produce an intermediate file with the default file suffix `_ipl_3D.nc`).

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

1.5 BlockingDevs

```
Usage: BlockingDevs <parameter list>
Parameters:
  --inlist <string>  [""]
  --avg <string> [""]
  --varname <string> [""]
  --avgname <string> [""]
  --pv
  --z500
  --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 `--z500`
Boolean which specifies whether the vertically averaged potential anomaly (`--pv`) or 500 mb geopotential height (`--z500`) anomaly will be calculated. One or the other must be specified, not both.

1.5.1 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 1.6.2). There are two options for calculations, 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 2 output variables: the unsmoothed anomaly (`DIPV/DGH`) and the anomaly with 2-day smoothing (`ADIPV/ADGH`).

1.6 Other Tools: 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`

1.6.1 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.

1.6.2 Other Tools: 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`).

1.6.3 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.

1.6.4 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".

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).