

THE UNIVERSITY OF MELBOURNE
CYCLONE DETECTION AND TRACKING
ALGORITHM

Manual V1.0

written and edited by

**Kevin Keay, Ahmad Galea, Martina Messmer,
Acacia Pepler, Irina Rudeva and Ian Simmonds**

School of Earth Sciences, The University of Melbourne
Parkville, Melbourne, Australia

[doi:10.13140/RG.2.2.15518.56645](https://doi.org/10.13140/RG.2.2.15518.56645)

CONTENTS

1	run-scripts	3
2	Compilation	4
	Dependencies:	4
	Fortran Portability:	4
3	read_nc2cmp	5
	Usage:	5
	Examples:	5
	Fortran code to compare cmp (output) with netcdf (input) file	6
4	cycloc.man	8
	Usage:	8
	Summary of options:	8
	Description	8
	Options:	11
	namelist input:	13
	Data formats:	20
5	track.man	24
	Usage:	24
	Summary of options:	24
	Files:	24
	Description:	24
	Options:	25
	Namelist input:	25
	Data formats:	31
6	Troubleshooting and hints	37
	General:	37
	How to define rproj:	37
A	README.mp	39

CYCLONE DETECTION AND TRACKING ALGORITHM

The papers given in the references page form a good starting point for a description of the cyclone detection and tracking algorithm. Below are descriptions of the run-scripts and selected programmes: `read_nc2cmp` (convert NetCDF to conmap files), `cycloc.man` (cyclone finding programme) and `track.man` (cyclone tracking programme). The last section provides some troubleshoots and hints.

Acknowledgements:

Kevin Keay has provided most of the text for this manual.

Ahmad Galea, Martina Messmer, Acacia Pepler, Irina Rudeva and **Ian Simmonds** have updated the code and edited the manual to increase readability.

1 RUN-SCRIPTS

After downloading and compiling the source code (you can also try to use the pre-compiled binaries) the cyclone detection and tracking algorithm can be started using the different run-scripts (combination of bash and csh). The user can decide to run the general scrip *run-tool*, which initiates all the other run-scripts automatically. Please note, that the paths need to be set according to the user's needs and settings in all of the available run-scripts. An other option is to run the scripts individually:

- *run-nc2cmp.topo* converts the topography of the corresponding data set from netcdf into a cmp file format.
- *run-nc2cmp* converts the input netcdf files (e.g., mean sea level pressure) to files with cmp format.
- *run-cyc* identifies the cyclone locations in the given input data set
- *run-trk* connects cyclone locations to full cyclone tracks. The script takes dates from the previous and next year, to make sure that cyclone tracks that exceed the year boundary are also tracked. At the moment these dates are set to 01-Nov-(Year-1) and 31-Jan-(Year+1). This can be adjusted to every user's needs.

Finally, the *read_trk.ncl* script is written in the NCAR Command Language (NCL). It reads in the output of the tracking algorithm and provides an output file that is often easier to read for recent command languages. Please note that even if these programs are all tested, they might still be subject to errors.

2 COMPILATION

Makefiles are used for compilation of all programmes. Core programme features have been tested on Linux and Mac using the *gfortran* compiler. *gfortran* Makefiles have been supplied for all core applications (*source/src.dir*), along with helper bash scripts for unpacking and compiling. Bash scripts have been supplied for compiling all supplied utilities (*source/utls.dir*) using *gfortran/gcc*. See README.mp supplied with source for more information.

NOTE: Files with a '.F' extension pass through a cc preprocessor to collect and check changes in included files.

In order to keep track of modifications to the code, variations to routines and executables are given version numbers. This does not yet apply to the include files themselves. Only the currently used versions of the routines are provided.

DEPENDENCIES:

Utility *read_nc2cmp* requires *netcdf* and *udunits* libraries. *udunits* has dropped Fortran support since v1.12. The last compatible version (v1.12) has been provided as part of this package. For detailed build instructions see README.mp supplied with source.

FORTTRAN PORTABILITY:

While most core programme features have been tested using *gfortran* compiler, software was originally developed on SUNOS and SUN Solaris machines. SUN Fortran allows some constructions which other platforms do not permit, and attention is drawn to the following types of syntax or statement which may cause trouble in untested edge cases.

1. Undefined variables are set to zero on the SUN but may return NaN on other platforms or may give a compilation error. These may still exist in rarely used diagnostics and may be set to zero if found in such places.
2. Some machines do not recognise data format fields such as '1pe3.10.1', which may need to be converted to '1pe3.10', 'e3.10', or something similar.
3. Null do loops may exist in subroutines "cycio", depending on the ending index, e.g.,

```
read ( .... (xv(k),yv(k),ii=2,itabc2),(sc(k,isup),isup=1,nsuplt)
```

where $itabc2 < 2$ or $nsuplt < 1$. *gfortran*, SunOS and Sun Solaris systems skip such loops; but some systems may process the first item of the loop (i.e., $itabc2 = 2$ and $isup = 1$), in which case conditional statements must be included in the code.

3 READ_NC2CMP

This may be used to convert certain kinds of NetCDF data to the 'conmap' format which is required for the first phase of the cyclone tracking scheme (determining cyclone locations). The NetCDF file should have a single variable e.g. mean sea level pressure, or the same variable at multiple levels e.g. geopotential height at 500, 800 hPa. The dimensions should be (time, latitude, longitude) or (level, time, latitude, longitude).

USAGE:

```
read_nc2cmp [-help][-D idbg][-i ncfile][-o cmpfile] [-d "lon,lat,time"][-u uservars][-U vunits][-v vtypes] [-l
levelvars][-L ilev][-g gridtype][-r rtype][-s uscal] [-p udunits][-m no_maps][-M "map1,map2"][-t "date1,date2"]
[-T "ttype"][-G][-S]
```

EXAMPLES:

1. NCEP Mean sea level pressure

There is no level variable and the pressure needs to be rescaled from hPa to Pa, i.e., apply a scale of 0.01 (-s option). Maps 5-8 need to be decoded:

```
read_nc2cmp -i slp.2004.nc -o jj.cmp -u slp -r NCEP -v PMSL -s 0.01 -M "5,8"
```

2. NCEP 500 hPa specific humidity

There is one level, so the example is similar to the previous one:

```
read_nc2cmp -i shum.2005.500.nc -o j.cmp -d "lon,lat,time" -u shum -D 2 -r NCEP -m 2 -l level
-L 1 -v SHUM500 -U "kg/kg"
```

3. ERA40 500 hPa geopotential

For ERA40 the data are automatically rescaled to height (metres). There are 23 levels and level 17 is equal to 500 hPa:

```
read_nc2cmp -i ncdata/hgt.200208.nc -o j.cmp -d "longitude,latitude,time" -u z -D 3 -r ERA40
-l levelist -L 17 -m 2
```

4. Regional GCM 500 hPa heights

There is one level and maps 1-5 need to be decomposed:

```
read_nc2cmp -i z500.nc -o z500.cmp -u HGT_P -v Z500 -r RGCM -U m -d "LON,LAT,TIME"
-l PLEV10.10 -L 1 -M "1,5"
```

5. ERA5 Mslp

ERA5 MSLP has no levels. -i specifies the input file and -o the name of the output file. -d gives the dimension names, please make sure that they exactly agree with the ones in the NetCDF file. -u gives the variable name and this can vary. If ERA5 is downloaded as grib and then converted to netcdf the variable name is different to the one when directly downloading a NetCDF file from the Copernicus Climate Change Service. Please be sure that the variable name here (var1)

and the one in your NETCDF file agrees. -s is again a scale factor as seen in example 1. -r gives the name of the data set and in -g the resolution can be given. This is mainly for the user to keep track on what has been used for the cyclone detection and tracking. -p provides the path to the udunits.dat:

```
read_nc2cmp -I -i era5.slp.1979.nc -o era5.Mslp.1979.cmp -d "lon,lat,time" -u var1 -s 0.01 -v
PMSL -U MB -r ERA5 -g "0.25x0.25DEG" -p ../udunits/bin/udunits.dat
```

FORTRAN CODE TO COMPARE CMP (OUTPUT) WITH NETCDF (INPUT) FILE

This code can be used to compare some variables of the cmp file to the original netcdf file. It helps to understand if the conversion process was successful or not:

Fortran

Program read_cmp

```
integer nlat,nlon
real, allocatable, dimension(:)::lat,lon
real, allocatable::var(:,:)
character*80 head

open(20,file='cmp/erain.Mslp.2008.cmp',form='unformatted')
!open(20,file='mslp.2012.cmp',form='unformatted')

do i=1,5
read(20)nlat; print*, nlat
if(i.eq.1)allocate (lat(nlat))
read(20)lat
read(20)nlon; print*, nlon
if (i.eq.1)allocate (lon(nlon), var(nlon,nlat))
read(20)lon
print*,lon
read(20) head
print*,head
read(20) var
print*,var
print*, "max: ",maxval(var)
print*, "min: ",minval(var)
end do
```

```
close(20)
```

end

The corresponding ncl code is:

NCL (NCAR Command Language)

```
begin
```

```
  fcmp = addfile("erain.slp.2008.nc", "r")
```

```
  slp = short2fft(fcmp->mssl)/100. ; Store some data from the three netCDF
```

```
  print(dimsizes(slp))
```

```
  print(max(slp(0, :, :)))
```

```
  print(min(slp(0, :, :)))
```

```
end
```

4 CYCLOC.MAN

USAGE:

cycloc3x [-d (1 or 2 digits)] [-i infile] [-p ffile] [-c cycfile] [-P] [-C] [-z zfile [-x] [-X] [-Y] [-Z] [-s ffile] [-S ffile] [-F] [-L llfile] [-g zfile] [-u ufile [-v vfile]] [-V 'svfiles' [-N 'svnames']] data

SUMMARY OF OPTIONS:

-d: idiagx,idiagy	diagnostic output levels
-i: infile	namelist input file
-p: ffile	PS output data file
-c: cycfile	cyclone output data file
-P	input PS data (otherwise lat.-lon.)
-C	cyclone diagnostics only
-z: zfile	topography input file
-Z	input PS topog. (otherwise lat.-lon.)
-Y	write PS topography (zfile + .st)
-x	write smoothed PS topog. (zfile + .sl)
-X	write smoothed LL topog. (zfile + .sl)
-s: ffile	smoothed PS output file
-S: ffile	smoothed LL output file
-L: llfile	input file for new lat-lon
-F	input file formatted (mapx)
-g: gfile	geopot.ht. for steering
-u: ufile	u steering vel. file
-v: vfile	v steering vel. file (otherwise as ufile but with v)
-V: svfile	supplementary var. files
-N: svname	supplementary var. names

Further information about the options is given under 'OPTIONS' below.

DESCRIPTION

Cycloc finds positions of maxima and minima in array data. It may be used for any type of field data; but is mainly designed for locating meteorological lows and highs on the sphere, and particularly lows of mean sea level pressure (MSLP or PMSL). For this reason, most of the references are to 'pressure' and 'lows' or 'cyclones'. Some facilities and format specification are particular to this interest and may not be useful or may need to be modified for other applications. These include

1. Open depressions. Lows are normally regarded as pressure minima and are called closed depressions because they are surrounded by closed isobars; but cyclonic disturbances may have

locally large vorticity and may be meteorologically important without being associated with closed isobars. They can be located as maxima of cyclonic relative vorticity or maxima of the pressure Laplacian; or alternatively, in this scheme, they are identified with points of inflexion or minimum gradient in a trough and associated with a maximum of the Laplacian of pressure. When vorticity itself is used as the field variable, this category is not relevant.

2. Strength criterion. Most numerical analyses tend to contain many shallow and meteorologically weak systems, especially in the tropics and in regions of slack pressure gradients. These can be weeded out by applying a criterion of minimum strength. A good measure of the strength is the Laplacian of the pressure. To allow for the fact that a low will normally extend over a radius of several degrees and that the depth of the centre below that of the environment is proportional to the pressure Laplacian and the area, the strength is normally taken from an average of the Laplacian of the pressure over a nominated radius.
3. Weak depressions. For tracking purposes, it is useful to retain a proportion of the weaker systems, but to identify them as such and to encourage a matching of strong systems with strong and weak with weak, other conditions being equal. Some of these weak systems may develop into stronger systems or help to make the track of a strong system continuous over an analysis time in which, possibly to a defect in the analysis, it may have weakened somewhat.
4. Data smoothing. Higher resolution analyses (2.5 deg. or finer) tend to produce rather too many cyclones along trough lines, and some of these may be due to noise in the analysis or undulations in the spectral harmonic representation. Diffusive smoothing, which is most satisfactorily performed on the PS grid, is very effective in reducing the number systems and coalescing closely spaced minima; it has been found to produce a better and more reliable result than simply interpolating from a coarser resolution analysis.
5. Steering velocities. An approximation to the thermal steering velocities may be obtained geostrophically from the PMSL field by averaging gradients over a certain radius. Because the gradients are usually greater in the warm sector, this normally results in a nett easterward or ESE movement and the the correct sort of rotation of systems in complex lows. The calculate velocities are scaled by a suitable factor. The $1/f$ factor is graded down near the Equator, but is not really accurate at low latitudes.

Because the scheme has mainly been used for finding systems in the mid- and high latitudes of the northern or the southern hemisphere, and in order to avoid problems with meridian convergence at high latitudes, depressions are sought on a polar stereographic (PS) array. This is not the optimum arrangement for studies in which only the tropics are of interest; however, the programme does provide for obtaining cyclone positions globally (or over any chosen band of latitudes spanning the equator) and rationalising duplicate locations in the equatorial belt.

Since data are normally available in lat.-lon. form, the programme has two main functions: the interpolation of lat.-lon. data to PS form and the search for lows on the PS grid. By using the appropriate options, these functions can be performed separately. One reason for doing this is that allows low-finding parameter sensitivity and optimisation to be carried out without requiring the grid

transformation to be repeated each time. The main CPU cost is in the bicubic spline interpolation, which is used in both the transformation to PS form and in the differential methods employed in the cyclone search. When obtaining higher order diagnostics to identify programme defects, the programme stops after one analysis time; the '-C' option may be used if a cyclone position file be not required.

Some basic usages are:

<code>cycloc3x -c cyldata latlndata</code>	Find lows from lat.-lon. data
<code>cycloc3x -p psdata -c cyldata latlndata</code>	Write PS data file and also find lows from lat.-lon. data
<code>cycloc3x -p psdata latlndata</code>	Write PS data file only
<code>cycloc3x -P -c cyldata psdata</code>	Find lows from PS data
<code>cycloc3x -P -C psdata</code>	Perform low finding from PS data but do not write cyclone data file

In addition to these functions, it has been found beneficial with some data sets and some applications to include also the following:

1. Retrieval of smoothed data. Like interpolation, smoothing is somewhat CPU intensive, because of the need to observe the diffusive stability criterion. To reduce unnecessary interpolation and smoothing in optimisation runs, it is efficient to write smoothed data files. These may be in either PS or lat.-lon. form. The latter may be used for plotting isobars along with cyclone positions to show the smoothed form of the data from which the latter were derived. This lat.-lon. output need not be of the same resolution as the input data. If the resolution is to be changed, a file of lats. and lons. is provided. Some usages are

<code>cycloc3x -S smlldata -L llfile latlndata</code>	Write smoothed lat.-lon. data file 'smlldata on grid given by 'llfile'
<code>cycloc3x -s smpsdata latlndata</code>	Write smoothed PS data file
<code>cycloc3x -P -c cyldata smpsdata</code>	Find lows from smoothed PS data

2. Topography filtering. Because reductions of pressure to mean sea level is an uncertain procedure which tends to give rise to spurious lows, it is best to filter lows found below elevated topography. There is also a tendency for lows to appear wear the topography is highly curved, i.e., the Laplacian of the topography is large. Often it is when the topogragphy is convex that this happens, and such spurious systems can occur at sea level. These lows may be filtered by applying a penalty to the cyclone strength (the pressure Laplacian) proportional to the topographic height Laplacian. Topographic variation can also affect calculations of geostrophic steering velocities, which need to be weighted down over elevated topography. These filtering procedures require a topography file. It is best to keep the topography used for filtering consistent with the pressures by smoothing it if the pressures be smoothed. The topography may be smoothed during the cyclone run (since it need only be done once, or in advance and written to in PS form.

<code>cycloc3x -c cyldata -z zsdata latlndata</code>	Find lows using zsdata topog.
<code>cycloc3x -x smpszdata zsdata</code>	Write smoothed PS topog. data
<code>cycloc3x -c cyldata -Z -z smpszdata</code>	Find lows using smpszdata

smoothed topography

3. Steering velocities. These may be interpolated from upper level data, either from actual velocities or geostrophically from upper level geopotential heights. In either case, the data are read from separate concatenated files. If the component files differ only by a change from u to v in the first letter, the v file argument need not be given.

```
cyclocx -c cycdata -g zdata latlondata      Calculate steering from z data
cyclocx -c cycdata -u udata latlondata      Calculate steering from u,v data
cyclocx -c cycdata -u udata -v vdata        Calculate steering from u,v data latlondata
```

4. Supplementary variables. These may be interpolated from additional data files, one concatenated file per variable, e.g.

```
cyclocx -c cycdata -V't500.dat z500-850.dat' -n't500 z500-850' latlondata
```

OPTIONS:

- d This option, if used, specifies the diagnostic output required. This is done by giving a number with one or two digits, which are interpreted as the numbers `idiagx` and `idiagy`

<code>idiagx = 1:</code>	Reports main stages of programme flow
<code>idiagy = 4:</code>	Diagnostics of <code>llexpand</code> depending on value of <code>idiagy</code>
<code>idiagy = 5:</code>	Diagnostics of <code>brspls</code> depending on value of <code>idiagy</code>
<code>idiagx = 6:</code>	Diagnostics of <code>cycp</code> depending on value of <code>idiagy</code>
<code>idiagy = 1,2,3</code>	progressively more detailed diagnostics on <code>cycp</code>
<code>idiagy = 4</code>	diagnostics of "fmi"
<code>idiagy = 5</code>	diagnostics of "frmin"
<code>idiagy = 6</code>	diagnostics of "cvave"
<code>idiagy = 7</code>	diagnostics of "steer1"
<code>idiagy = 8</code>	refinement of vorticity centre
<code>idiagy = 9</code>	diagnostics of topographic steering penalties
<code>idiagx = 7</code>	Diagnostics of <code>steer2</code> depending on value of <code>idiagy</code>
<code>idiagx = 8</code>	Diagnostics of <code>supvar</code> depending on value of <code>idiagy</code>

-i infile

Namelist input file for parameters controlling cyclone finding. (See section 'NAMELIST INPUT' below.)

-p ffile

If specified, the field data interpolated to the PS (used for the cyclone search) is written to the file "ffile". This file may be used as input to the cyclone finding programme

- P If specified, the field data are taken to be PS arrays; otherwise, latitude longitude data are assumed. PS data are in the same form as written using the -p option; the format is defined under 'DATA FORMATS' below.

- c cycfile If specified, a cyclone (or anticyclone) search is performed and the cyclone positions are written to "cycfile"

- C If specified, a cyclone (or anticyclone) search is performed but no cyclone position data file is written.
- z zfile
If specified, a topography input file is read. The topography may then be used for topographic filtering of low positions and/or steering velocities or may be interpolated to cyclone positions and entered as a column in the cyclone position file.
- Y If specified, the topography is interpolated to the PS array and written to the file “zfile”.st. The -z option must be used.
- x If specified, the topography is smoothed on the PS array and written in PS form to the file “zfile”.sp. The -z option must be used.
- X If specified, the topography is smoothed on the PS array, reinterpolated to lat.-lon. form, and written to the file “zfile”.sl. The -z option must be used.
- s fsfile
If specified, field data smoothed on the PS array are written to “fsfile”
- S flfile
If specified, field data smoothed on the PS array and reinterpolated to the lat.-lon. are written to “fsfile”.
- L llfile
File containing lats. and lons. for reinterpolating fields of smoothed topography and/or pressure to lat.-lon. form. If not specified, the original lats. and lons. supplied (in the topography and pressure fields, respectively, will be used.
- F Read latitudes and longitude as an ASCII file in the format used by ‘mapx -f’; otherwise, read as an unformatted CIF file.
- g gfile
If specified, steering velocities are calculated geostrophically from geopotential heights given in “gfile”
- u ufile
If specified, steering velocities are read directly. The u-component is taken to be “ufile”.
- v vfile
In conjunction with the -u option, the v-component is taken from “vfile”; if not specified thus, the v-component is taken to have the same name as the u-component file except that the first letter of the file name is ‘v’.
- V svfile
A list of filenames containing supplementary variables whose values are to be interpolated to the cyclone positions are entered into extra columns of the tabulation. If more than one, filenames should be enclosed in quotes
- n svname
Names of supplementary variables to be used as column heads, each consisting of a single word of up to 10 characters. If more than one, header names should be enclosed in quotes.

NAMELIST INPUT:

Instruction parameters are given in a file containing one or both of the namelist records “nlmtrangp” and “nmlcycloc”. When the programme is being used only to interpolate, smooth, and/or reinterpolate data arrays, only “nlmtrangp” need be used; when cyclone finding is to be performed only from a PS array, only “nmlcyc” need be used. For the full programme of interpolating and finding, both may be used and in the order indicated. Note that time limit parameters (ddummn, ddhmmx, dastrt, hrstrt, dastop, hrstop) may be entered in either namelist (If entered in both, only the latter will be used).

```

namelist /nlmtrangp/quant,level,lunit,source,unit,dmode,
* ddhmmn,ddhmmx,dastrt,hrstrt,dastop,hrstop,ni,nj,
* xcen,ycen,hemis,rproj,rdiff0,rdiff,rdifz0,rdifz,
* intopt,noneg,expNS,spval

namelist /nmlcycloc/hilo,iopmxc,istmxc,area,cunit,latmnc,
* latmxc,lonmnc,lonmxc,imnc,imxc,jmnc,jmxc,nshell,mscrn,sdrmx,
* drmx1,drmx2,itmx1,itmx2,difft1,difft2,iconcv,cmnh,cmnc0,
* cmnc1,cmnc2,dpmn,swvmn,fcmn,fmxc,frmxc,frmxc,
* icendp,cvarad,rdincr,nrddir,sphtgr,
* rdpgrd, rdustr,npgdir,alatgv,rhoa,upfact,iavsup,rdsupv,
* zsmax,zscr1,zscr2,ftopeq,itabc,itabc2,itabc3,itabc4,
* ddhmmn,ddhmmx,dastrt,hrstrt,dastop,hrstop,
* cmnhw,cmncw,dpmnw,swvmnw,fmxcw,fcmnw

```

Time quantities appearing in both namelists:

These quantities determine the range of times at which cyclones are found. The meanings of the dates and times of day depend on “dmode”. This needs to be set in the interpolation phase, but not when data interpolated to PS is used, since this variable is carried in the PS data file format.

ddhmmn = 10000	Minimum time interval between successive data fields, in days, hours, minutes (ddhmm); e.g. 010000=1 day, 000600=6 hrs.
ddhmmx = 10000	Maximum time interval between successive data fields (0=inf, no check made)
dastrt = 0	Starting date (0=first data field used; otherwise the meaning depends on dmode. If dmode='DDDHM', it is the day no; if dmode='YMDHM', 910311= 11th Mar. 1991)
hrstrt = 0	Starting time
dastop = 0	Finishing date (0=last data field used)
hrstop = 0	Finishing time

Quantities appearing in 'nlmtrangp':

(a) Character variables

These quantities may need to be set in `nmltrngp` if not created or inferred from the data headers by the local lat.-lon. data reading routine. If set, they will override `llmaprd`.

```
(character quant*8,level*9,lunit*10,source*10,unit*12,dmode*6)
quant = ' '      Quantity of data array
level = ' '      Level of data array
lunit = ' '      Quantity of level of data array
source = ' '     Source or identification code of data
unit = ' '       Units for data, and (lat-lon) grid/region used
dmode = ' '      Mode for interpretation of dates and times, e.g.,
                 ymdhm year, month, day, hour, minute (yymmddhhmm)
                 ydd year, day (1-365) (yyddd)
                 dddd day (d), decimals of day (D) (ddddddDDDD)
```

(b) Projection information

When finding cyclones, the hemisphere of the projection is normally set by the predominant hemisphere specified by `latmnc` and `latmxc`. If the range extend more than 10 deg. into both hemispheres, PS arrays will be constructed for both. When the programme is only used to create PS array files, the hemisphere will need to be set by writing `hemis = 'N'` or `'S'`. If `hemis = 'G'`, arrays for both hemispheres will be found. By default, the position of the pole (`xcen,ycen`) is normally the central point of the array; however, it may be specified otherwise. The resolution of the grid is determined `rproj`, the number of intervals between the pole and the equator and the size of the array by `ni` and `nj`. The default setting is:

```
rproj = 30. ni = nj = 61 (and thus xcen = ycen = 31)
```

This makes the equator touch the extremities of the array ($i=1,ni$ and $j=1,nj$) at the centres of the sides of the array, i.e., at 0, 90, 180, and 270 deg.E., as does any other setting in which $ni=nj=2.*rproj+1$. This is adequate for searches of mid-latitude systems up to 80 deg. from the pole. If cyclones are to be sought to or beyond the Equator on either projection or globally, it is best to make allow an extra margin of $0.2*rproj$, i.e., about 10 deg or 5–10 grid points by increasing $ni=nj$ by 10–20 points.

```
ni = 61          Size of PS array in 90degE directions
nj = 61          Size of PS array in GM directions
xcen = 0.        Grid coordinates of pole of projection (not
ycen = 0.        necessarily integral values). (If not otherwise defined, these will be set
                 to the float(int(1+ni)/2) and float(int(1+nj)/2).)
rproj = 30.      No. of grid point spaces between pole and equator
```

hemis = ' ' Projection: S or s = PS SH, N or n = PS NH

(c) Smoothing radii

Smoothing of pressures and topography is applied by setting non-zero values of the smoothing radii, `rdiff` and `rdifz`, respectively. This is done by applying a number of passes of a finite impulse filter on the PS array using a diffusive coefficient scaled to give a uniform spatial smoothing. If data, e.g., pressure data, have already been smoothed by an amount `rdiff0`, the relevant radius is set to zero, otherwise further smoothing will be added, and the total smoothing will be:

$$\text{rdiff_total} = \sqrt{\text{rdiff0}^2 + \text{rdiff}^2}$$

`rdiff0` and `rdifz0` may be used as declarations of smoothing previously applied for the purpose of calculating the total amount of smoothing. These values will only stand if not overridden by smoothing values contained in the data files. There is provision for this information in PS files but not CIF (lat.-lon.) files.

The amount of smoothing needed will depend upon the type of system being sought and the characteristics and resolution of the data set. A radius of 2 degrees has been found suitable for mid-latitude lows from data sets covering a variety of resolutions from 1–2.5 deg. For anti-cyclones, the smoothing radius could be rather larger. It is not wise to make smoothing radii much larger than the finest part of the grid resolution, since it is this that determines the number of diffusive passes required for computational stability. For consistency in applying penalties using the Laplacian of the surface height, topography should receive exactly the same smoothing.

<code>rdiff0 = 0.</code>	Diffusive radius ($\sqrt{K \Delta t}$) previously used for smoothing input data
<code>rdiff = 0.</code>	Diffusive radius ($\sqrt{K \Delta t}$) for smoothing PS fields = $\sqrt{\text{rdiff0}^2 + \text{rdiff1}^2}$ <code>rdiff1</code> is applied diffusion (<code>rdiff</code> in input)
<code>rdifz0 = 0.</code>	Diffusive radii for topography
<code>rdifz = 0.</code>	

Quantities appearing in 'nmlcycloc':

(a) Highs or lows

`hilo = 'L'` Highs (H) or lows (L) required

(b) Cyclone status

The basic choice is to search only for minima in the data field, i.e., closed depressions. In mid-latitude cyclone studies, the positions of open depressions may also be of interest, and they will be found also by setting `iopmxc=1`. The type of cyclone is indicated by the units (i.e. second) digit of the status variable.

Criteria for cyclone selection are set by `cmnh`, `cmnc0`, etc. Of those selected, a proportion may be deemed ‘strong’ if they meet additional criteria, given by the values of `cmncw`, etc., which will need to be higher. Weak cyclones are indicated by the addition of 10, i.e., a 1 in the tens column of the status variable. The combinations are thus:

0	strong closed
1	weak closed
10	strong weak
11	weak weak

`istmxc` specifies the maximum value of this variable that may be allowed. Thus:

<code>iopmxc = 0</code>	<code>istmxc = 11</code>	allows: 0 , 10
<code>iopmxc = 1</code>	<code>istmxc = 10</code>	allows: 0, 1, 10
<code>iopmxc = 1</code>	<code>istmxc = 11</code>	allows: 0 , 1, 10, 11
<code>iopmxc = 1</code>	<code>istmxc = 1</code>	allows: 0, 1

For certain datasets, `iopmxc = 2` allows the programme to find open lows near strong vorticity centres where the pressure laplacian is negative, subject to certain conditions.

<code>iopmxc = 1</code>	Closed features only (0) or both open & closed (1)
<code>istmxc = 01</code>	Strong closed (0) strong closed and open (1) all strong and weak closed (10), etc.

(c) Area of search

Normally only latitude limits are used, with the sign convention that southern latitudes are negative. Common types of setting are

<code>latmnc = -90.</code> , <code>latmxc = -15.</code>	SH to 15S
<code>latmnc = -90.</code> , <code>latmxc = 5.</code>	SH to 5N
<code>latmnc = -90.</code> , <code>latmxc = 90.</code>	Global
<code>latmnc = 10.</code> , <code>latmxc = 90.</code>	NH from 10N

<code>area = ‘</code>	Geographical region covered by cyclone search (in cyclone record header line) (NOT USED)
<code>latmnc = -90.</code>	S limit of latitude
<code>latmxc = 90.</code>	N limit of latitude
<code>lonmnc = 0.</code>	W limit of longitude (lonmnc may be < 0.)
<code>lonmxc = 360.</code>	E limit of longitude (lonmxc may be < lonmnc.)
<code>imnc = 0</code>	Limits of search in i direction
<code>imxc = 0</code>	(no checks whether <code>imxc = imnc</code>)

jmnc = 0	Limits of search in j direction
jmxc = 0	(no checks whether jmxc = jmnc)

(d) Search control

As a preliminary screening, the value of a data function is compared with that at the surrounding data points. The function may be the pressure, the pressure gradient, or the pressure Laplacian, depending on the value of mscrn. For pressure or pressure gradient, the central value should be a minimum; for the Laplacian, it should be a maximum. A ring of normally 4 or 8 points will be used for the screening, depending on the values of nshell; however, templates for rings of up to 24 are provided.

nshell = 8	No. (4,8,12,20,24) of surrounding points for scanning
mscrn = 2	Screen with min (0), min grad (1), or max delsq (2)
sdrmx = 10.	Maximum total distance (sigma dr) of new position from starting position in deg.lat.
drmx1 = 0.7	Maximum distance of movement in one iteration, for closed and open depressions, respectively(grid units)
drmx2 = 0.3	
itm1 = 5	Maximum number of new grid squares to be traversed in searching for a closed or open depression, resp.
itm2 = 5	
dift1 = 0.5	Closest allowable spacing of systems (grid units)
dift2 = 6.0	Minimum separation of starting grid point and closed system position for allowing a parallel search for an open depression (grid units)
iconcv = 2	1=pressure surface must be concave at start of fmin iteration,2= for frmin iteration also.

(e) Strength criteria

Weak or shallow systems are rejected on a number of criteria related to strength. For quantities which over most of the domain are close to zero or a constant amount, a maximum value of the function (fmxc) is appropriate; these might include vorticity maxima or, in the tropics, pressure minima. For mid-latitude lows, whose central pressure tends to reflect that of its environment or latitude rather than its strength, the strength is associated with a minimum of the Laplacian of the pressure through parameters beginning with the letter 'c'. The search can be terminated at any of a number of points—by the Laplacian being less than cmnh at the starting grid point, less than cmnc1 at the depression centre, or less than cmnc1 or cmnc2 (for closed and open depressions) in the final strength calculation. This last may be different from the central value because (a) the strength may be calculated from the average value of the Laplacian over a radius of several degrees and/or the point at which it is determined (or about which it is averaged) may not be the centre itself but the position of the Laplacian maximum closest to the cyclone centre. A number of other criteria (swvmn, fmxc, frmxc, frcmxc) are not normally used now.

dpmn cannot be used at present since the depth calculation is disabled. The parameter fccmn is used in conjunction with iopmxc=2.

cmnh = 0.	Minimum value Laplacian at grid pt. max. (deg.lat.**2)
cmnc0 = 0.	Minimum central value of Laplacian (deg.lat.**2)
cmnc1 = 0.	Minimum area averaged Laplacian, closed depressions
cmnc2 = 0.	Minimum area averaged Laplacian, open depressions
swvmn = 0.	Minimum third derivative at inflection point
dpmn = 0.	Minimum depth (if calculated) in mb.
fmxc = -999.	Maximum function value at centre
frmxc = 0.	Maximum absolute gradient of function at centre of open depression (function units/deg.lat.) (if.ne.0.)
frmcx = 0.	Maximum ratio of gradient/laplacian(deg.lat.)(if.ne.0.)
fccmn = 0.	Minimum cross gradient 2nd deriv.

(f) Cyclone strength and position controls

The parameter icendp determines the position that is used (a) in the programme cvave, which calculates an average of the Laplacian over a radius cvarad degrees around it and (b) the position that is tabulated in the cyclone position file. An averaging radius of 2 degrees has generally been used.

icendp = 1	cyclone	cyclone
icendp = 2	Laplacian	cyclone
icendp = 3	Laplacian	cyclone (Laplacian for separation check)
icendp = 4	Laplacian	Laplacian
icendp = 5	Laplacian	Laplacian (No minimum search, all Laplacian maxima recorded)

rdincr, nrddir, and sphtrg relate to a calculation of cyclone depth and radius, which is not enabled in this implementation.

icendp = 1	Use (1) cyclone centre or (2) vorticity centre for cvave, radius, and depth calculations.
cvarad = 4.0	Radius around centre over which average the Laplacian of the function is found (deg.lat.)
rdincr = 0.5	Radial increment in deg.lat. for radius/depth
nrddir = 12	No. of radial directions for radius/depth
sphtrg = .false.	Calculate radii using (T) spherical trigonometry, (F) map scale of projection at cyclone centre

(g) Steering velocities

Steering velocities may be found from the MSL pressure field by calculating the geostrophic velocity from the pressure gradient averaged over a radius of rdpgrd, usually 5 deg. This makes

use of the fact that even though the wind is zero at the cyclone centre, the average around it is not. Scalings are made appropriate for geopotential heights when these are implied by the initial letter 'Z' in the data header. To allow for the diminishing correlation of geostrophic velocity with closeness to the Equator, the 1/f factor is modified at low latitudes so as to pass linearly through zero at the Equator, and the exact form of the function is determined by alatgv. The steering velocities may be arbitrarily scaled here, but it is normal to do this in the tracking programme.

rdustr = -999.	Radius (deg.lat.) for steering velocity or pressure gradient averaging circle (If unaltered by namelist this is set to 0. for use in steer2 rdpgrd for use in steer1.)
rdpgrd = 5.	Default averaging radius for PMSL derived steering
npgdir = 12	No. of radial directions for p gradient calculation
alatgv = 8.	Characteristic latitude for reducing geostrophic velocities near equator
rhoa = 1.2	Air density characteristic of level (Kg./cu.m.)
upfact = 1.	Factor for scaling steering flow from msl values

(h) Tabulation specifiers

The following determine the cyclone data columns that will be present in the tabulation. itabc is provided for backward compatibility, but is overridden by itabc3 and itabc4. Normally itabc1 and itabc2 are set to unity. itabc3=0 allows only position information; increasing it to 1 gives pressure and to 2 gives also the Laplacian. itabc3=3 is also supposed to give radius and depth, but these have been disabled for the present.

itabc = 2	1=calc min.p; 2=also cv; 3 = also cxc, dpc, rdc; 4=also upc, vpc; 5=also sc.
itabc1 = 1	1=include open/closed status
itabc2 = 1	1=lon. & lat. of centre; 2=also vort. centre
itabc3 = -1	1=calc min.p; 2=also cv; 3 = also cxc, dpc, rdc; 4=also zs; -1= use itabc
itabc4 = -1	1=also upc, vpc; -1= use itabc

(i) Supplementary variables

iavsupx(isup)=0	1=average supp. variables over radius rdsupv
rdsupv = 5.	averaging radius for supplementary variables

(j) Topographic screening

With the default values, no screening is carried out. Cyclones found at positions where the topographic height is greater than zsmax are rejected. When ftopeq is set, the calculated strength

(used also with the strenght criteria) is reduced by an amount proportional to the absolute value of the topographic Laplacian, i.e.,

$$\text{delsq p (reduced)} = \text{delsq p} - \text{ftopeq} * |\text{delsq sz}$$

The calculated steering velocities are tapered from their full values at zscr1 to zero at zscr2 to prevent unreliable velocities being found. For many MSLP datasets, zsmax = 1000 m, ftopeq = 0.005, zscr1 = 200 m, zscr2 = 1000 m are suitable values. Some model data may not need screening, nor will sigma level vorticities or 500 mb. geopotential heights.

zsmax = 1.e+10 Maximum topographic height for finding cyclones
 zscr1 = 1.e+10 / Lower and upper topographic heights for phasing
 zscr2 = 1.e+10 down cyclone velocities from PMSL data.
 ftopeq = 0. Error in delsq-f units per unit of delsq-zs caused
 by reduction from topographic height zs
 (0.01 suitable for f in mb., zs in m.)

(k) Strong cyclone status criteria

The Laplacian or other quantities need to be greater or less than the following quantities for the cyclone to be considered 'strong'. Normally only cmncw is used MSLP cyclones (0.2 mb./deg.**2 is suitable) of fmxcw for vorticity centres.

cmnhw = 0. Minimum value Laplacian at grid pt. max. for strong
 cmncw = 0. Minimum central or ave. of Laplacian | status
 swvmnw = 0. Minimum third derivative at inflection | (iop≥10)
 dpmnw = 0. Minimum depth (if calculated) in mb. |
 fmxcw = -999. Maximum function value at centre |
 fccmnw = 0. Minimum cross gradient 2nd deriv.

DATA FORMATS:

(a) Field input data

The programme is capable of using a number of types of data fields. The most basic are the fields of the variable in which maxima or minima are to be found. For meteorological applications this will normally be MSL pressure or geopotential height; occasionally it may be vorticity or the Laplacian of pressure. In addition there may be a single field of topographic height, which may be used for screening out PMSL or lower level lows where the topography is elevated. Steering velocities, when employed, may be inferred from the PMSL fields, from wind velocities analysed at a suitable level, or geostrophically from upper level geopotential heights. Additionally, values of selected supplementary variables may be supplied, to be interpolated to the positions of the lows or highs.

Pressure, steering, and supplementary data must be supplied in sequential order. Pressures may be supplied as a single file of concatenated records, as multiple files containing a single record, or

as multiple files each containing one or more sequential concatenated records. When specifying multiple files using wild-cards, the filenames should be chosen such that the periods that they cover follow sequentially. Steering and supplementary variable data, if used, can only be read as one file of sequentially concatenated records for each steering velocity component and/or supplementary variable.

Pressure and topography may be given as either lat.-lon. arrays or as PS arrays; steering and supplementary data, if supplied, are always read as lat.-lon. data. Each PS array only covers a single hemisphere (except for a little of the opposite hemisphere). This is adequate for finding mid-latitude systems in one hemisphere, but, for finding systems globally, the data on two hemispheres are required. Global data for a particular time, when written out or read in in PS form consist of a NH array followed by a SH array. The dimensions of the input data are not particular to the programme executable and can be any size up to the total allowed by the programmes data arrays; these are given in 'cyc5.h'.

Latitude-longitude data are written to and read from unformatted files in what has been called conmap input format (CIF) or conmap form. This is defined by the fortran statements:

read (iunit) nlat	No. of latitudes
read (iunit) (lat(j),j=1,nlat)	Latitudes
read (iunit) nlon	No. of longitudes
read (iunit) (lon(i),i=1,nlon)	Longitudes
read (iunit) head	80 character header
read (iunit) ((f(i,j),i=1,nlon),j=1,nlat)	Array data

The data header is interpreted in the following way:

quant = head(1:8)	Quantity (e.g. PMSL, Z, VOR)
level = head(11:14)	Height or sigma level (e.g. 500, or blank)
source = head(31:42)	Data source (e.g. ECMWF, or expt. no.)
da = head(43:50)	Date (e.g. 19970701, or model day number)
hr = head(52:55)	Time of day (e.g., 0000, 1200)
unit = head(60:72)	Units of quant (e.g. MB, M, S**-1)
grid = head(73:82)	Grid resolution (e.g. 3.0x5.0DEG)

e.g.:

PMSL	ECMWF	19940701	1200	MB	2.5x2.5DEG
Z500	ECMWF	19940701	1200	M	2.5x2.5DEG

Pressures data and headers are read by "llmaprd", which may be modified to according to the format of the data available to the user. Other CIF reads and writes occur in various parts of the code. Polar stereographic data are written using the following statements:

```
integer da,hr
character quant*8,level*9,lunit*10,source*10,grid*17,
* unit*12,dmode*6
```

```

character hemis*1
character*10 fileid
data fileid/'psdata'/

write (iunit) fileid,quant,level,lunit,source,dmode,unit,grid,
* ni,nj,hemis,xcen,ycen,rproj,da,hr,spval,rdiff
write (iunit) ((fj(i,j),i=1,ni),j=1,nj)

```

where

```

hemis = hemisphere of projection (N or S)
spval = special value indicator (default = 99999.9)
rdiff = radius of smoothing applied
ni,nj,xcen,ycen,rproj - as above

```

(b) Cyclone input data

The cyclone position file consists of a concatenated series of single analysis time records of variable length depending on the number of cyclones found. Each record contains a header line, a record of control parameters and quantity information in namelist form, a row of column headers, and one row of numerical data for each cyclone position. The header line should conform to the format given by statement 240 below. The number `nnmlc` is needed for reading the namelist record, `itabc*` are needed for interpreting the tabulation, and `da` and `hr` are needed for time management.

The Fortran statements are:

```

namelist /nmlcydat/quant,level,lunit,source,unit,cunit,area,
* dmode,rdiff,hilo,feat,iopmxc,istmxc,latmnc,latmxc,lonmnc,lonmxc,
* nshell,mscrn,sdrmx,drmx1,drmx2,itmx1,itmx2,dift1,dift2,iconcv,
* icendp,cvarad,cmnh,cmnc0,cmnc1,cmnc2,swvmn,dpmn,fcmn,
* fmx,frmxc,frcmxc,cmnhw,cmncw,dpmnw,swvmnw,rdincr,nrddir,sphtrg,
* zsmax,zscr1,zscr2,qsteer,rdustr,npgdir,alatgv,rhoa,upfact
c ...
write (iunit) ! blank line separating headers
write (iunit,'(a80)' chead ! header line
if (lnblnk(chead).eq.0) go to 400
c ...
write (chead,240,err=440,end=430) da,hr,nnmlc,nnmlcp4,
* itabc1,itabc2,itabc3,itabc4,itabc5,nk
240 format (' CENTRES: ',i8,x,i4,' (NNML=',i2,',',i2;',ITABC=',
* 2i1,i2,i1,i2,')',',i3,x,a)
c ...
write (iunit,'()',end=460)

```

```

write (iunit,'(a100)',end=470)      ! namelist record written to file and
* (lnmlc(inmlc),inmlc=1,nnmlc)    ! read/written as 80 character strings
c ...
write (iunit,'(a/)',end=510)      ! tabulation headers
* tabhead
do 500 k = 1,nk
write (iunit,fmt)
* k,                                ! sequential number
* iopc(k),                          ! cyclone status
* (xc(k),yc(k),                    ! (if itabc2 ≥ 1) cyclone position
* fc(k),                            ! if itabc3 ≥ 1 field variable
* cc(k),                            ! if itabc3 ≥ 2 averaged Laplacian
* dpc(k),                          ! if itabc3 ≥ 3 depth
* rdc(k),                          ! " " radius
* zsc(k),                          ! if itabc3 ≥ togographic height
* (upc(k),vpc(k),                  ! if itabc4 ≥ 1 steering velocity
* (xv(k),yv(k),                    ! if itabc2 ≥ 2 vorticity centre
* (sc(k),isup),isup=1,itabc5)      ! if itabc5 ≥ 1 supplementary vars.
500 continue

```

5 TRACK.MAN

USAGE:

trackx [-d idiagt] [-i infile] [-c cfile] [-a zfile] [-b afile] [-z zfile] [-w wfile] [-h] [-j zfile] [-U] [-N] [-O] [-o]

SUMMARY OF OPTIONS:

-i: ifile	instruction file (default intrack)
-c: cfile	cyclone data file (default cycdat)
-a: afile	zonal average prediction vel. file
-b: afile	grid point prediction vel. file
-z: zfile	z500 file for geostrophic velocities
-w: wfile	parameter weighting stats. output file
-h	continue tracks from thist?.1 files
-j: jfile	continue from concatenated history file
-d: idiagt	diagnostic output level
-U	unformatted files to be used
-N	unformatted files with namelist
-O	force overwriting of thist files
-w	not used
-F	not used
-o	not used

Further information about the options is given under 'OPTIONS' below.

FILES:

track.dir/{*F,*f,*h,Makefile}

DESCRIPTION:

Some basic usages are:

trackx -i ifile -c cdata -FO	Track cyclones from cdata and force overwrite of thist*, writing formatted output
trackx -i ifile -c cdata -a afile	Track cyclones from cdata using prediction velocities in afile writing unformatted output
cat thist[123].1 > trackdata	Concatenate component track history files

Tracking can be continued by addition to an existing track file or files. The tracking resumes from the last date/time in the existing track file. NOTE: Continuation options have not been checked since format changes were made.

trackx -j oldtrackdata -i ifile -c cdata Add tracks of cyclones from cdata to tracks in oldtrackdata

trackx -h -i ifile -c cdata Add tracks of cyclones from cdata to tracks in thist[123].1

OPTIONS:

- d idiagt This option, if used, uses an integer argument to specify the diagnostic output required
- idiagy >= 1: Tracking stages at each time noted
- idiagy >= 2: Summary tables for each stage
- idiagy >= 3: Further diagnostics of predict, match
- idiagy >= 4: Further diagnostics of match
- idiagy >= 5: Formatted track histories at each time step
- i ifile Namelist input file for parameters controlling cyclone tracking, default is 'intrack'. (See section 'NAMELIST INPUT' below.)
- c cfile Cyclone data file (default cycdat)
- a afile zonal average prediction velocity file
- b afile grid point prediction velocity file
- Climatological cyclone velocity prediction file. The option -a indicates that zonal average velocities are provided; -b, that grid point velocities are provided. See 'DATA FORMATS' below. If neither option be used, the programme will work without prediction velocities.
- z zfile z500 file for geostrophic velocities. (NO LONGER USED - the information is provided in the cyclone position file now.)
- h This option indicates that tracks are to be sequentially continued from the tracks contained in the set of files named thist[123].1
- j jfile This option indicates that tracks are to be sequentially continued from the tracks contained in jfile. Options -h and -j must not be used together.
- U The programme writes to unformatted track history files and reads from unformatted files if -h or -j be used.
- N The programme writes and reads unformatted files but with a namelist record incorporated in character strings.
- O By default, the programme will exit if existing thist* files be found. This option will force overwriting of thist files.

NAMELIST INPUT:

Instruction parameters are given in a file containing the namelist record "nlmtrack":

```

namelist /nmltrack/ ddhmmn,ddhmmx,dastrt,hrstrt,dastop,hrstop,
* da0,hr0,iopmxt,latmnt,latmxt,lonmnt,lonmxt,cmnt1,cmnt2,cmnt3,
* refdt,wsteer,fsteer,asteer,wmotn,wpten,dequiv,rprob,rpbell,
* qmxopn,qmxwek,qmxnew,nsort,irevmx,merget,qmerge,itabt3,itabt4,
* rsttks,pstrak,hemis,xcen,ycen,rproj,requiv

```

(a) Time quantities

These quantities determine the range of times at which tracking is done. The meanings of the dates and times of day depend on “dmode”, which is set in the cyclone file.

Certain aspects of programme control require a base time or epoch to be set so that decimal times (and time differences) in units of days can be calculated. These times may also be useful for track plotting. The epoch is defined by the date and time of day, da0 and hr0.

If the time difference between successive analyses exceed ddhmmx, the programme will either exit or continue if rsttks=.true. In the latter case, all existing tracks are finished off and new ones are begun.

ddhmmn = 010000	Minimum time interval between successive data fields, in days, hours, minutes (ddhmm); e.g. 010000=1 day, 000600=6 hrs.
ddhmmx = 020000	Maximum time interval between successive data fields (0=inf, no check made)
dastrt = 0	Starting date (0=first data field used; otherwise the meaning depends on dmode. If dmode='DDDHM', it is the day no; if dmode='YMDHM', 910311= 11th Mar. 1991)
hrstrt = 0	Starting time
dastop = 0	Finishing date (0=last data field used)
hrstop = 0	Finishing time
da0 = 000000	Epoch date (decimal dates)
hr0 = 0000	Epoch time
rsttks = .true.	Start new tracks after data gap(1)

(b) Screening of cyclone data

The screening of depressions on the basis of status, strength, or geographical position is normally performed in the cyclone finding programme; but further screening may be performed in the tracking programme if desired.

Cyclones are classified as open or closed and strong or weak, as in programme cycloc. The status variable for the possible combinations is 0 (strong closed), 1 (strong open), 10 (weak closed), 11 (weak open). The parameter iopmxt will limit the allowed value of this variable. Weak cyclones may also be culled by setting limits on the strength of open and/or closed depressions.

Only latitude limits are used for delimiting the area of tracking, with the sign convention that southern latitudes are negative; longitude limits are not enabled.

	iopmxt = 99	Open depressions allowed (=1), or not
	latmxt = 90.	Max. latitude
	latmnt = -90.	Min. latitude
c	lonmxt = 0.	Max. longitude
c	lonmnt = 360.	Min. longitude
	cmnt1 = 0.	Max. Laplacian for closed depressions (0=use cmnc1)
	cmnt2 = 0.	Max. Laplacian for open depressions (0=use cmnc2)

(c) Prediction velocities

The projected displacement of a cyclone is based on one or more of three velocity predictors: a continuation of the movement of the cyclone during the previous analysis interval (uM), the climatological cyclone velocity at the cyclone's latitude or latitude and longitude (uL), and a scaled steering velocity (uS). If more than one be used, the prediction velocity is a weighted combination of them, viz.,

$$u_{\text{pred}} = wtM * uM + wtL * uL + wtS * uS, \quad wtM + wtL + wtS = 1.$$

When the programme was first written, data for certain times was missing, and it was considered necessary to allow for non-uniform intervals between analysis times by reducing the weighting of climatology from the parameter value (wmotn) when the time difference (dt) exceeded the reference analysis interval (refdt). The principal behind this is that the influence of inertia is greatest at short time intervals, but decreases relative to climatology with increasing time interval. In practice this does not work very well because of noise in the analysed cyclone positions; however, the details are given below anyway. The effect of the motion weighting parameter is referred to reference time interval, refdet. The time dependence may be switched off by setting refdt=0; in this case the previous motion component of the projected displacement is just wmotn times a linear extrapolation in time of the previous velocity.

The displacement (dxnextM) and corresponding velocity (unextM=dxnextM/dtnext) predicted from previous movement (dxlast, ulast) are

$$\begin{aligned} dx_{\text{nextM}} &= wMdx * dx_{\text{last}}, \\ un_{\text{extM}} &= wMdx * ulast * (dt_{\text{last}} / dt_{\text{next}}), \\ &= wMu * ulast, \text{ where} \\ wMdx &= (1 - wmotn^{**} (dt_{\text{next}} / refdt)) / (wmotn^{**} (-dt_{\text{last}} / refdt) - 1.), \\ wMu &= wMdx * (dt_{\text{last}} / dt_{\text{next}}). \end{aligned}$$

For $dt_{\text{last}} = dt_{\text{next}} = n * refdt$, the memory of previous movement decreases exponentially with interval length (unless wmotn=1), and the climatological component increases correspondingly,

$$wMdx = wmotn^{**n} = wMu$$

For wmotn = 1, the projected movement is a linear extrapolation, and for wmotn ~ 1,

$$wMdx = wmotn^{**n} * dt_{\text{next}} / dt_{\text{last}}, \quad wMu = wmotn^{**n}$$

Removing the time dependence by setting $\text{refdt} = 0$, causes $w\text{Mu} = w\text{motn}$ for all time intervals, i.e.

$$\text{dxnextM} = (\text{dtnext}/\text{dtlast}) * \text{dxlast}.$$

No time dependence has been instituted for steering velocities (i.e., it is the same as for motion, with $\text{refdt} = 0$).

The quantity $w\text{Mu}$ only controls the relative weighting of previous movement and climatology; $w\text{steer}$ controls the weighting of steering versus the other two, so:

$$\begin{aligned} w\text{tS} &= w\text{steer} \\ w\text{tM} &= (1.-w\text{steer})*w\text{Mu} \\ w\text{tM} &= (1.-w\text{steer})*(1.-w\text{Mu}) \end{aligned}$$

With the introduction of steering velocities, the usefulness of climatological velocities has declined, and is normally not included. Climatological velocities will be used in place of movement in the case of newly formed systems (for which the latter is not available) if a prediction velocity file be given on the command line, even if:

$$w\text{motn} = 1.$$

The steering velocity is scaled by a factor $f\text{steer}$ from the raw value given in the cyclone data file, which represents the wind velocity at the steering level or a raw geostrophic velocity obtained geostrophically from a pressure or geopotential field. For predicting the velocities of mid-latitude MSLP lows from geostrophic velocities obtained from 5 deg. averaged PMSL gradients, a factor of 2.0 has been found suitable; for predicting them from actual or geostrophic 500 mb. winds, a factor of 0.5 or 0.6 is usually assumed.

The steering displacement is calculated in two parts. The last position is projected by a proportion $(1-\text{asteer})$ and the next is regressed by a proportion asteer . The projected and regressed displacements are thus:

$$\begin{aligned} \text{dxproj} &= w\text{steer}*f\text{steer}*(1.-\text{asteer})*u_flow * dt \\ \text{dxlast} &= w\text{steer}*f\text{steer}* \text{asteer} *u_flow * dt \end{aligned}$$

where u_flow is the calculated flow at the steering level and dt is the analysis time interval. The logical and default value to use is $\text{asteer}=0.5$, which is centred in time. Note that $w\text{steer}$ and $f\text{steer}$ are both needed since $w\text{steer}$ affects the weighting of the other components (by $1-w\text{steer}$) whereas $f\text{steer}$ does not. When no climatological velocities are given and a cyclone is new, one could argue that it would be logical to make the steering weighting 1; this does not improve the skill of the tracking (which may be indicative of the finite skill of steering information) and provision for augmenting the weighting has not been included.

Provision also exists for using projected pressures based on previous pressure tendency in the probability calculation,

$$\begin{aligned} w\text{Pdash} &= (1.-w\text{pten}**(\text{dtnext}/\text{refdt})) / (w\text{pten}**(-\text{dtlast}/\text{refdt})-1.), \\ dp &= \text{dtlast}/\text{dtnext} * w\text{Pdash}*(p(n) - p(n-1)). \end{aligned}$$

This has not been found to contribute skill to the tracking of MSLP lows and is not recommended; it may be of some use when tracking from vorticity fields.

refdt = 1.	Reference time interval for wmotn, wpten, dequiv, rprobb, rpbell, qmxopn, qmxwek, qmxnew
wsteer = 0.	Weighting factor for steering velocity
fsteer = 1.	Scaling factor for steering velocity
asteer = 0.5	Weighting (0. or 0.5) for steering velocity at time tc (weighting at tb is 1-asteer)
wmotn = 0.36	Weighting factor for movement relative to the total of movement and latitude applicable to an interval of refdt
wpten = 0.3	Weighting factor for p tendency (1 day)

(d) Probability calculation

Probabilities of identification are calculated between all pairs of lows at the last and next times that fall within a critical or pass radius of each other. Distances can either be calculated on the sphere or on a PS projection (if pstrak=.true.). The pass radius is maximum equal to rprobb (degrees) for strong closed depressions that are at least two intervals in age. The probability function (or actually the logarithm of the probability) is given by

$$q = \begin{aligned} &= qmx - r^{**2}/(rpbell + (1.-rpbell)*r^{**2}), \\ &= qmx - r^{**2} \text{ if } rpbell = 1, \end{aligned}$$

where r is the distance between the centres divided by rprobb. The 'distance' may include an extra dimension, a scaling of the difference between the predicted and actual pressures at the new time (dp) when dequiv is non-zero, in which case

$$r = (1/rprobb) * \text{sqrt}(\text{dist}^{**2} + (\text{dp}*\text{dequiv})^{**2}).$$

The function q is a maximum (qmx) when the lows are coincident (r=0) and zero when they are displaced at or greater than rprobb (r >= 1). The maximum is 1 for an old strong closed low but is reduced for a weak, open, or new low according to the following rules:

qmx = qage*qstat	
qage = 1	for a low more than one interval old
qmxnew	for a newly formed low
qstat = 1	for a strong closed low
qmxopn	for a strong open low
qmxwek	for a weak (open or closed) low

The parameter rpbell determines the shape of the function: rpbell = 1 gives an inverted parabolic shape, rpbell = xx gives a Cressman function shape, and lower values give a bell-shaped curve rapidly diminishing away from r=0.

Associations that have a log. probability greater than zero are passed to the matching routine for sorting.

dequiv = 0.35	Deg. lat. equiv to 1 mb. pressure diff.
rprobb = 12.5	Pass radius (deg.lat.) for prob. fun.
rpbell = 1.	c ² /R (where P = R ² /(R ² + r ²) in prob.f.
qmxopn = 0.80	Max. value of prob.fn.(q) open depress.

qmxwek	= 0.60	Max. value of prob.fn.(q) weak depress.
qmxnew	= 0.75	Max. value of prob.fn. for new cyclones
requiv	= -1.	(formerly reciprocal of dequiv - NOT USED)

(e) Matching calculation

Associations are arranged into groups such that each old and new cyclone only figures in the associations in a particular group. The matching calculation finds the possible combinations of associations such that any one cyclone shall occur in only one association within the combination. The combination with the greatest probability (i.e., with the maximum sum of log. probabilities of the associations in the combination) gives the matching for the group. Unmatched cyclones are deemed to have been born or to have decayed.

The matching calculation is performed in six nested do loops, one for each association added in a particular combination. The number of loops can be limited to nsort by making this number less than 6. This is not very helpful. Frequently more than six are required, and it is not really practicable to nest more loops because the computational effort increases exponentially with the number of associations in a group. To overcome this difficulty, the number of grouped associations can be reduced artificially by confirming the match of the highest probability association without reference to the summed functions of the other associations possible with it. After this sorting revision, the matching is attempted with the associations which do not involve either of the matched pair of cyclone positions. The process may be iterated up to a maximum of irevmx sorting revisions.

Provision for division and merging of cyclone tracks using the parameters merget and qmerge has not been written into the algorithm.

Matching of combinations occurs within

nsort	= 6	Max.no. of simultaneous associations
irevmx	= 9	Max. no. of artificial sorting revisns.
merget	= 0	Merging of tracks (not yet available (unused))
qmerge	= 0.	Min. prob. fn. for merging/dividing (unused)

(f) Tabulation specifiers

The tabulation columns normally contain the same numbers as in the cyclone file, except for the addition of decimal days, track end status, and association probabilities, and also except for the fact that the cyclone data are reordered into tracks. If desired, certain columns may be deleted and itabt3 and itabt4 have the same effect as itabc3 and itabc4 in cycloc.

itabt3	= 100	Tabuln. of cyclone variables in history file
	= 0	use (itabt3=itabc)
itabt4	= 100	Tabuln. of additional variables in history file

(g) Projection information

Many of the tracking calculations require a knowledge of angular separations on the sphere. This is now normally performed using spherical trigonometry, which is enabled by the logical variable pstrak. It is more accurate and only takes only slightly more computational effort

than alternative (when `pstrak=.false.`) of computing the distances on a PS projection; it also allows tracks to be constructed without any consideration of which hemisphere is to be used for projection.

The use of the PS projection dates from a time when all phases of the programme and all data files used positions in PS units. This PS calculation makes use of the conformal property and a map scale, viz.

$$\begin{aligned} \text{dist} &= \text{scallat} * \sqrt{\text{deltax}^{**2} + \text{deltay}^{**2}} \\ \text{scallat} &= \text{rproj} * \tan(\text{phidash}/2.) \\ &= 2.*57.2958/\text{rproj} * \text{R}^{**2}/(\text{R}^{**2} + \text{rproj}^{**2}) \\ \text{R} &= \sqrt{(\text{x-xcen})^{**2} + (\text{y-ycen})^{**2}} \end{aligned}$$

For this purpose, projection parameters `hemis`, `xcen`, `ycen`, and `rproj`. Since the cyclone data is read from and the tracks are written to files containing positions as latitudes and longitudes, the projection scale (`rproj`) is arbitrary and there is no array size (`ni,nj`). It is set by default to 30 to make for reasonable numbers in the diagnostic tabulations.

<code>pstrak</code>	= <code>.true.</code>	Calculate tracks on PS projection
<code>xcen</code>	= 31	Position on array of pole of projection. (This will normally be at the centre of the array, but may be at any grid point or intermediate location.)
<code>ycen</code>	= 31	
<code>rproj</code>	= 30.	No. of grid point spaces between pole and equator
<code>hemis</code>	= ‘	The hemisphere in which the projection is done will depend on the latitude range of the cyclone data unless specified ‘N’ or ‘S’.

DATA FORMATS:

(a) Cyclone input data

The cyclone position file consists of a concatenated series of single analysis time records of variable length depending on the number of cyclones found. Each record contains a header line, a record of control parameters and quantity information in namelist form, a row of column headers, and one row of numerical data for each cyclone position. The header line should conform to the format given by statement 240 below. The number `nnmlc` is needed for reading the namelist record, `itabc*` are needed for interpreting the tabulation, and `da` and `hr` are needed for time management.

The Fortran statements are:

```
namelist /nmlcydat/quant,level,lunit,source,unit,cunit,area,
* dmode,rdiff,hilo,feat,iopmxc,istmxc,latmnc,latmxc,lonmnc,lonmxc,
* nshell,mscrn,sdrmx,drmx1,drmx2,itmx1,itmx2,dift1,dift2,iconcv,
* icendp,cvarad,cmnh,cmnc0,cmnc1,cmnc2,swvmn,dpmn,fccmn,
* fmx,frmxc,frmxc,cmnhw,cmncw,dpmnw,swvmnw,rdincr,nrddir,sphtgr,
* zsmx,zscr1,zscr2,qsteer,rdustr,npgdir,alatgv,rhoa,upfact
```

c ...

```

400 continue                                ! can skip blank lines
read (iunit,'(a80)' chead                   ! header line
if (lnblnk(chead).eq.0) go to 400
c ...
read (chead,240,err=440,end=430) da,hr,nnmlc,nnmlcp4,
* itabc1,itabc2,itabc3,itabc4,itabc5,nk
240 format (' CENTRES: ',i8,x,i4,' (NNML=',i2,',',i2,',';ITABC=',
* 2i1,i2,i1,i2,')', ',i3,x,a)
c ...
read (iunit,'()',end=460)
read (iunit,'(a100)',end=470)              ! strings read/written to file and
* (lnmlc(inmlc),inmlc=1,nnmlc)           ! read as namelist record
c ...
read (iunit,'(//)',end=510)                ! first line has tabulation headers
do 500 k = 1,nk
read (iunit,fmt,err=520,end=510)
* k,                                     ! sequential number
* iopc(k),                               ! cyclone status
* (xc(k),yc(k),                          !(if itabc2 >= 1) cyclone position
* fc(k),                                  ! if itabc3 >= 1 field variable
* cc(k),                                  ! if itabc3 >= 2 averaged Laplacian
* dpc(k),                                 ! if itabc3 >= 3 depth
* rdc(k),                                 ! " " radius
* zsc(k),                                 ! if itabc3 >= 4 topographic height
* (upc(k),vpc(k),                         ! if itabc4 >= 1 steering velocity
* (xv(k),yv(k),                           ! if itabc2 >= 2 vorticity centre
* (sc(k,isup),isup=1,itabc5)              ! if itabc5 >= 1 supplementary vars.
500 continue

```

(b) Prediction velocity input data

Prediction velocities are read as zonal averages from the a formatted file, the argument of a -a option, or as u and v lat.-lon. arrays from an unformatted file the argument of a -b option. The velocities should be in m/s, although they are converted to deg./day in subroutine "predvl". If no such files be provided, the programme will function without and give full weighting to the motion component (i.e., make wmotn=1.).

The Fortran statement for zonal average velocities are:

```

read (iunit,'(a80)') card
read (iunit,*) nja
do 120 ja = 1,nja
read (iunit,*) lat(ja),u(ja),v(ja)
120 continue

```


Regional velocities are read from a single unformatted file containing two concatenated CIF (conmap input file) records, for the u (east) and v (north) components; the u component record is read by

```

read (iunit) nlat           ! No. of latitudes
read (iunit) (lat(j),j=1,nlat) ! Latitudes
read (iunit) nlon          ! No. of longitudes
read (iunit) (lon(i),i=1,nlon) ! Longitudes
read (iunit) head          ! 80 character header
read (iunit) ((u(i,j),i=1,nlon),j=1,nlat)
                               ! Array data

```

(c) Track output data

The track output is initially written to several files: thist1.[12], which contain the parameter and date information, and thist2.1, and thist3.[12], which contain the track data. The “.1” and “.2” versions of thist1 and thist3 are written at successive analysis times. thist2.1 contains only completed tracks, and thist3.[12] contain all the tracks from the first numbered track which is still growing to the most recent track. At the end of tracking, the most recent versions of thist1, thist2, and thist3 are concatenated into a single file.

The track file (and the partial files from which it is compiled) contains two types of record, a header block (taken from thist1.[1 or 2]J) and a large number (“lstrk2”) of variable length track records, taken from thist2.1 and thist3.[1 or 2], which are in the same format.

The track file may be formatted or unformatted. For most applications, a formatted file is more convenient, and it may be easily compressed. An unformatted file may have two possible forms of the parameter list, one a straight listing (option -U) and the other a namelist listing (option -N). When a namelist is used with an unformatted data file, the namelist is initially written to a scratch file; the individual lines of this file are stored as character strings in the data file.

(a) File header formats

The fortran statements for the file header are as follows:

Formatted files

```

write (iunit,'(a/)') ' TRACK HISTORY FILE'
write (iunit,nmltrdata)
write (iunit,'()')

```

The namelist is:

```

namelist /nmltrdata/
* quant,level,lunit,source,dmode,unit,cunit,area,
* hilo,feat,latmnc,latmxc,lonmnc,lonmxc,
* sdrmx,drmx1,drmx2,itmx1,itmx2,nshell,cvarad,
* fmx,frmxc,frmxc,da0,hr0,ddhmt,iopmxt,cmnt1,cmnt2,

```

```
* iadvtp, afile, refdt, wsteer, fsteer, asteer, wmotn, wpten,
* dequiv, rprob, rpbell, qmxopn, qmxwek, qmxnew,
* nsort, irevmx, merget, qmerge, itabt3, itabt4, rsttks,
* t1, da1, hr1, ic, tc, dac, hrc, nkc, ndt, lstrk1, lstrk2
```

In addition to the parameters, the namelist includes some start and finish information, viz.,

t1	Decimal time in days of track start after an arbitrary ‘epoch’, da0,hr0
da1,hr1	Date and time of start of tracking
ic	Current analysis time number
tc	Decimal time in days of current time after da0,hr0
dac,hrc	Current date and time
nkc	No. of cyclones at current time
ndt	Time interval (days) of last analysis interval
lstrk1	No. of last track before first uncompleted track (i.e., the last track of file thist2.1)
lstrk2	No. of last track in track file

Unformatted files

The track parameters can be written simply as a single straight record in the track file, using option -U, viz.,

```
fileid = 'thist1'
write (iunit) fileid,
* quant, level, lunit, source, dmode, unit, cunit, area,
* hilo, feat, latmnc, latmxc, lonmnc, lonmxc,
* sdrmx, drmx1, drmx2, itmx1, itmx2, nshell, cvarad,
* fmx, frmx, frcmxc, da0, hr0, ddhmt,
* iopmxt, cmnt1, cmnt2, iadvtp, afile, refdt, wsteer, fsteer,
* asteer, wmotn, wpten, dequiv, rprob, rpbell, qmxopn, qmxnew,
* nsort, irevmx, merget, qmerge, itabt3, itabt4, rsttks
write (iunit) t1, da1, hr1, ic, tc, dac, hrc, nkc, ndt,
* lstrk1, lstrk2
```

Alternatively the parameters can be written as the namelist nmltrdata, defined above, which is written to the track file as character strings lnmlt(inmlt). This has the advantage that extra variables may be added without the format needing to be changed; however, it may not be convenient to use it on some operating systems.

```
fileid = 'thnml'
```

```

write (iunit) fileid,nnmlt
do 80 inmlt = 1,nnmlt
write (iunit) lnmlt(inmlt) 80
continue

```

(b) Formats for individual track records

Formatted files

```

write (iunit,310) trk,statf,statl,ifst,ilst,nit,
* idafst,ihrfst,idalst,ihrlst,itabt3z,itabt4z 310
format (' Track ',i4,': stat = ',i1,i1,' ifst = ',i4,
* ', ilst = ',i4,' nit = ',i3,x,i8,i4,' - ',i8,i4,
* '. (itab=',2i2,')')
write (iunit,'(/a)') trtab(1:lntrtab)
write (iunit,fmt) ((vart(m,isw(ivar)),ivar=1,nvar),m=1,nit)
write (iunit,'()')

```

Unformatted files

```

fileid = 'thist2'
write (iunit) fileid,trk,statf,statl,ifst,ilst,nit,
* idafst,ihrfst,idalst,ihrlst,itabt3z,itabt4z
write (iunit) ((vart(m,isw(ivar)),ivar=1,nvar),m=1,nit)

```

The track record consists of a track header and the data for each individual time. The information in the track header is as follows:

trk	Sequential track number
statf	Status of system at the start of the track record
statk	Status of system at the end of the track record
ifst	Analysis time number for the start of the track
ilst	Analysis time number for the end of the track
nit	Number of analysis times that the track covers
idafst,ihrfst	Date and time of cyclogenesis
idalst,ihrlst	Date and time of cyclolysis
itabt3z	Tabulation specifier for cyclone characteristics
itabt4z	Tabulation specifier for steering velocity

The data is stored in an array `vart(m,isw(ivar))`. The first subscript (“m” or “it”) is the analysis period relative to cyclogenesis (m=1) and the second is a pointer to the array location for each variable required in the specified tabulation. The tabulation is ordered as

follows:

t	Decimal time after da0,hr0	
da	Date	
hr	Hour	
stat	Status	
k	Cyclone no. in cyclone file	
iop	Open/closed status	
q	Matching probability	
x	Longitude	
y	Latitude	
p	Central pressure	if itabt3 \geq 1
c	Delsq central pressure	if itabt3 \geq 2
dp	Cyclone depth	if itabt3 \geq 3
rd	Cyclone radius	if itabt3 \geq 3
up	Steering E'wds velocity	if itabt4 \geq 1
vp	Steering N'wds velocity	if itabt4 \geq 1

Depending on the extent of the tabulation specified by itabt3 and itabt4, only certain variables will be plotted. The switching variable, "isw" points from a sequential column number to the required variable.

6 TROUBLESHOOTING AND HINTS

GENERAL:

1. The cycloc code has the maximum size of the array built in as a parameter, so if you start using high resolution data you have to manually edit that to allow larger grids. At the moment the maximum size is 721 (lat) x 1440 (lon) and thus, it is able to digest the ERA5 data set with its original resolution as input.
2. Sometimes the cubic spline code in cyclocx (on high res data) can take forever because it tries to output way too much info and generates a multi-gigabyte output file. The reason for this is unclear, but if it happens try to pipe the output to `/dev/null`.
3. It is really important to set the variable and units correctly in the header for the file with the gridded data. If you accidentally leave it as MSLP/MB but are using a Z file then things can go very weird!
4. Sometimes the cyclone identification method finds laplacians that are way way too high. These end up being written to the cyclone file as "*****" which makes the tracking part crash, so you need to manually remove those. This issue has been noted when using the algorithm with CMIP GPH (especially over higher topography) or regional climate model output (not well constrained tracking boundaries).
5. The cyclone code uses 2-digit years, so if you're trying to track across years like 1999-2000 you have to be careful with the dates you give in the header file. [This problem has been solved, as the algorithm now uses 4-digit years.](#)

HOW TO DEFINE RPROJ:

The best way to think of the PS grid is that it is *fundamentally defined by rproj*, the number of grid point spaces between the pole and equator. There are some variations on this as mentioned in `cycloc.man` but for almost any 'sensible' application, one would choose `rproj` to reflect the native grid resolution and then the PS grid is defined by:

$$\begin{aligned} n_i &= 2 * rproj + 1 \\ n_j &= 2 * rproj + 1 \end{aligned}$$

The pole $(x_{cen}, y_{cen}) = ((n_i+1)/2, ((n_j+1)/2)$ and the equator touches the extremities of the grid. Furthermore, Ross Murray derived a relationship between `rpole` and the latitudinal native grid resolution (GR)(not documented but is used in the [1999 FROST paper](#):

$$GR = 114.6 / r_{pole} * \cos^{**2} (90. - \text{phi}) / 2)$$

where `phi` is latitude (in degrees) and `GR` is deg.lat. (same as deg. when just looking along meridians). So if you want to construct a PS grid with a resolution comparable to the native (input) resolution you can use this formula as a guide. It does vary with latitude. At the pole (`phi=90`) `GR= 114.6/pole`

and at the equator ($\phi=0$) $GR = 114.6/r_{pole} * 0.5$, i.e., half that at the pole. Hence to ensure that the PS resolution is at least as good as the native grid use:

$$r_{pole} = 114.6/GR$$

For a 0.5 x 0.5 degree native grid you can set $GR=0.5$ and therefore $r_{pole} = 114.6/0.5 = 229.2$, i.e., use an integral value larger than this. $r_{proj}=240$ detects really small systems, a larger value of r_{proj} is better, i.e., $r_{proj}=360$. The smaller value would be fine in general but the larger value helped to detect more of those little systems. Note that $r_{proj}=360$ gives a PS grid resolution of $114.6/360 = 0.32$ degrees at the pole, 0.16 at the equator (double the resolution).

If you make r_{proj} larger than the value suggested by the formula, then you are effectively interpolating to a finer grid than your input data. For 2.5 x 2.5 degree data we usually use $r_{proj}= 60$, i.e., a PS grid resolution of $114.6/60 = 1.91$ at the pole, 0.955 at the equator.

To summarise, the value of r_{proj} determines the desired resolution of the PS grid. If you choose r_{proj} as suggested by the formula then the PS grid will have the same (or comparable) latitudinal resolution as your input (native) grid at the pole and double the latitudinal resolution at the equator. However, especially if you are looking for smaller systems, a larger value (maybe 50% bigger) may help to detect these, but in general, the suggested value are adequate.

A README.MP

```
#####
#####
# Unimelb Cyclone Tracking
#####
#####
```

Cyclone tracking scheme developed at the University of Melbourne

```
#####
# Dependencies
#####
```

Below are lists of dependencies for core applications and utilities respectively.

```
#####
# Core applications
#####
```

gfortran

```
#####
# Utilities
#####
```

gfortran, gcc, netcdf, udunits

udunits:

udunits has dropped Fortran support since v1.12. The last compatible version (v1.12) has been provided as part of this package. To compile read_nc2cmp, ensure that:

netcdf.inc, udunits.inc, udunits.dat and libudunits.a

are all present in the directory scripts/utills.dir. They can be found in the folders:

source/dependencies/, source/dependencies/udunits/bin

libudunits.a may need to be recompiled for your specific platform.

The source can be found in the folder dependencies/udunits/source, along with compilation instructions at dependencies/udunits/README.md.

NOTE: read_nc2cmp is the only library which depends on netcdf and udunits.

```
#####
# Compilation
#####
```

Run source/make-all to build all applications and utilities.

Core applications and utilities can be build separately as follows:

```
#####
# Core applications
#####
```

To compile, navigate to scripts/src.dir and execute ?run-make? script.

Will unpack source from .zip files and use make to compile binaries in source/cycbin folder.

```
#####
# Utilities
#####
```

To compile, navigate to source/utls.dir and execute ?run-compile? script.

Will compile all binaries in source/ubin folder