Software for Unified Model Output
SUMO Version 1.5
User's Guide

otherwise known as

The SUMO-wrestler's handbook

Summary of changes

v1.0/1.1 May 1996 : Original version of document — Roger Brugge

v1.2 October 1996 : Moisture input, additional SG fields, UM-format input, PV-surface output — RB

v1.3 May 1998 : Diabatic heating, masked transient fluxes, column-integrated water vapour, output level changes, Fortran 90 — RB

v1.4 February 1999 : Streamfunction and velocity potential, code to make use of UM RECONA files, various Fortran changes — RB

v1.5 June 1999 : NetCDf version 3 included, code runs on SGI Fermat, improvement to i/o library routines— RB

THIS DOCUMENT IS IN DRAFT FORMAT ONLY - IT MAKES REFERENCE TO FEATURES OF THE PROGRAM WHICH HAVE YET TO ENABLED.

Contents

[1] Denotes at what version of SUMO the section was last changed.

1 Introduction

This purpose of this document is to describe SUMO — the (UGAMP-written) analysis Software for Unified Model Output — which was designed for use with output from the Met. Office Unified Model and the Met. Office Assimilation datasets of the UARS Project. The program was developed from the UMAP package, and therefore bears a striking similarity in style and use to both SUMO and SMAP.

The analysis program is 'front-ended' by a Fortran-based user-interface (the 'data pre-processor') to facilitate the input of data required to control the analysis program. This pre-processor performs a variety of checks on the data, and produces namelists for input into the analysis program. The preprocessor can be by-passed by the user; in this case a namelist input file will need to be created manually for the analysis program.

This document should be viewed as a continuously evolving description of the diagnostic program and its pre-processor. Changes made to either program should (if appropriate) be described herein. Note that the two programs themselves contain inline documentation designed to help the programmer to understand the workings of the individual routines.

At version 1.2 some argument-driven SUMO-code, together with i/o routines supplied by Jeff Cole, was placed into pre-compiled library files.

Version 1.3 makes use of Fortran 90 syntax, which has advantages insofar as apsects of portability are concerned. New variables introduced in the SG diagnostic block include column-integrated water vapour and diabatic heating, along with streamfunction and velocity potential (available through the bugfix file). Transient fluxes can also be computed in a more general manner in this version, using land-air or isentropic mass masks. Finally in this version the user is able to set up the vertical interpolation such that the number of input and output levels may be different in value.

Version 1.4 contains further Fortran 90 changes, the permanent inclusion of streamfunction and velocity potential calculation code, the ability to use Unified Model RECONA files in the calculation of vertical level information. This version was also ported to the CSAR/CFS Fujitsu, which was a replacement for the RAL Cray J90, and was also front-ended by the XSUMO windows interface software. Version 1.2 of the SUMOLIB library is required to run version 1.4 of SUMO.

Version 1.5 includes modifications to enable it to run the the CSAR/CFS SGI (Fermat), with improvements to the library i/o routines made by Jeff Cole. Version 1.3 of SUMOLIB is required by this version of Sumo, which also uses netCDF version 3 routines.

2 The data pre-processor

The data pre-processor (PRESUMO) is designed to facilitate the input of data into the diagnostic program. The pre-processor is written in Fortran and produces a sequence of namelist input statements for the diagnostic program. Given the large number of namelists that are read (in a specific order) by the diagnostic program (and the large number of diagnostics that are available within the diagnostic program), the pre-processor can be employed by the user (1) to reduce the amount of input he/she needs to specify (2) to minimise the number of variable names that need to be remembered, and (3) to eliminate the necessity of defining the input data in a pre-determined order (to match the namelist order).

2.1 Data pre-processor syntax

The input required by the pre-processor consists of a sequence of block structures — embedded within each are a series of one or more keyword phrases. The input to the pre-processor takes the following form:

```
BLOCK_NAME
  KEYWORD=integer_value
  KEYWORD=real_value
  KEYWORD=logical_value
  KEYWORD=character_value
  KEYWORD=value1,value2,.........,valuej,
          valuej+1,......valuen.
  KEYWORD=ALL
END_OF_BLOCK
```

Notes:
1. No blanks must follow the '=' sign, or appear after the first digit in a continuation line, unless the keyword value is of type character. Leading blanks can be used to make the input easier to read.
2. Except when specifying file pathnames, all keywords and their corresponding values must be specified in upper case only
3. Keywords cannot be abbreviated. Keywords can be specified in any order within a data block.
4. No blank lines are allowed.
5. Multiple data entries must be separated by commas. If more than one line is used to specify a sequence of multiple data entries (the values to be assigned to an array, for example) then the last character on any line that is followed by a continuation line must be a comma. This comma can be in any of the first 80 columns on the input line.
6. The value `ALL` can be used to select all possible values (for example, to select all types of diagnostic fields for a particular type of diagnostic).
7. Logical values can take one of the following forms: `T`, `.T.`, `TRUE`, `.TRUE.`, `F`, `.F.`, `FALSE` or `.FALSE.`.
8. Real values can be expressed in Fortran E or F format.
9. Character values should be specified without any surrounding quotes.
10. Each keyword in a particular data block corresponds to a specific Fortran namelist variable or array in the pre-processor and diagnostic program. Any keyword not assigned a value by the user takes a default value – this will in turn be passed to the corresponding namelist variable or array.
11. The COMMON (or global) data block (beginning with the line `COMMON` and ending with the line `ENDCOMMON`) must be the first block to appear in the user-written data file. The second block in the data file must be the MODELCONSTANTS data block (beginning with `MODELCONSTANTS` and ending with `ENDMODELCONSTANTS`) — both these first two blocks must appear.
12. Any diagnostic data blocks should follow the MODELCONSTANTS data block. The diagnostic data blocks can be specified in any order. Each begins with a line containing the phrase `$XX` and ends with a line containing `$ENDXX`, where `XX` is the standard two character code used throughout the diagnostic program and pre-processor to denote the diagnostic type; for example, `TF` represents the transient flux diagnostics.
13. A data block may be specified more than once. In the event of a particular keyword from a given data block appearing more than once, data from the earlier specifications may be overwritten.

The data blocks of the pre-processor can be visualised as being part of a tree-like structure, with switches contained in blocks near the top of the tree being used to enable/disable whole data blocks lower down the tree. Figure 2.1 shows the structure of this tree.

The following is a summary of the purpose of each data block:

- COMMON – contains keywords controlling the overall function of the diagnostic program.
- MODELCONSTANTS – contains keywords relating to the model/analysis that created the data used by the diagnostic program, and to the format of the data itself.
- UMFILENAME – contains keywords necessary to specify the output filenames from the Unified Model, if these are to be used.
- OR – controls the output of the model orography, and the production of orographic masks.
- LV – data block to enable the user to specify the levels to which fields will be interpolated, if input and output data levels differ.
- SG – diagnostics of the three-dimensional gridpoint fields, and of certain single level (eg surface) fields.
  - GH – gridpoint field diagnostics on horizontal surfaces, and also column totals.
  - GZ – zonal (north-south-type) output of gridpoint fields.
  - GM – meridional (west-east-type) output of gridpoint fields.
  - GP – profile diagnostics of the gridpoint fields.
  - G3 – allows the output of complete gridpoint fields to a separate data file, in a user-specified format (for use in subsequent analysis programs, for example).
- TF – transient flux diagnostics.
  - FH – transient flux diagnostics on horizontal surfaces.
  - FZ – zonal (north-south-type) output of transient fluxes.
  - FM – meridional (west-east-type) output of transient fluxes.
  - FP – profile diagnostics of the transient fluxes.
  - F3 – allows the output of complete transient flux arrays to a separate data file, in a user-specified format (for use in different analysis programs, for example).
- OD – optional dynamical field diagnostics.
  - OH – optional dynamical field diagnostics on horizontal surfaces.
  - OZ – zonal (north-south-type) output of the optional dynamical fields.
  - OM – meridional (west-east-type) output of the optional dynamical fields.
  - OP – profile diagnostics of the optional dynamical fields.
  - O3 – allows the output of complete optional dynamical field arrays to a separate data file, in a user-specified format (for use in different analysis programs, for example).
- ZF – zonally-averaged flux diagnostics.
- XD – zonal-mean cross-sections of (thermo)dynamical fields.

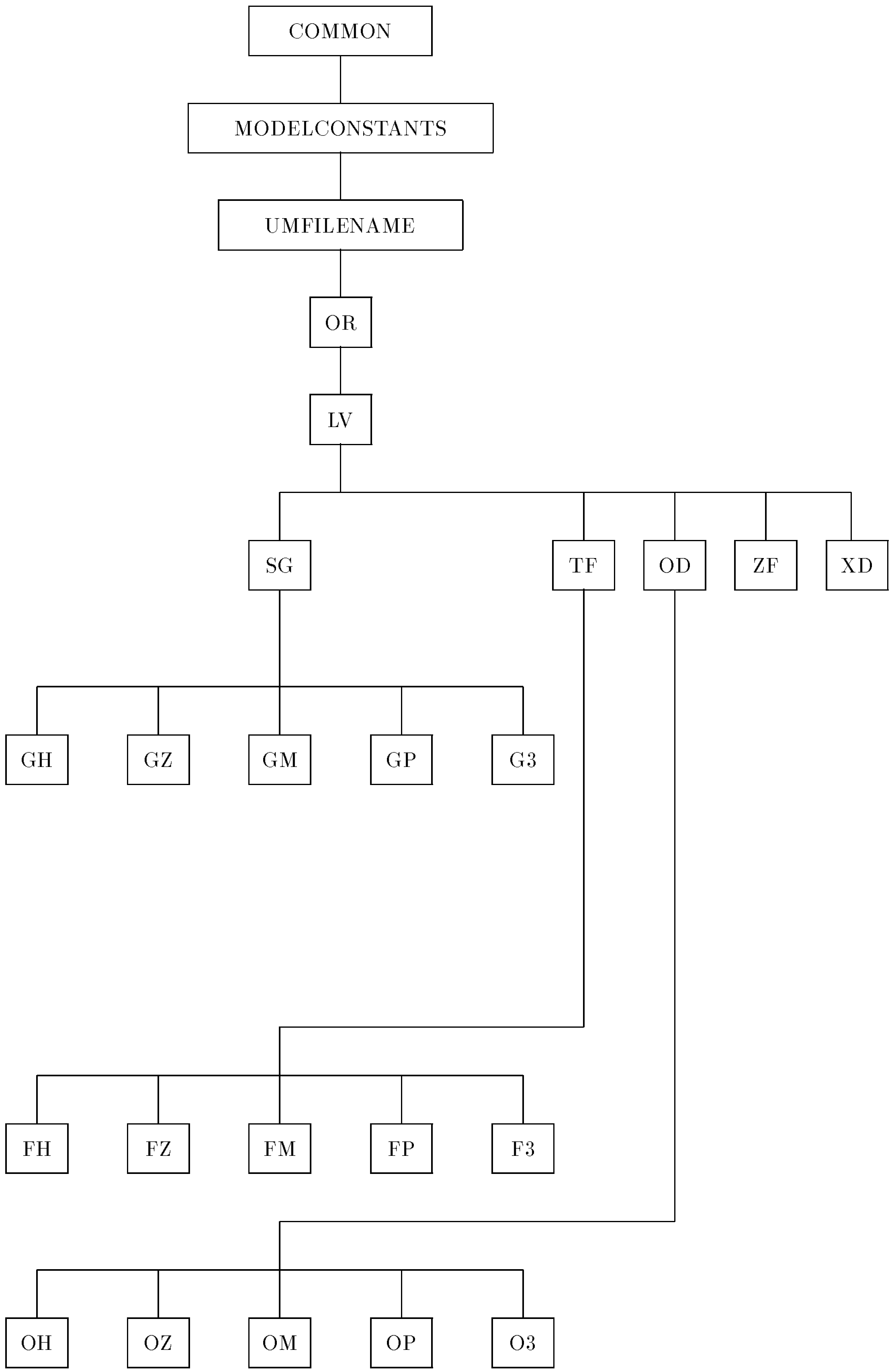

Figure 2.1: Flow chart showing the data blocks used by the pre-processor, and their hierarchy.

2.2 Keywords for the COMMON data block

The following are the keywords (and their associated default namelist values) that can appear in the COMMON data block (note that the COMMON data block must appear as the first data block in the input data file to the pre-processor):

- `ASCIICHARS` – integer value denoting the number of characters used to represent each array element value in the UTF. Namelist default is `NASCII=2`.
- `BEGCENTURY` – two digit integer specifying the century number of the start time of the diagnostic period. Default namelist value `BEGCEN=19`.
- `BEGDAY` – real value denoting the start time (format -*yymmdd* or *dd.dd*) of the diagnostic period. This value must be specified by the user for namelist variable `BEGDAY`.
- `BEGHOUR` – two digit integer specifying the hour number of the start time of the diagnostic period. Default namelist value `BEGHR=12`.
- `CURRENTDIR` – character value specifying the full pathname of the current directory. This value must be specified by the user for the namelist variable `YCPTHC`.
- `DIAGNOSTICS` – character array denoting the type(s) of diagnostics to be computed. Permissible values are `SG`, `TF`, `OD`, `ZF` and `XD`. Multiple diagnostics may be selected by separating values with commas. The array is used to define variables `LDSG`, `LDTF`, `LDXD` etc.
- `DRSOUTPUT` – logical value to switch on the output of DRS format data. The default namelist value `LDRS=.FALSE.`. Note that SUMO does not allow more than one of `LDRS`, `LNET` and `LASCII` to be set to `.TRUE.` in the same job. See also section 3.10.
- `ENDCENTURY` – two digit integer specifying the century number of the end time of the diagnostic period. Default namelist value `ENDCEN=19`.
- `ENDDAY` – real value denoting the end time (format -*yymmdd* or *dd.dd*) of the diagnostic period. This value must be specified by the user for namelist variable `ENDDAY`.
- `ENDHOUR` – two digit integer specifying the hour number of the end time of the diagnostic period. Default namelist value `ENDHR=12`.
- `FREQPRINT` – integer value indicating how often (in timesteps) the output fields will be printed. The default namelist value is `NFREQP=0`, which will suppress the instantaneous and time series output.
- `FREQPLOT` – integer value indicating how often (in timesteps) the output data is written to the UTF or DRS or NetCDF files. The default namelist value is `NFREQA=0`, which will suppress the instantaneous and time series output.
- `FREQSAMPLING` – integer value indicating how often (in timesteps) the input data is sampled during the running of the diagnostic program. This value must be specified by the user; namelist variable `NFREQD`. It does not apply to trajectory data files.
- `HOMEDIR` – character value denoting the full pathname of the home directory. This value must be specified by the user for namelist variable `YCPTHH`.
- `INITIALDIR` – character value denoting the full pathname of the directory containing the initial data. This value must be specified by the user for namelist variable `YCPTHI`.
- `JOBNAME` – character*8 array of two elements used to specify the name of the diagnostic job. The default is to specify no name for namelist variable `YONAME`.
- `JOBNUMBER` – real number to be used to specify the diagnostic job number. The default namelist value is `RNTAPE=0.0`.

- `LEVELTYPE` – character value denoting the type of surface on which the output will be produced. Permissible values are `ET` (model (η) surfaces), `TH` (isentropic surfaces), `IS` (isobaric surfaces) or `PV` (potential vorticity surfaces). Default namelist value `YTYPSF=ET`. NOTE THAT CURRENTLY A VALUE OF 'ET' IS NOT ALLOWED.

- `DATADIR` – character value specifying the full pathname of the history data directory. This value must be specified by the user for namelist variable `YCPTHM`.

- `MASKAVERAGE` – logical switch to enable land/air mask weighting when computing spatial and temporal averages. Currently only applies to output on isentropic surfaces. Default namelist value `LAVMSK=.FALSE.`.

- `MASSWEIGHT` – logical switch to enable gridpoint fields to be mass-weighted (ie multiplied by σ) when interpolated to isentropic surfaces. Default namelist value `LMASSW=.FALSE.`.

- `NETCDFOUTPUT` – logical value to switch on the output of NetCDF format data. The default namelist value `LNET=.FALSE.`. Note that SUMO does not allow more than one of `LDRS`, `LNET` and `LASCII` to be set to `.TRUE.` in the same job. See also section 3.11. NOTE: this has not yet been enabled.

- `OROGRAPHY` – logical switch to control the inclusion of orography in certain calculations. Default namelist value `LROGR=.TRUE.`.

- `PLOT` – logical value to switch on the creation of UTFs for use in later plotting programs. Default namelist value `LASCII=.TRUE.`. Note that SUMO does not allow more than one of `LDRS`, `LNET` and `LASCII` to be set to `.TRUE.` in the same job.

- `PRINT` – logical value used to switch on the printing of selected diagnostics. Default namelist value `LWRITE=.TRUE.`.

- `PRINTSIGFIG` – integer value denoting the number of significant digits to be used when printing output arrays. The namelist default value is `NSIGFG=2`, values in the range 2 to 5 are permissible.

- `RUNNINGDAYS` – real value denoting the time period (days) used in any running-mean calculations. Default namelist value `DYRUNM=1.0`.

- `SCRATCH` – logical value to switch on the use of scratch storage for certain large arrays within the diagnostic program. Default namelist value `LSCRAT=.FALSE.`.

- `SIGMAAVERAGE` – logical switch to enable σ-weighted averages when output uses isentropic surfaces. Default namelist value `LAVSIG=.FALSE.`.

- `TEMPDIR` – character value denoting the full pathname of the work directory. This value must be specified by the user for namelist variable `YCPTHT`.

- `UTF` – real value (currently 1.4) specifying the version of UTF to be used. Default namelist value `LUTF14=.TRUE.`.

2.2.1 Use of days/day numbers to control diagnostics

SUMO allow the user to control the start and end times of the diagnostic job using either day numbers or dates. This subsection illustrates this feature.

Internally, SUMO keeps track of time using decimal day numbers. These are converted to/from dates when required using the information specified in the MODELCONSTANTS data block.

Usually, the period of the diagnosis will be specified in terms of dates. In this case, input to the preprocessor will be of the following form

```
BEGDAY=-920428
ENDDAY=-920430
BEGCENTURY=19
ENDCENTURY=19
BEGHOUR=12
ENDHOUR=12
STARTCENTURY=19
STARTYEAR=92
STARTMONTH=04
STARTDAY=21
STARTHOUR=12
STARTMIN=00
CALENDAR=GR
```

Note that `BEGDAY` and `ENDDAY` are negative. The first six entries in the list define the diagnostic period, while the latter six may be used to indicate the start time of a model integration (ie. the time corresponding to day 0.0). (In the case of the diagnosis of analysed fields, it is usual to set the values of `STARTCENTURY`...`STARTHOUR` to correspond to the values of `BEGDAY`, `BEGCENTURY` and `BEGHOUR`). `CALENDAR` specifies whether the 30-day month or Gregorian calendar is being used.

SUMO determines that the start time (in century, year, month, day, hour format) is 1992042812, in the example above. By subtracting off the values of `STARTCENTURY`...`STARTMIN` (ie. 1992042112) the internal day-number counter is initialised to day 7.0. Note that the value of `STARTMIN` is currently used only to form the header of the UTF file (if used), along with the values of `STARTCENTURY`...`STARTHOUR`.

Alternatively, (and especially for output from model integrations) the values of `BEGDAY` and `ENDDAY` may be specified using (positive-valued) day numbers (as determined from the timestep counter and the time increment between model timesteps). In this case, the starting date of the analysis period is calculated by adding the value of `BEGDAY` to the date determined from the values of `STARTCENTURY`...`STARTHOUR` — similarly for the end time of the diagnosis.

2.3 Keywords for the MODELCONSTANTS data block

The following are the keywords (and their associated default namelist values) that can appear in the MODELCONSTANTS data block (note that the MODELCONSTANTS data block must appear in the input data file to the pre-processor, immediately after the COMMON data block):

- `CALENDAR` – specifies either the 30-day month calendar (`30`) or the Gregorian calendar (`GR`) – the latter is the default.
- `DATATYPE` – character variable denoting the type of input field data to be analysed. Permitted values are `UARS` (for UKMO assimilated data for the UARS project), `UARSRUNMEAN` (for running mean data generated originally by SUMO from `UARS`-type data), `UNIFIEDMODEL` (for data on pressure levels from the Unified Model), and `UNIFIEDMODELRUNMEAN` (for data generated originally by SUMO from `UNIFIEDMODEL`-type data).
- `MODELLEVELS` – integer denoting number of levels of input data to SUMO.
- `STARTCENTURY` – two digit integer value representing the starting century of the analysis data/model run, not that of the restart job. The default is 19.
- `STARTDAY` – two digit integer value representing the starting day number of the analysis data/model run, not that of the restart job. The default is 15.
- `STARTHOUR` – two digit integer value representing the starting hour number of the analysis data/model run, not the restart job. The default is 12.
- `STARTMIN` – two digit integer value representing the minutes part of the starting time of the analysis data/model run, not the restart job. The default is 00.
- `STARTMONTH` – integer value representing the starting month of the analysis data/model run, not that of the restart job. The default is 01.
- `STARTYEAR` – two digit integer value representing the starting year of the analysis data/model run, not that of the restart job. The default is 87.
- `STEPS/DAY` – real value specifying the number of timesteps per day. This must be specified by the user since the default namelist value is `TSPD=0.0`.

The user-supplied values for keywords `STARTYEAR`, `STARTMONTH`, `STARTDAY`, `STARTHOUR` and `STARTMIN` are combined together to form three namelist variables that appear in the UTF header, with default values, given by `NYEAR=8701`, `NDAY=1512` and `NMIN=0`.

2.3.1 Specification of Unified Model filenames

Beginning with version 1.2, SUMO will only accept output files from the Unified Model that are named according to the UM naming convention outlined below. Each file must contain data for a single timestep.

Filenames must be of the form `ccfmaa.pjj4110` where
`ccfm` is the experiment id (no default value),
`a` is job id (no default),
`a` signifies submodel producing the data (assumed '`a`' ('atmos') by default),
`.` is the time standard indicator (assumed '`.`' (standard absolute time) (by default),
`p` is the file type indicator (assumed '`p`' (instantaneous data) by default),
`j` indicates the filestream (assumed '`j`' by default),
`j4` denotes the year where 1800 is coded as `00`, 1810 as `10`, 1820 as `20`, 1890 as `90`, 1900 as `a0`, 1990 as `j0`, 1991 as `j1`, 1999 as `j9`, etc.,
`1` is the month with `a` meaning October, etc.,
`1` is the day with `a` meaning the 10th, etc.,
`0` is the hour with `a` meaning 10GMT, etc.

The corresponding keywords for the UMFILENAME data block are as follows:

- `EXPID` – 4-character experiment id (assumed not to change across restarts).
- `JOBID` – 1 character job id (assumed not to change across restarts).
- `SUBMODELID` – 1 character submodel id.
- `TIMESTANDARD` – 1 character time period indicator.
- `FILETYPE` – 1 character filetype indicator.
- `FILESTREAM` – 1 character filestream indicator.

The UM files are assumed to be direct access, starting with global header(s), followed by field headers(s) and finally the field data.

2.4 Keywords for diagnostic data blocks

Identical keywords are used for many of the diagnostic data blocks, reflecting the fact that each of the diagnostic types produce similar output (eg. latitude-longitude cross-sections, zonal-height slices, time series, etc.). The fact that identical keywords are used throughout the input file (these being converted into unique Fortran namelist elements in the pre-processor output file) makes it easier to prepare the data for the diagnostic program, than if the pre-processor did not exist. A description of all these keywords now follows — note that the default values may vary from diagnostic type to diagnostic type, and that not all keywords apply to every data block (or diagnostic type).

- `3DOUTPUT` – logical constant specifying whether output consisting of 3D field arrays will be produced.
- `CI` – contour interval of plotted fields (either in the order corresponding to the fields as ordered by the keyword `PLOTFIELD`, or in a predetermined order — see the individual diagnostic sections below for more details.)
- `COLPROFILE` – `.TRUE.` if vertical profile information to be produced.
- `COLTOTAL` – `.TRUE.` if vertically-summed column totals are to be computed.
- `CUTOFF` – real value denoting the the separation period between low- and high-pass components of the transient flux fields.
- `EDDY` – `.TRUE.` if fields are to be output in the form of eddy fields (ie. departures from a zonal mean).
- `EDDYFLUX` – `.TRUE.` if eddy fluxes are to be output.
- `FORMULA` – used to define formula for user-programmable diagnostic fields (eg. for use by transient flux routines).
- `HIGHPASS` – logical variable used to indicate the output of highpass transient fluxes.
- `HORIZMASK` – logical constant used to control the output of a lat-long orographic mask in the UTF.
- `HORIZONTAL` – logical constant specifying whether horizontal (ie. latitude-longitude) diagnostics will be produced.
- `HOVMOLLER` – 2 real values (degrees) representing the southern and northern latitude limits over which Hovmöller plots will be generated. (These are longitude-time sections, averaged over the specified latitude range.)
- `INSTANTANEOUS` – if `.TRUE.`, then instantaneous fields will be produced, at intervals given by the `FREQPRINT` (for printed output) and `FREQPLOT` (for UTF output) keyword values.
- `LATLONG` – if `.TRUE.`, latitude-longitude slices will be produced.
- `LEVELMEAN` – if `.TRUE.` then the average value of the field on levels will be computed. The nature of the levels (eg model/eta, isentropic, isobaric or potential vorticity) is determined by the `LEVELTYPE` keyword value.
- `LOWPASS` – logical variable used to indicate the output of lowpass transient fluxes.
- `MASK` – a series of characters, corresponding in order to the user-specified transient flux equations, defining the mask to be used in the calculation of equation transient flux formula. Permissible values are `B`, `S` and `O` – denoting β (the air-land mask), isentropic σ, and no-mask respectively. The default is `X` – the user must ovcerride the default.
- `MEANFLUX` – `.TRUE.` if mean (or total) fluxes are to be output.
- `MERIDIONAL` – logical constant specifying whether meridional diagnostics will be produced.
- `MERIDMEAN` – if `.TRUE.`, area-weighted meridional-mean averages will be produced.

- `NSAVERAGE` – two real values, specifying a pair of longitudes (in the range 0 to 360 degrees) between which the data will be averaged to produce an north-south slice. The first value is assumed to lie to the west of the second value. The zonal mean is a special case, when the longitude limits are 0 and 360 degrees. Values of -999.0 are used to indicate that the feature is not required.
- `NSMASK` – logical constant used to control the output of a north-south orographic mask in the UTF.
- `NSSLICE` – a real value, specifying a single longitude (in the range 0 to 360 degrees) at which an north-south slice of data will be extracted. A value of -999.0 is used to indicate that the feature is not required.
- `NUMBEROFCOORDS` – integer value denoting the number of (latitude, longitude) coordinate pairs specified for a particular purpose.
- `NUMBEROFFIELDS` – used to specify the number of different fields selected by the user for a particular type of diagnostic.
- `O/PCHANNEL` – integer value denoting the output channel to which the printed diagnostic output will be written.
- `O/PLEVELS` – Real values indicating the values of surfaces onto which the input level fields will be interpolated. These levels are in addition to certain default levels — see section 2.5.5.
- `ODLEVELS` – integer value, values or the value `ALL` defining eta levels for which latitude-longitude output of OD diagnostics occur.
- `ODSURFACES` – real values indicating non-eta surfaces for which latitude-longitude output of OD diagnostics occur.
- `PLOTFIELD` – integer value, values or the value `ALL` defining which fields are to be processed to produce UTF output for later plotting. These values can be specified in any order.
- `PLOTOROG` – logical value denoting whether the orography (initial geopotential) will be passed to the UTF.
- `PRINTFIELD` – integer value, values or the value `ALL` defining which fields are to be processed to produce printed output. These values can be specified in any order.
- `PRINTOROG` – if `.TRUE.` the orography (initial geopotential) will be printed out.
- `PROFILE` – if `.TRUE.` vertical profile diagnostics will be produced.
- `PROFILECOORDS` – a sequence of real (latitude, longitude) coordinates (in degrees), typically used for the production of vertical coordinates.
- `RUNNINGMEAN` – logical value to indicate if running-mean output required.
- `SGLEVELS` – integer value, values or the value `ALL` defining eta levels for which latitude-longitude output of gridpoint fields occurs.
- `SGSURFACES` – real values indicating non-eta surfaces for which latitude-longitude output of gridpoint fields occurs.
- `SLICE` – `.TRUE.` if latitude-longitude fields are to be produced.
- `TAVERAGED` – if `.TRUE.`, then time averages since the start of the diagnostic period will be computed. The sampling period is given by the `FREQSAMPLING` keyword value.
- `TFLEVELS` – integer value, values or the value `ALL` defining eta levels for which latitude-longitude output of transient fluxes occur.

- `TFSURFACES` – real values indicating non-eta surfaces for which latitude-longitude output of transient fluxes occur.
- `TIMESLICE` – `.TRUE.` if information in the form of time-slices to be output.
- `TOTALFLUX` – `.TRUE.` if total flux (ie than computed using instantaneous data) is to be output.
- `TSERIES` – if `.TRUE.`, then a times-series of instantaneous values will be produced, at intervals given by the `FREQPRINT` (for printed output) and `FREQPLOT` (for UTF output) keyword values.
- `WEAVERAGE` – two real values, specifying a pair of latitudes (in the range -90.0 to 90.0 degrees) between which the data will be averaged to produce an east-west slice. The southernmost latitude should be the first of the pair. The meridional mean is a special case, when the latitude limits are -90.0 and 90.0 degrees. Values of -999.0 are used to indicate that the feature is not required.
- `WEMASK` – logical constant used to control the output of a west-east orographic mask in the UTF.
- `WESLICE` – a real value, specifying a single latitude (in the range -90.0 to 90.0 degrees) at which an east-west slice of data will be extracted. A value of -999.0 is used to indicate that the feature is not required.
- `WINDVECTOR` – integer value, values or the value `ALL` indicating the code values of contour fields on which wind vectors will be superimposed.
- `ZFLEVELS` – integer value, values or the value `ALL` defining eta levels for which latitude-longitude output of zonal mean fluxes occurs.
- `ZFSURFACES` – real values indicating non-eta surfaces for which latitude-longitude output of zonal mean fluxes occurs.
- `ZONAL` – logical constant specifying whether zonal diagnostics will be produced.
- `ZONALMEAN` – if `.TRUE.`, then the zonal-mean averages will be produced.

2.5 Diagnostics

The following subsections outline the data input options for each of the diagnostic types. These elaborate on the information given in section 2.4. Table 2.0 illustrates the availability of output formats from SUMO. Option B may be selected with one of the options A, C or D, but options A, C and D must not be selected together.

Option	A (see * below)	B	C	D
	`PLOT=TRUE` `DRSOUTPUT=FALSE` `NETCDFOUTPUT=FALSE` `UTF=1.4`	`PRINT=TRUE` `NETCDFOUTPUT=FALSE`	`PLOT=FALSE` `DRSOUTPUT=TRUE` `NETCDFOUTPUT=FALSE`	`PLOT=FALSE` `DRSOUTPUT=FALSE` `NETCDFOUTPUT=TRUE`
	UTF format	printed output	DRS format	NetCDF format – NOT YET ENABLED
Diagnostics				
OR (currently disabled)	yes	yes	no	no
SG	yes (G3*)	yes (no time series if C used)‡	yes (G3, GH, GZ only)	no
TF	yes (F3*)	yes	yes (F3, FH, FZ only)	no
OD	yes (O3*)	yes (no time series if C used)	yes (O3, OH, OZ only)	no
ZF	yes	yes (no time series if C used)	yes	no
XD	yes	yes	yes	no

Table 2.0: Illustrating the availability of output data formats. Note that options A, C and D are not permitted together in the same job. * output in UTF format, except for G3, T3, F3 and O3 diagnostics which, by default, produce files containing fortran formatted records (UKMO PP-formatted records for G3 output. ‡ - in addition, running-mean output of selected SG-type fields is available.

IMPORTANT: When the `LDRS` switch is set to `TRUE`, SUMO can currently only be used to generate diagnostics of 3-dimensional fields, global latitude-longitude fields and zonal-mean fields, in DRS format. The same job may, however, be used to produce printed information of other spatial subsets of the output data (eg slices, meridional means). It should be noted that time-series output is, generally, turned off for NetCDF-formatted, DRS-formatted and printed output when this switch is set. See sections 3.10 and 3.11.

2.5.1 LV level selection

The LV data block enables the user to specify the values of the levels to which the input fields will be interpolated, for diagnostic jobs when the output is not on the input levels (see the `LEVELTYPE` keyword in the COMMON data block.

When the user doesn't provide a set of surface values for the interpolation algorithms, a default set of levels is selected by the user. These correspond (see sections 5.2.2 to 5.2.4) to profiles derived from the coordinate values of the input data. When the user provides a set of (up to `NL`) interpolation surfaces, these are merged with the default surfaces (by removing those default surfaces that lie closest to the user-specified surfaces to maintain `NL` surfaces for interpolation).

Note that this merging process can mean that, for example, zonal cross-section plots may vary slightly from job to job, unless the same interpolation surfaces have been specified. Also, latitude-longitude output on non-eta surfaces can only be produced for those surfaces specified by the user in the LV data block.

Meaning	Keyword name	Namelist variable	Default value
Surfaces to which vertical interpolation will be done, entered in units of millibars, degK or pvu (potential vorticity units). **Important: if the switch `*DEF JPOUT` has been selected, then the user MUST supply as many output levels as are specified by the fortran parameter `JPOUT`.**	`O/PLEVELS`	`OUTLEV`	`NL`*0.0

Table 2.1: Pre-processor keywords for the vertical level (LV) data block, together with the corresponding diagnostic program variable name, and the default variable value that is assigned in the case when the user does not supply a value. `NL` is the number of model levels, or the number of levels of input data.

Details of the LV data block are shown in Table 2.1. Note that the surface values can be specified in any order.

Beginning with version 1.3, SUMO allows the user to output data on `JPOUT` levels, this number of levels being different from the number of input levels `JPNL` – the switch `JPOUT` must be set to enable this. However, if the output is to be on model levels, then the parameter `JPOUT` must be equal in value to the parameter `JPNL`, and the switch `*DEF JPOUT` MUST NOT be set.

2.5.2 OR diagnostics

The OR diagnostics relate to the production of orographic masks, and to the output of the orography in the form of either a printed or UTF version of the orographic height. To enable the OR diagnostics, set the keyword `OROGRAPHY` to `.TRUE.` in the COMMON data block — this is the default. Details of the OR data block are shown in Table 2.2.

Meaning	Keyword name	Namelist variable	Default value
Orographic height included in the UTF output file	`PLOTOROG`	`LROGPL`	`.TRUE.`
Printed orographic height produced	`PRINTOROG`	`LROGPR`	`.TRUE.`
Contour interval for orography passed to the UTF (metres)	`CI`	`CINTOR`	200.0
Surface pressure (orographic) mask passed to UTF when output is on pressure surfaces, for use with latitude-longitude plots	`HORIZMASK`	`LMSKH`	`.TRUE.`
Surface pressure (orographic) mask passed to UTF when output is on pressure surfaces, for use with zonal mean cross-section plots	`NSMASK`	`LMSKNS`	`.TRUE.`
Surface pressure (orographic) mask passed to UTF when output is on pressure surfaces, for use with meridional mean cross-section plots	`WEMASK`	`LMSKWE`	`.TRUE.`

Table 2.2: Pre-processor keywords for the orography (OR) data block, together with the corresponding diagnostic program variable name, and the default variable value that is assigned in the case when the user does not supply a value.

Note that the orography is read from the initial gridpoint data file. The orographic masks are currently only produced if the diagnostic program output is defined to be on pressure levels. With this proviso, the orography and orographic masks are switched on by default if the `OROGRAPHY` keyword in the COMMON data block is assigned the value `.TRUE.` — this keyword is used to specify the presence of orography beneath the lower boundary of the model, and its inclusion in certain diagnostic calculations.

2.5.3 SG diagnostics

The (SG) diagnostics refer to the three-dimensional gridpoint field diagnostics (along with a few surface fields). These are calculated on the input data surfaces. If the diagnostic output is not on the same surfaces (see, for example, the keyword `LEVELTYPE` in the COMMON data block, then the fields are interpolated to the output surfaces, using surface data values provided via the LV data block.

The SG diagnostics are switched on by setting `SG` as one of the keyword values for the `DIAGNOSTICS` keyword in the COMMON data block. The user must then set the appropriate keywords in the SG data block (see Table 2.3) and in at least one of the GH, GZ, GM, GP or G3 data blocks – the daughter SG data blocks.

Meaning	Keyword name	Namelist variable	Default value
Switch to allow horizontal (ie. latitude-longitude) SG diagnostics. Setting this switch to `.TRUE.` enables information provided in the GH data block to be used by the program.	`HORIZONTAL`	`LDGH`	`.FALSE.`
Switch to allow zonal SG diagnostics. Setting this switch to `.TRUE.` enables information provided in the GZ data block to be used by the program.	`ZONAL`	`LDGZ`	`.TRUE.`
Switch to allow meridional SG diagnostics. Setting this switch to `.TRUE.` enables information provided in the GM data block to be used by the program.	`MERIDIONAL`	`LDGM`	`.FALSE.`
Switch to allow profile SG diagnostics. Setting this switch to `.TRUE.` enables information provided in the GP data block to be used by the program.	`PROFILE`	`LDGP`	`.FALSE.`
Switch to allow output of three-dimensional SG fields. Setting this switch to `.TRUE.` enables information provided in the G3 data block to be used by the program.	`3DOUTPUT`	`LDG3`	`.FALSE.`
Switch to allow time-averaging of SG fields.	`TAVERAGED`	`LDSGTD`	`.TRUE.`
Switch to allow time series of SG fields. (Not allowed when DRS or NetCDF output has been selected.)	`TSERIES`	`LDSGTS`	`.FALSE.`
Switch to allow output of instantaneous SG fields.	`INSTANTANEOUS`	`LDSGIN`	`.FALSE.`
Switch to allow output of running-mean fields.	`RUNNINGMEAN`	`LDSGRM`	`.FALSE.`
Fortran channel for printed output	`O/PCHANNEL`	`NCSG`	3
Integers denoting levels on which latitude-longitude slices will be produced (if output is on eta surfaces)	`SGLEVELS`	`NVSGHR`	NL*0
Real values denoting levels at which non-eta surface latitude-longitude slices will be output	`SGSURFACES`	`VSGSFC`	NL*0

Table 2.3: Pre-processor keywords for the gridpoint field (SG) data block, together with the corresponding diagnostic program variable name, and the default variable value that is assigned in the case when the user does not supply a value. `NL` is the number of model levels, or the number of levels of input data.

Note that the time-control switches in the SG data block affect all the five daughter SG data blocks.

2.5.3.1 SG(GH) diagnostics

The GH subset of the SG diagnostics controls the production of diagnostics in the latitude-longitude plane, including horizontal slices at specified levels, Hovmöller plots and vertically integrated (mass-weighted) column totals. The fields analysed by this subset can be specified independently of the fields used by the other SG diagnostic subsets. Table 2.4 illustrates the keywords and their associated namelist variables in the GH data block.

Note that to enable these diagnostics the `DIAGNOSTICS` switch in the COMMON data block must have been set to include the `SG` option, and in the SG data block the `HORIZONTAL` switch must have been set to `.TRUE.`. Other switches in the SG data block must be set to enable the type of time-control (eg time averages, time series etc) required.

The field codes appropriate to the SG(GH) diagnostics can be found in Table 6.1.

Meaning	Keyword name	Namelist variable	Default value
Contour interval to be used for each field type when plotting the UTFs	`CI`	`CINTGH`	`JPSGMX`*-0.1
Field codes for the fields whose diagnostics are to be printed	`PRINTFIELD`	`NGHPR`	`JPSGMX`*0
Field codes for the fields whose diagnostics are to be plotted	`PLOTFIELD`	`NGHPL`	`JPSGMX`*0
Number of different fields to be processed	`NUMBEROFFIELDS`	`NGH`	0
Codes of fields for which windvector information will be sent to the UTF	`WINDVECTOR`	`NGHVEC`	`JPSGMX`*0
Switch to enable the output of data on latitude-longitude slices — these are the slice specified by `NVSGHR` or `VSGSFC` (see Table 2.3).	`SLICE`	`LDGHSL`	`.FALSE.`
Switch to enable the output of mass-weighted column totals. (This switch is only valid when output is on eta levels.)	`COLTOTAL`	`LDGHCT`	`.FALSE.`
Switch to output horizontal slice information in the form of eddy values (ie. with the zonal mean subtracted off)	`EDDY`	`LDGHED`	`.FALSE.`
Real values denoting the latitude range between which data on latitude-longitude slices (see `SLICE` above) will be averaged to produce Hovmöller output. `SLICE` must be set to `.TRUE.`. (Not allowed when DRS or NetCDF output has been selected.)	`HOVMOLLER`	`VGHTS`	2*−999.0

Table 2.4: Pre-processor keywords for the gridpoint field (GH) data block, together with the corresponding diagnostic program variable name and the default variable value, that is assigned in the case when the user does not supply a value. `JPSGMX` is the maximum number of fields that can be processed by the SG diagnostics.

2.5.3.2 SG(GZ) diagnostics

The GZ subset of the SG diagnostics controls the production of diagnostics in the zonal plane, including zonal slices at specified longitudes, zonal means and averages of fields in a specified longitude range. The fields analysed by this subset can be specified independently of the fields used by the other SG diagnostic subsets. Table 2.5 illustrates the keywords and their associated namelist variables in the GZ data block.

Note that to enable these diagnostics the `DIAGNOSTICS` switch in the COMMON data block must have been set to include the `SG` option, and in the SG data block the `ZONAL` switch must have been set to `.TRUE.`. Other switches in the SG data block must be set to enable the type of time-control (eg time averages, time series etc) required.

The field codes appropriate to the SG(GZ) diagnostics can be found in Table 6.1.

Meaning	Keyword name	Namelist variable	Default value
Contour interval to be used for each field type when plotting the UTFs	`CI`	`CINTGZ`	`JPSGMX`*-0.1
Field codes for the fields whose diagnostics are to be printed	`PRINTFIELD`	`NGZPR`	`JPSGMX`*0
Field codes for the fields whose diagnostics are to be plotted	`PLOTFIELD`	`NGZPL`	`JPSGMX`*0
Number of different fields to be processed	`NUMBEROFFIELDS`	`NGZ`	0
Codes of fields for which windvector information will be sent to the UTF — THIS IS CURRENTLY DISABLED	`WINDVECTOR`	`NGZVEC`	`JPSGMX`*0
Switch to enable the output of zonal mean data	`ZONALMEAN`	`LDGZZM`	`.FALSE.`
Switch to enable the extraction of data along a specified longitude	`NSSLICE`	`VGZNS`	-999.0
Two longitudes between which averages (as a function of latitude) will be computed	`NSAVERAGE`	`VGZNSA`	2*-999.0
Switch to enable the output of zonal-mean time series at the levels specified by `NVSGHR` or `VSGSFC` (see Table 2.3). `ZONALMEAN` must be set to `.TRUE.`. (Not allowed when DRS or NetCDF output has been selected.)	`TIMESLICE`	`LDGZTS`	`.FALSE.`

Table 2.5: Pre-processor keywords for the gridpoint field (GZ) data block, together with the corresponding diagnostic program variable name, and the default variable value that is assigned in the case when the user does not supply a value. `JPSGMX` is the maximum number of fields that can be processed by the SG diagnostics.

2.5.7.3 SG(GM) diagnostics

The GM subset of the SG diagnostics controls the production of diagnostics in the meridional plane, including meridional slices at specified latitudes, meridional means and averages of fields in a specified latitude range. The fields analysed by this subset can be specified independently of the fields used by the other SG diagnostic subsets. Table 2.6 illustrates the keywords and their associated namelist variables in the GM data block.

Note that to enable these diagnostics the `DIAGNOSTICS` switch in the COMMON data block must have been set to include the `SG` option, and in the SG data block the `MERIDIONAL` switch must have been set to `.TRUE.`. Other switches in the SG data block must be set to enable the type of time-control (eg time averages, time series etc) required.

The field codes appropriate to the SG(GM) diagnostics can be found in Table 6.1.

2.5.3.4 SG(GP) diagnostics

The GP subset of the SG diagnostics controls the production of column profile diagnostics at a user-specified series of locations. These diagnostics consist of profile values and column-integrated, mass-weighted totals — the latter only when output is on eta surfaces. The fields analysed by this subset can be specified independently of the fields used by the other SG diagnostic subsets. Table 2.7 illustrates the keywords and

Meaning	Keyword name	Namelist variable	Default value
Contour interval to be used for each field type when plotting the UTFs	CI	CINTGM	JPSGMX*-0.1
Field codes for the fields whose diagnostics are to be printed	PRINTFIELD	NGMPR	JPSGMX*0
Field codes for the fields whose diagnostics are to be plotted	PLOTFIELD	NGMPL	JPSGMX*0
Number of different fields to be processed	NUMBEROFFIELDS	NGM	0
Codes of fields for which windvector information will be sent to the UTF — THIS IS CURRENTLY DISABLED	WINDVECTOR	NGMVEC	JPSGMX*0
Switch to enable the output of meridional mean data	MERIDMEAN	LDGMMM	.FALSE.
Switch to enable the extraction of data along a specified latitude	WESLICE	VGMEW	-999.0
Two latitudes between which averages (as a function of longitude) will be computed	WEAVERAGE	VGMEWA	2*-999.0
Switch to enable the output of meridional-mean time series at the levels specified by NVSGHR or VSGSFC (see Table 2.3). MERIDMEAN must be .TRUE.. (Not allowed when DRS or NetCDF output has been selected.)	TIMESLICE	LDGMTS	.FALSE.

Table 2.6: Pre-processor keywords for the gridpoint field (GM) data block, together with the corresponding diagnostic program variable name, and the default variable value that is assigned in the case when the user does not supply a value. JPSGMX is the maximum number of fields that can be processed by the SG diagnostics.

their associated namelist variables in the GP data block.

Note that to enable these diagnostics the `DIAGNOSTICS` switch in the COMMON data block must have been set to include the `SG` option, and in the SG data block the `PROFILE` switch must have been set to `.TRUE.`. Other switches in the SG data block must be set to enable the type of time-control (eg time averages, time series etc) required.

The field codes appropriate to the SG(GP) diagnostics can be found in Table 6.1.

Meaning	Keyword name	Namelist variable	Default value
Contour interval to be used for each field type when plotting the UTFs	`CI`	`CINTGP`	`JPSGMX`*-0.1
Field codes for the fields whose diagnostics are to be printed	`PRINTFIELD`	`NGPPR`	`JPSGMX`*0
Field codes for the fields whose diagnostics are to be plotted	`PLOTFIELD`	`NGPPL`	`JPSGMX`*0
Number of different fields to be processed	`NUMBEROFFIELDS`	`NGP`	0
Number of different coordinates at which profile calculations will be performed; must not be greater than `JPGPPF`	`NUMBEROFCOORDS`	`NGPPF`	0
Pairs of (latitude, longitude) profile coordinates	`PROFILECOORDS`	`VGPPF`	`2*JPGPPF`*-999.0
Switch to enable the output of vertical profile values	`COLPROFILE`	`LDGPCP`	`.FALSE.`
Switch to enable the output of column-integrated totals — valid only when output is on eta levels	`COLTOTAL`	`LDGPCT`	`.FALSE.`

Table 2.7: Pre-processor keywords for the gridpoint field (GP) data block, together with the corresponding diagnostic program variable name, and the default variable value that is assigned in the case when the user does not supply a value. `JPSGMX` is the maximum number of fields that can be processed by the SG diagnostics, while `JPGPPF` is the maximum number of profiles that the program can handle in a single job.

2.5.3.5 SG(G3) diagnostics

The G3 subset of the SG diagnostics allows the user to output three-dimensional (two-dimensional in the case of surface fields) SG-type fields (either instantaneous or time averaged) to a file. This facility is useful if these fields are required as input to another program, or to a plotting package that does not accept UTFs as input. The user is able to re-program the appropriate routine in the diagnostic program to allow this output to be any required format, or to output a subregion of the 3D fields. The output file, by default, will contain data either in UKMO-PP formatted record format, or DRS format (see Table 2.0). However, the former may be changed to a Fortran formatted record format but substituting the output routine with another (`OPSG3D`) held in the program library. Table 2.8 illustrates the keywords and their associated namelist variables in the G3 data block.

Note that to enable these diagnostics the `DIAGNOSTICS` switch in the COMMON data block must have been set to include the `SG` option, and in the SG data block the `3DOUTPUT` switch must have been set to `.TRUE.`. Other switches in the SG data block must be set to enable the type of time-control (eg time averages or instantaneous) required.

The field codes appropriate to the SG(G3) diagnostics can be found in Table 6.1.

Meaning	Keyword name	Namelist variable	Default value
Field codes for the fields to be output	PLOTFIELD	NG3PL	JPSGMX*0
Number of different fields to be processed	NUMBEROFFIELDS	NG3	0
Fortran channel for output	O/PCHANNEL	NCG3	1

Table 2.8: Pre-processor keywords for the gridpoint field (G3) data block, together with the corresponding diagnostic program variable name, and the default variable value that is assigned in the case when the user does not supply a value. `JPSGMX` is the maximum number of fields that can be processed by the SG diagnostics.

2.5.4 TF diagnostics

The TF diagnostics refer to the three-dimensional transient flux diagnostics, which may be computed using a user-programmable formula involving the three-dimensional (thermo)dynamical fields. These are calculated on the output surfaces, the (thermo)dynamical fields having been first vertically-interpolated if required; in this case the output surfaces are setup using surface data values provided via the LV data block.

The TF diagnostics are switched on by setting `TF` as one of the keyword values for the `DIAGNOSTICS` keyword in the COMMON data block. The user must then set the appropriate keywords in the TF data block (see Table 2.9) and in at least one of the FH, FZ, FM, FP or F3 data blocks – the daughter TF data blocks.

2.5.4.1 TF(FH) diagnostics

The FH subset of the TF diagnostics controls the production of diagnostics in the latitude-longitude plane, currently horizontal slices at specified levels. The transient fluxes analysed by this subset can be specified independently of the fields used by the other TF diagnostic subsets. Table 2.10 illustrates the keywords and their associated namelist variables in the FH data block.

Note that to enable these diagnostics the `DIAGNOSTICS` switch in the COMMON data block must have been set to include the `TF` option, and in the TF data block the `HORIZONTAL` switch must have been set to `.TRUE.`.

The field codes (in the range 1 to `JPTFMX`, subject to a maximum of `NTF`) used in the FH data block refer to the transient fluxes specified in the TF data block.

2.5.4.2 TF(FZ) diagnostics

The FZ subset of the TF diagnostics controls the production of diagnostics in the zonal plane, including zonal slices at specified longitudes, zonal means and averages of fields in a specified longitude range. The transient fluxes analysed by this subset can be specified independently of the fields used by the other TF diagnostic subsets. Table 2.11 illustrates the keywords and their associated namelist variables in the FZ data block.

Note that to enable these diagnostics the `DIAGNOSTICS` switch in the COMMON data block must have been set to include the `TF` option, and in the TF data block the `ZONAL` switch must have been set to `.TRUE.`.

The field codes (in the range 1 to `JPTFMX`, subject to a maximum of `NTF`) used in the FZ data block refer to the transient fluxes specified in the TF data block.

2.5.9.2 TF(FM) diagnostics

The FM subset of the TF diagnostics controls the production of diagnostics in the meridional plane, including meridional slices at specified latitudes, meridional means and averages of fields in a specified latitude range. The transient fluxes analysed by this subset can be specified independently of the fields used by the other TF diagnostic subsets. Table 2.12 illustrates the keywords and their associated namelist variables in the FM data block.

Meaning	Keyword name	Namelist variable	Default value
Switch to allow horizontal (ie. latitude-longitude) TF diagnostics. Setting this switch to `.TRUE.` enables information provided in the FH data block to be used by the program.	`HORIZONTAL`	`LDFH`	`.FALSE.`
Switch to allow zonal TF diagnostics. Setting this switch to `.TRUE.` enables information provided in the FZ data block to be used by the program.	`ZONAL`	`LDFZ`	`.TRUE.`
Switch to allow meridional TF diagnostics. Setting this switch to `.TRUE.` enables information provided in the FM data block to be used by the program.	`MERIDIONAL`	`LDFM`	`.FALSE.`
Switch to allow profile TF diagnostics. Setting this switch to `.TRUE.` enables information provided in the FP data block to be used by the program.	`PROFILE`	`LDFP`	`.FALSE.`
Switch to allow output of three-dimensional TF fields. Setting this switch to `.TRUE.` enables information provided in the F3 data block to be used by the program.	`3DOUTPUT`	`LDF3`	`.FALSE.`
Fortran channel for printed output	`O/PCHANNEL`	`NCTF`	3
Integers denoting levels on which latitude-longitude slices will be produced (if output is on eta surfaces)	`TFLEVELS`	`NVTFHR`	`NL`*0
Real values denoting levels at which non-eta surface latitude-longitude slices will be output	`TFSURFACES`	`VTFSFC`	`NL`*0
Switch to enable output of total transient fluxes (ie the sum of the high-pass and low-pass components)	`TOTALFLUX`	`LTFTOT`	`.TRUE.`
Switch to enable output of high-pass transient flux components	`HIGHPASS`	`LTFHP`	`.TRUE.`
Switch to enable output of low-pass transient flux components	`LOWPASS`	`LTFLP`	`.TRUE.`
Real value denoting cut-off time (days) between the low- and high-pass components of the transient fluxes	`CUTOFF`	`TFCUT`	3.0
Number of different fields to be processed	`NUMBEROFFIELDS`	`NTF`	0
User-specified transient flux formulae (see section 7.3); only one formula allowed per line of the input file.	`FORMULA`	`NFLDTF`	`4JPTFMX`*0
		`FACTTF`	`2JPTFMX`*0
Mask factor employed in each equation	`MASK`	`YTFMSK`	`JPTFMX*X`

Table 2.9: Pre-processor keywords for transient flux (TF) data block, together with the corresponding diagnostic program variable name, and the default variable value that is assigned in the case when the user does not supply a value. `NL` is the number of model levels, or the number of levels of input data, and `JPTFMX` is the maximum number of transient flux formulae allowed.

Meaning	Keyword name	Namelist variable	Default value
Contour interval to be used for each field type when plotting the UTFs	CI	CINTFH	JPTFMX*-0.1
Codes for the fluxes whose diagnostics are to be printed, in the range 1 – NTF	PRINTFIELD	NFHPR	JPTFMX*0
Codes for the fluxes whose diagnostics are to be plotted, in the range 1 – NTF	PLOTFIELD	NFHPL	JPTFMX*0
Switch to enable the output of data on latitude-longitude slices — these are the slice specified by NVSGHR or VSGSFC (see Table 2.9.	SLICE	LDFHSL	.FALSE.

Table 2.10: Pre-processor keywords for the transient flux (FH) data block, together with the corresponding diagnostic program variable name and the default variable value, that is assigned in the case when the user does not supply a value. JPTFMX is the maximum number of transient fluxes that can be processed by the TF diagnostics.

Meaning	Keyword name	Namelist variable	Default value
Contour interval to be used for each field type when plotting the UTFs	CI	CINTFZ	JPTFMX*-0.1
Codes for the fluxes whose diagnostics are to be printed, in the range 1 – NTF	PRINTFIELD	NFZPR	JPTFMX*0
Codes for the fluxes whose diagnostics are to be plotted, in the range 1 – NTF	PLOTFIELD	NFZPL	JPTFMX*0
Switch to enable the output of zonal mean data	ZONALMEAN	LDFZZM	.FALSE.
Switch to enable the extraction of data along a specified longitude	NSSLICE	VFZNS	-999.0
Two longitudes between which averages (as a function of latitude) will be computed	NSAVERAGE	VFZNSA	2*-999.0

Table 2.11: Pre-processor keywords for the transient flux (FZ) data block, together with the corresponding diagnostic program variable name, and the default variable value that is assigned in the case when the user does not supply a value. JPTFMX is the maximum number of transient fluxes that can be processed by the TF diagnostics.

Note that to enable these diagnostics the `DIAGNOSTICS` switch in the COMMON data block must have been set to include the `TF` option, and in the TF data block the `MERIDIONAL` switch must have been set to `.TRUE.`.

The field codes (in the range 1 to `JPTFMX`, subject to a maximum of `NTF`) used in the FM data block refer to the transient fluxes specified in the TF data block.

Meaning	Keyword name	Namelist variable	Default value
Contour interval to be used for each field type when plotting the UTFs	`CI`	`CINTFM`	`JPTFMX`*-0.1
Codes for the fluxes whose diagnostics are to be printed, in the range `1` – `NTF`	`PRINTFIELD`	`NFMPR`	`JPTFMX`*0
Codes for the fluxes whose diagnostics are to be plotted, in the range `1` – `NTF`	`PLOTFIELD`	`NFMPL`	`JPTFMX`*0
Switch to enable the output of meridional mean data	`MERIDMEAN`	`LDFMMM`	`.FALSE.`
Switch to enable the extraction of data along a specified latitude	`WESLICE`	`VFMEW`	-999.0
Two latitudes between which averages (as a function of longitude) will be computed	`WEAVERAGE`	`VFMEWA`	2*-999.0

Table 2.12: Pre-processor keywords for the transient flux (FM) data block, together with the corresponding diagnostic program variable name, and the default variable value that is assigned in the case when the user does not supply a value. `JPTFMX` is the maximum number of transient fluxes that can be processed by the TF diagnostics.

2.5.4.4 TF(FP) diagnostics

The FP subset of the TF diagnostics controls the production of column profiles of transient fluxes at a user-specified series of locations. The transient fluxes analysed by this subset can be specified independently of the fields used by the other TF diagnostic subsets. Table 2.13 illustrates the keywords and their associated namelist variables in the TP data block.

Note that to enable these diagnostics the `DIAGNOSTICS` switch in the COMMON data block must have been set to include the `TF` option, and in the TF data block the `PROFILE` switch must have been set to `.TRUE.`.

The field codes (in the range 1 to `JPTFMX`, subject to a maximum of `NTF`) used in the FP data block refer to the transient fluxes specified in the TF data block.

2.5.4.5 TF(F3) diagnostics

The F3 subset of the TF diagnostics allows the user to output three-dimensional transient flux arrays to a file. This facility is useful if these fields are required as input to another program, or to a plotting package that does not accept UTFs as input. The user is able to re-program the appropriate routine in the diagnostic program to allow this output to be any required format, or to output a subregion of the transient flux arrays. The output file, by default, will contain data either in Fortran-formatted record format, or DRS format. Table 2.14 illustrates the keywords and their associated namelist variables in the F3 data block.

Note that to enable these diagnostics the `DIAGNOSTICS` switch in the COMMON data block must have been set to include the `TF` option, and in the TF data block the `3DOUTPUT` switch must have been set to `.TRUE.`.

The field codes (in the range 1 to `JPTFMX`, subject to a maximum of `NTF`) used in the F3 data block refer to the transient fluxes specified in the TF data block.

Meaning	Keyword name	Namelist variable	Default value
Codes for the fluxes whose diagnostics are to be printed, in the range `1` – `NTF`	`PRINTFIELD`	`NFPPR`	`JPTFMX`*0
Codes for the fluxes whose diagnostics are to be plotted, in the range `1` – `NTF`	`PLOTFIELD`	`NFPPL`	`JPTFMX`*0
Number of different coordinates at which profile calculations will be performed; must not be greater than `JPFPPF`	`NUMBEROFCOORDS`	`NFPPF`	0
Pairs of (latitude, longitude) profile coordinates	`PROFILECOORDS`	`VFPPF`	`2*JPFPPF`*-999.0
Switch to enable the output of vertical profile values	`COLPROFILE`	`LDFPCP`	`.FALSE.`

Table 2.13: Pre-processor keywords for the transient flux (FP) data block, together with the corresponding diagnostic program variable name, and the default variable value that is assigned in the case when the user does not supply a value. `JPTFMX` is the maximum number of transient fluxes that can be processed by the TF diagnostics, while `JPFPPF` is the maximum number of profiles that the program can handle in a single job.

Meaning	Keyword name	Namelist variable	Default value
Codes for the fluxes to be output, in the range `1` – `NTF`	`PLOTFIELD`	`NF3PL`	`JPTFMX`*0
Fortran channel for output	`O/PCHANNEL`	`NCF3`	1

Table 2.14: Pre-processor keywords for the transient flux (F3) data block, together with the corresponding diagnostic program variable name, and the default variable value that is assigned in the case when the user does not supply a value. `JPTFMX` is the maximum number of transient fluxes that can be processed by the TF diagnostics.

2.5.5 OD diagnostics

The (OD) diagnostics refer to the optional three-dimensional dynamical field diagnostics. These are not calculated on the input data surfaces, but on the output surfaces using interpolated gridpoint fields (if necessary).

The OD diagnostics are switched on by setting `OD` as one of the keyword values for the `DIAGNOSTICS` keyword in the COMMON data block. The user must then set the appropriate keywords in the OD data block (see Table 2.15) and in at least one of the OH, OZ, OM, OP or O3 data blocks – the daughter OD data blocks.

Meaning	Keyword name	Namelist variable	Default value
Switch to allow horizontal (ie. latitude-longitude) OD diagnostics. Setting this switch to `.TRUE.` enables information provided in the OH data block to be used by the program.	`HORIZONTAL`	`LDOH`	`.FALSE.`
Switch to allow zonal OD diagnostics. Setting this switch to `.TRUE.` enables information provided in the OZ data block to be used by the program.	`ZONAL`	`LDOZ`	`.TRUE.`
Switch to allow meridional OD diagnostics. Setting this switch to `.TRUE.` enables information provided in the OM data block to be used by the program.	`MERIDIONAL`	`LDOM`	`.FALSE.`
Switch to allow profile OD diagnostics. Setting this switch to `.TRUE.` enables information provided in the OP data block to be used by the program.	`PROFILE`	`LDOP`	`.FALSE.`
Switch to allow output of three-dimensional OD fields. Setting this switch to `.TRUE.` enables information provided in the O3 data block to be used by the program.	`3DOUTPUT`	`LDO3`	`.FALSE.`
Switch to allow time-averaging of OD fields.	`TAVERAGED`	`LDODTD`	`.TRUE.`
Switch to allow time series of OD fields. (Not allowed when DRS or NetCDF output has been selected.)	`TSERIES`	`LDODTS`	`.FALSE.`
Switch to allow output of instantaneous OD fields.	`INSTANTANEOUS`	`LDODIN`	`.FALSE.`
Fortran channel for printed output	`O/PCHANNEL`	`NCOD`	3
Integers denoting levels on which latitude-longitude slices will be produced (if output is on eta surfaces)	`ODLEVELS`	`NVODHR`	`NL`*0
Real values denoting levels at which non-eta surface latitude-longitude slices will be output	`ODSURFACES`	`VODSFC`	`NL`*0

Table 2.15: Pre-processor keywords for the optional dynamical diagnostic (OD) data block, together with the corresponding diagnostic program variable name, and the default variable value that is assigned in the case when the user does not supply a value. `NL` is the number of model levels, or the number of levels of input data.

Note that the time-control switches in the OD data block affect all the five daughter OD data blocks.

2.5.5.1 OD(OH) diagnostics

The OH subset of the OD diagnostics controls the production of diagnostics in the latitude-longitude plane, including horizontal slices at specified levels, Hovmöller plots and vertically integrated (mass-weighted) column totals. The fields analysed by this subset can be specified independently of the fields used by the other OD diagnostic subsets. Table 2.16 illustrates the keywords and their associated namelist variables in the OH data block.

Note that to enable these diagnostics the `DIAGNOSTICS` switch in the COMMON data block must have been set to include the `OD` option, and in the OD data block the `HORIZONTAL` switch must have been set to `.TRUE.`. Other switches in the OD data block must be set to enable the type of time-control (eg time averages, time series etc) required.

The field codes appropriate to the OD(OH) diagnostics can be found in Table 8.1.

Meaning	Keyword name	Namelist variable	Default value
Contour interval to be used for each field type when plotting the UTFs	`CI`	`CINTOH`	`JPODMX`*-0.1
Field codes for the fields whose diagnostics are to be printed	`PRINTFIELD`	`NOHPR`	`JPODMX`*0
Field codes for the fields whose diagnostics are to be plotted	`PLOTFIELD`	`NOHPL`	`JPODMX`*0
Number of different fields to be processed	`NUMBEROFFIELDS`	`NOH`	0
Codes of fields for which windvector information will be sent to the UTF	`WINDVECTOR`	`NOHVEC`	`JPODMX`*0
Switch to enable the output of data on latitude-longitude slices — these are the slice specified by `NVODHR` or `VODSFC` (see Table 2.15).	`SLICE`	`LDOHSL`	`.FALSE.`
Switch to enable the output of mass-weighted column totals. (This switch is only valid when output is on eta levels.)	`COLTOTAL`	`LDOHCT`	`.FALSE.`
Switch to output horizontal slice information in the form of eddy values (ie. with the zonal mean subtracted off)	`EDDY`	`LDOHED`	`.FALSE.`
Real values denoting the latitude range between which data on latitude-longitude slices (see `SLICE` above) will be averaged to produce Hovmöller output. `SLICE` must be set to `.TRUE.`. (Not allowed when DRS or NetCDF output has been selected.)	`HOVMOLLER`	`VOHTS`	2*–999.0

Table 2.16: Pre-processor keywords for the optional dynamical field (OH) data block, together with the corresponding diagnostic program variable name and the default variable value, that is assigned in the case when the user does not supply a value. `JPODMX` is the maximum number of fields that can be processed by the OD diagnostics.

2.5.5.2 OD(OZ) diagnostics

The OZ subset of the OD diagnostics controls the production of diagnostics in the zonal plane, including zonal slices at specified longitudes, zonal means and averages of fields in a specified longitude range. The fields analysed by this subset can be specified independently of the fields used by the other OD diagnostic subsets. Table 2.17 illustrates the keywords and their associated namelist variables in the OZ data block.

Note that to enable these diagnostics the `DIAGNOSTICS` switch in the COMMON data block must have been set to include the `OD` option, and in the OD data block the `ZONAL` switch must have been set to `.TRUE.`. Other switches in the OD data block must be set to enable the type of time-control (eg time averages, time series etc) required.

The field codes appropriate to the OD(OZ) diagnostics can be found in Table 8.1.

Meaning	Keyword name	Namelist variable	Default value
Contour interval to be used for each field type when plotting the UTFs	CI	CINTOZ	JPODMX*-0.1
Field codes for the fields whose diagnostics are to be printed	PRINTFIELD	NOZPR	JPODMX*0
Field codes for the fields whose diagnostics are to be plotted	PLOTFIELD	NOZPL	JPODMX*0
Number of different fields to be processed	NUMBEROFFIELDS	NOZ	0
Codes of fields for which windvector information will be sent to the UTF — THIS IS CURRENTLY DISABLED	WINDVECTOR	NOZVEC	JPODMX*0
Switch to enable the output of zonal mean data	ZONALMEAN	LDOZZM	.FALSE.
Switch to enable the extraction of data along a specified longitude	NSSLICE	VOZNS	-999.0
Two longitudes between which averages (as a function of latitude) will be computed	NSAVERAGE	VOZNSA	2*-999.0
Switch to enable the output of zonal-mean time series at the levels specified by NVODHR or VODSFC (see Table 2.15). ZONALMEAN must be set to .TRUE.. (Not allowed when DRS or NetCDF output has been selected.)	TIMESLICE	LDOZTS	.FALSE.

Table 2.17: Pre-processor keywords for the optional dynamical field (OZ) data block, together with the corresponding diagnostic program variable name, and the default variable value that is assigned in the case when the user does not supply a value. JPODMX is the maximum number of fields that can be processed by the OD diagnostics.

2.5.5.3 OD(OM) diagnostics

The OM subset of the OD diagnostics controls the production of diagnostics in the meridional plane, including meridional slices at specified latitudes, meridional means and averages of fields in a specified latitude range. The fields analysed by this subset can be specified independently of the fields used by the other OD diagnostic subsets. Table 2.18 illustrates the keywords and their associated namelist variables in the OM data block.

Note that to enable these diagnostics the DIAGNOSTICS switch in the COMMON data block must have been set to include the OD option, and in the OD data block the MERIDIONAL switch must have been set to .TRUE.. Other switches in the OD data block must be set to enable the type of time-control (eg time averages, time series etc) required.

The field codes appropriate to the OD(OM) diagnostics can be found in Table 8.1.

2.5.5.4 OD(OP) diagnostics

The OP subset of the OD diagnostics controls the production of column profile diagnostics at a user-specified series of locations. These diagnostics consist of profile values and column-integrated, mass-weighted totals — the latter only when output is on eta surfaces. The fields analysed by this subset can be specified independently of the fields used by the other OD diagnostic subsets. Table 2.19 illustrates the keywords and their associated namelist variables in the OP data block.

Note that to enable these diagnostics the DIAGNOSTICS switch in the COMMON data block must have

Meaning	Keyword name	Namelist variable	Default value
Contour interval to be used for each field type when plotting the UTFs	`CI`	`CINTOM`	`JPODMX`*-0.1
Field codes for the fields whose diagnostics are to be printed	`PRINTFIELD`	`NOMPR`	`JPODMX`*0
Field codes for the fields whose diagnostics are to be plotted	`PLOTFIELD`	`NOMPL`	`JPODMX`*0
Number of different fields to be processed	`NUMBEROFFIELDS`	`NOM`	0
Codes of fields for which windvector information will be sent to the UTF — THIS IS CURRENTLY DISABLED	`WINDVECTOR`	`NOMVEC`	`JPODMX`*0
Switch to enable the output of meridional mean data	`MERIDMEAN`	`LDOMMM`	`.FALSE.`
Switch to enable the extraction of data along a specified latitude	`WESLICE`	`VOMEW`	-999.0
Two latitudes between which averages (as a function of longitude) will be computed	`WEAVERAGE`	`VOMEWA`	2*-999.0
Switch to enable the output of meridional-mean time series at the levels specified by `NVODHR` or `VODSFC` (see Table 2.15). `MERIDMEAN` must be `.TRUE.`. (Not allowed when DRS or NetCDF output has been selected.)	`TIMESLICE`	`LDOMTS`	`.FALSE.`

Table 2.18: Pre-processor keywords for the optional dynamical field (OM) data block, together with the corresponding diagnostic program variable name, and the default variable value that is assigned in the case when the user does not supply a value. `JPODMX` is the maximum number of fields that can be processed by the OD diagnostics.

been set to include the `OD` option, and in the OD data block the `PROFILE` switch must have been set to `.TRUE.`. Other switches in the OD data block must be set to enable the type of time-control (eg time averages, time series etc) required.

The field codes appropriate to the OD(OP) diagnostics can be found in Table 8.1.

Meaning	Keyword name	Namelist variable	Default value
Contour interval to be used for each field type when plotting the UTFs	`CI`	`CINTOP`	`JPODMX*`-0.1
Field codes for the fields whose diagnostics are to be printed	`PRINTFIELD`	`NOPPR`	`JPODMX*`0
Field codes for the fields whose diagnostics are to be plotted	`PLOTFIELD`	`NOPPL`	`JPODMX*`0
Number of different fields to be processed	`NUMBEROFFIELDS`	`NOP`	0
Number of different coordinates at which profile calculations will be performed; must not be greater than `JPOPPF`	`NUMBEROFCOORDS`	`NOPPF`	0
Pairs of (latitude, longitude) profile coordinates	`PROFILECOORDS`	`VOPPF`	`2*JPOPPF*`-999.0
Switch to enable the output of vertical profile values	`COLPROFILE`	`LDOPCP`	`.FALSE.`
Switch to enable the output of column-integrated totals — valid only when output is on eta levels	`COLTOTAL`	`LDOPCT`	`.FALSE.`

Table 2.19: Pre-processor keywords for the optional dynamical field (OP) data block, together with the corresponding diagnostic program variable name, and the default variable value that is assigned in the case when the user does not supply a value. `JPODMX` is the maximum number of fields that can be processed by the OD diagnostics, while `JPOPPF` is the maximum number of profiles that the program can handle in a single job.

2.5.5.5 OD(O3) diagnostics

The O3 subset of the OD diagnostics allows the user to output three-dimensional (two-dimensional in the case of surface fields) OD-type fields (either instantaneous or time averaged) to a file. This facility is useful if these fields are required as input to another program, or to a plotting package that does not accept UTFs as input. The user is able to re-program the appropriate routine in the diagnostic program to allow this output to be any required format, or to output a subregion of the 3D fields. The output file, by default, will contain data either in Fortran formatted record format, or DRS format. Table 2.20 illustrates the keywords and their associated namelist variables in the O3 data block.

Note that to enable these diagnostics the `DIAGNOSTICS` switch in the COMMON data block must have been set to include the `OD` option, and in the OD data block the `3DOUTPUT` switch must have been set to `.TRUE.`. Other switches in the OD data block must be set to enable the type of time-control (eg time averages or instantaneous) required.

The field codes appropriate to the OD(O3) diagnostics can be found in Table 8.1.

2.5.6 ZF diagnostics

The (ZF) diagnostics refer to the zonally-averaged flux diagnostics, which may be computed using a user-programmable formula involving the three-dimensional (thermo)dynamical fields. These are calculated

Meaning	Keyword name	Namelist variable	Default value
Field codes for the fields to be output	PLOTFIELD	NO3PL	JPODMX*0
Number of different fields to be processed	NUMBEROFFIELDS	NO3	0
Fortran channel for output	O/PCHANNEL	NCO3	1

Table 2.20: Pre-processor keywords for the optional dynamical field (O3) data block, together with the corresponding diagnostic program variable name, and the default variable value that is assigned in the case when the user does not supply a value. `JPODMX` is the maximum number of fields that can be processed by the OD diagnostics.

on the output surfaces in gridpoint form from the (thermo-)dynamical fields. If the diagnostic output is not on eta surfaces (see keyword `LEVELTYPE` in the COMMON data block), then the fluxes are computed using interpolated SG-type fields appropriate to the output surface, using surface data values provided via the LV data block.

The ZF diagnostics are switched on by setting `ZF` as one of the keyword values for the `DIAGNOSTICS` keyword in the COMMON data block. The user must then set the appropriate keywords in the ZF data block (see Table 2.21).

2.5.7 XD diagnostics

The XD diagnostics enable the user to produce zonal-mean cross-sections of certain thermodynamical and dynamical fields, including those pertaining to transformed Eulerian mean diagnostics — code that was originally written by John Thuburn for the Flux program (unpublished).

The XD diagnostics are switched on by setting `XD` as one of the keyword values for the `DIAGNOSTICS` keyword in the COMMON data block. The user must then set the appropriate keywords in the XD data block (see Table 2.22). These diagnostics are only available if the user has specified that the output is required on pressure levels, ie. that keyword `LEVELTYPE` has been set to `IS` in the COMMON data block. Moreover, the transformed Eulerian mean (TEM) diagnostics may only be produced as a set of time-averaged XD diagnostics, since they involve 'rates of change'-type quantities.

Meaning	Keyword name	Namelist variable	Default value
Switch to allow output of eddy ZF diagnostics.	EDDYFLUX	LZFEDF	.FALSE.
Switch to allow output of mean ZF diagnostics.	MEANFLUX	LZFMNF	.FALSE.
Switch to allow output of zonal-mean ZF diagnostics.	ZONALMEAN	LDZFZM	.FALSE.
Switch to allow output of level-mean ZF diagnostics.	LEVELMEAN	LDZFLM	.FALSE.
Fortran channel for printed output	O/PCHANNEL	NCZF	3
Contour interval to be used for each field type when plotting the UTFs	CI	CINTZF	JPZFMX*-0.1
Codes for the fluxes whose diagnostics are to be printed, in the range 1 – NZF	PRINTFIELD	NZFPR	JPZFMX*0
Codes for the fluxes whose diagnostics are to be plotted, in the range 1 – NZF	PLOTFIELD	NZFPL	JPZFMX*0
Integers denoting levels on which zonal-mean time-series will be produced (if output is on eta surfaces)	ZFLEVELS	NVZFHR	NL*0
Real values denoting levels at which non-eta surface zonal-mean time-series will be output	ZFSURFACES	VZFSFC	NL*0
Number of different fields to be processed	NUMBEROFFIELDS	NZF	0
User-specified flux formulae (see section 9.3); only one formula allowed per line of the input file	FORMULA	NFLDZF	4JPZFMX*0
		FACTZF	2JPZFMX*0
Switch to allow time-averaging of ZF fields.	TAVERAGED	LDZFTD	.TRUE.
Switch to allow time series of ZF fields. (Not allowed when DRS or NetCDF output has been selected.)	TSERIES	LDZFTS	.FALSE.
Switch to allow output of instantaneous ZF fields.	INSTANTANEOUS	LDZFIN	.FALSE.

Table 2.21: Pre-processor keywords for zonal-mean flux (ZF) data block, together with the corresponding diagnostic program variable name, and the default variable value that is assigned in the case when the user does not supply a value. NL is the number of model levels, or the number of levels of input data, and JPZFMX is the maximum number of flux formulae allowed.

Meaning	Keyword name	Namelist variable	Default value
Fortran channel for output	O/PCHANNEL	NCXD	3
Switch to enable time-average (and TEM) diagnostic output	TAVERAGED	LDXDTD	.TRUE.
Switch to enable instantaneous (non-TEM) diagnostic output	INSTANTANEOUS	LDXDIN	.FALSE.
Codes for the fluxes whose diagnostics are to be printed, in the range 1 – NXD	PRINTFIELD	NXDPR	JPXDMX*0
Codes for the fluxes whose diagnostics are to be plotted, in the range 1 – NXD	PLOTFIELD	NXDPL	JPXDMX*0
Number of different fields to be processed	NUMBEROFFIELDS	NXD	0
Contour interval to be used for each field type when plotting the UTFs	CI	CINTXD	JPXDMX*-0.1
Switch to allow output of zonal-mean XD diagnostics.	ZONALMEAN	LDXDZM	.FALSE.
Switch to allow output of level-mean XD diagnostics.	LEVELMEAN	LDXDLM	.FALSE.

Table 2.22: Pre-processor keywords for zonal-mean (thermo)dynamical (XD) data block, together with the corresponding diagnostic program variable name, and the default variable value that is assigned in the case when the user does not supply a value.

2.6 Example use of the pre-processor

The data pre-processor is designed to read the user's keyword data from Fortran channel 10 and to create a namelist file for input to the diagnostic program on channel 11. Channel 12 is used as a temporary work file.

Figure 2.2 illustrates the format of the input file for the data pre-processor. Note that the indentation of the keyword lines is for clarity only — however, it is suggested that the user does likewise to distinguish between the different data blocks.

The first five lines after the keyword `COMMON` define which directories are to be used. Data for 12GMT on 4 November 1998 to 6 November 1998 will be used for the diagnostic calculations, and the SG diagnostics are switched on. Data will be sampled every 24 timesteps (ie daily), and any three-dimensional output or time-averaged output will contain data at the same time interval. The diagnostic job is given the name `SUMO JOB` and the jobnumber 1.0.

Output will be in the form of DRS files. The vertical coordinate for the output will be that of isobaric surfaces, orography is switched off (the data is from the UARS assimilated dataset and provides input to SUMO on pressure surfaces).

Since the pre-processor is stored as a nupdate library (under the name `presumo15.npl`), the following commands must be executed (where the pathnames are appropriate if the job is run on the Reading Sun workstations) to create the diagnostic program namelist — note that the number of options on each command has been minimised in this example:

```
nupdate -p /expt2/brugge/umdiagn/presumo15.npl -i presumo.updates -f -c cpl
f90 cpl.f
a.out
grep -F '*****END-OF-NAMELIST-FILE***' fort.11 > eoftest
if [ -s eoftest]
then
   echo "namelist file correctly formed"
   rm eoftest
else
   echo "namelist file incomplete; job stopped"
   exit 1
fi
```

After execution of a.out, a check is made to see that the namelist file has been correctly formed by searching for a phrase that should appear in the last record of the file. The final line in this command sequence writes a copy of the namelist file to stdout; the result is shown in Fig. 2.3.

```
COMMON
  INITIALDIR=/claus/andy/assim
  TEMPDIR=/tmp/brugge
  HOMEDIR=/home/brugge
  DATADIR=/tmp
  CURRENTDIR=/tmp/brugge
  BEGDAY=-981104
  ENDDAY=-981106
  BEGCENTURY=19
  ENDCENTURY=19
  BEGHOUR=12
  ENDHOUR=12
  RUNNINGDAYS=1.0
  DIAGNOSTICS=SG
  UTF=1.4
  LEVELTYPE=IS
  PRINT=F
  PLOT=F
  DRSOUTPUT=T
  NETCDFOUTPUT=F
  PRINTSIGFIG=2
  ASCIICHARS=2
  SCRATCH=F
  OROGRAPHY=F
  FREQSAMPLING=24
  FREQPRINT=24
  FREQPLOT=24
  JOBNAME=SUMO_JOB
  JOBNUMBER=1.0
  MASSWEIGHT=F
  MASKAVERAGE=F
  SIGMAAVERAGE=F
ENDCOMMON
MODELCONSTANTS
  STEPS/DAY=24.0
  STARTCENTURY=19
  STARTYEAR=98
  STARTMONTH=11
  STARTDAY=04
  STARTHOUR=12
  STARTMIN=00
  MODELLEVELS=22
  DATATYPE=UARS
  CALENDAR=GR
ENDMODELCONSTANTS
\$SG
  HORIZONTAL=F
  ZONAL=F
  MERIDIONAL=F
  PROFILE=F
  3DOUTPUT=T
  TAVERAGED=T
  TSERIES=F
  INSTANTANEOUS=F
  RUNNINGMEAN=F
  O/PCHANNEL=6
SGSURFACES=1000.0,100.0,10.0,1.0,0.316
\$ENDSG
```

Figure 2.2: Example input data file to the data pre-processor.

```
/tmp/brugge
/home/brugge
/tmp/brugge
/tmp
/claus/andy/assim
 &GLOBAL  BEGDAY = -981104., ENDDAY = -981106., YTYPSF = 'IS', LWRITE = F, LASCII = F, LSCRAT = F, LDRS = T, LNET = F, LUTF14 = T,
  LROGR = F, NSIGFG = 2, NFREQD = 24, NFREQP = 24, NFREQA = 24, RNTAPE = 1., YNAME = 'SUMO_JOB',  '        ', NASCII = 2, LMASSW = F
  LAVMSK = F, LAVSIG = F, BEGCEN = 19., ENDCEN = 19., BEGHR = 12., ENDHR = 12., DYRUNM = 1. /
 &FLXTYP  LDSG = T, LDTF = F, LDOD = F, LDZF = F, LDXD = F /
 &UMSPEC  TSPD = 24., NYEAR = 9811, NDAY = 412, NMIN = 0, NCENT = 19, LUARS = T, LUARRM = F, LUM = F, LUMRM = F, YCALEN = 'GR' /
 &FNDATA  YEXPUM = 'xxxx', YJOBUM = 'x', YSUBUM = 'a', YTIMUM = '.', YTYPUM = 'p', YSTMUM = 'j' /
....some namelists deleted....
 &SGDATA
   LDGH = F,
   LDGZ = F,
   LDGM = F,
   LDGP = F,
   LDG3 = T,
   LDSGTD = T,
   LDSGTS = F,
   LDSGIN = F,
   LDSGRM = F,
   NCSG =        6,
  NVSGHR =
          0,       0,       0,       0,       0,       0,
          0,       0,       0,       0,       0,       0,
          0,       0,       0,       0,       0,       0,
          0,       0,       0,       0,
  VSGSFC =
    0.10000E+04, 0.10000E+03, 0.10000E+02, 0.10000E+01, 0.31600E+00, 0.00000E+00,
    0.00000E+00, 0.00000E+00, 0.00000E+00, 0.00000E+00, 0.00000E+00, 0.00000E+00,
    0.00000E+00, 0.00000E+00, 0.00000E+00, 0.00000E+00, 0.00000E+00, 0.00000E+00,
    0.00000E+00, 0.00000E+00, 0.00000E+00, 0.00000E+00,
 /
....remaining namelists deleted....
```

Figure 2.3: Namelist output produced by the pre-processor input file in Fig. 2.2.

3 General notes

3.1 Contour intervals

Data on the input data files are assumed to be dimensional and the diagnostic program converts the data to 'meteorological-type' output units. The user is able to specify a contour interval in terms of these output units, that will be passed to the UTF. (Note, however, that depending upon the plotting package employed to process these UTFs, it may be possible to override such contour intervals during the plotting process.)

Elements `CINTXX` are the contour intervals for plotting fields from the XX diagnostics, if positive. If negative, the fields are both plotted normalised as follows:

- `CINTXX` < -1
  - for plotting normalise the range of the field to 1000, the contour interval being $-$`CINTXX` in normalised units. Eg. `CINTXX=-200` gives a contour interval 1/5th of the field range.
- $-1 \leq$ `CINTXX` < 0
  - For plotting normalise the range of the field to lie between 0.1 and 1.0 by removing powers of 10 only, the contour interval being $-$`CINTXX` in normalised units. Eg. `CINTXX=-0.2` gives a contour interval 1/5th of the next power of 10 greater than the field range.

3.2 Fortran parameters

In order to control (ie. minimise) the amount of memory required by the diagnostic program, use if made of Fortran parameters to dimension many variables. It is important that these are chosen to be consistent with the originating GCM run or input dataset.

In this section reference will only be made to those that need be altered during runs of the program as it stands at present. Users wishing to rewrite parts of the program should refer to the parameter documentation within the source code.

Parameters that the user frequently may need to change can be found in `COMDECK PARAM1`:

- `JPNHEM` Determines the domain of the atmosphere — 1: hemispheric with flow symmetric about equator, 2: full global domain. For a GCM run the value 2 applies.
- `JPNL` Number of model (η) levels (or the number of levels of input data) (N_L) in the vertical.
- `JPOUT` Number of levels of output data when output is vertically interpolated. If used, and `JPOUT` is not equal to `JPNL` then `OUTLEVS` must be defined with `JPOUT` values — see section 2.5.5.
- `JPMG` Number of longitudes of input data.
- `JPJG` Number of latitudes between pole and equator (velocity gridpoints).
- `JPWORD` Number of bytes per word – 8 on the 64-bit Cray and 4 on the 32-bit Sun workstations.

3.3 UGAMP transfer files — UTFs

The UGAMP transfer file is a means of passing output data from the diagnostics program to a remote site for plotting purposes. The program currently supports an extended form of UTF version 1.4 — see section 3.7. For a definition of these standards, the user should read Blackburn *et al.* (1992).

To use the UTF output facility in the program, the user must (at the very least) ensure that `LUTF14` is set to `.TRUE.` and that `LASCII` is also set to the value `.TRUE.` — see the COMMON data block (section 2.2).

3.4 Channel numbers

Table 3.2 shows the channel numbers currently used by the diagnostic program.

Channel number	Channel name	Description
1	NCG3	SG 3-D output
1	NCF3	TF 3-D output
1	NCO3	OD 3-D output
3	NCSG	SG printed diagnostic output
3	NCOD	OD printed diagnostic output
3	NCZF	ZF printed diagnostic output
3	NCXD	XD printed diagnostic output
9	NCHIST	History data file
11	NCNAME	Input channel for namelist data
14	NCAIRM	Horizontal land/air masks
15	NCROGG	Gridpoint orography data
16	NCROGS	Initial gridpoint data file
17 – 24		Reserved for user-written code, and for reassignment of files assigned to channel 3 by default.
29	NCGHDI	SG/GH DRS dictionary file
30	NCGHDA	SG/GH DRS data file
31	NCZMDI	Zonal mean DRS dictionary file
32	NCZMDA	Zonal mean DRS data file
33	NCNS2G	Spectral to grid transform work file
34	NCNG2S	Grid to spectral transform work file
35	NCWORK	spectral to grid transform work file
41	NCGP(1)	3-D gridpoint data
42	NCGP(2)	Surface gridpoint data
43	NCGP(3)	Model half-level gridpoint data
60	NCTF3	Tf work file
63	NCRUNM	Running-mean output file
65	NCUTF	UTF
66	NCZFWK	ZF time series work file
68	NCGPAC(1)	3-D time accumulated gridpoint data
69	NCGPAC(2)	Surface time accumulated gridpoint data
70	NCGPAC(3)	Model half-level time accumulated gridpoint data
71	NCGPAV(1)	3-D time averaged gridpoint data
72	NCGPAV(2)	Surface time averaged gridpoint data
73	NCGPAV(3)	Model half-level time averaged gridpoint data
74	NCGPTS	GP time series work file
74	NC4DAT	4-D DRS data file
75	NCGHTS	GH time series work file
75	NC4DIC	4-D DRS dictionary file
76	NCGZTS	GZ time series work file
76	NC3TDA	3-D DRS time average data file
77	NCGMTS	GM time series work file
77	NC3TDI	3-D DRS time average dictionary file
78	NC3DAT	3-D DRS data file
79	NC3DIC	3-D DRS dictionary file
80	NC2TDA	2-D DRS time average data file

Table 3.1: Fortran channel numbers used by the diagnostic program.

Channel number	Channel name	Description
81	NC2TDI	2-D DRS time average dictionary file
82	NCOPTS	OP time series work file
82	NCFHDI	TF/FH DRS dictionary file
83	NCOHTS	OH time series work file
83	NCFHDA	TF/FH DRS data file
84	NCOZTS	OZ time series work file
84	NCOHDI	OD/OH DRS dictionary file
85	NCOMTS	OM time series work file
85	NCOHDA	OD/OH DRS data file
86	NCODAC	3-D time accumulated OD data
87	NCODAV	3-D time averaged OD data
88	NCODC	OD data
89	NCETWK	Used to store miscellaneous data on input surfaces
90	NCTF1	TF work file
91	NCTF2	TF work file
92	NCRECN	RECONA file from Unified Model

Table 3.1(cont.): Fortran channel numbers used by the diagnostic program.

3.5 SUMO coding styles/conventions

The following guidlines on the standards and styles used within SUMO should be read in conjunction with the UGAMP Fortran programming standards documentation (Steenman-Clark, 1993). Notes on the standards/styles contained in SUMO – this will eventually be incorporated into the documentation.

The SUMO nupdate source code library is stored in the following order:

- COMDECK OUTLINE
- COMDECK DATE
- Various VERSION COMDECKs
- COMDECK TITLE
- Parameter comdecks
- /Common/ comdecks — inserted alphabetically.
- Program library decks — typically each new diagnostic type will create one new deck; initialisation and checking code are added to existing decks. Decks are added in chronological order; subroutines within each deck are usually listed in alphabetical order.

The following is an example of the sequencing of the source code within the program library:

```
*COMDECK OUTLINE
C  MAINAP                      - entry/controlling routine
Cdata input and checking -----------------------------------------------
C    PARCHK                    - check PARAMETERs in PARAM1
C    CHANNL                    - define fortran channel numbers, pathnames
C      CHOPEN                  - open work files
C    HEADER                    - print out header info about the job
C    DATAIN                    - read input and namelist data
C    DATES                     - convert between dates and days
C    OVRRID                    - overrides certain input data
```

```
C    DATCHK                   - check global input and namelist data
C      AVGCHK                 - check mask and weighting input
C      CHKLV                  - check LV data
C      CHKOR                  - check OR data
         .
         .
         .
Cutility routines used
C  ABORT                      - premature termination routine after error
C  ADD0                       - dummy routine for TF calculations
         .
         .
         .
CSUN-specific routines
C  ETIME                      - calendar routine
C  FREE                       - routine to release heap space
C  MALLOC                     - heap allocation routine
C
CCray-specific routines
C  ASNFILE                    - file assign routine
C  ASNUNIT                    - unit assign routine
C  CLOCK                      - returns current time
         .
         .
         .
Cother routines required
C  DRS library functions
C  NetCDF library routines
C
Cother non-standard features used
*IF DEF,SUN,OR,DEF,CRAY
C  POINTER
*ENDIF
C  Named COMMON items defined using DATA, rather than in blockdata
C        subprogram
C  Mixing of CHARACTER and non-CHARACTER items in COMMON
C
*COMDECK DATE
**    *                      DATE : MAY 1996                       *
*COMDECK TITLE
**
**    **************************************************************
**    *   U.K. UNIVERSITIES GLOBAL ATMOSPHERIC MODELLING PROJECT   *
**    *                                                            *
**    *           DIAGNOSTIC PROGRAM FOR  UKMO ASSIMILATED         *
**    *                       DATA ANALYSIS                        *
**    *    AND FOR THE ANALYSIS OF SELECTED UNIFIED MODEL OUTPUT   *
**    *                                                            *
*CALL VERSIONC
**    *                                                            *
**    *                      AUTHOR : R.BRUGGE                     *
**    *                                                            *
**    * History:                                                   *
```

```
**    *   1.0   : Initial release                       Apr 1996  *
**    *   1.1   : Addition of running means             May 1996  *
*CALL DATE
**    **************************************************************
**
```

The code is then followed by comdecks and decks in order given above.

There are two types of comdecks within the nupdate library, namely parameter comdecks and common block comdecks. Of the parameter comdecks, `PARAM1` contains GCM/data-dependent parameters which may need to change from job to job. `PARAM2` contains parameters derived from those in `PARAM1`, and should not require alteration. `PARAM3` contains parameters that may need to change as the source code is altered (eg the number of fields for each type of diagnostic). The style of each parameter comdeck is similar to the style of each common block comdeck — see below.

Common block comdecks should each contain related variables/constants. The structure of each is as follows:

```
*COMDECK DIAGTP
      LOGICAL LDSG,LDTF,LDOD,LDZF,LDXD
      COMMON /DIAGTP/LDSG,LDTF,LDOD,LDZF,LDXD
*IF DEF,DOC
C-----------------------------------------------------------------------
C     DIAGTP  -  DIAGNOSTIC TYPES
C
C     NAME    MEANING
C     ----    -------
C     LDSG    .TRUE. if 3-D/surface fields to be output
C     LDTF    .TRUE. transient fluxes to be output
C     LDOD    .TRUE. optional dynamical diagnostics to be output
C     LDZF    .TRUE. zonal mean/eddy fluxes to be output
C     LDXD    .TRUE. if zonal-mean diagnostics required
C-----------------------------------------------------------------------
*ENDIF
```

Note that integer and real variable types are not explicitly defined. Any logical and/or character declarations appear before the `COMMON` statement. A one line description of the common block precedes a description of the meaning of all the variables.

The initial letter(s) of variables are used to defines the type and origin of the variables themselves, as follows:

- `LO` : local logical
- `LP` : logical passed into a routine as an argument
- `L` : other logicals
- `YO` : local character
- `YP` : character passed into a routine as an argument
- `Y` : other characters
- `Z` : local real
- `P` : real passed into a routine as an argument
- `I` : local integer

- `K` : integer passed into a routine as an argument
- `J` : do loop counter
- `M,N` : other integers
- `NC` : fortran channel/unit name
- `JP` : parameter
- `NR` : record number for data on direct access files.

6-character (maximum) variable names are used throughout, and hollarith variables are not used.

Diagnostic type codes (eg `SG`, `OD`) included in variable names to differentiate between similar variables for different diagnostics, eg `LDSG`, `LDOD`.

The outline of each subroutine or function is as follows — note that common decks and parameters are accesses via the appropriate comdeck:

```
      SUBROUTINE TIMINI
**    Function - to initialise SG scratch file for time-average
**          calculations
**    Args in - none
**    Args out - none
**    Args for work - none
**    Com used - /COMIOC/NCGPAC
**    Com changed - /CWORK/WGP2D1,/VARTIM/NFLGAV
**    Params used - /PARAM1/JPJG,/PARAM2/JPIGC,JPIGD,JPNLP,
**    Called by - SGINIT,TRINIT
**    Calls - none
**    Files read - none
**    Files written - NCGPAC(1),NCGPAC(2),NCGPAC(3)
**    Author - R Brugge, University of Reading

*CALL PARAM1
*CALL PARAM2
*CALL COMIOC
*CALL CWORK
*CALL VARTIM
*IF DEF,RECL
      DIMENSION Z(JPIGD),ZS(JPIGC*JPJG),ZH(JPIGC*JPNLP)
*ELSE
      DIMENSION Z(JPIGD*JPJGG),ZS(JPIGC*JPJGG),ZH(JPIGC*JPNLP*JPJGG)
*ENDIF
      EQUIVALENCE(ZS(1),WGP2D1(1,1))
:
:
Executable code
:
:
      RETURN
      END
```

The following points should be noted about the coding style:

- Blank lines do not start with a `C`.

- Comment lines start with a `**`.
- `C` in the first column is used for certain 'temporary' comments, eg. to comment out call statements that may be withdrawn at a later date.
- Comments are written in mixed case.
- Fortran is written in upper case.
- Contents of `IF...ENDIF` loops are indented by 3 columns.
- Continuation lines contain a minus (-) character in column 6.
- Continuation lines are indented by 6 characters.
- `FORMAT` statements appear after the first line referencing the statement.
- `DO` loops end with a separate `CONTINUE` statement.
- `ENDIF` coded as a single word.
- Section/subsection numbering not used.
- Labels 6000 to 6999 used (generally) for channel 6 output.
- Comments should be used liberally within the code to define variables and methods.

3.5.1 Fortran 90 style changes

Beginning with version 1.3, it is envisaged that major changes to any of the SUMO subroutines will be accompanied by style changes to convert such routines into fortran 90 code. Changes made to the style when using fortran 90 include the following (fixed 72-column width lines are still used, rather than the free-format option):

- Comment lines have `!` as the first significant character. The `!` appears in the first column when used at the head of each subprogram to describe the subprogram characteristics, otherwise it is indented to be aligned with any following code.
- No new `COMMON` blocks will be added — `MODULE`s are used instead.
- The `DO...CONTINUE` construct is replaced by the `DO...ENDDO` construct.
- Contents of `DO...ENDDO` loops are indented by 2 columns.

3.6 SUMO skeleton — the DIY guide

One of the requirements underpinning the development and style of SUMO is the facility for users to incorporate their own changes into the program. While such changes may be of a temporary nature, it is anticipated that many will evolve into 'standard' SUMO diagnostics. This section details the underlying skeleton of SUMO, and should enable the user to decide where to insert his/her changes.

Figure 3.1 illustrates the basic SUMO skeleton – for the up to date flow-tree of SUMO the reader is referred to comdeck OUTLINE.

To facilitate understanding of the code, use has been made of the nupdate software to group all routines that relate to a specific diagnostic type together in a single deck. These deck names may also be found in comdeck OUTLINE. Note that the deck and routine names are not the same — decks are used to group together sets of related routines.

Figure 3.1 illustrates that SUMO consists of five separate sections of code, with entry into each being controlled by the main program `MAINAP`.

```
   MAINAP                    - entry/controlling routine
-------------data input and checking
     PARCHK                  - check PARAMETERs in PARAM1
     CHANNL                  - define fortran channel numbers, pathnames
       DACNST                - define direct access file constants
       CHOPEN                - open work files
     HEADER                  - print out header information about the job
     DATAIN                  - read input and namelist data
     OVRRID                  - overrides certain input data
     DATES                   - convert between dates and days
     DATCHK                  - check global input and namelist data
       CHKLV                 - check LV data
               and similar calls for other diagnostic types
     CHHEAD                  - print out channel header information
-------------derive constants
     CONST                   - define physical/maths constants
     INIT                    - calculated derived constants - some i/o
     DINIT                   - controls initialisation for specific diagnostics
       SGINIT                - initialise SG diagnostics, constants etc
               and similar calls for other diagnostic types
-------------initial output prior to the time loop
     UTF14                   - interface to INIU14, initialise and write header
     INIOP                   - initial (non-time-dependent) field output
       INIROG                - read, print or plot the orography(geopotential)
-------------start of time loop
     ANALYS                  - controls time loop and time analysis
-------------end of time loop
     ENDOP                   - produce output at end of diagnostics
       ENDSG                 - terminate the SG time-diagnostics
               and similar calls for other diagnostic types
     EUTF14                  - write UTF 1.4 tail
     CHCLOS                  - control the closing of work files
```

Figure 3.1: Skeleton routines defining the structure of SUMO.

• Initially, the data is input and checked. This section also contains a routine to define the fortran channels, to open these channels and to set up direct access file pointers for the relevant gridpoint fields. The default namelist values are set in `DATAIN`, where the PRESUMO- (or user-) supplied namelist data are also read. The routine `DATCHK` initiates calls to a tree of data-checking routines (eg `CHKSG`) which are used to terminate the diagnostic job, in the event that the input data is incorrect. Note that this is required in case the user has not made use of the checking facilities in PRESUMO.

• The next step is to define any constants (physical, model resolution- or data-dependent) required by the program. Again each type of diagnostic is associated with an initialisation routine, wherein arrays containing information such as field titles, ordered arrays of contour intervals, printing and plotting switches etc, are set up.

• The program should then be ready to run. The initial output consists of the UTF headers, and the output of any data that does not vary with time (eg the initial data fields – if applicable).

```
-------------start of time loop
     ANALYS                  - controls time loop and time analysis
       RDDATA                - controls reading of field data
       SGFLUX                - compute gridpoint fields -- see Figure 3.3
       INIMSK                - make lat-long, zonal and merid mean orogr masks
       SGCALC                - SG diagnostics controller
         SGINST              - instantaneous and time average calcs and o/p
           MSKPL             - copy orographic masks to UTF
           SGVEC             - copy wind vector information to UTF
           HROUT             - horizontal SG diagnostics
           HRDRS             - horizontal SG DRS o/p
           HRNET             - horizontal SG NetCDF o/p
           GZOUT             - zonal SG diagnostics
             ZMDRS           - zonal SG DRS o/p
             ZMNET           - zonal SG NetCDF o/p
           GMOUT             - meridional SG diagnostics
           GPOUT             - profile SG diagnostics
         TIMACC              - accumulate SG fields for time-averages
         DRS3D               - output 3-D data in DRS format
         OPSG3D              - output 3D SG fields
       ODCALC                - OD diagnostics controller
       XDFLUX                - XD diagnostics controller
       ZFCALC                - ZF diagnostics controller
-------------end of time loop
```

Figure 3.2: Illustrating the structure of routine `ANALYS`, (SUMO version 1.1).

• The time-stepping is controlled by routine `ANALYS` — see Figure 3.2 for the structure of this section of the code that existed at version 1.1. The routine keeps track of the timestep counter, and computes whether instantaneous printouts or UTFs of output data are required for each step. Note that a time-series is considered to be a sequence of instantaneous fields, so the values of variables `NFREQP` and `NFREQA` will also control the contents of the printed and UTF time-series — which can consequently be different. Thus the contents of a time-series is not solely dependent upon the diagnostic sampling frequency `NFREQD`.

`ANALYS` calls routine `RDDATA` which reads in the gridpoint data as required.

The next routine, `SGFLUX` may be considered central to the whole SUMO program – see Figure 3.3. It

```
       SGFLUX                 - Gridoint fields computed
-------------start of latitude pair loop
         GRMULT               - computes various gridpoint fields on eta levels
           PCNST              - derive various pressure-related arrays
             PSGCE            - computes input data-level pressures
           NTEND              - computes non-linear momentum tendencies
           GHGCR              - compute geopotential height
           WCALC              - vertical velocity computed
           SRFCR              - various surface and low-level fields found
           PVCR               - theta and potential vorticity found
         INTPR                - control interpolation to pressure levels
           INTERP             - interpolate one field to pressure levels
         INTTH                - control interpolation to theta levels
           DPDTHG             - compute mass weighting term on eta half levels
           INTTH2             - extra/interpolate one field to theta levels
           INTERT             - interpolate one field to theta levels
         INTPV                - control interpolation to PV surfaces
           INTERE             - interpolate one field to PV surfaces
         MSSFAC               - multiply isentropic fields by mass-weighting
         TFCONT               - TF calculations at current latitude (if possible)
         ZFCONT               - ZF calculations at current latitude (if possible)
         ODFLXL               - single latitude OD calculations
-------------end of latitude pair loop
         TFCONT               - TF global calculations if required
         ZFCONT               - ZF global calculations if required
         ODFLXG               - OD global calculations
         LLAIR                - create lat-long land/air masks in COMMSK or scratch
```

Figure 3.3: Illustrating the structure of routine `SGFLUX`, (SUMO version 1.1).

also acts as a location where the user may access all the gridpoint fields latitude-by-latitude – immediately after the call to routine `MSSFAC`. The user should ensure that `SGFLUX` will be called if changes are to made to `SGFLUX` – this can be done in routine `ANALYS`.

Note that `TFCONT` may be called from one of two locations within `SGFLUX` — dependent upon whether vertically-interpolated surface fields of vorticity, streamfunction, divergence and/or velocity potential are required by the TF diagnostics. After the second call to `TFCONT` the user may insert routines to compute diagnostics that need to access multiple latitudes of given field(s) — including whole 3-D SG fields. In this case it is worth checking the contents of routine `WRITSG` to see how the gridpoints fields were written to the direct access scratch files (substituting Fortran READs for WRITEs will enable the fields to be copied into memory). Routine `ZFCONT` is, likewise, called in two places to compute the ZF diagnostics.

The arrays in common block `/CWORK/` may be used to hold 2-D or 3-D fields.

Returning to Figure 3.2, routine `SGCALC` controls the calculation of the output fields (typically slices and sections), and could be followed by any user-written code to extract particular slices or sections from a 3-D field.

It should also be pointed out that the entire 3-D SG fields are accessed in routine `OPSG3D`. This routine can be used to (for example) extract particular elements of these fields. In this case the user should ensure that the SG diagnostic type has been selected, and that within the PRESUMO SG data block `3DOUTPUT` and `INSTANTANEOUS` are both set to `.TRUE.`. Also `PLOT` must be set to `.TRUE.` in this case, since the 3-D output routines are invoked by this flag.

It is assumed that virtually any changes to SUMO will involve the SG (dynamical) fields.

- Finally, returning to Figure 3.1, routine `ENDOP` controls the output of time-average and time-series. In this case `ENDSG` will access the time-averaged 3-D SG fields — likewise for the TR fields. In this case, note that the switch `TIMEAVERAGE` must be set to `.TRUE` in the SG.

3.7 Extensions to the UTF1.4 standard

SUMO allows the user to output data on surfaces of constant potential vorticity. Such a possibility was not foreseen when the UTF version 1.4 standard (Blackburn *et. al.*, 1992) was defined. As a result, SUMO now employs an extension to this standard.

Accordingly, Table 6 of Blackburn *et. al.* (1992) should now be as follows:

LTYP	Co-ordinate type	Units	Typical Range
0	sigma	sigma*1000.0	0.0 to 1000.0
1	hybrid sigma/pressure	eta*1000.0	0.0 to 1000.0
2	pressure	mb	0.0 to 1000.0
3	isentropic	Kelvin	7000.0 to 300.0
4	hybrid sigma/isentropic	eta*1000.0	0.0 to 1000.0
5	potential vorticity	pvu	10000.0 to 0.0

Table 3.2: Revised table 6 of Blackburn *et. al.* (1992) as used in SUMO to enable output on potential vorticity surfaces.

3.8 A brief guide to SUMO input and output data

A succession of developments of SUMO means that the program is able to diagnose data from a variety of sources, and to generate output in a variety of different formats. As a result, the number of switches, libraries, code changes, etc., that a user may need to alter when changing the SUMO jobdeck to run in a different mode is rapidly growing. To clarify these changes, the user should refer to Tables 3.3 and 3.4 when setting up SUMO to run in a different mode.

SUMO expects filenames to be of a set structure (naming convention), namely

- `ukmon` — UARS assimilated data (`n` is a 1 to 4-digit integer defining the 'UARS day number')
- `ukmormccyymmddhh` — UARS running-mean data, generated from UARS assimilated data file; (`ccyymmddhh` is the start-time of the running-mean period of each file)
- `ccfmaa.pjj4110` — output from the Unified Model (on pressure surfaces) — see section 2.3.1.
- `umrmccyymmddhh` — Unified Model running-mean data (on pressure surfaces); (`ccyymmddhh` is the start-time of the running-mean period of each file)

Note that UM output can be in IEEE- or Cray-format. The former can be read on all platforms, the latter only on a Cray.

3.9 SUMO and nupdate conditional directives

Each successive development of SUMO tends to increase the size of the source code and the number of diagnostics available, with a consequent increase in the compilation time and memory usage. There is also, arguably, a reduction in the ease with which the user can incorporate personal code changes into the program.

Consequently, use is made of the nupdate conditional directives to enable control to be exercised over the code that is extracted from the nupdate program and inserted into the user's fortran compile file. These directives are illustrated in Table 3.4 — in general they are of the form of two-letter keywords that match the

Model type/data source	**Libraries required**	**PRESUMO input**	**PRESUMO code changes**	**SUMO code changes**	**Diagnostics disallowed**
UKMO/UARS assimilated data	†	`DATATYPE=UARS`			
UKMO/UARS running-mean assimilated data	†	`DATATYPE=` `UARSRUNMEAN`		`*DEF RUNMEAN`	
Output from Unified Model	†	`DATATYPE=` `UNIFIEDMODEL`		`*DEF MLEVELS`	
Unified Model output in running-mean format	†	`DATATYPE=` `UNIFIEDMODELRUNNINGMEAN`		`*DEF RUNNMEAN` `*DEF MLEVELS`	
SUMO output data format					
Printed results		`PRINT=T`			
UTF and/or 3-D arrays in a fortran unformatted file (uf)§		`PLOT=T` `DRSOUTPUT=F` `NETCDFOUTPUT=F`			CO. Only G3, O3 and F3 allowed in uf.
					Diagnostics allowed
DRS files§	libdrs.a	`DRSOUTPUT=T` `PLOT=F` `NETCDFOUTPUT=F`		`*DEF DRS`	GH, GZ, G3, FH, FZ, F3, OH, OZ, O3, ZF, XD
NetCDF files§ (not yet implemented)	libnetcdf.a	`DRSOUTPUT=F` `PLOT=F` `NETCDFOUTPUT=T`		`*DEF NET`	GH, GZ, FH, FZ, OH, OZ, ZF, XD
UKMO PP files		`PLOT=T` `DRSOUTPUT=F` `NETCDFOUTPUT=F`			G3
UKMO PP files		`RUNNINGMEAN=T`		`*DEF RUNMEAN`	GH (input files on input levels only)

Table 3.3: Requirements for running SUMO in different modes. § — Only one of NetCDF, UTF and DRS output available from the same SUMO job. † — use `sumolib.o`, together with `ffio.o` and `libgrib.a` on Sun workstations; use `libsumo12.a`, together with `libffio32.a`, `libblas32.a`, `libumgrib32.a`, `libutil32.a` and `libemos32.a` on the Cray.

Switches, etc.	Consequence	Notes
Unix job script settings		
SGMR='no'	fort.41 and fort.68 reside on disk	Reduces memory use but increases i/o times
SGMR='yes'	fort.41, ie `NCGP(1)`, memory resident	Reduces i/o times while increasing memory usage if SG-type fields are used
SGMR='yes', TIMAVG='yes'	fort.41 and fort.68, ie `NCGPAC(1)`, memory resident	Reduces i/o times while increasing memory usage if SG-type time-averages are required (Setting any of the above switches to 'yes' can significantly reduce cpu time needed and is most useful when job memory can be kept below 10Mw on the RAL Cray.)
DRSOUTPUT='T'	Used to access additional DRS library code	
NETCDFOUTPUT='T'	Used to access additional NetCDF library code	
USMMMODELDATA='.TRUE.'	Used to specify USMM-generated input data	
PRESUMO data input		
`MODELLEVELS=n`	`n` must match the value assigned to SUMO parameter `JPNL`	

Table 3.4: Notes on the setting of switches, etc., that apply to all SUMO jobs.

Switches, etc.	Consequence	Notes
SUMO nupdate conditional directives		
`*DEF nn`, for each of the nn (= TF, OD, ZF and XD) diagnostics used	Includes appropriate source code in the SUMO fortran file.	Each 'definition' will increase memory usage and compilation time. see section 3.9.
Modify lines `PARAM1.2` to match input data resolution parameters.	Ensures data arrays are correctly dimensioned.	
`*DEF RECL` (Not yet checked - DO NOT USE)	If used, then files containing 3-D accumulated field values contain data for one latitude pair per file record.	Can be used to reduce memory usage, but increases i/o times. Useful for high resolution data analysis.
`*DEF DRS`	Includes DRS-related source code in the SUMO fortran file.	
`*DEF DOC`	No need to use this switches	See SUMO source listing for details.
`*DEF NET`	Activate code to generate NetCDF output.	DO NOT USE - NOT YET IMPLEMENTED
`*DEF RUNMEAN`	Activate output code to create running-mean output in PP-format.	
`*DEF JPOUT`	Enable fewer output levels that input.	
`*DEF CRAY`	UM-Cray, UM-IEEE and UARS-pp formatted data can be used	MUST BE USED IF RUNNING SUMO ON A CRAY
`*DEF SUN`	UM-IEEE and UARS-pp formatted data can be used	MUST BE USED IF RUNNING SUMO ON A SUN WORKSTATION
`*DEF HP`	UM-IEEE and UARS-pp formatted data can be used	MUST BE USED IF RUNNING SUMO ON A HP WORKSTATION
`*DEF SGI`	UM-IEEE and UARS-pp formatted data can be used	MUST BE USED IF RUNNING SUMO ON THE CSAR SGI
`*DEF FUJI`	UM-IEEE and UARS-pp formatted data can be used	MUST BE USED IF RUNNING SUMO ON THE NERC FUJITSU

Table 3.4(cont.): Notes on the setting of switches, etc., that apply to all SUMO jobs.

diagnostic keywords. Note that the use of certain of these directives may result in a few warning messages from the loader (ie segldr) to the effect that there are 'unsatisfied externals'. These warnings can be ignored.

3.10 DRS output files and their contents

It must be noted that the diagnostics that may be output with the DRS option set to `.TRUE.` are only a subset (currently 3-D fields, lat-long output on specified surfaces, and zonal-means) of those available in UTF format, since it is envisaged that the DRS-user will wish to subsequently sub-sample or manipulate the DRS data interactively (eg. using VCS).

Table (3.5) illustrates the DRS filenames that are generated internally by SUMO, together with their current contents. This table should be used as a reference to the DRS-formatted diagnostics that may be produced by SUMO.

Channel number – see also Table 3.1	DRS filenames	File contents	Description
NC4DAT, NC4DIC	sg4d.dat, sg4d.dic	SG/G3, OD/O3 three-dimensional arrays	Instantaneous 3-D fields at different times, generating 4-D time-series arrays
NC3DAT, NC3DIC	sg3d.dat, sg3d.dic	SG/G3 two-dimensional arrays	Instantaneous 2-D fields (for data arrays that exist only at a single level) at different times, generating 3-D time-series arrays
NC3TDA, NC3TDI	sg3t.dat, sg3t.dic	SG/G3, OD/O3, TF/F3 three-dimensional arrays	Time-averaged 3-D fields
NC2TDA, NC2TDI	sg2t.dat, sg2t.dic	SG/G3 two-dimensional arrays	Time-averaged 2-D fields (for data arrays that exist only at a single level)
NCGHDA, NCGHDI	sggh.dat, sggh.dic	SG/GH lat-long arrays	Instantaneous and time-averaged lat-long arrays at one or more levels
NCZMDA, NCZMDI	zm.dat, zm.dic	SG/GZ, TF/FZ, OD/OZ ZF, XD zonal-mean arrays	Instantaneous and time-averaged zonal-mean arrays
NCFHDA, NCFHDI	tffh.dat, tffh.dic	TF/FH lat-long arrays	Lat-long arrays at one or more levels
NCOHDA, NCOHDI	oddh.dat, oddh.dic	OD/OH lat-long arrays	Instantaneous and time-averaged lat-long arrays at one or more levels

Table 3.5: DRS output files, their filenames and contents. These filenames must not exist prior to execution of SUMO.

3.11 NetCDF output files and their contents

NOTE THAT NetCDF OUTPUT IS NOT YET AVAILABLE FROM SUMO.

It must be noted that the diagnostics that may be output with the NetCDF option set to `.TRUE.` are only a subset (currently lat-long output on specified surfaces, and zonal-means) of those available in UTF format, since it is envisaged that the NetCDF-user will wish to subsequently sub-sample or manipulate the NetCDF data interactively by means of a graphics package.

Table 3.6 illustrates the NetCDF filenames that are generated internally by SUMO, together with their current contents. This table should be used as a reference to the NetCDF-formatted diagnostics that may be produced by SUMO.

NetCDF filenames	File contents	Description
sumosg.nc	SG/GH lat.-long. arrays, SG/GZ zonal-mean arrays	Instantaneous 2-D fields at different times, and time-averaged 2-D fields.
sumotf.nc	TF/FH lat.-long. arrays, TF/FZ zonal-mean arrays	Instantaneous 2-D fields at different times, and time-averaged 2-D fields.
sumood.nc	OD/OH lat.-long. arrays, OD/OZ zonal-mean arrays	Instantaneous 2-D fields at different times, and time-averaged 2-D fields.
sumozf.nc	ZF zonal arrays	Instantaneous 2-D fields at different times, and time-averaged 2-D fields.
sumoxd.nc	XD zonal arrays	Instantaneous 2-D fields at different times, and time-averaged 2-D fields.

Table 3.6: NetCDF output filenames and contents. These filenames must not exist prior to execution of SUMO.

3.12 Calendars

SUMO uses 'day numbers' internally, as a means of keeping track of time. Beginning with version 1.2, a 30-day calendar (ie. 360 days per year) is assumed to be appropriate for data output by the climate version of the UM, while a true calendar (365 or 366 days per year) is assumed for the UARS assimilated data. These defaults can be overwritten (and it is better to always do so) by means of the variable `L30DAY` in the MODELCONSTANTS data block.

4 OR diagnostics

OROGRAPHIC OUTPUT IS DISABLED AT PRESENT - THIS SECTION SHOULD THEREFORE BE SKIPPED.

4.1 Introduction

4.2 Implementation description

4.3 Summary of options available

5 Calculation of gridpoint fields

5.1 Introduction

This section describes the three-dimensional (and surface) fields computed by the diagnostics program in order to generate the SG diagnostics.

In view of the number and size of the three-dimensional arrays, all gridpoint arrays are stored in memory for just one latitude pair at a time. Gridpoint arrays are saved 'permanently' by copying their contents to disk storage using direct access scratch files. The array values for the 'current' latitude can be found in the following common blocks

- `/COMGRM/` – vertical gradient and interpolation arrays,
- `/COMMSK/` – orographic masks and flags,
- `/COMPRL/` – pressure-related arrays and surface gridpoint values,
- `/GRIDP/` – gridpoint data at a single latitude,

Common block `/COMNGP/` contains record pointer information enabling the program to easily access the gridpoint arrays held in direct access scratch files.

5.2 Implementation description

Figure 5.1 shows the principal routines used to calculate (and store on scratch files) the gridpoint fields.

The gridpoint fields are computed if at least one of the requested diagnostics needs to make use of some/all of these fields. The appropriate logical variables will be set according to inputs read into the program in routine `DATAIN`. The direct access scratch files required to store the three-dimensional fields are specified in `CHANNL`, with pointers to the initial record numbers for each field being set in routine `DACNST`. Routine `CHKSG` is used to check the validity of the user-input data to the program; `LDSG` may be set to `.FALSE.` in this routine if the analysis of gridpoint fields is inappropriate.

5.2.1 Derivation of gridpoint fields

The computation of the gridpoint fields begins with the reading of the input fields using routine `RDDATA`. The data is read into SUMO on the Arakawa 'B'-grid, with the three velocity components (u, v and ω – the latter in pressure coordinates) held at the same gridpoints. Currently data for the u, v, ω, geopotential height (h), specific humidity (q) and temperature (T) fields are read in via `RDDATA`, along with the pressure field. If SUMO detects that one or more of these fields is not stored in the input field dataset, then the field will be preset to a mask value (see routine `GFILLM`).

Routines `GLOBALE` and `GLOBALP` control the calculation of vorticity, divergence and potential vorticity (P) (based on code supplied by the UK Meteorological Office) on input surfaces/levels:

$$\zeta = f + \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}$$

$$D = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}$$

$$P = -g\zeta\frac{\partial \theta}{\partial p}$$

The streamfunction and velocity potential on input surfaces are then computed by the method outlined in section 5.2.5.

Subroutine `SGFLUX` is the routine that controls the production of gridpoint fields, with routines `GRMULT` to `INTPV`, along with the routine `WRITSG`, being called latitude-by-latitude. In `GRMULT`, T and h are interpolated (using horizontal bi-linear interpolation) on to the velocity gridpoints.

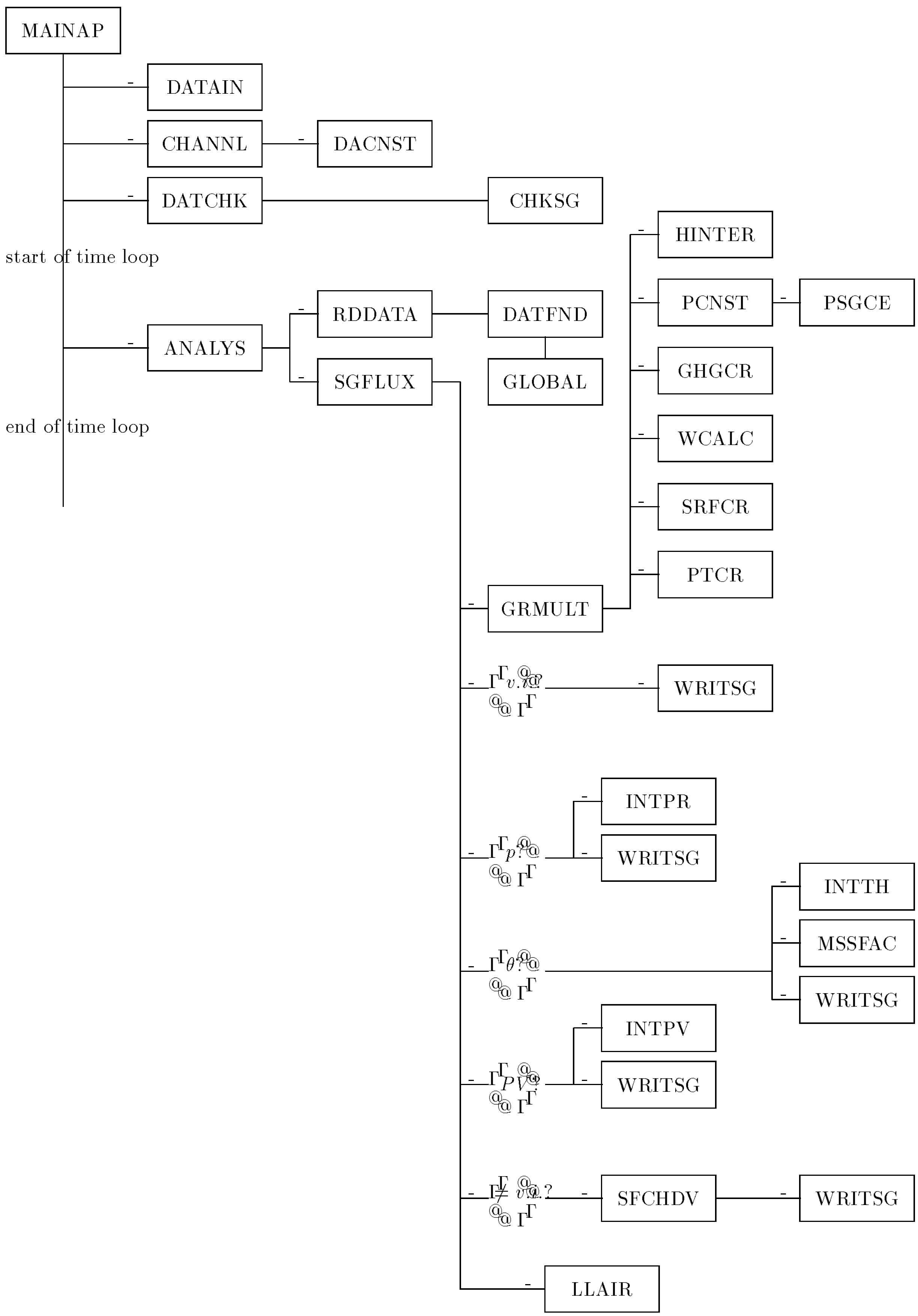

Figure 5.1: Flow chart showing the principal subroutines used to compute the gridpoint fields. If *v.i.* is `.TRUE.` then vertical interpolation is required.

Once the 'basic' gridpoint fields have been determined, routine GRMULT controls the production of further, derived, gridpoint fields. Within GRMULT, the land/air mask is specified (currently set to unity at all gridpoints since it the orography is currently not used explicitly). Surface pressure is currently unavailable and is set to a mask value.

Routine PSGCE computes the half-level pressures ($p_{k+\frac{1}{2}}$) according to

$$\left.\begin{array}{l} p_{\frac{1}{2}} = 0 \\ p_{k+\frac{1}{2}} = \frac{1}{2}\left(p_k + p_{k+1}\right) \quad \text{for} \quad k = 1, N_L - 1 \\ p_{N_L+\frac{1}{2}} = \text{undefined} \end{array}\right\} \tag{5.1}$$

Here, N_L is the number of full-levels. The Montgomery streamfunction is defined by

$$M_\psi = \phi + C_p T \tag{5.2}$$

Routine WCALC is used to compute w, the height-coordinate vertical velocity. From the hydrostatic equation,

$$\frac{\partial p}{\partial z} = -\rho g$$

and

$$p = \rho R_d T$$

we obtain

$$w = -\frac{\omega R_d T}{p g} \tag{5.3}$$

The vertical integral of the water vapour (precipitable water content, W) is determined in routine VINTQ according to

$$W = \frac{1}{g}\sum_{k=1}^{N_L} q(p_{k+\frac{1}{2}} - p_{k-\frac{1}{2}})$$

Subroutine SRFCR computes various surface and low-level fields for use when extrapolating below the ground. Currently, surface temperature, surface pressure, surface potential temperature are set to a mask value, with surface ω being defined at identically zero. Surface potential vorticity is currently not computed. Finally, potential temperature (θ) is computed according to

$$\theta = T\left(\frac{p_0}{p}\right)^{R_d/C_p} \tag{5.4}$$

The diabatic heating (H) is computed on model levels (if the input data is specified on model levels) in routine DIABTH according to the equation

$$H = \frac{\partial \theta}{\partial t} + \underline{v}.\underline{\nabla}\theta + \dot{\eta}\,\frac{\partial \theta}{\partial \eta}$$

where $\dot{\eta}$ is computed using the vertical integral of the continuity equation. The term $\frac{\partial \theta}{\partial t}$ is evaluated using a backward time-difference, ie.

$$\frac{\partial \theta}{\partial t}^n = \frac{\theta^n - \theta^{n-1}}{t^n - t^{n-1}}$$

where n is the time-sampling counter. When the input data is specified on pressure levels, the diabatic heating (H) is computed according to

$$H = \frac{\partial \theta}{\partial t} + \underline{v}.\underline{\nabla}\theta + \omega\frac{\partial \theta}{\partial p}$$

and $\frac{\partial \theta}{\partial t}$ is specified as above.

In both cases, the user must ensure, consequently, that upper air data at one sampling time period prior to the initial diagnostic time is made available to SUMO.

The gridpoint fields having now been computed for the current latitude, they are then written out to a direct access scratch file for later use (see routine WRITSG).

5.2.2 Interpolation to pressure surfaces

This section is only applicable if the user has requested that output from the diagnostic program be evaluated on surfaces of constant pressure, that differ to those on which the input data fields are defined.

The pressure levels to which the gridpoint data is to be interpolated are computed in subroutine `INIPLV`. By default, these levels are set equal to the input pressure surface values, such that one output pressure level is produced for each input pressure level. These default values are then merged with the user-supplied levels (variable `OUTLEV`) in the following manner: the `OUTLEV` values are sorted into order of increasing pressure, then added to the array of default values one-by-one, each time removing the remaining default p value closest to the `OUTLEV` value to be added.

Routine `INTPR` is the controlling routine for the interpolation of gridpoint fields from input to output pressure surfaces, interpolation being performed for one latitude of data values at a time. Interpolation is linear in $\ln p$ between adjacent input pressure levels. Thus for a variable $X(p)$ this gives

$$X(p) = X(p_k) + \frac{\ln \frac{p}{p_k}}{\ln \frac{p_{k+1}}{p_k}}[X(p_{k+1}) - X(p_k)], \;\; p_k < p < p_{k+1} \tag{5.5}$$

where the subscript k is a counter of the input data-levels.

Various forms of extrapolation could be used above the topmost input-level, and below the lowermost input-level — see routine `INTERP`; currently, linear extrapolation is applied.

This interpolation to pressure surfaces is carried out for the following gridpoint fields: geopotential height, ω, u, v, temperature, specific humidity, diabatic heating, w and potential vorticity. Potential temperature is then computed from the pressure and interpolated temperature, and the Montgomery streamfunction is re-computed using the geopotential height and the temperature on the pressure surfaces. Finally in routine `INTPR`, the land/air mask array is determined using the surface pressure field (when available).

The pressure level and surface (or single level) fields in `/GRIDP/` are then written out to direct access scratch files, using routines `WRITSG` and `WRITTR`.

At this stage, the pressure level streamfunction, velocity potential, divergence and relative vorticity on the output pressure levels have not been computed. This is done when gridpoint velocity information from all latitude pairs is available — see section 5.2.5.

5.2.3 Interpolation to potential temperature surfaces

This section is only applicable if the user has requested that output from the diagnostic program be evaluated on surfaces of constant potential temperature.

The potential temperature levels to which the gridpoint data is to be interpolated are computed in subroutine `INTHLV`. By default, when the input data to SUMO is on pressure levels, these levels are defined according to

$$\theta_k = \begin{cases} 280 + \frac{20}{500}(1000 - p_{in}) & p_{in} \geq 500.0 \\ 300 + \frac{120}{400}(500 - p_{in}) & 500.0 > p_{in} \geq 100.0 \\ 420 + \frac{100}{50}(100 - p_{in}) & 100.0 > p_{in} \geq 50.0 \\ 520 + \frac{320}{40}(50 - p_{in}) & 50.0 > p_{in} \geq 10.0 \\ 840 + \frac{1060}{9}(10 - p_{in}) & p_{in} < 10.0 \end{cases}$$

where p_{in} are the input pressure levels.

These default values are then merged with the user-supplied levels (variable `OUTLEV`) in the following manner: the `OUTLEV` values are sorted into order of decreasing potential temperature, then added to the array of default values one-by-one, each time removing the remaining default θ value closest to the `OUTLEV` value to be added.

Routine `INTTH` is the controlling routine for the interpolation of gridpoint fields from input levels to potential temperature surfaces, interpolation being performed for one latitude pair of data values at a time. Interpolation is linear in θ between adjacent input full levels. Thus for a variable $X(\theta)$ this gives

$$X(\theta) = X(\theta_k) + \frac{\theta - \theta_k}{\theta_{k+1} - \theta_k}[X(\theta_{k+1}) - X(\theta_k)], \;\; \theta_k \leq \theta < \theta_{k+1} \tag{5.6}$$

where the subscript k is a counter of the input data-levels.

Note that the interpolation is carried out from the uppermost level downwards; if a profile is multi-valued in θ then the field value corresponding to the upper θ location will be selected.

Routine `INTTH` also calls routine `DPDTHG` which computes that isentropic mass-weighting function σ according to

$$\sigma = -\frac{1}{g}\frac{\partial p}{\partial \theta}, \tag{5.7}$$

this field being evaluated using data on the input data levels, with the σ value resulting from the use of θ_k and θ_{k+1} being assumed to be known at potential temperature $0.5 \times (\theta_k + \theta_{k+1})$. σ at these θ points is then interpolated (again, linearly in θ) to the required output isentropic levels.

Various forms of extrapolation are used above the topmost full-level, and below the lowermost full-level — see routine `INTERT` for complete details. For many fields, it should be noted that gridpoint values beneath the lower boundary (if this location is known) are set to a large (masking) value, since isentropic-surface diagnostics frequently make use of either the land/air mask, or else use variables weighted by σ.

In addition to σ, the interpolation to isentropic surfaces is carried out for the following gridpoint fields: geopotential height, ω, u, v, temperature, specific humidity, diabatic heating, w and potential vorticity. Pressure is then computed from the potential temperature and interpolated temperature, and the Montgomery streamfunction is re-computed using the geopotential height and the temperature on the θ surfaces. Finally in routine `INTTH`, the land/air mask array is determined using the surface potential temperature field.

Routine `MSSFAC` allows the user to weight (ie multiply) the gridpoint fields on isentropic surfaces by σ. Currently, the following variables are multiplied by σ provided that `LMASSW=.TRUE.`: U, V, ω, temperature, geopotential height, specific humidity, diabatic heating, potential vorticity, w and the Montgomery streamfunction. The inclusion in this list of U and V has implications for the computation of the streamfunction, velocity potential, divergence and relative vorticity on isentropic surfaces (see section 5.2.5).

The θ-level and surface (or single level) fields in `/GRIDP/` are then written out to direct access scratch files, using routine `WRITSG`.

At this stage, the isentropic level streamfunction, velocity potential, divergence and relative vorticity have not been computed. This is done when gridpoint velocity information from all latitude pairs is available — see section 5.2.5.

5.2.4 Interpolation to potential vorticity surfaces

This section is only applicable if the user has requested that output from the diagnostic program be evaluated on surfaces of constant potential vorticity.

The potential vorticity levels to which the gridpoint data is to be interpolated are computed in subroutine `INPVLV`. By default, these levels are set up such that level `JPNL` is assigned a value of 0.5 pvu; the lowest `JPNL/3` levels are spaced at intervals of 0.5 pvu; the middle `JPNL/3` levels are spaced at intervals of 5.0 pvu; and the upper `JPNL/3` levels are spaced at intervals of 10.0 pvu. One potential vorticity level is produced for each full input data-level. These default values are then merged with the user-supplied levels (variable `OUTLEV`) in the following manner: the `OUTLEV` values are sorted into order of decreasing potential vorticity, then added to the array of default values one-by-one, each time removing the remaining default PV value closest to the `OUTLEV` value to be added.

Routine `INTPV` is the controlling routine for the interpolation of gridpoint fields from input surfaces to potential vorticity surfaces, interpolation being performed for one latitude pair of data values at a time. Interpolation is linear in potential vorticity between adjacent input full levels. Thus for a variable $X(P)$ this gives

$$X(P) = X(P_k) + \frac{P - P_k}{P_{k+1} - P_k}[X(P_{k+1}) - X(P_k)], \quad P_k \le P < P_{k+1}$$

where the subscript k is a counter of the input data-levels and P denotes potential vorticity.

Note that the interpolation is carried out from the uppermost level downwards; if a profile is multi-valued in potential vorticity then the field value corresponding to the upper potential vorticity location will be selected. Also, the interpolation is only performed using data on input data-levels between which the magnitude potential vorticity increases with height.

If the chosen output surface value of P is greater (less) than the values of potential vorticity on all input data surfaces surfaces at that location in the northern (southern) hemisphere, then the field value that is assigned to that output surface is a copy of that field value at the top full input data-level. (This follows the assumption that potential vorticity generally increases (decreases) with increasing height in the northern (southern) hemisphere.) Similarly, if the chosen output surface value of P is less (greater) than the values of potential vorticity on all input data surfaces at that location in the northern (southern) hemisphere, then the field value that is assigned to that output surface is a copy of that field value at the bottom full input data-level.

No extrapolation is performed above/below the extreme full input data-levels — see routine `INTERE` for complete details.

Routine `INTPV` also defines a constant `PVLATO` (set to 5 degrees by default) which specifies the latitude range (`PVLATO`°N to `-PVLATO`°S) in which no interpolation is performed; instead, a mask value is applied to field values on potential vorticity surfaces in this region. This is because of possible complications that may arise when positive (negative) valued potential vorticity is entrained south (north) of the equator. The user may, however, reset this value of `PVLATO` within the SUMO code. If `PVLATO` is set such that the fields consist of masked values on at least one latitude, then the fields of streamfunction, divergence, vorticity and velocity potential are undefined over the entire globe.

The interpolation to isentropic surfaces is carried out for the following gridpoint fields: geopotential height, ω, U, V, temperature, specific humidity, diabatic heating, vertical velocity, and pressure. Potential temperature is then computed from the interpolated pressure and temperature, and the Montgomery streamfunction is re-computed using the geopotential height and the temperature on the potential vorticity surfaces. Finally in routine `INTPV`, the land/air mask array is set to unity everywhere (this may be changed at a later date).

The potential vorticity-level and surface (or single level) fields in `/GRIDP/` are then written out to direct access scratch files, using routines `WRITSG`.

At this stage, the isentropic level streamfunction, divergence, vorticity and velocity potential have not been computed. This is done when gridpoint velocity information from all latitude pairs is available — see section 5.2.5.

5.2.5 Globally-computed fields on non-input levels

Routine `SFCHDV` is used to compute the streamfunction, divergence, vorticity and velocity potential on non-input levels. At version 1.3 of SUMO, the calculation of streamfunction and velocity potential is done by means of a bugfix — at version 1.4 it is part of the SUMO library code.

Note that the form of all four fields at locations below the ground surface may be critically dependent upon the way u and v are extrapolated below the surface.

The vorticity and divergence are computed according to the equations given in section 5.2.1

The streamfunction and velocity potential are derived using the relative vorticity and divergence (respectively) as a starting point. These starting fields are then transformed to spectral space at which point ∇^{-2} of the field is computed to produce spectral streamfunction and velocity potential. Finally, the inverse transformation back to gridpoint coordinates is carried out. SUMO uses code based on that from the UK Meteorological Office PP-Package, which follows the ideas of Machenhauer and Daley (1972) and Haseler (1979).

Note that the method requires the one-off calculation of potentially large work files (which are slow to generate). These files are dependent upon the gridpoint resolution and the spectral truncation being used. If a known model/data configuration is likely to be re-used, it may be best to save these workfiles for subsequent re-use (channels `NCNS2G` and `NCNG2S`).

5.2.6 Land/air masks for horizontal averaging

The final routine called by `SGFLUX` is `LLAIR`. The latter routine sets up horizontal arrays of land/air masks that are used when horizontal-averaging or weighting of arrays is required. Currently, these masks are used during the following procedures:

- creation of west-east averaged values between two specified latitudes (for gridpoint fields),
- creation of north-south averaged values between two specified latitudes (for gridpoint fields).

Currently, the mask is only used as an input argument to routine `BOXAV`.

If output from the diagnostic job is required on η levels, then all gridpoints are located above the earth's surface, and thus the mask is set to unity everywhere.

If output from the job is required on pressure levels then, depending upon the pressure levels specified by the user, some of the gridpoint fields will lie beneath the earth's surface. Consequently, the mask should be a mixture of values in the range 0 to 1. However, to ensure compatibility with 'standard' treatment of data on pressure surfaces, the mask is currently set to unity everywhere.

Likewise, when output from the job is required on surfaces of constant potential vorticity, the mask is currently set to unity everywhere except in the equatorial band (if it exists) where no interpolation to potential vorticity surfaces is performed.

Only in the case of isentropic output surfaces is the mask value currently set to values in the range 0 to 1.

Note that for instantaneous masks, values of either 0 or 1 apply. For time averaged masks, values in the range 0 to 1 apply, since the mask is then the average of all the instantaneous values at the given gridpoint on the given surface.

6 SG diagnostics

6.1 Introduction

The SG diagnostics permit the production of diagnostics pertaining to 'basic' gridpoint fields. Instantaneous, time averaged and time series diagnostics can be produced, and three-dimensional gridpoint arrays can be copied to an output file for use in other packages. (In the latter case, it is easy for the user to specify the file format.) Various slices (both vertical and/or horizontal) of the fields can be extracted, together with global-type averages (eg. zonal means) or local, regional averages. It is also possible to output files in PP-format containing running-mean values of fields that were present in the input data files.

For a complete summary of the options available, see section 2.5.3. Diagnostics can be produced on eta levels, isobaric surfaces, potential vorticity surfaces or on isentropic surfaces. Table 6.1 illustrates the fields that can be diagnosed by the SG diagnostics.

Field	units	Field code	
Streamfunction	m^2s^{-1}	1	$*$
Relative vorticity	s^{-1}	2	$*$
Divergence	s^{-1}	3	$*$
Omega	mb hr^{-1}	4	$*$
Zonal wind	ms^{-1}	5	$*$
Meridional wind	ms^{-1}	6	$*$
Temperature (T)	°C	7	$*$
Geopotential height (z)	m	8	$*$
Specific humidity	g kg^{-1}	9	$*$
Diabatic heating	K day^{-1}	10	$*$
Velocity potential	m^2s^{-1}	11	$*$
Pressure (p)	mb	12	
Potential temperature (θ) — replaced by $(-1/g)\frac{\partial P}{\partial \theta}$ on isentropic surfaces	K	13	
Ertel potential vorticity	pvu	14	$*$
Surface pressure	mb	15	
MSL pressure	mb	16	
Vertical velocity (w)	ms^{-1}	17	$*$
Temperature	K	18	$*$
$g.z + C_p.T$	J kg^{-1}	19	$*$
Surface potential temperature	K	20	
Land/air mask (=1 if gridpoint located on or above surface, else 0)		21	
Column water amount	kgm^{-2}	22	

Table 6.1: Three-dimensional and surface fields available via the SG diagnostics. Fields marked $*$ are multiplied by $(-1/g)\frac{\partial P}{\partial \theta}$ if output is requested on isentropic surfaces and if the mass-weighting switch `LMASSW` is set to `.TRUE.`; in this case the fields are produced with MKS units. Note, however, that the streamfunction, relative vorticity, divergence and velocity potential fields are computed from the mass-weighted horizontal velocity fields in this case.

6.2 Implementation description

Figure 6.1 shows the principal routines used to perform the SG diagnostics. The user's input is read into the program in routine `DATAIN` and then checked by routines `DATCHK` and `CHKSG`; the latter calls routines to check the input relating to horizontal surface (`CHKGH`), zonal (`CHKGZ`), meridional (`CHKGM`), profile (`CHKGP`) and three-dimensional output (`CHKG3`) diagnostics.

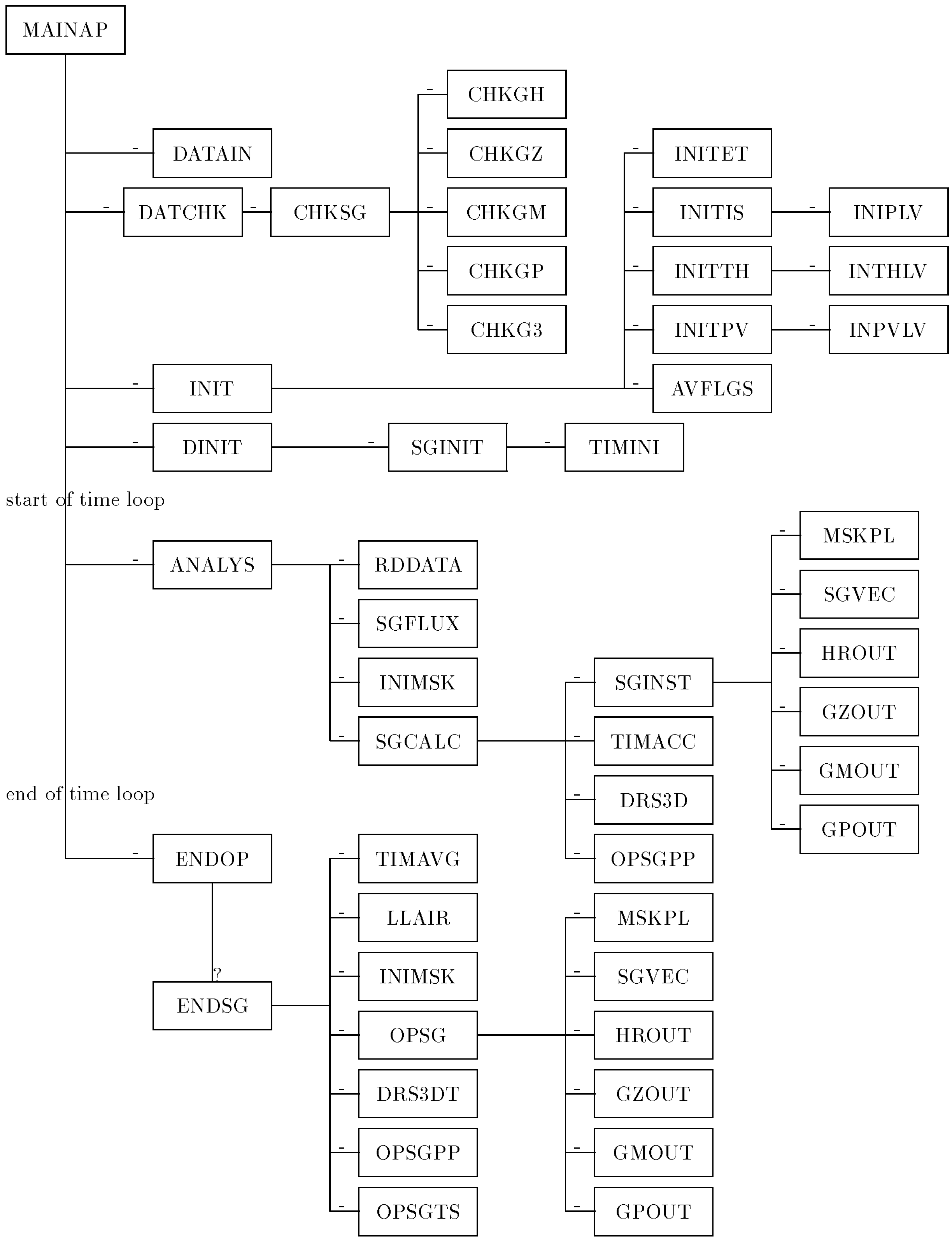

Figure 6.1: Flow chart showing the principal subroutines used to perform the SG diagnostics.

Routine INIT computes non-physical constants and arrays required by the program, and calls routines that set up output surface-type dependent arrays. In particular, INIPLV, INTHLV and INPVLV determine the non-η output levels to which the gridpoint fields may be interpolated when output on non-eta levels has been requested. (See the source code and sections 5.2.2 to 5.2.4 of this document for further details.)

Routine DINIT controls the initialisation of constants for the various diagnostics. When SG-type diagnostics are requested, it calls SGINIT, a routine that sets up the array names and diagnostic constants for the fields listed in Table 6.1. Contour intervals for these fields (to be passed to the UTF, if this option is selected) and ordered logical arrays to control the production of printed and plotted output and wind-vector information, are set up. Routine TIMINI initialises the scratch files required by the time averaging routines (if time averaging of the SG fields has been requested — arrays on these scratch files that will hold accumulated gridpoint values are preset to zero).

ANALYS is the routine that controls the incrementing of the time loop, and the calculations at each timestep. RDDATA controls the reading of the input field data, and SGFLUX computes the gridpoint fields at each timestep — see section 5.

Routine INIMSK computes the current orographic masks required if the user has requested UTF output on pressure surfaces, and has set any of the mask variables in the OR data block (see section 4) to .TRUE.. Note that currently UTF mask arrays are only produced when output on pressure surfaces has been requested — due to restrictions defined by the UTF1.4 specification. Also, the diagnostic program will only produce masks for zonal- and meridional-mean fields; zonal and meridional masks at single longitudes or latitudes cannot be produced. Both these shortcoming should be overcome by a later release of this program.

SGCALC is the routine that controls the SG diagnostics at each timestep, calling in turn, SGINST (to produce instantaneous output, and to save data for the later production of time series), TIMACC (to accumulate data for the calculation of time averages) and OPSGPP (to save any of the SG arrays — see Table 6.1 — in the UKMO-PP format; replace this with a call to OPSG3D for user-specified data format and/or file-format), or DRS3D (to save the SG-type arrays in a DRS file).

To reduce the amount of i/o associated with the direct access files that store the gridpoint fields, each of the fields is read once-only by SGINST; the routines HROUT, GZOUT, GMOUT and GPOUT are then called in turn (if necessary) to compute any GH, GZ, GM or GP diagnostics, respectively. Routines MSKPL and SGVEC are called to write pressure UTF masks and wind vector components to the UTF, if so required.

It should be noted that immediately after each field is read into SGINST it is converted from MKS to meteorological units (see routine SGCONV).

At the end of the time loop, routine ENDSG controls the production of time averaged fields, and also of any time series. TIMAVG computes the time averages, including the effects of any user-requesting masking, and LLAIR (see section 5.2.6) calculates the land/air masks required by any horizontal averaging. Routine INIMSK computes the time averaged orographic masks for use with UTF output on pressure surfaces (see above).

OPSG controls the production of the time averaged SG diagnostics in the same manner as SGINST controls the instantaneous diagnostics. OPSGPP is used to save any time averaged SG fields in a user-specified data format and/or file format), DRS3DT saves the time-averaged SG-type fields in a DRS file, and OPSGTS creates and outputs any SG time series requested by the user.

6.3 Summary of options available

Tables 2.3 – 2.8 illustrate the options available to the user when setting up the program to compute the SG diagnostics. These are now explained in a little more detail, together with some of the possible pitfalls. Reference is made in this section to the Fortran namelists variables, rather than to the data block keywords. Use should be made of the data pre-processor, which will help to eradicate inconsistent input data, before running the diagnostic program.

When diagnosing the gridpoint fields, the user should ensure that appropriate values have been specified for the variables LMASSW, LAVMSK and LAVSIG (amongst others) since these can affect significantly the results that will be produced. Masking and weighting are explained more fully in section 6.4. By default mass-weighting and σ-weighted averages (both mainly applicable to isentropic diagnostics) are switched off, and the flag LAVMSK is set to .FALSE..

The DIAGNOSTICS keyword in the COMMON data block must be set to SG to enable the SG diagnostics, and the appropriate value must be set for output level type (character variable YTYPSF — the default is to produce output on model (η) levels). Attention is also drawn to the use of the OR data block to produce UTF masks when output on pressure surfaces is requested.

If the fields are to be vertically-interpolated, then array OUTLEV must be defined — unless the default output levels are to be used (see sections 5.2.2 to 5.2.4). While the default values for OUTLEV may be satisfactory if only vertical slices are to be diagnosed, they will probably be unsatisfactory for the production of diagnostics on latitude-longitude surfaces. As a general rule, it should be assumed that any non-eta output surfaces for which latitude-longitude diagnostics are needed should be specified by setting non-default values for OUTLEV — see section 2.5.1. The OUTLEV levels are also used to define the output surfaces to be used in the production of zonal- mean and meridional-mean time series.

The SG data block may be envisaged as being at the top of the SG diagnostics tree — see Fig. 2.1. This data block must be used to specify the type of spatial and temporal diagnostics required — by default zonal, time averaged diagnostics will be produced.

Also to be specified in this data block are the levels for which latitude-longitude diagnostics are needed; if the diagnostic output is on model (eta) levels, then NVSGHR should be filled with the integer η level numbers required (in the range 1 to N_L), else if the output is on non-eta levels, then VSGSFC should be filled with one (or more) values copied from those contained in the array OUTLEV.

If runningmean output corresponding to the input data fields is required, the user must ensure that the appropriate value is assigned to RUNNINGDAYS in the COMMON data block, and that RUNNINGMEAN is set to .TRUE. in the SG data block.

Entities in data block GH (Table 2.4) need only be specified if output on horizontal (latitude-longitude) surfaces is required.

The fields codes of the gridpoint fields (see Table 6.1) for which printed and/or plotted output is required should be specified using variables NGHPR and NGHPL respectively. The field codes may be specified in any order for each variable, and the total number of different fields selected for either type of output must be specified using NGH. By default, no fields are selected.

The contour intervals (to be passed to the UTF if appropriate) for the NGHPL fields should be specified in order corresponding to the field codes in NGHPL.

The UTF standard (version 1.4) allows for wind vector information to be included in the UTF in a manner that facilitates it's incorporation into selected plotted fields (see Blackburn *et al.* 1992). The user can take advantage of this facility by assigning to variable NGHVEC the field code values corresponding to the fields on which horizontal wind vectors will be plotted.

Logical variables LDGHSL, LDGHCT and LDGHED need to be set to .TRUE. to produce GH diagnostic output, while Hovmöller output can be enabled by specifying the latitude range required (variable VGHTS). LDGHCT can only be selected when output is on model (η) levels, provided that LDGHSL has been set to .TRUE..

Entities in data block GZ (Table 2.5) need only be specified if diagnostics in the zonal plane are required.

The fields codes of the gridpoint fields (see Table 6.1) for which printed and/or plotted output is required should be specified using variables NGZPR and NGZPL respectively. The field codes may be specified in any order for each variable, and the total number of different fields selected for either type of output must be specified using NGZ. By default, no fields are selected.

The contour intervals (to be passed to the UTF if appropriate) for the NGZPL fields should be specified in order corresponding to the field codes in NGZPL.

The UTF standard (version 1.4) allows for wind vector information to be included in the UTF in a manner that facilitates it's incorporation into selected plotted fields (see Blackburn *et al.* 1992). The user can take advantage of this facility by assigning to variable NGZVEC the field code values corresponding to the fields on which zonal wind vectors will be plotted. (Note: this facility is currently disabled for wind components in the zonal plane.)

Logical variables LDGZZM and LDGZTS need to be set to .TRUE. to produce GZ diagnostic output; the former will enable zonal- mean fields to be produced, while the latter will allow the production of zonal-mean time series on the surfaces specified by the variables NVSGHR or VSGSFC, as appropriate, provided that LDGZZM is set to .TRUE.. Longitude slices and west-east averages over a specified longitude range can be computed provided VGZNS and VGZNSA are defined.

Variables in data block GM (Table 2.6) need only be specified if diagnostics in the meridional plane are required.

The fields codes of the gridpoint fields (see Table 6.1) for which printed and/or plotted output is required should be specified using variables NGMPR and NGMPL respectively. The field codes may be specified in any order for each variable, and the total number of different fields selected for either type of output must be specified using NGM. By default, no fields are selected.

The contour intervals (to be passed to the UTF if appropriate) for the NGMPL fields should be specified in order corresponding to the field codes in NGMPL.

The UTF standard (version 1.4) allows for wind vector information to be included in the UTF in a manner that facilitates it's incorporation into selected plotted fields (see Blackburn *et al.* 1992). The user can take advantage of this facility by assigning to variable NGMVEC the field code values corresponding to the fields on which meridional wind vectors will be plotted. (Note: this facility is currently disabled for wind components in the meridional plane.)

Logical variables LDGMMM and LDGMTS need to be set to .TRUE. to produce GM diagnostic output; the former will enable meridional-mean fields to be produced, while the latter will allow the production of meridional-mean time series on the surfaces specified by the variables NVSGHR or VSGSFC, as appropriate, provided that LDGMMM is set to .TRUE.. Longitude slices and west-east averages over a specified longitude range can be computed provided VGMEW and VGMEWA are defined.

Variables in data block GP (Table 2.7) need only be specified if column profile diagnostics are required.

The fields codes of the gridpoint fields (see Table 6.1) for which printed and/or plotted output is required should be specified using variables NGPPR and NGPPL respectively. The field codes may be specified in any order for each variable, and the total number of different fields selected for either type of output must be specified using NGP. By default, no fields are selected.

The contour intervals (to be passed to the UTF if appropriate) for the NGPPL fields should be specified in order corresponding to the field codes in NGPPL. Note that these contour intervals will be used when time series of column profiles (or totals) are produced (in the event that LDGP and LDSGTS are both .TRUE..

To produce column profile and/or column total diagnostics (the latter only when output is requested on model (η) levels) requires that LDGPCP and/or LDGPCT be set to .TRUE.. By default, both are set to .FALSE.. The user must also specify the profile coordinates of up to JPGPPF locations on the globe where these diagnostics will be evaluated (array VGPPF), along with the number of coordinate pairs thus selected, NGPPF.

The final gridpoint data block that needs to be set up by the user is the G3 data block. This will allow the user to save the full fields (or subsets thereof) for use in other programs/packages.

If LASCII is .TRUE., then the output will be in the form of a UKMO-PP format file. This may be altered — see use of routine OPSG3D above. If LDRS is .TRUE., then a DRS format file will be produced. If LNET is .TRUE., then a NetCDF file will be produced. Note that setting either LDRS = .TRUE. or LNET = .TRUE. prevents the output of time-series data other than that in DRS or NetCDF format.

The user needs to specify values for the output channel for the data, the fields codes of the fields to be saved, and the total number of fields to be saved (see Table 2.8).

6.4 Use of σ and mass-weighted masks

This section describes the use of the land/air and sigma masks/weights in the diagnostic program. Data block COMMON allows the user to input values to three logical variables describing the weighting flags as follows:

- `LMASSW` – `.TRUE.` if variables to be are to be mass-weighted when diagnostic output is on θ-surfaces. In this case, the fields marked with an asterisk ($*$) in Table 6.1 are multiplied by $(-1/g)\frac{\partial p}{\partial \theta}(=\sigma)$.

- `LAVMSK` – `.TRUE.` if spatial and temporal averages are to be computed using a land/air mask weighting. Currently, this only applies to if diagnostic output is on isentropic surfaces.

- `LAVSIG` – `.TRUE.` if spatial and temporal averages are weighted by the isentropic mass factor (σ); this only applies to isentropic surface output.

It is, however, possible for these three flags to be set inconsistently by the user. For example, σ-weighted averages are not of much use if the fields themselves have already been weighted by σ, resulting in an average of the form

$$\overline{\sigma X}^{\sigma} = \frac{\sum \sigma.(\sigma X)}{\sum \sigma}$$

where X is the un-weighted field (eg velocity). Consequently, routine `AVGCHK` is used to eliminate these inconsistencies.

Within `AVGCHK`, the following sequence of checks (in the order shown) are carried out:

- Unless output is on isentropic surfaces, `LMASSW` is set to `.FALSE.`.

- `LAVSIG` is set to `.FALSE.` unless the output is on isentropic surfaces.

- Since land/air masks are not used currently when output is on either pressure or potential vorticity surfaces, `LAVMSK` is set to `.FALSE.` if output is on p-surfaces.

- If `LMASSW` and `LAVSIG` are both `.TRUE.` the program will eventually be aborted with an error message, since it cannot cope with σ-weighted averages of σ-weighted variables.

- The program will also be aborted if both `LAVMSK` and `LAVSIG` are `.TRUE.`, since it is not possible to use both σ-weighting and land/air-mask weighting when computing averages.

Routine `AVFLGS` subsequently assigns a value to the integer variable `NAVTYP` as follows (the variable be used throughout the program to define the type of averaging to be applied to the fields):

- `NAVTYP=1` – arithmetic mean over all data values, if `LAVMSK` and `LAVSIG` are both `.FALSE.`. Note that if some of these data values have been set to (very large) masking-type values, then the arithmetic mean will be meaningless.

- `NAVTYP=2` – land/air mask-weighted average, if `LAVMSK` is `.TRUE.` and `LAVSIG` is `.FALSE.`.

- `NAVTYP=3` – isentropic σ-weighted average, if `LAVMSK` is `.FALSE.` and `LAVSIG` is `.TRUE.`.

If `LMASSW` is `.TRUE.` then routine `MSSFAC` multiplies the appropriate gridpoint fields by σ. In this case, when obtaining the relative vorticity from the absolute vorticity in gridpoint space (eg routine `SGCONV`), the planetary vorticity is multiplied by σ. This results in a quantity that is not exactly the same as the curl of the $\sigma-$weighted velocity. **The user may wish to change this.**

6.5 Mathematical definitions

In this section the spatial interpolation and averaging schemes using to produce the SG diagnostics are outlined, along with an explanation of the method used to compute column totals. It should be noted that the exact format of the schemes is dependent upon the choice of masking (if any) to be applied during the calculations. The vertical interpolation schemes are described in sections 5.2.2 to 5.2.4.

6.5.1 Latitude-longitude diagnostics

Hovmöller time series are computed between latitudes ϕ_1 and ϕ_2 (Fortran variables `VGHTS(1)` and `VGHTS(2)`) over the 360 degree range of longitude, by computing the area-weighted average value of variable Φ in gridboxes along each line of longitude.

Now the area enclosed by latitudes ϕ_j and ϕ_{j+1} and longitudes λ_i and λ_{i+1} (A) is given by

$$A = (\sin\phi_{j+1} - \sin\phi_j)(\lambda_{i+1} - \lambda_i)R_e^2$$

where R_e is the earth's surface.

Thus the contribution to the Hovmöller time series bounded by longitudes λ_1 and λ_2 is given by

$$\overline{\Phi}^A = \frac{\sum_{i,j} a_{i,j}\Phi_{i,j}M_{i,j}^h}{\sum_{i,j} a_{i,j}M_{i,j}^h} \tag{6.1}$$

where $a_{i,j}$ is that area of the (i,j)'th gridbox enclosed with the region defined by ϕ_1, ϕ_2, λ_1 and λ_2 — computed in the same way as A — and $M_{i,j}^h$ is the land/air mask applicable to each gridpoint. The mask value is equal to unity if the field Φ is a surface field, or if the calculation is being carried out on a model η level. Otherwise, M^h is computed according to section 5.2.6 — in particular, for instantaneous fields on θ surfaces it takes the value 1 on or above the ground surface, and 0 below it. If the denominator is equal to 0, then the weighted average is set equal to a large positive number.

Mass-weighted column totals of a variable Φ over the entire global domain can only be computed if the user has requested that output from the diagnostics program be presented on model (η) levels; values of Φ are combined with the half-level pressures to compute column totals according to

$$\Phi_{Total} = \frac{1}{g}\sum_{k=1}^{N_L}\Phi_k(p_{k+\frac{1}{2}} - p_{k-\frac{1}{2}}) \tag{6.2}$$

Thus the dimensions of Φ_{Total} are equal to those of Φ multiplied by kg m^{-2}. (This is based upon work by Lucic (1991).) Note that column totals are only available for instantaneous fields.

Slices of data on latitude-longitude surfaces can be obtained either of the eddy field, or of the total field. The eddy field is defined as the total field minus the zonal mean, where the zonal mean is given by

$$\Phi^{zonal} = \frac{\sum_{i=1}^{\mathrm{MG}} M_i\Phi_i}{\sum_{i=1}^{\mathrm{MG}} M_i} \tag{6.3}$$

Here, M_i is defined according to Fortran variable `NAVTYP`, see section 6.4. In particular, M_i is equal to unity for surface fields, and for zonal means computed on model (eta) levels. The zonal mean of land/air masks and the σ mass-weighting function always employs a mask of unity at all gridpoints. If the denominator is zero, then the zonal mean is set to a very large value.

6.5.2 Zonal diagnostics

Zonal means are computed according to Eq. (6.3) above, and include a masking function, if appropriate. The zonal means are also output in the form of time series, for those levels where `LSGLEV` is `.TRUE.` (corresponding to the levels defined by `NVSGHR` or `VSGSFC` for non-η and η surfaces respectively).

Field slices at longitude `VGZNS` are calculated assuming that

$$\frac{\partial\Phi}{\partial\lambda} = \text{constant}$$

North-south slices of area-averaged fields between two specified longitudes λ_1 and λ_2 (Fortran variables `VGZNSA(1)` and `VGZNSA(2)`) for each latitude in the entire range of latitudes $+90$ to -90 degrees are computed using

$$\overline{\Phi}^{ns} = \frac{\sum_{i,j} a_{i,j}\Phi_{i,j}M^h_{i,j}}{\sum_{i,j} a_{i,j}M^h_{i,j}}$$

where $a_{i,j}$ is that area of the (i,j)'th gridbox enclosed with the region defined by ϕ_1, ϕ_1, λ_1 and λ_2 (where ϕ_1 and ϕ_2 are the latitude boundaries of a given gridbox) — computed in the same way as A in the previous section – and $M^h_{i,j}$ is the land/air mask applicable to each gridpoint. The mask value is equal to unity if the field Φ is a surface field, or if the calculation is being carried out on a model η level. Otherwise, M^h is computed according to section 5.2.6 — in particular, for instantaneous fields on θ surfaces it takes the value 1 on or above the ground surface, and 0 below it. If the denominator is equal to 0, then the weighted average is set equal to a large positive number.

6.5.3 Meridional diagnostics

Meridional mean fields are found according to Eqn. (6.1), with the areas $a_{i,j}$ being the gridbox areas summed from pole to pole for a given longitude. The mask $M^h_{i,j}$ are defined according to the Fortran variable `NAVTYP`, see section 6.4. In particular, $M^h_{i,j}$ is equal to unity for surface fields, and for meridional means computed on model (eta) levels. The meridional mean of land/air masks and the σ mass-weighting function always employs a mask of unity at all gridpoints. If the denominator is zero, then the meridional mean is set to a very large value.

The meridional means are also output in the form of time series, for those levels where `LSGLEV` is `.TRUE.` (corresponding to the levels defined by `NVSGHR` or `VSGSFC` for non-η and η surfaces respectively).

Field slices at longitude `VGMEW` are calculated assuming that

$$\frac{\partial\Phi}{\partial\sin\phi} = \text{constant}$$

West-east slices of area-averaged fields between two specified latitudes ϕ_1 and ϕ_2 (Fortran variables `VGMEWA(1)` and `VGMEWA(2)`) for each longitude in the entire range of longitudes $\lambda = 0$ to $\lambda = 360$ degrees are computed using

$$\overline{\Phi}^{we} = \frac{\sum_{i,j} a_{i,j}\Phi_{i,j}M^h_{i,j}}{\sum_{i,j} a_{i,j}M^h_{i,j}}$$

where $a_{i,j}$ is that area of the (i,j)'th gridbox enclosed with the region defined by ϕ_1, ϕ_2, λ_1 and λ_2 (where λ_1 and λ_2 are the longitude boundaries of a given gridbox) — computed in the same way as A in the section 6.5.1 – and $M^h_{i,j}$ is the land/air mask applicable to each gridpoint. The mask value is equal to unity if the field Φ is a surface field, or if the calculation is being carried out on a model η level. Otherwise, M^h is computed according to section 5.2.6 — in particular, for instantaneous fields on θ surfaces it takes the value 1 on or above the ground surface, and 0 below it. If the denominator is equal to 0, then the weighted average is set equal to a large positive number.

6.5.4 Profile diagnostics

The vertical profiles of Φ at the coordinates `VGPPF` are computed by horizontal interpolation of Φ at each of the N_L output levels, assuming that

$$\frac{\partial\Phi}{\partial\sin\phi} = \text{constant}$$

and

$$\frac{\partial\Phi}{\partial\lambda} = \text{constant}$$

Note that this differs to the approach used by Lucic (unpublished) in the Flux Program.

Mass-weighted column totals of Φ at the same locations can only be computed if the user has requested that output from the diagnostics program be presented on model (η) levels; interpolated values of Φ_{int} are combined with the half-level pressures to compute column totals according to

$$\Phi_{int,Total} = \frac{1}{g}\sum_{k=1}^{N_L} \Phi_{int,k}(p_{k+\frac{1}{2}} - p_{k-\frac{1}{2}}) \tag{6.4}$$

Both vertical profiles and column totals can be output as a time series. Note that column totals are only available for instantaneous fields.

6.5.5 Wind vectors

Routine `SGVEC` can currently only be used to output UTF wind vector information for latitude-longitude slices. For multi-level fields, horizontal wind information is output for the levels corresponding to those of the array Φ on which they will superimposed. For surface fields of Φ, horizontal wind vector information from the lowest output level is copied to the UTF.

Note that wind vector information at a given level (for a given time or time period) is copied only once to the UTf file. It is up to the user to merge together the correct Φ and wind information from the UTF when the data are subsequently plotted.

6.5.6 UTF pressure masks

Pressure level masking information is copied to the UTF via routine `MSKPL`. Currently, only latitude-longitude, zonal-mean and meridional-mean mask information is passed to the UTF. Any particular mask (for a given time or time period) will be output only once in a single UTF.

6.6 Time-averaging

The form of the time-averaging of the gridpoint fields is governed by the value of variable `NAVTYP` (see section 6.4). There are three types of time-averaging, defined as follows:

$$\overline{X}^t = \frac{1}{n}\sum_1^n X(t) \quad \text{(arithmetic average)} \tag{6.5}$$

$$\overline{X}^{\beta t} = \frac{\sum_1^n \beta(t)X(t)}{\sum_1^n \beta(t)} \quad \text{(land/air mask-weighted average)} \tag{6.6}$$

$$\overline{X}^{\sigma t} = \frac{\sum_1^n \sigma(t)X(t)}{\sum_1^n \sigma(t)} \quad (\sigma\text{-weighted average}) \tag{6.7}$$

where n is the number of samples of the variable $X(t)$ and β is the land/air mask (set to unity above the surface, and zero underground).

These three types of averaging correspond to `NAVTYP` being set to 1, 2 and 3 respectively. Note that time averages of σ and β are always computed using the first of these equations — essential when computing the denominator of the second and third of the equations.

6.7 Saving 3-D fields

Fortran routines (`OPSGPP`, `OPSG3D`, `DRS3D` and `DRS3DT`) are available for use as a template for those users wishing to save whole fields (or parts thereof) for later use in other packages. These routines can be edited to fit the user's requirements.

It is also possible to alter the form of the this entire file (written to Fortran channel `NCG3`) by means of the unicos `assign` command in, for example, the unicos shell script that runs the diagnostic job.

7 TF diagnostics

7.1 Introduction

The TF diagnostics permit the production of transient flux diagnostics. The equations defining these transient fluxes are specified by the user, and may involve any of the three-dimensional (thermo-)dynamical SG-type fields. Various slices (both vertical and/or horizontal) of the fields can be extracted, together with global-type averages (eg. zonal means) or local, regional averages.

For a complete summary of the options available, see section 2.5.4. The SG-type fields that may be used are those listed in Table 6.1 (excluding the surface pressure, MSL pressure, the surface potential temperature and the land/air mask).

The transient fluxes may be output in the form of the total flux, and also high- and low-pass components. The low-pass filter involves the summation of consecutive, non-overlapping n-day means; the high-pass filter removes the variances and covariances of these n-day means, and is computed as the difference between the total and the low-pass fluxes. The cut-off value n can be specified by the user.

The transient fluxes are computed using interpolated (thermo-)dynamical fields if the output is not specified to be on η surfaces.

7.2 Implementation description

Figure 7.1 shows the principal routines used to compute the TF diagnostics. The user's input is read into the program in routine `DATAIN` and then checked by routines `DATCHK` and `CHKTF`; the latter calls routines to check the input relating to horizontal surface (`CHKFH`), zonal (`CHKFZ`), meridional (`CHKFM`), profile (`CHKFP`) and three-dimensional output (`CHKF3`) diagnostics.

Routine `DINIT` controls the initialisation of constants for the various diagnostics. In particular, it calls `TFINIT`, a routine that sets up the array names and diagnostic constants for the transient flux types specified by the user. Contour intervals for these fields (to be passed to the UTF if appropriate) and ordered logical arrays to control the production of printed and plotted output and wind- vector information, are set up.

`ANALYS` is the routine that controls the incrementing of the time loop, and the calculations at each timestep. `SGFLUX` computes the gridpoint fields at each timestep — see section 5 — and calls `TFCONT`, which controls the step-by-step summations required to compute time-averages.

At the end of the time loop, routine `ENDTF` controls the production of the time averaged fields and any required output. `TFMEAN` computes the time averages, these being computed latitude-by-latitude in routine `TFLAT`, and stores them in a scratch file. Finally, `OPTF` controls the output of the transient fluxes.

7.3 Transient flux formulae

The SUMO transient flux diagnostics require the user to specify the formulae for the required fluxes. Note that there are no default formulae (see Table 2.9).

The general formulae for a flux (Φ) may be written in the form

$$\Phi = F_1\phi_1\phi_2 + F_2\phi_3\phi_4 \tag{7.1}$$

where F_1 and F_2 are numerical factors (either positive or negative) and the ϕ_i are three-dimensional (thermo-)dynamical fields. These four fields may or may not be different from each other.

Note that it is possible for the factor F_2 to be set to zero, and also for individual fields to be suppressed in the formula. Thus, each of the following are valid formulae capable of being processed by SUMO:

$$\Phi = 1.0\phi_1\phi_2 \tag{7.2}$$

$$\Phi = -1.0\phi_1\phi_2 + 1.0\phi_3 \tag{7.3}$$

$$\Phi = 4.0\phi_1 - 2.0\phi_3\phi_4 \tag{7.4}$$

$$\Phi = 1.0\phi_1\phi_1 + 1.0\phi_3\phi_3 \tag{7.5}$$

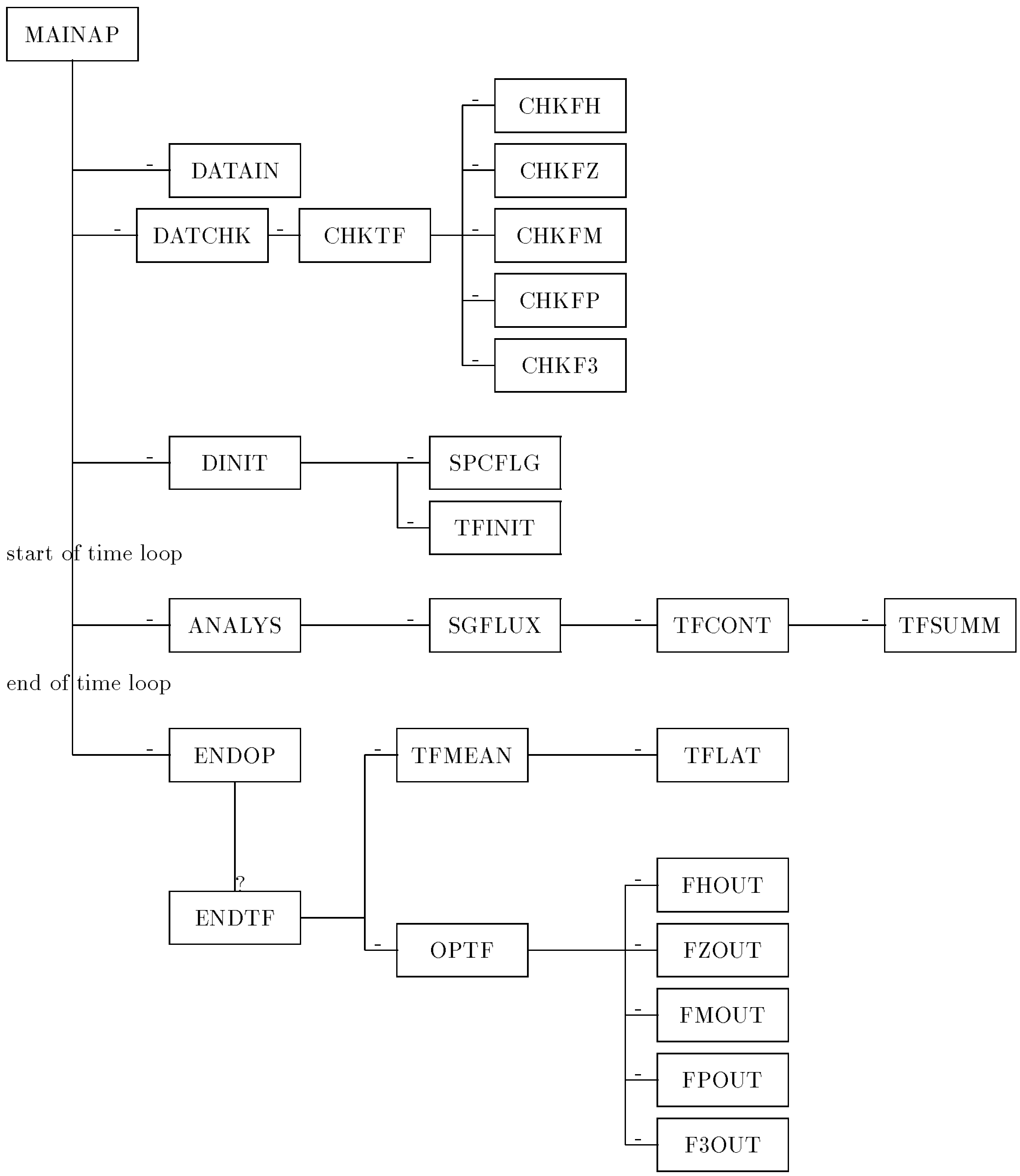

Figure 7.1: Flow chart showing the principal subroutines used to perform the TF diagnostics.

Whether or not these formulae are physically meaningful or not is open to question. It up to the user to ensure that any specified equations are, for example, dimensionally correct.

Formulae are entered (in upper-case only) into the program via the TF data block, where they are specified in the form

```
FORMULA=FACTOR1XFIELD1XFIELD2+FACTOR2XFIELD3XFIELD4
```

Multiple formulae are separated by commas. Here, FIELD1 ... FIELD4 are the SG field codes (see Table 6.1). Thus, specifying

```
FORMULA=1.0X6X6-1.0X5X5,
   -1.0X5X6+0.0X0X0,
   0.5X5X5+0.5X6X6,
   1.0X5X7,
   1.0X5X9
```

will result in the following transient fluxes being computed

$$\overline{v'v'} - \overline{u'u'}$$

$$-\overline{u'v'}$$

$$\frac{1}{2}\overline{u'u'} + \frac{1}{2}\overline{v'v'}$$

$$\overline{u'T'}$$

$$\overline{u'\mathrm{q}'}$$

Only one field formula is allowed per line of the input data file. Note that field codes 5, 6, 7 and 9 correspond to the zonal wind, meridional wind, temperature and specific humidity, respectively. The use of field code 0 results in no field being used at that point in the transient flux equation, while if the second term in the equation is omitted (or if `FACTOR2` is zero) then the resulting transient flux will contain only one term.

7.4 Summary of options available

Tables 2.9 – 2.14 illustrate the options available to the user when setting up the program to compute the TF diagnostics. These are now explained in a little more detail, together with some of the possible pitfalls. Reference is made in this section to the Fortran namelists variables, rather than to the data block keywords. Use should be made of the data pre-processor, which will help to eradicate inconsistent input data, before running the diagnostic program.

When computing the gridpoint fields, the user should ensure that appropriate values have been specified for the variable `LMASSW` since this can affect significantly the results that will be produced. Masking and weighting are explained more fully in section 6.4. When output is on isentropic surfaces field code 13 will refer to the quantity σ. When output is on isentropic surfaces and this mass-weighting factor is switched on all quantities appearing in the transient flux formulae are automatically multiplied by σ.

The `DIAGNOSTICS` keyword in the COMMON data block must be set to `TF` to enable the TF diagnostics, and the appropriate value must be set for output level type (character variable `YTYPSF` — the default is to produce output on model (η) levels).

If the fields are to be vertically-interpolated, then array `OUTLEV` must be defined — unless the default output levels are to be used (see sections 5.2.2 to 5.2.4). While the default values for `OUTLEV` may be satisfactory if only vertical slices are to be diagnosed, they will probably be unsatisfactory for the production of diagnostics on latitude-longitude surfaces. As a general rule, it should be assumed that any non-eta output surfaces for which latitude-longitude diagnostics are needed should be specified by setting non-default values for `OUTLEV` — see section 2.5.1.

Note that the number of times that the data is sampled to produce the TF diagnostics must be an integer multiple of the cut-off timescale specified in the TF data block. Thus for a 3.0 day cut-off timescale and

However, with a 18-hour data sampling frequency, valid start and end days would be 90.75 and 180.00. Thus would give a total of 120 samples, with 4 samples in each low-pass block.

The TF data block may be envisaged as being at the top of the TF diagnostics tree — see Fig. 2.1. This data block must be used to specify the type of spatial diagnostics required — by default zonal diagnostics will be produced.

Also to be specified in this data block are the levels for which latitude-longitude diagnostics are needed; if the diagnostic output is on model (eta) levels, then `NVTFHR` should be filled with the integer η level numbers required (in the range 1 to N_L), else if the output is on non-eta levels, then `VTFSFC` should be filled with one (or more) values copied from those contained in the array `OUTLEV`.

By default, `LTFTOT`, `LTFHP` and `LTFLP` are set to `.TRUE.` to enable total, high- and low-pass transient fluxes to be produced. These values may be altered, as can the value of `TFCUT` (3.0 by default) which defines the cut-off timescale between low-and high-pass components. `NTF` must be set equal to the number of formulae specified.

Entities in data block FH (Table 2.10) need only be specified if output on horizontal (latitude-longitude) surfaces is required.

The field codes of the transient fluxes for which printed and/or plotted output is required should be specified using variables `NFHPR` and `NFHPL` respectively — these are simply integer values corresponding to the position on the required transient flux in the `FORMULA` sequence. The field codes may be specified in any order for each variable. By default, no fields are selected.

The contour intervals (to be passed to the UTF if appropriate) for the `NFHPL` fields should be specified in the order corresponding to the field codes in `NFHPL`. Finally, logical variable `LDFHSL` must be set to `.TRUE.` to enable latitude-longitude output.

Entities in data block FZ (Table 2.11) need only be specified if diagnostics in the zonal plane are required.

The fields codes of the transient fluxes for which printed and/or plotted output is required should be specified using variables `NFZPR` and `NFZPL` respectively — these are simply integer values corresponding to the position on the required transient flux in the `FORMULA` sequence. The field codes may be specified in any order for each variable. By default, no fields are selected.

The contour intervals (to be passed to the UTF if appropriate) for the `NFZPL` fields should be specified in the order corresponding to the field codes in `NFZPL`. Logical variable `LDFZZM` needs to be set to `.TRUE.` to produce zonal-mean fields. Longitude slices and west-east averages over a specified longitude range can be computed provided `VFZNS` and `VFZNSA` are defined.

Variables in data block FM (Table 2.12) need only be specified if diagnostics in the meridional plane are required.

The fields codes of the transient fluxes for which printed and/or plotted output is required should be specified using variables `NFMPR` and `NFMPL` respectively — these are simply integer values corresponding to the position on the required transient flux in the `FORMULA` sequence. The field codes may be specified in any order for each variable. By default, no fields are selected.

The contour intervals (to be passed to the UTF if appropriate) for the `NFMPL` fields should be specified in the order corresponding to the field codes in `NFMPL`.

Logical variable `LDFMMM` needs to be set to `.TRUE.` to produce meridional-mean fields. Longitude slices and west-east averages over a specified longitude range can be computed provided `VFMEW` and `VFMEWA` are defined.

Variables in data block FP (Table 2.13) need only be specified if column profile diagnostics are required.

The fields codes of the transient fluxes for which printed and/or plotted output is required should be specified using variables `NFPPR` and `NFPPL` respectively — these are simply integer values corresponding to the position on the required transient flux in the `FORMULA` sequence. The field codes may be specified in any order for each variable. By default, no fields are selected.

To produce column profile diagnostics requires that `LDFPCP` be set to `.TRUE.`. The user must also specify the profile coordinates of up to `JPFPPF` locations on the globe where these diagnostics will be evaluated (array `VFPPF`), along with the number of coordinate pairs thus selected, `NFPPF`.

The final gridpoint data block that needs to be set up by the user is the F3 data block. This will allow the user to save the full transient flux arrays (or subsets thereof) for use in other programs/packages.

If `LASCII` is `.TRUE.`, then the output will be in the form of a fortran formatted file. If `LDRS` is `.TRUE.`, then a DRS format file will be produced.

The user needs to specify values for the output channel for the data and the fields codes of the fields to be saved (see Table 2.14).

7.5 Mathematical definitions

The filter was originally designed to be as similar as possible to a 2.5 to 6 day bandpass filter used by Blackmon et al. (1977), who showed that the transient eddies with such frequencies had properties that were associated with baroclinic instability. Further descriptions of the filter can be found in Lau et al. (1981) and in White (1983).

The program computes the total transient flux $\overline{u'\phi'}$ according to

$$\overline{u'\phi'} = \overline{u\phi} - \overline{u}\,\overline{\phi} \tag{7.6}$$

where the time average of a field χ(denoted by the overbar) is found according to

$$\overline{\chi} = \frac{\sum_{i=1}^{n} \chi_i M_i}{\sum_{i=1}^{n} M_i} \tag{7.7}$$

and where n represents the number of samples in the averaging period and M_i is the mask (one of β (the land-air mask), isentropic σ or unity (in the case when no mask is requested)). Samples are taken every `NFREQD` timesteps for the total transient flux. When no mask is requested, the equation reduces to

$$\overline{\chi} = \frac{1}{n}\sum_{i=1}^{n} \chi_i$$

The separation of the fluxes into low- and high-pass components requires the specification of a cut-off frequency, which by default in the diagnostic program is taken to be 3 days. This results in total of n_l ($= 3 \times$ timesteps per day/`NFREQD`) samples being used to compute the variance and covariance of successive, non-overlapping 3-day means, while n_s is the number of samples in the total sampling period. Note that n_l/n_s must be an integer.

The low pass transient flux (subscript L) of ϕ by u is found according to

$$\left(\overline{u'\phi'}\right)_L = \sum_{j=1}^{n_s/n_l} \left\{ N_T \cdot \frac{1}{N_T} \left(\sum_{i=1}^{n_l} u_{n_l(j-1)+i} M_{n_l(j-1)+i} \right) \cdot \frac{1}{N_T} \left(\sum_{i=1}^{n_l} \phi_{n_l(j-1)+i} M_{n_l(j-1)+i} \right) \right\} / \sum_{j=1}^{n_s} M_j - \overline{u}\,\overline{\phi} \tag{7.8}$$

(the final term being averaged over the total sampling period), with

$$N_T = \sum_{i=1}^{n_l} M_{n_l(j-1)+i}$$

When no mask is used, $N_T = n_l$ and

$$\left(\overline{u'\phi'}\right)_L = \frac{n_l}{n_s} \sum_{j=1}^{n_s/n_l} \left\{ \frac{1}{n_l} \left(\sum_{i=1}^{n_l} u_{n_l(j-1)+i} \right) \frac{1}{n_l} \left(\sum_{i=1}^{n_l} \phi_{n_l(j-1)+i} \right) \right\} - \overline{u}\,\overline{\phi}$$

The high-pass component is given by

$$\left(\overline{u'\phi'}\right)_H = \overline{u'\phi'} - \left(\overline{u'\phi'}\right)_L \tag{7.9}$$

Many of the mathematical definitions of the SG diagnostic calculations also apply to the TF diagnostics. The former are described in section 6.5. In this section only those calculations that differ from the SG-type counterparts, are are additions to the latter, are described. Note that no time-series output is currently available.

7.5.1 Latitude-longitude diagnostics

Slices of data on latitude-longitude surfaces are available only for the full transient flux field (i.e. the zonal mean cannot be removed).

7.5.2 Zonal diagnostics

Zonal means — see section 6.5.2.

Slices at longitude `VFZNS` — see section 6.5.2.

North-south slices of area-averaged transient fluxes between longitudes `VFZNSA(1)` and `VFZNSA(2)` — see section 6.5.2.

7.5.3 Meridional diagnostics

Meridional mean fields — see section 6.5.3.

Slices at longitude `VFMEW` — see section 6.5.3.

West-east slices of area-averaged transient fluxes between latitudes `VFMEWA(1)` and `VFMEWA(2)` — see section 6.5.3.

7.5.4 Profile diagnostics

Vertical transient flux profiles at coordinates `VFPPF` — see section 6.5.4.

7.6 Saving 3-D fields

A Fortran routine (`F3OUT`) is available for use as a template for those users wishing to save whole fields (or parts thereof) for later use in other packages. This routine can be edited to fit the user's requirements.

It is also possible to alter the form of the this entire file (written to Fortran channel `NCF3`) by means of the unicos `assign` command in, for example, the unicos shell script that runs the diagnostic job.

8 OD diagnostics

8.1 Introduction

The OD diagnostics control the production of certain diagnostic fields that are derived from those in the SG subset, and they permit the user to incorporate additional gridpoint diagnostics into SUMO. Instantaneous, time averaged and time series diagnostics can be produced, and three-dimensional gridpoint arrays can be copied to an output file for use in other packages. (In the latter case, it is easy for the user to specify the file format.) Various slices (both vertical and/or horizontal) of the fields can be extracted, together with global-type averages (eg. zonal means) or local, regional averages.

It should be noted that there is a major difference between the fields computed by the OD code and those computed using the SG code, namely that the OD diagnostics are computed using data on the SUMO output surfaces. Thus any input level data will already have been interpolated it the output is not to be presented on levels corresponding to the input data. This needs to be considered when computing fields (eg. geostrophic velocity components) whose formula may be dependent upon the type of surface on which the data is held.

For a complete summary of the options available, see section 2.5.5. Diagnostics can be produced on model (eta) levels, isobaric surfaces, isentropic surfaces or potential vorticity surfaces. Table 8.1 illustrates the fields that can be diagnosed by the OD diagnostics. The equivalent potential vorticity latitude diagnostics are based upon code written by David Lary; for a full explanation of these diagnostics users are referred to Lary *et. al.* (1994). Lary (1993) also gives an insight into equivalent potential vorticity latitude diagnostics.

Field	units	Field code	Surfaces on which the field is output			
			η	p	θ	PV
Relative humidity	%	1	yes	yes	yes	yes
$g.z + C_p.T + L_v.q$	J kg^{-1}	2	yes	yes	yes	yes
Wind speed $(= \sqrt{u^2+v^2})$	ms^{-1}	3	yes	yes	yes	yes
Equivalent PV latitude	deg	4	no	no	yes	no
Normalised PV area	[0-1]	5	no	no	yes	no
Geostrophic zonal wind (u_g)	ms^{-1}	6	yes	yes	yes	no$*$
Geostrophic meridional wind (v_g)	ms^{-1}	7	yes	yes	yes	no$*$
Rossby wave source (filtered) (S)	s^{-2}	8	no	yes	no	no
Equivalent potential temperature	K	9	yes	yes	yes	yes
Equivalent saturated potential temperature	K	10	yes	yes	yes	yes

Table 8.1: Three-dimensional fields available via the OD diagnostics. Fields that will be added in the future are denoted by $*$.

In isentropic coordinates, none of the fields are currently mass-weighted (i.e. multiplied by σ). Since meaningless results are liable to result if the SG fields (see Table 6.1) have been multiplied by σ, the user is advised not to combine this option with the OD diagnostics.

Users wishing to incorporate additional fields into SUMO, via the OD diagnostics, should read section 8.7.

8.2 Implementation description

Figure 8.1 shows the principal routines used to perform the OD diagnostics. The user's input is read into the program in routine `DATAIN` and then checked by routines `DATCHK` and `CHKOD`; the latter calls routines to check the input relating to horizontal surface (`CHKOH`), zonal (`CHKOZ`), meridional (`CHKOM`), profile (`CHKOP`) and three-dimensional output (`CHKO3`) diagnostics. Note that routine `CHKOD` currently switches off the OD diagnostics if the SUMO output is to be on isentropic surfaces and the mass-weighting switch has been turned on.

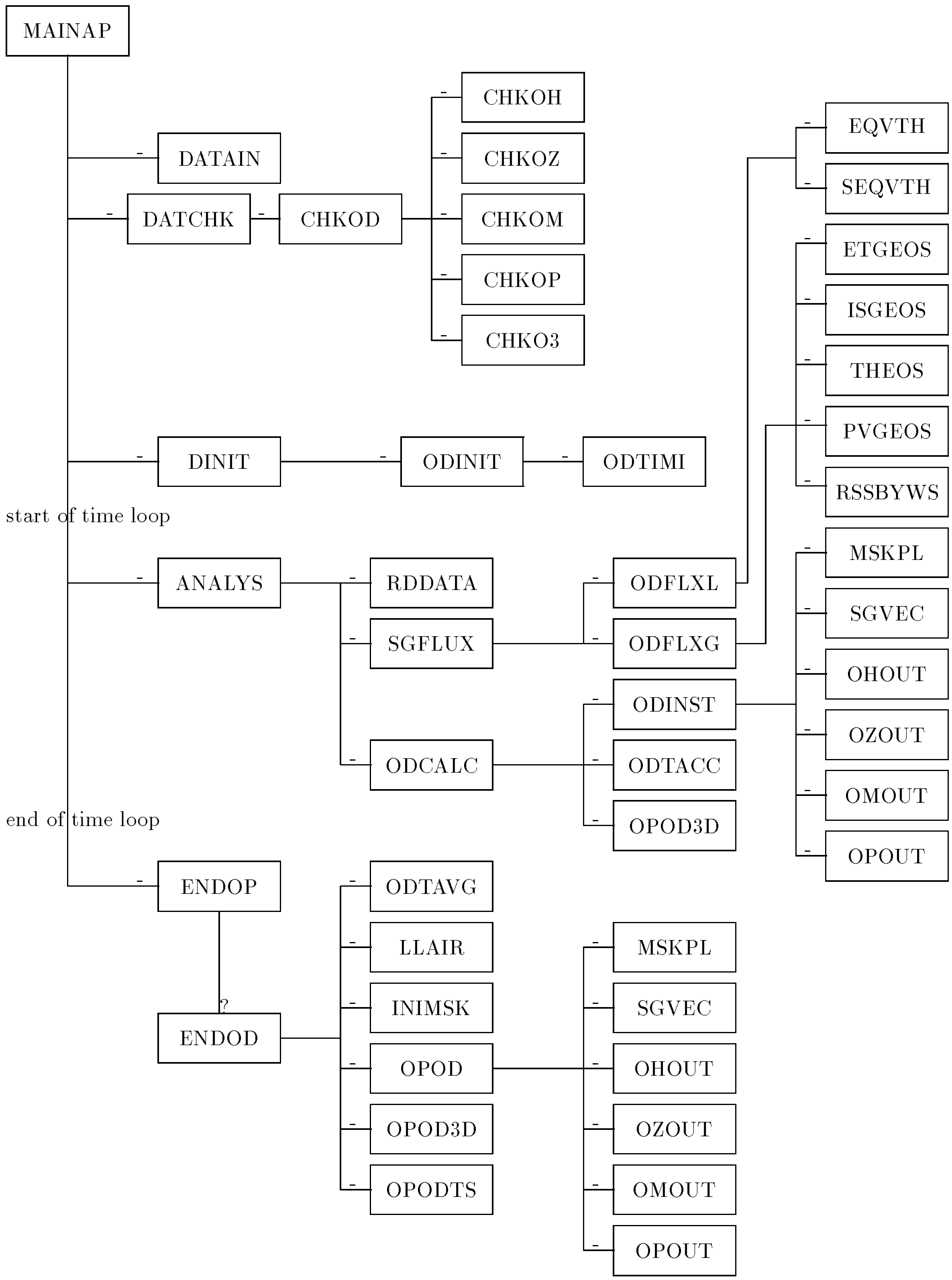

Figure 8.1: Flow chart showing the principal subroutines used to perform the OD diagnostics.

Routine DINIT controls the initialisation of constants for the various diagnostics. When OD-type diagnostics are requested, it calls ODINIT, a routine that sets up the array names and diagnostic constants for the fields listed in Table 8.1. Contour intervals for these fields (to be passed to the UTF if appropriate) and ordered logical arrays to control the production of printed and plotted output and wind-vector information, are set up. Routine ODTIMI initialises the scratch files required by the time averaging routines.

ANALYS is the routine that controls the incrementing of the time loop, and the calculations at each timestep. RDDATA controls the reading in of the input data fields, and SGFLUX computes the gridpoint fields at each timestep — see section 5. After any required vertical interpolation has been done, routine SGFLUX calls two routines which compute the instantaneous OD fields; ODFLXL computes fields that can be computed using SG fields at a single latitude (eg, those which require no differentiation of SG fields in the meridional direction), while ODFLXG computes fields that require a knowledge of global SG fields. The latter routine computes the geostrophic velocity components, and calls routines ETGEOS, ISGEOS, THGEOS and PVGEOS to perform calculations on model (eta) surfaces, pressure, potential temperature surfaces and potential vorticity surfaces, respectively. RSSBYWS computes the Rossby wave source field.

ODCALC is the routine that controls the output of OD diagnostics at each timestep, calling in turn, ODINST (to produce instantaneous output, and to save data for the later production of time series), ODTACC (to accumulate data for the calculation of time averages) and OPOD3D (to save any of the OD arrays — see Table 8.1 — in a user-specified data format and/or file-format).

To reduce the amount of i/o associated with the direct access files that store the gridpoint fields, each of the fields is read once-only by ODINST; the routines OHOUT, OZOUT, OMOUT and OPOUT are then called in turn (if necessary) to compute any OH, OZ, OM or OP diagnostics, respectively. Routines MSKPL and SGVEC are called to write pressure UTF masks and wind vector components to the UTF, if so required.

At the end of the time loop, routine ENDOD controls the production of time averaged fields, and also of any time series. ODTAVG computes the time averages, including the effects of any user- requesting masking, and LLAIR (see section 5.2.6) calculates the land/air masks required by any horizontal averaging. Routine INIMSK computes the time averaged orographic masks for use with UTF output on pressure surfaces (see above).

OPOD controls the production of the time averaged OD diagnostics in the same manner as ODINST controls the instantaneous diagnostics. OPOD3D is used to save any time averaged OD fields in a user-specified data format and/or file format), and OPODTS creates and outputs any OD time series requested by the user.

8.3 Mathematical definition of fields

This section defines mathematically the fields produced by the OD diagnostics. It should be remembered that calculations are performed on the surfaces on which fields are to output by the SUMO job, i.e. one of η, pressure and θ. In this way the OD diagnostics differ from the SG diagnostics where the majority of the fields are computed on model (eta) levels, and are then interpolated to the output surfaces.

The relative humidity (RH) is determined according to the following method. First, the saturation vapour vapour pressure $e_s(T)$ is found according to

$$e_s(T) = C_1 \exp\left(\frac{C_3(T-T_0)}{T-C_4}\right) \tag{8.1}$$

where

$$\begin{aligned} T_0 &= 273.16 \\ C_1 &= 610.78 \\ C_3 &= \begin{cases} 17.269 & \text{for } T \geq T_0 \\ 21.875 & \text{for } T < T_0 \end{cases} \\ C_4 &= \begin{cases} 35.86 & \text{for } T \geq T_0 \\ 7.66 & \text{for } T < T_0 \end{cases} \end{aligned}$$

This formulation takes account of both the ice and water phases. Next, the saturation specific humidity q_s is determined according to

$$q_s = \frac{q_s^*}{c} \tag{8.2}$$

where

$$q_s^* = \varepsilon e_s(T)/p,$$

with

$$c = 1 - (1-\varepsilon)e_s(T)/p$$

and

$$\varepsilon = R_d/R_v.$$

The relative humidity is then computed according to

$$RH = 100 \times q/q_s \tag{8.3}$$

This approach produces problems as $p \to (1-\varepsilon)e_s$ with c tending towards zero in this case, while in regions where $p \to 0$ the method gives a negative value for q_s. To overcome this, Blackburn *et. al.* (1989) modified c according to

$$c = \max(10^{-10}, |c|).$$

for use with the UGAMP GCMs, and this correction is therefore applied in SUMO.

The relative humidity is computed in routine `TETENS`; this routine contains code that allows the user to force the relative humidity to lie in the range 0 – 100 per cent, as follows:

```
LOFORC=.TRUE.
ZRHMIN=0.0
ZRHMAX=100.0
```

If `LOFORC` is set to `.TRUE.` (the default) then RH is forced to lie between the values `ZRHMIN` and `ZRHMAX` inclusive. The user can alter these if required – the program writes out a warning when these variables are used.

The moist static energy Q_s is derived from the Montgomery streamfunction M_ψ according to

$$Q_s = gz + C_pT + L_vq = M_\psi + L_vq \tag{8.4}$$

where L_v is the latent heat of vapourisation.

The wind speed $|u|$ is determined from the vector sum of the zonal and meridional wind components,

$$|u| = \sqrt{u^2 + v^2} \tag{8.5}$$

The equivalent PV latitude (ϕ_e) and normalised PV area (A_N) calculations are carried out, on one θ-surface at a time, in routine `ODFLXG`. Note that if the output surface type does not correspond to a potential temperature surface, then both these quantities will be set to zero. If any isentropic surface is subterranean over any region of the globe, then both ϕ_e and A_N will be set to a mask value at all gridpoints on that surface.

The first step is to determine the range of PV values on a given isentropic surface. Next, a set of PV 'bins' are constructed – the number used being the larger of 180 and the value of the parameter `JPJGG`. Each bin is then assigned a range of PV values (the same range in each bin) such that all gridpoints on the isentropic surface can be assigned to a particular 'bin' according to their PV.

As this assignment is done, the area of each gridbox (A_G) is computed according

$$A_G = (\sin\phi_2 - \sin\phi_1)(\lambda_2 - \lambda_1)R_e^2$$

where R_e is the radius of the earth, and the ϕ and λ are the latitudes and longitudes bounding the gridsquare. The total area of the gridsquares assigned to each 'bin' may then the determined. Note that areas in the southern hemisphere are taken to be positive.

When this process has been completed the area enclosed by each PV isopleth (as defined by the boundary PV values for each 'bin') is then known. Assuming that the most positive PV bin corresponds to data adjacent to the north pole, and the most negative PV bin to data adjacent to the south pole, the cumulative area from the south pole towards the north pole can then be determined.

The equivalent PV latitude can then be computed from the cumulative 'bin' areas (A_{accum}) according to

$$\phi_e = \frac{180}{\pi} \arcsin\left(-1.0 + A_{accum}/(2\pi R_e^2)\right) \tag{8.6}$$

with resulting values in the range -90 to 90 degrees.

The normalised PV area is determined by finding the area enclosed between the equivalent PV latitude and the nearest pole, the area being normalised with respect to the area of a hemisphere. Values typically range between 0 near both poles to 1 at the equator – see Lary (1993).

The formulae used to compute the geostrophic velocity components are output level-type-dependent, and are computed in one of the routines `ETGEOS`, `ISGEOS`, `THGEOS` or `PVGEOS`. (Values on potential vorticity surfaces are currently not defined.)

THESE DIAGNOSTICS ARE NOT CURRENTLY AVAILABLE ON MODEL LEVELS.

On pressure surfaces, the following expressions are used:

$$u_g = -\frac{1}{R_e f}\frac{\partial \Phi}{\partial \phi} \tag{8.7}$$

$$v_g = \frac{1}{R_e f \cos\phi}\frac{\partial \Phi}{\partial \lambda} \tag{8.8}$$

while for output on θ surfaces the Montgomery streamfunction is used to compute the geostrophic velocity components according to

$$u_g = -\frac{1}{R_e f}\frac{\partial M_\psi}{\partial \phi} \tag{8.9}$$

$$v_g = \frac{1}{R_e f \cos\phi}\frac{\partial M_\psi}{\partial \lambda} \tag{8.10}$$

Finite differencing in the meridional direction of terms involved in the calculation of v_g is performed using a third-order centred scheme written by J. Thuburn. At interior gridpoints (i.e. away from the poles) 5 gridpoints are involved in each $\partial/\partial y$ evaluation, (the scheme becomes fourth-order on a regular grid) with one-sided differences using 3 gridpoints being employed at latitudes adjacent to the poles (routine `DBDY`).

In the zonal direction, differences are computed using centred differences over 2 gridlengths being used when differencing is performed in pressure or θ surfaces.

Since the Coriolis parameter vanishes at the equator, the geostrophic velocity components between latitudes 10°N and 10°S (or the nearest gridbox latitudes outside this range) are extrapolated linearly with respect to the sine of latitude.

THE ROSSBY WAVE SOURCE IS CURRENTLY NOT COMPUTED.

The calculation of the equivalent potential temperature (θ_E) and the equivalent saturated potential temperature (θ_{Es}) follows the approach of Bolton (1980). Thus

$$\epsilon = \frac{R_d}{R_v} \times 1000$$

where R_d and R_v are the gas constants for dry air and water vapour. The mixing ratio (r, g kg^{-1}) is given by

$$r = \left(\frac{q}{1-q}\right) \times 1000$$

where q is defined in terms of SI units. The vapour pressure e (millibars) is given by

$$e = \frac{pr \times 10^{-2}}{\epsilon + r}$$

p being in SI units. The temperature at the lifting condensation level (T_L, K) is found using the formula

$$T_L = \frac{2840}{3.5 \ln T - \ln e - 4.805} + 55.0$$

Finally, θ_E is evaluated using

$$\theta_E = T\left(\frac{10^5}{P}\right)^{0.2854(1.0-0.00028r)} \times \exp\left[\left(\frac{3.376}{T_L} - 0.00254\right) \times r\left(1 + 0.81 \times 10^{-3}r\right)\right] \tag{8.11}$$

To compute θ_{Es}, r is replaced by the saturated mixing ratio of the air at the gridpoint with the temperature T using Eqn. (8.1), while T_L is set equal to the gridpoint temperature T (no lifting is required to saturate an already-saturated air parcel). Eqn (8.11) is then used to compute θ_{Es} with these modifications.

8.4 Summary of options available

Tables 2.15 – 2.20 illustrate the options available to the user when setting up the program to compute the OD diagnostics. These are now explained in a little more detail, together with some of the possible pitfalls. Reference is made in this section to the Fortran namelists variables, rather than to the data block keywords. Use should be made of the data pre-processor, which will help to eradicate inconsistent input data, before running the diagnostic program.

When diagnosing the gridpoint fields, the user should ensure that appropriate values have been specified for the variables `LMASSW`, `LAVMSK` and `LAVSIG` (amongst others) since these can affect significantly the results that will be produced. It is recommended that `LMASSW` is set to `.FALSE.` (the default).

The user should refer to section 6.5 for a description of the mathematics underlying the extraction of slices, profiles, etc. from the three-dimensional OD arrays; the method is identical to that applied to the SG-type fields.

The `DIAGNOSTICS` keyword in the COMMON data block must be set to `OD` to enable the OD diagnostics, and the appropriate value must be set for output level type (character variable `YTYPSF` — the default is to produce output on model (η) levels). Attention is also drawn to the use of the OR data block to produce UTF masks when output on pressure surfaces is requested.

If the input fields are to be interpolated to other surfaces, then array `OUTLEV` must be defined — unless the default output levels are to be used (see sections 5.2.2 to 5.2.4). While the default values for `OUTLEV` may be satisfactory if only vertical slices are to be diagnosed, they will probably be unsatisfactory for the production of diagnostics on latitude-longitude surfaces. As a general rule, it should be assumed that any non-eta output surfaces for which latitude-longitude diagnostics are needed should be specified by setting non-default values for `OUTLEV` — see section 2.5.1. The `OUTLEV` levels are also used to define the output surfaces to be used in the production of zonal-mean and meridional-mean time series.

The OD data block may be envisaged as being at the top of the OD diagnostics tree — see Fig. 2.1. This data block must be used to specify the type of spatial and temporal diagnostics required — by default zonal, time averaged diagnostics will be produced.

Also to be specified in this data block are the levels for which latitude-longitude diagnostics are needed; if the diagnostic output is on input data levels, then `NVODHR` should be filled with the integer level numbers required (in the range 1 to N_L), else if the output is on non-model levels, then `VODSFC` should be filled with one (or more) values copied from those contained in the array `OUTLEV`.

Variables in data block OH (Table 2.16) need only be specified if output on horizontal (latitude-longitude) surfaces is required. The fields codes of the gridpoint fields (see Table 8.1) for which printed and/or plotted output is required should be specified using variables `NOHPR` and `NOHPL` respectively. The field codes may be specified in any order for each variable, and the total number of different fields selected for either type of output must be specified using `NOH`. By default, no fields are selected.

The contour intervals (to be passed to the UTF if appropriate) for the `NOHPL` fields should be specified in order corresponding to the field codes in `NOHPL`.

The UTF standard (version 1.4) allows for wind vector information to be included in the UTF in a manner that facilitates it's incorporation into selected plotted fields (see Blackburn *et al.* 1992). The user can take advantage of this facility by assigning to variable `NOHVEC` the field code values corresponding to the fields on which horizontal wind vectors will be plotted.

Logical variables `LDOHSL`, `LDOHCT` and `LDOHED` need to be set to `.TRUE.` to produce OH diagnostic output, while Hovmöller output can be enabled by specifying the latitude range required (variable `VOHTS`). `LDOHCT` can only be selected when output is on model (η) levels, provided that `LDOHSL` has been set to `.TRUE.`.

Entities in data block OZ (Table 2.17) need only be specified if diagnostics in the zonal plane are required. The fields codes of the gridpoint fields (see Table 8.1) for which printed and/or plotted output is required should be specified using variables `NOZPR` and `NOZPL` respectively. The field codes may be specified in any order for each variable, and the total number of different fields selected for either type of output must be specified using `NOZ`. By default, no fields are selected.

The contour intervals (to be passed to the UTF if appropriate) for the `NOZPL` fields should be specified in order corresponding to the field codes in `NOZPL`.

The UTF standard (version 1.4) allows for wind vector information to be included in the UTF in a manner that facilitates it's incorporation into selected plotted fields (see Blackburn *et al.* 1992). The user can take advantage of this facility by assigning to variable `NOZVEC` the field code values corresponding to the fields on which zonal wind vectors will be plotted. (Note: this facility is currently disabled for wind components in the zonal plane.)

Logical variables `LDOZZM` and `LDOZTS` need to be set to `.TRUE.` to produce OZ diagnostic output; the former will enable zonal- mean fields to be produced, while the latter will allow the production of zonal-mean time series on the surfaces specified by the variables `NVODHR` or `VODSFC`, as appropriate, provided that `LDOZZM` is set to `.TRUE.`. Longitude slices and west-east averages over a specified longitude range can be computed provided `VOZNS` and `VOZNSA` are defined.

Variables in data block OM (Table 2.18) need only be specified if diagnostics in the meridional plane are required. The fields codes of the gridpoint fields (see Table 8.1) for which printed and/or plotted output is required should be specified using variables `NOMPR` and `NOMPL` respectively. The field codes may be specified in any order for each variable, and the total number of different fields selected for either type of output must be specified using `NOM`. By default, no fields are selected.

The contour intervals (to be passed to the UTF if appropriate) for the `NOMPL` fields should be specified in order corresponding to the field codes in `NOMPL`.

The UTF standard (version 1.4) allows for wind vector information to be included in the UTF in a manner that facilitates it's incorporation into selected plotted fields (see Blackburn *et al.* 1992). The user can take advantage of this facility by assigning to variable `NOMVEC` the field code values corresponding to the fields on which meridional wind vectors will be plotted. (Note: this facility is currently disabled for wind components in the meridional plane.)

Logical variables `LDOMMM` and `LDOMTS` need to be set to `.TRUE.` to produce OM diagnostic output; the former will enable meridional-mean fields to be produced, while the latter will allow the production of meridional-mean time series on the surfaces specified by the variables `NVODHR` or `VODSFC`, as appropriate, provided that `LDOMMM` is set to `.TRUE.`. Longitude slices and west-east averages over a specified longitude range can be computed provided `VOMEW` and `VOMEWA` are defined.

Variables in data block OP (Table 2.19) need only be specified if column profile diagnostics are required. The fields codes of the gridpoint fields (see Table 8.1) for which printed and/or plotted output is required should be specified using variables `NOPPR` and `NOPPL` respectively. The field codes may be specified in any order for each variable, and the total number of different fields selected for either type of output must be specified using `NOP`. By default, no fields are selected.

The contour intervals (to be passed to the UTF if appropriate) for the `NOPPL` fields should be specified in order corresponding to the field codes in `NOPPL`. Note that these contour intervals will be used when time series of column profiles (or totals) are produced (in the event that `LDOP` and `LDODTS` are both `.TRUE.`.

To produce column profile and/or column total diagnostics (the latter only when output is requested on model (η) levels) requires that `LDOPCP` and/or `LDOPCT` be set to `.TRUE.`. By default, both are set to `.FALSE.`. The user must also specify the profile coordinates of up to `JPOPPF` locations on the globe where these diagnostics will be evaluated (array `VOPPF`), along with the number of coordinate pairs thus selected, `NOPPF`.

The final gridpoint data block that needs to be set up by the user is the O3 data block. This will allow the user to save the full fields (or subsets thereof) for use in other programs/packages. The user needs to specify values for the output channel for the data, the fields codes of the fields to be saved, and the total number of fields to be saved (see Table 2.20).

If `LASCII` is `.TRUE.`, then the output will be in the form of a fortran formatted file. If `LDRS` is `.TRUE.`, then a DRS format file will be produced.

8.5 Time averaging

The time averaging of the OD fields is carried out in an identical manner to that for the SG gridpoint diagnostics – see section 6.6.

8.6 Saving 3-D fields

A Fortran routine (`OPOD3D`) is available for use as a template for those users wishing to save whole fields (or parts thereof) for later use in other packages. This routine can be edited to fit the user's requirements.

It is also possible to alter the form of the this entire file (written to Fortran channel `NCO3`) by means of the unicos `assign` command in, for example, the unicos shell script that runs the diagnostic job.

8.7 How to incorporate additional OD fields into SUMO

It is envisaged that many users will wish to incorporate their own diagnostics into SUMO; this section describes the steps that must be undertaken to modify the OD diagnostic code for this purpose – it asumes that additional three-dimensional fields are required.

1. Alter the Fortran parameter `JPODMX` in comdeck `PARAM3` so that it specifies the total number of fields available in the OD diagnostic subset.

2. For each additional OD variable modify commmon block `COMNGP` so that it contains one extra variable (eq `NR1***`) for each additional field.

3. Modify routine `DACNST` to define the initial record, of `JPJG` sequential records, for storing each additional field in direct access files (the variables `NR1***`). The ordering of these initial records will define the field number to be used in the OD data blocks for variables `NOHPR` etc.

4. Modify routine `ODINIT` to specify the local title variables `YONAME` and `YOWNAM`. The ordering of these titles must correspond to the ordering of the fields as specified by the `NR1***` variables in routine `DACNST`. Similar changes will need to be made in routine `ODDRSI`, to set up the variable short- and long-names, as well as the units of the new variable, for DRS output, and to routine `ODNETI` for NetCDF output.

5. At present the following routines exist to compute instantaneous OD gridpoint fields:

   - `ODFLXL` — fields computed for one latitude at a time, using SG fields of the 'current' latitude.
   - `ODFLXG` – fields computed globally (typically because the OD field at a given latitude requires SG-type data on several latitudes).
   - `ETGEOS`, `ISGEOS`, `THGEOS`, `PVGEOS` – fields computed globally but using different formulae on different output surface types.

   Any code to compute additional OD fields should be added to one of these routines; it may be necessary to create three routines corresponding to routines `ETGEOS`, `ISGEOS`, `THGEOS` and `PVGEOS` to be used when calculations can be performed latitude-by-latitude, but when different formulae are required on different output surface types. Remember to write out any additional OD arrays (global or at a single latitude) to the file `NCODC` in any of these routines. Fields should be calculated using MKS units.

If a field cannot be computed under certain conditions, ensure that it is set to a null or masked-value at all gridpoints. This will prevent later problems at the output stages of the program.

6. If the additional OD field is not to be output in MKS format, then code will need to be inserted in routines `ODINST` and `OPOD` to check for the field number and to do any conversion of units accordingly; refer to routine `SGCONV` for an example of how to do this.

7. All time-averaging and output processing should be handled without the need to change existing code.

8. Finally, modify comdeck `OUTLINE` to reflect the inclusion of any addition subprograms into SUMO, and update comdeck `GRIDOD` to enable storage of the new variables.

The following change will need to made to PRESUMO for compatibility:

- Alter comdeck `PARAMS` so that parameter `JPODMX` is identical to the value used in the SUMO program.

9 ZF diagnostics

9.1 Introduction

The ZF diagnostics permit the production of zonal mean flux diagnostics. The equations defining these fluxes are specified by the user, and may involve any of the three-dimensional (thermo-)dynamical SG-type fields. Both zonal means of the total flux and of the eddy flux can be computed. The diagnostic output can be in the form of zonal and/or level mean fluxes, and can consist of either instantaneous, time-average or time-series information.

For a complete summary of the options available, see section 2.5.6. The SG-type fields that may be used are those listed in Table 6.1 (excluding the surface pressure, MSL pressure, the surface potential temperature and the land/air mask).

The fluxes are computed using interpolated fields if the output is not specified to be on the same surfaces as the input data.

9.2 Implementation description

Figure 9.1 shows the principal routines used to compute the ZF diagnostics. The user's input is read into the program in routine `DATAIN` and then checked by routines `DATCHK` and `CHKZF`.

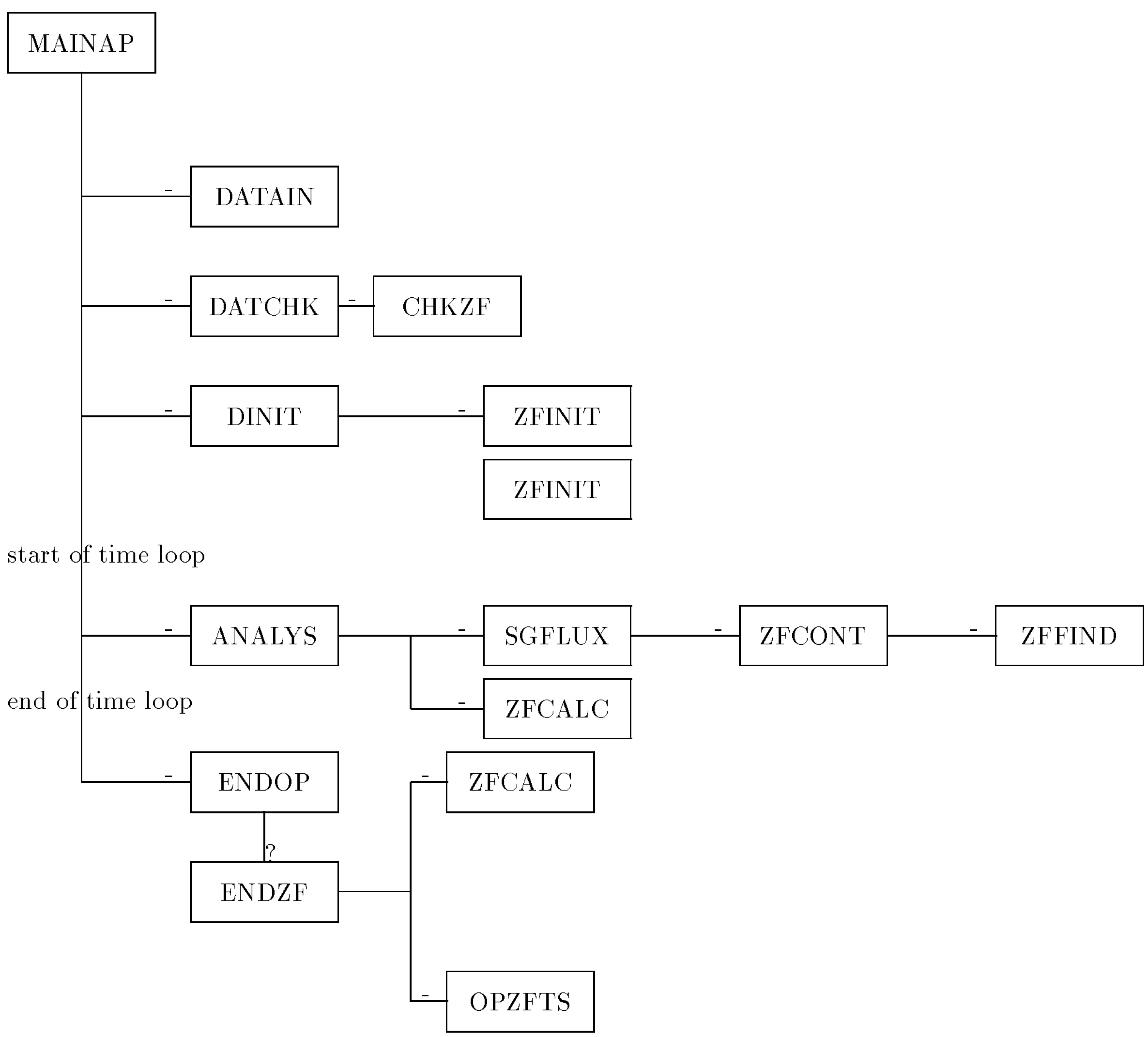

Figure 9.1: Flow chart showing the principal subroutines used to perform the ZF diagnostics.

Routine `DINIT` controls the initialisation of constants for the various diagnostics. In particular, it calls

ZFINIT, a routine that sets up the array names and diagnostic constants for the flux types specified by the user.

ANALYS is the routine that controls the incrementing of the time loop, and the calculations at each timestep. **SGFLUX** computes the SG-type gridpoint fields at each timestep — see section 5 — and calls **ZFCONT**, which controls the step-by-step calculation of the zonal-mean mean- and eddy-fluxes in routine **ZFFIND**. Routine **ZFCALC** outputs the instantaneous fluxes if they are required.

At the end of the time loop, routine **ENDZF** controls the production of the time-averaged and time-series fields and any required output. Routine **ZFCALC** now outputs the time-average fluxes, while **OPZFTS** controls the creation of the time series.

9.3 Flux formulae

Consider a three dimensional field ϕ_1. Let the symbol $\overline{\phi_1}^z$ represent the zonal mean of this field, such that

$$\phi_1 = \overline{\phi_1}^z + \phi_1^*$$

where ϕ_1^* is the perturbation from the zonal mean at a given height and latitude. Typically, if ϕ_1 represents a velocity component and the field ϕ_2 represents another scalar field, then the zonal-mean flux of $\phi_1\phi_2$ may be written as

$$\overline{\phi_1\phi_2}^z = \overline{\phi_1}^z\,\overline{\phi_2}^z + \overline{\phi_1^*\phi_2^*}^z \tag{9.1}$$

Here the left hand side of Eq. (9.1) will be referred to as the zonal-mean mean-flux, and the second term on the right hand side as the zonal-mean eddy-flux. Both these quantities can be output from SUMO.

The SUMO zonal-mean flux diagnostics require the user to specify the formulae of the required fluxes. Note that there are no default formulae (see Table 2.21). The general formulae for a ZF-type flux (Φ) may be written in the form

$$\Phi = F_1\phi_1\phi_2 + F_2\phi_3\phi_4 \tag{9.2}$$

where F_1 and F_2 are numerical factors (either positive or negative) and the ϕ_i are three-dimensional (thermo-)dynamical fields. These four fields may or may not be different from each other.

Note that it is possible for the factor F_2 to be set to zero, and also for individual fields to be suppressed in the formula. In the former case, this reduces to a formula typical of a simple flux (ie velocity component times scalar field) provided that $F_1 = 1$. Thus, each of the following are valid formulae capable of being processed by SUMO:

$$\Phi = 1.0\phi_1\phi_2 \tag{9.3}$$

$$\Phi = -1.0\phi_1\phi_2 + 1.0\phi_3 \tag{9.4}$$

$$\Phi = 4.0\phi_1 - 2.0\phi_3\phi_4 \tag{9.5}$$

$$\Phi = 1.0\phi_1\phi_1 + 1.0\phi_3\phi_3 \tag{9.6}$$

Whether or not these formulae are physically meaningful or not is open to question. It up to the user to ensure that any specified equations are, for example, dimensionally correct. When just one field is specified in either term in the formula (9.2), then the other (missing) field in the pair is taken as being a field containing zeroes – such a formulae needs to be treated with caution by the user.

Formulae are entered (in upper-case only) into the program via the ZF data block, where they are specified in the form

```
FORMULA=FACTOR1XFIELD1XFIELD2+FACTOR2XFIELD3XFIELD4
```

Multiple formulae are separated by commas. Here, **FIELD1**, ...**FIELD4** are the SG field codes (see Table 6.1). Thus, specifying

```
  FORMULA=1.0X6X6-1.0X5X5,
     -1.0X5X6+0.0X0X0,
     0.5X5X5+0.5X6X6,
     1.0X5X7,
```

will result in the following mean and eddy fluxes being computed

$$\begin{array}{cc} \overline{vv}^z - \overline{uu}^z & \overline{v^*v^*}^z - \overline{u^*u^*}^z \\ -\overline{uv}^z & -\overline{u^*v^*}^z \\ \frac{1}{2}\overline{uu}^z + \frac{1}{2}\overline{vv}^z & \frac{1}{2}\overline{u^*u^*}^z + \frac{1}{2}\overline{v^*v^*}^z \\ \overline{uT}^z & \overline{u^*T^*}^z \end{array}$$

Only one field formula is allowed per line of the input data file. Note that field codes 5, 6 and 7 correspond to the zonal wind, meridional wind and temperature, respectively. The use of field code 0 results in a field of zeroes being used at that point in flux equation, while if the second term in the equation is omitted (or if `FACTOR2` is zero) then the resulting flux will contain only one term.

9.4 Summary of options available

Table 2.21 illustrates the options available to the user when setting up the program to compute the ZF diagnostics. These are now explained in a little more detail, together with some of the possible pitfalls. Reference is made in this section to the Fortran namelists variables, rather than to the data block keywords. Use should be made of the data pre-processor, which will help to eradicate inconsistent input data, before running the diagnostic program.

When computing the ZF fields, the user should ensure that appropriate values have been specified for the variable `LMASSW` since this can affect significantly the results that will be produced. Masking and weighting are explained more fully in section 6.4. When output is on isentropic surfaces field code 13 will refer to the quantity σ. When output is on isentropic surfaces and this mass-weighting factor is switched on **all** fields appearing in the zonal-mean flux formulae are automatically multiplied by σ.

The `DIAGNOSTICS` keyword in the COMMON data block must be set to `ZF` to enable the ZF diagnostic calculations, and the appropriate value must be set for output level type (character variable `YTYPSF` — the default is to produce output on levels corresponding to those of the input data).

If the fields are to be interpolated to non-η surfaces, then array `OUTLEV` must be defined — unless the default output levels are to be used (see sections 5.2.2 to 5.2.4). While the default values for `OUTLEV` may be satisfactory if only vertical slices are to be diagnosed, they will probably be unsatisfactory for the production of diagnostics on horizontal (i.e. time-series) surfaces. As a general rule, it should be assumed that any non-eta output surfaces for which time-series diagnostics are needed should be specified by setting non-default values for `OUTLEV` — see section 2.5.1.

The ZF data block is currently the only data block contained in the 'ZF tree' — see Fig. 2.1. This data block must be used to specify the form of ZF diagnostics required.

Logical variables `LZFEDF` and `LZFMNF` should be set to `.TRUE.` to allow the output of eddy- and/or mean-fluxes respectively. These fluxes can be in the form of zonal-mean flux cross-sections (`LDZFZM=.TRUE.`) and/or a profile of level-mean fluxes (`LDZFLM=.TRUE.`).

Arrays `NFLDZF` and `FACTZF` are used to define the `NZF` individual flux formulae; arrays `NZFPR` and `NZFPL` specify which of these individual fluxes are printed and/or converted to UTF/DRS format. Printed arrays are output to channel `NCZF` while array `CINTZF` defines the contour interval to be passed to the UTF (if appropriate) for each of the fields specified by `NZFPL`. Output can be in the form of time-series (`LDZFTS=.TRUE.`), time-averages (`LDZFTD=.TRUE.`) and/or instantaneous fields (`LDZFIN=.TRUE.`). Time series can be in the form of a set of zonal-means at levels specified by either `VZFSFC` or `NVZFHR` as appropriate, and as a sequence of level-mean profiles.

9.5 Mathematical definitions

The relationship between the zonal-mean mean- and eddy-fluxes is defined in section 9.3 above.

The zonal mean of a latitude(ϕ)-longitude(λ) field Φ held at gridpoints (i,j) is defined as

$$\overline{\Phi(\phi)}^{\lambda}(j) = \frac{1}{\mathrm{MG}} \sum_{i=1}^{\mathrm{MG}} \Phi(i,j) \tag{9.7}$$

where MG is the number of gridpoints along each line of latitude.

The level-mean of a flux is determined by an area-weighting method, and is found from the zonal-mean:

$$\overline{\Phi}^{\phi\lambda} = \frac{1}{2}\sum_{j=1}^{\mathrm{JGG}} (\sin\phi_{j+1} - \sin\phi_j)\overline{\Phi(\phi)}^{\lambda}(j) \tag{9.8}$$

where the ϕ represent the gridbox latitude boundaries from pole to pole.

10 XD diagnostics

10.1 Introduction

The XD diagnostics permit the production of zonal mean cross-section diagnostics, including transformed Eulerian-mean diagnostics. The diagnostic output can be in the form of zonal and/or level mean fluxes, and can consist of either instantaneous or time-averaged fields.

For a complete summary of the options available, see section 2.5.7. The fields may only be computed on pressure surfaces, and use SG-type fields interpolated to these surfaces.

Table 10.1 illustrates the XD fields that can be output from SUMO. Note that the transformed Eulerian-mean (TEM) fields may only be output if time-averages have been requested, since they include rate of change calculations, requiring a time-range of data. **(If additional zonal-mean fields are added to this list at a later date, then it is important that the order of 'non-TEM fields', then 'TEM fields', is maintained.)**

10.2 Implementation description

Figure 10.1 shows the principal routines used to compute the XD diagnostics. The user's input is read into the program in routine `DATAIN` and then checked by routines `DATCHK` and `CHKXD`.

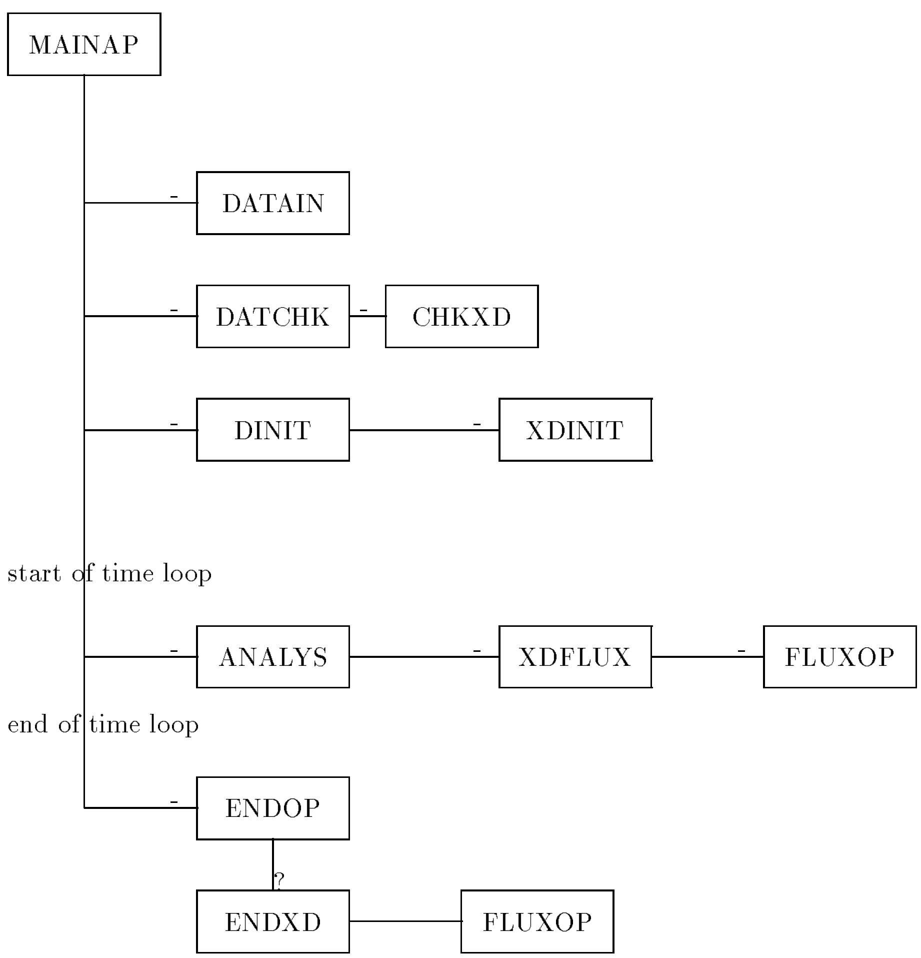

Figure 10.1: Flow chart showing the principal subroutines used to perform the XD diagnostics.

Routine `DINIT` controls the initialisation of constants for the various diagnostics. In particular, it calls `XDINIT`, a routine that sets up the array names and diagnostic constants for the XD fields.

`ANALYS` is the routine that controls the incrementing of the time loop, and the calculations at each timestep. `XDFLUX` computes the zonal-mean fields at each timestep and calls `FLUXOP`, which controls the

Field	units	Field code
Mean meridional circulation	kgs^{-1}	1
Horizontal eddy momentum flux	m^2s^{-2}	2
Vertical eddy momentum flux	$ms^{-1}mb.hr^{-1}$	3
Horizontal eddy heat flux	$K.ms^{-1}$	4
Vertical eddy heat flux	$K.mb.h^{-1}$	5
Kinetic energy of zonal-mean wind	Jkg^{-1}	6
Eddy kinetic energy	Jkg^{-1}	7
Scaled northward E-P flux	$m^3.RAD$	8
Scaled upward E-P flux	m^3Pa	9
Horizontal E-P flux divergence	m^3	10
Vertical E-P flux divergence	m^3	11
Total E-P flux divergence	m^3	12
Square of the Brunt-Väisälä frequency	s^{-2}	13
PV y-gradient for varying N^2	$m^{-1}s^{-1}$	14
Total permitted wavenumber (phase speed=0)		15
Vertical mean heat flux	$K.mb.hr^{-1}$	16
Horizontal mean heat flux	$K.ms^{-1}$	17
Horizontal eddy moisture flux	$ms^{-1}gkg^{-1}$	18
Vertical eddy moisture flux	$mb.hr^{-1}g.kg^{-1}$	19
Transformed Eulerian mean fields		
Zonal wind tendency	$ms^{-1}day^{-1}$	20
Zonal-mean residual vertical velocity, $\overline{w}^*$	ms^{-1}	21
Temperature tendency	$K.day^{-1}$	22
Residual in u budget	$m.s^{-1}day^{-1}$	23
Residual in T budget	$K.day^{-1}$	24
Residual in v budget	$m.s^{-1}day^{-1}$	25
$\overline{v}^*$	ms^{-1}	26
$\overline{\omega}^*$	Pas^{-1}	27
$-\overline{v}^* \times$ absolute vorticity	$ms^{-1}day^{-1}$	28
$\overline{\omega}^* \times d(\overline{u})/dp$	$ms^{-1}day^{-1}$	29
Horizontal E-P flux (divided by Earth's radius)	$(ms^{-1})^2$	30
Vertical E-P flux (divided by Earth's radius)	$Pa.m/(s^2)$	31
$-$divergence of horizontal E-P flux	$ms^{-1}day^{-1}$	32
$-$divergence of vertical E-P flux	$ms^{-1}day^{-1}$	33
$-$total E-P flux divergence	$ms^{-1}day^{-1}$	34
$\overline{v}^* \times d(\overline{T})/dy$	$K.day^{-1}$	35
$\overline{\omega}^* \times d(\overline{T})/dp$	$K.day^{-1}$	36
$-\kappa\overline{\omega}^*\overline{T}/p$	$K.day^{-1}$	37
Eddy terms in the thermodynamic equation	$K.day^{-1}$	38
Departure from gradient wind balance	$ms^{-1}day^{-1}$	39
$\overline{u} \times (f + \overline{u}.\tan \mathrm{latitude})$	$ms^{-1}day^{-1}$	40
Northward wind tendency	$ms^{-1}day^{-1}$	41
$d(\mathrm{geopotential})/dy$	$ms^{-1}day^{-1}$	42
$\overline{v} \times d(\overline{v})/dy$	$ms^{-1}day^{-1}$	43
$\overline{\omega} \times d(\overline{v})/dp$	$ms^{-1}day^{-1}$	44
$(d(\overline{v'v'})/d\phi)/R_e$	$ms^{-1}day^{-1}$	45
$d(\overline{\omega'v'})/dp$	$ms^{-1}day^{-1}$	46
$(\overline{u'u'} - \overline{v'v'}) \times \tan\phi/R_e$	$ms^{-1}day^{-1}$	47

Table 10.1: Zonal-mean cross-section fields available via the XD diagnostics.

output of instantaneous fields.

At the end of the time loop, routine ENDXD controls the production of the time-averaged fields and any required output. Routine FLUXOP finally outputs the time-average fluxes. Calculations are carried out using SI units; fields are converted to meteorological units immediately prior to output.

10.3 Mathematical definitions

All calculations are performed using data on pressure (p_k) levels – the latter may be either the default levels or a mixture of default and user-specified levels. For certain calculations a series of half-level pressures is required; these are defined according to

$$p_{1/2} = 0.0$$

$$p_{k+1/2} = 0.5 \times (p_k + p_{k+1}) \quad \text{for} \;\; 1 \leq k \leq N_L - 1$$

$$p_{N_{L+1/2}} = 1000.0\text{mb, if } p_{N_L} \leq 1000.0$$

$$p_{N_{L+1/2}} = 2p_{N_L} - p_{N_{L-1/2}}\text{mb, if } p_{N_L} > 1000.0$$

In all the formulae illustrated below, the overbar ($\overline{x}$) symbol is used to denote the zonal mean (in this case of quantity x) around a latitude circle.

Mean meridional circulation

The mean meridional streamfunction (Ψ)is evaluated by computing the mass flux due to a zonal-mean meridional velocity component $\overline{v}$ into a gridbox of width $2\pi R_e \cos\phi$ (ie. the length of a latitude circle) and depth Δz. Substituting for Δz in terms of Δp, and integrating downwards from $p = 0$ gives

$$\Psi = \frac{2\pi R_e \cos\phi}{g} \int_0^p \overline{v} dp$$

Note that R_e is the Earth's radius.

Numerically, $\Psi = 0$ at the upper boundary. The RHS of the previous equation is evaluated by integrating downwards to each successive half-level pressure; the value on each full-level surface is then taken to be the arithmetic mean of the adjacent two half-level surface values.

Horizontal eddy momentum flux

The zonal mean horizontal eddy momentum flux is defined according to

$$\overline{u'v'} = \overline{uv} - \overline{u}\,\overline{v}$$

where u and v are the zonal and meridional velocity components.

Vertical eddy momentum flux

The zonal mean vertical eddy momentum flux is defined according to

$$\overline{u'\omega'} = \overline{u\omega} - \overline{u}\,\overline{\omega}$$

where ω is the omega velocity component.

Horizontal eddy heat flux

The zonal mean horizontal eddy heat flux is defined according to

$$\overline{v'T'} = \overline{vT} - \overline{v}\,\overline{T}$$

where T is the temperature.

Vertical eddy heat flux

The zonal mean vertical eddy heat flux is defined according to

$$\overline{\omega' T'} = \overline{\omega T} - \overline{\omega}\,\overline{T}$$

Kinetic energy of the zonal mean wind

The kinetic energy due to the zonal mean wind is calculated using

$$K = \frac{1}{2} \times (\overline{u}.\overline{u} + \overline{v}.\overline{v})$$

Eddy kinetic energy

The eddy kinetic energy due is defined as

$$K_{eddy} = \frac{1}{2} \times (\overline{u'u'} + \overline{v'v'})$$

Northward and upward E-P fluxes

This section is taken from notes by Mike Blackburn (unpublished). Eliassen and Palm (1961) showed that for linear conservative disturbances of a geostrophic state, the eddy fluxes obey the relationship

$$\frac{\partial F_y}{\partial y} + \frac{\partial F_p}{\partial p} = 0$$

where

$$F_y = \frac{1}{\sigma}\overline{u_p}\,\overline{v'\Phi'_p} - \overline{u'v'}$$

$$F_p = \frac{1}{\sigma}(f - \overline{u_y})\overline{v'\Phi'_p} - \overline{u'\omega'}$$

$$\sigma = -\overline{\theta_p}/(\rho_0\overline{\theta})$$

and Φ is the geopotential, θ is the potential temperature, f the Coriolis parameter and ρ_0 a density profile.

In terms of the nomenclature of Edmon *et al* (1980) this becomes for the sphere

$$\nabla.\underline{F} = \frac{1}{R_e \cos\phi}\frac{\partial}{\partial\phi}(F_\phi \cos\phi) + \frac{\partial F_p}{\partial p} = 0$$

where

$$F_\phi = R_e \cos\phi \left(-\overline{u'v'} + \frac{\overline{v'\theta'}\,\overline{u_p}}{\overline{\theta_p}} \right)$$

$$F_p = R_e \cos\phi \left(-\overline{u'\omega'} + \frac{\overline{v'\theta'}\,\overline{\zeta}}{\overline{\theta_p}} \right)$$

and

$$\overline{\zeta} = f - \partial\overline{u}/\partial y$$

For graphical purposes, Edmon *et al.* (1980) use the following conventions (which are also applied to fields 8 and 9 of the XD diagnostics): the arrows on a (ϕ, p) cross-section are given by

$$\underline{\hat{F}} = \left\{\hat{F}_\phi, \hat{F}_p\right\} \equiv \frac{2\pi R_e^2}{g}\cos\phi \left\{\frac{F_\phi}{R_e}, F_p\right\}$$

such that

$$\hat{F}_\phi = \frac{2\pi R_e^2}{g}\cos^2\phi \left(-\overline{u'v'} + \frac{\overline{v'\theta'}\,\overline{u_p}}{\overline{\theta_p}} \right)$$

$$\hat{F}_p = \frac{2\pi R_e^2}{g}\cos^2\phi \left(-\overline{u'\omega'} + \frac{\overline{v'\theta'}\,\overline{\zeta}}{\overline{\theta_p}} \right)$$

SUMO generates the two components of $\underline{\hat{F}}$. $\hat{F}_\phi$ points towards increasing latitude and $\hat{F}_p$ to increasing pressure if positive. $\hat{F}_\phi$ (units of m^3.rad) is multiplied by the distance occupied by 1 radian on the diagram and $\hat{F}_p$ (units of m^3.Pa) is multiplied by the distance occupied by 1Pa.

The divergence of the flux (Δ) (XD field code 12) is calculated according to

$$\Delta = \frac{\partial \hat{F}_\phi}{\partial \phi} + \frac{\partial \hat{F}_p}{\partial p}$$

(in units of m^3), and is related to the divergence of the E-P flux $\underline{F}$ by

$$\int \nabla . \underline{F} dm = \int \Delta d\phi dp$$

where $dm = 2\pi R_e^2 \cos\phi d\phi dp/g$ is the mass of an annular ring of dimensions $d\phi$ and dp.

By following these conventions, the flux vectors, when plotted, **look** non-divergent if, and only if, $\nabla.\underline{F} = 0$ and in addition give the direction of the total flux of wave activity for the wave-like eddies (Edmon *et al.* (1980)).

Since the forcing of the zonal flow depends upon $\nabla.\underline{F}/R_e \cos\phi$, a drawback of this form of the E-P cross-section is that the zonal wind tendency must be obtained from the divergence of the E-P flux, Δ, by multiplying by a factor which depends upon latitude (ie $\nabla.\underline{F} = g\Delta/(2\pi R_e^2 \cos\phi)$).

E-P flux divergence

The horizontal and upward divergence components of the E-P flux are defined according to

$$\text{horizontal divergence} = \frac{\partial \hat{F}_\phi}{\partial \phi}$$

$$\text{upward divergence} = -\frac{\partial \hat{F}_p}{\partial p}$$

while the total divergence is the sum of these two components.

Square of the Brunt-Väisälä frequency

The square of the Brunt-Väisälä frequency ($\overline{N^2}$) is defined according to the equation

$$\overline{N^2} = \frac{g}{\overline{\theta}} \frac{\partial \overline{\theta}}{\partial z}$$

where $\overline{\theta}$ is the zonal-mean potential temperature. Expanding this gives the zonal-mean static stability in terms of the zonal-mean temperature ($\overline{T}$):

$$\overline{N^2} = -\frac{pg^2}{R_d T^2} \left(\frac{\partial \overline{T}}{\partial p} - \frac{\kappa \overline{T}}{p} \right)$$

where p is the pressure value of the output surface(s). The vertical derivatives are computed in terms of η.

Potential vorticity gradient

The following is based upon work by Edmons et. al. (1980), and upon notes by Mike Blackburn (unpublished). On a Cartesian β-plane, the zonal-mean north-south gradient of quasi-geostrophic potential vorticity ($\overline{q}_y$) is given by

$$\overline{q}_y = \beta - \overline{u}_{yy} + f \left(\frac{\overline{\theta}_y}{\overline{\theta}_p} \right)_p$$

where β is the planetary vorticity gradient, and the final term relates to vortex stretching associated with the variation of isentropic slopes with altitude. On pressure surfaces, assuming that the thermal wind equation holds for zonal-mean quantities, it may be shown that

$$\frac{\partial \overline{u}}{\partial p} = \frac{R_d}{fp} \frac{\partial \overline{T}}{\partial y}$$

reduces to

$$\frac{\partial \overline{u}}{\partial p} = \frac{R_d T}{p\theta} \frac{\partial \overline{\theta}}{\partial y}$$

and hence

$$\overline{q}_y = \beta - \overline{u}_{yy} + \frac{f^2}{R_d} \left(\frac{p\theta}{T} \frac{\overline{u}_p}{\overline{\theta}_p} \right)_p$$

(in units of $\mathrm{m}^{-1}\mathrm{s}^{-1}$).

On a sphere, the equation for the quantity $\partial\overline{q}/\partial\phi$ may be written

$$\overline{q}_\phi = 2\Omega \cos\phi - \left[\frac{1}{R_e \cos\phi} \frac{\partial(\overline{u}\cos\phi)}{\partial\phi} \right]_\phi + f \left[\frac{\overline{\theta}_\phi}{\overline{\theta}_p} \right]_p$$

Since

$$\frac{\partial \overline{u}}{\partial p} = \frac{R_d T}{f p R_e \theta} \frac{\partial \overline{\theta}}{\partial \phi}$$

the former equation becomes

$$\overline{q}_\phi = 2\Omega\cos\phi - [\overline{u}_y]_\phi + \frac{f^2 R_e}{R_d} \left(\frac{p\theta}{T} \frac{\overline{u}_p}{\overline{\theta}_p} \right)_p$$

(in units of $\mathrm{s}^{-1}\mathrm{rad}^{-1}$). Combining the first two terms on the right hand side of this equation gives

$$\overline{q}_\phi = [\,\overline{\zeta}\,]_\phi + \frac{f^2 R_e}{R_d} \left(\frac{p\theta}{T} \frac{\overline{u}_p}{\overline{\theta}_p} \right)_p$$

This quantity is the potential vorticity gradient multiplied by R_e. The quantity output by SUMO is actually $\overline{q}_{R_e\phi}$, where

$$\overline{q}_{R_e\phi} = [\,\overline{\zeta}\,]_{R_e\phi} + \frac{f^2}{R_d} \left(\frac{p\theta}{T} \frac{\overline{u}_p}{\overline{\theta}_p} \right)_p$$

Total permitted wavenumber (phase speed = 0)

(Note that the phase speed (a constant) may be adjusted by altering lines of code at the start of routine `XDFLUX`.) For further details of the total permitted wavenumber, the reader is referred to Hoskins and Karoly (1981).

The total permitted wavenumber (or "stationary wavenumber" if the phase speed is zero or neglected) is defined according to

$$T_{wn} = \sqrt{\frac{\overline{q}_{R_e\phi}\cos\phi}{\overline{u}_m - c_m}}$$

(units of m^{-1}), where the Mercator velocity $\overline{u}_m$ is given by the ratio $\overline{u}/\cos\phi$ and $\overline{c}_m$ is given by the ratio $\overline{c}/\cos\phi$, $\overline{c}$ being the phase speed of the wave.

In fact, SUMO outputs a scaled (dimensionless) version of T_{wn}, where the right hand side of the previous equation is multiplied by the earth's radius. In the event of either the denominator or the right hand side of the previous equation being negative, then T_{wn} will be assigned a value of 100.

Mean heat fluxes

The vertical mean heat flux is evaluated using the quantity

$$(\overline{\omega} - \overline{\omega}^{\phi\lambda})(\overline{T} - \overline{T}^{\phi\lambda})$$

while the horizontal mean heat flux is given by the term

$$\overline{v}(\overline{T} - \overline{T}^{\phi\lambda})$$

where $\overline{X}^{\phi\lambda}$ denotes the global mean of X on a given pressure surface.

Eddy moisture fluxes

The horizontal eddy moisture flux is given by the quantity

$$\overline{vq} - \overline{v}\,\overline{q}$$

while the vertical eddy moisture flux is evaluated, similarly, using the expression

$$\overline{\omega q} - \overline{\omega}\,\overline{q}$$

Transformed Eulerian-mean fields

This section is taken from notes by John Thuburn (unpublished).

Andrews *et. al.* (1987) (p. 128) give expressions for the TEM equations on spherical geometry, log-pressure form. Changing the variable from log-pressure to pressure, and defining a residual mean meridional circulation with components

$$\overline{u}^* = 0 \tag{10.1}$$

$$\overline{v}^* = \overline{v} - \frac{\partial}{\partial p}\left(\frac{\overline{v'\theta'}}{\overline{\theta}_p}\right) \tag{10.2}$$

$$\overline{\omega}^* = \overline{\omega} + \frac{1}{R_e\cos\phi}\frac{\partial}{\partial\phi}\left(\cos\phi\frac{\overline{v'\theta'}}{\overline{\theta}_p}\right) \tag{10.3}$$

where the overbar again denotes the zonal mean of a quantity, allows the TEM set of equations to be written as

$$\overline{u}_t - \overline{v}^*\left(f - \frac{1}{R_e\cos\phi}\frac{\partial}{\partial\phi}(\overline{u}\cos\phi)\right) + \overline{\omega}^*\frac{\partial\overline{u}}{\partial p} - \overline{X} = \frac{1}{R_e\cos\phi}\nabla.\underline{F} \tag{10.4}$$

$$\overline{u}\left(f + \frac{\overline{u}}{R_e}\tan\phi\right) + \frac{1}{R_e}\frac{\partial\overline{\Phi}}{\partial\phi} = G \tag{10.5}$$

$$\frac{\partial\overline{\Phi}}{\partial p} = -\frac{RT}{p} \tag{10.6}$$

$$\frac{1}{R_e\cos\phi}\frac{\partial}{\partial\phi}\left(\cos\phi\,\overline{v}^*\right) + \frac{\partial\overline{\omega}^*}{\partial p} = 0 \tag{10.7}$$

$$\overline{\theta}_t + \frac{\overline{v}^*}{R_e}\frac{\partial\overline{\theta}}{\partial\phi} + \overline{\omega}^*\overline{\theta}_p - Q = -\frac{\partial}{\partial p}\left(\frac{\overline{v'\theta'}\,\overline{\theta}_\phi}{R_e\overline{\theta}_p} + \overline{\omega'\theta'}\right) \tag{10.8}$$

(See equation (10.12) for the thermodynamic equation in terms of temperature.) Here, $\overline{X}$ is the zonal mean of unspecified frictional (or other non-conservative) forcing in the eastward direction, and the components of F are given by

$$F^\phi = R_e\cos\phi\left(\frac{\overline{v'\theta'}}{\overline{\theta}_p}\overline{u}_p - \overline{u'v'}\right) \tag{10.9}$$

$$F^p = R_e\cos\phi\left(\frac{\overline{v'\theta'}}{\overline{\theta}_p}\left[f - \frac{1}{R_e\cos\phi}(\overline{u}\cos\phi)_\phi\right] - \overline{\omega'u'}\right) \tag{10.10}$$

and

$$\nabla.\underline{F} = \frac{1}{R_e\cos\phi}\frac{\partial}{\partial\phi}(\cos\phi F^\phi) + \frac{\partial F^p}{\partial p} \tag{10.11}$$

Also,

$$G = -\overline{v}_t - \frac{\overline{v}}{R_e}\frac{\partial\overline{v}}{\partial\phi} - \overline{\omega}\frac{\partial\overline{v}}{\partial p} + \overline{Y} - \frac{1}{R_e\cos\phi}\frac{\partial}{\partial\phi}\left(\overline{v'v'}\cos\phi\right) - \frac{\partial}{\partial p}\left(\overline{\omega'v'}\right) - \frac{1}{R_e}\overline{u'u'}\tan\phi$$

(ie. the departure from gradient wind balance) with $\overline{Y}$ being used to denote the zonal mean of unspecified frictional (or other non-conservative) forcing in the northward direction. Note that the equations do not apply if any pressure surface intersects the ground.

The following quantities may be determined (providing that the time-averaged, not the instantaneous, fields have been selected) by SUMO (with data on surfaces that intersect orography being assigned a masked value):

TEM tendency fields – codes 20, 22 and 41

The time-mean tendency of the $\overline{u}$, $\overline{T}$and $\overline{v}$ fields between diagnostic times t_1 and t_2 are given by

$$(\overline{u}_2 - \overline{u}_1)/(t_2 - t_1)$$

(with similar expressions for the other two fields).

TEM residual circulation fields – codes 21, 26 and 27

Equations (10.2) and (10.3) define the quantities $\overline{v}^*$ and $\overline{\omega}^*$. Assuming that the variation of pressure with height (z) may be represented using

$$p = p_0 \exp^{-z/H} \text{ with } \frac{\partial p}{\partial z} = -p/H = -\rho g$$

where p_0 and H (= 7000m) are constants with ρ the density, and that the vertical component of the residual circulation (expressed in height coordinates) is given by

$$\overline{w}^* = -\overline{\omega}^*/(\rho g)$$

it can be shown that

$$\overline{w}^* = -\overline{\omega}^* H/p$$

TEM E-P flux fields – codes 30 – 34

Referring to equations (10.9) and (10.10), SUMO can output the fields F^ϕ/R_e (field 30) and F^p/R_e (field 31). The two components of the term on the right hand side of equation (10.4) may be output using field codes 32 and 33, while field 34 is the sum of these two. (In fact, it is -1 times these latter three fields that is actually output.)

TEM Zonal momentum budget terms – codes 28, 29 and 23

Fields 28 and 29 are, respectively, the fields

$$\overline{v}^* \left(f - \frac{1}{R_e \cos\phi} \frac{\partial}{\partial \phi} (\overline{u} \cos\phi) \right)$$

and

$$\overline{\omega}^* \frac{\partial \overline{u}}{\partial p}$$

Finally, $\overline{X}$ (field code 23) is determined as the residual obtained by summing the other terms in the horizontal momentum equation (10.4).

TEM Thermodynamic equation terms – codes 24 and 35 – 38

The thermodynamic equation may be written in terms of temperature (rather than potential temperature) as

$$\frac{\partial \overline{T}}{\partial t} + \frac{\overline{v}^*}{R_e} \frac{\partial \overline{T}}{\partial \phi} + \overline{\omega}^* \frac{\partial \overline{T}}{\partial p} - \frac{\kappa \overline{\omega}^* \overline{T}}{p} + \frac{\overline{T}}{\overline{\theta}} \frac{\partial}{\partial p} \left\{ \frac{\overline{v'\theta'}}{\overline{\theta}_p} \frac{\overline{\theta}_\phi}{R_e} + \overline{\omega'\theta'} \right\} = \overline{H} \tag{10.12}$$

The first three terms on the left hand side of equation (10.12) may be output using field codes 22, 35 and 36 respectively. Field 37 corresponds to

$$-\frac{\kappa \overline{\omega}^* \overline{T}}{p}$$

while the eddy terms

$$\frac{\overline{T}}{\overline{\theta}} \frac{\partial}{\partial p} \left\{ \frac{\overline{v'\theta'}}{\overline{\theta}_p} \frac{\overline{\theta}_\phi}{R_e} + \overline{\omega'\theta'} \right\}$$

may be output using field code 38. Finally, the residual of the thermodynamic equation (H – field code 24) is determined by summing the left hand side of equation (10.12).

TEM Gradient wind balance – codes 39, 40 and 42

The first two terms of equation (10.5) may be output using field codes 40 and 42 respectively, while the sum of thee two (ie. the departure of the zonal-mean flow from gradient wind balance) can be output by specifying field code 39.

TEM Meridional momentum equation terms – codes 25 and 40 – 47

The meridional momentum equation (10.5) may be re-written in the form

$$\begin{aligned}\frac{\partial \overline{v}}{\partial t} + \overline{u}\left(f + \frac{\overline{u}}{R_e}\tan\phi\right) + \frac{\overline{v}}{R_e}\frac{\partial \overline{v}}{\partial \phi} + \overline{\omega}\frac{\partial \overline{v}}{\partial p} + \frac{1}{R_e}\frac{\partial}{\partial \phi}\left(\overline{v'v'}\right) \\ + \frac{\tan\phi}{R_e}\left(\overline{u'u'} - \overline{v'v'}\right) + \frac{\partial}{\partial p}\left(\overline{\omega'v'}\right) + \frac{1}{R_e}\frac{\partial \overline{\Phi}}{\partial \phi} = \overline{Y}\end{aligned} \tag{10.13}$$

Note that the two terms

$$-\frac{1}{R_e\cos\phi}\frac{\partial}{\partial\phi}\left(\overline{v'v'}\cos\phi\right) - \frac{1}{R_e}\overline{u'u'}\tan\phi$$

on the right hand side of the definition of G become undefined near the poles, so the sum of these terms is rearranged to give

$$-\frac{1}{R_e}\frac{\partial}{\partial\phi}\left(\overline{v'v'}\right) + \frac{1}{R_e}\left(\overline{v'v'} - \overline{u'u'}\right)\tan\phi$$

Now both these terms tend to zero at the poles.

The individual terms on the left hand side of equation (10.13) may be output using field codes 41, 40, 43, 44, 45, 47, 46 and 42 respectively. The residual term of this equation, $\overline{Y}$, evaluated by summing the left hand side, may be output using field code 25.

10.4 Summary of options available

Table 2.22 illustrates the options available to the user when setting up the program to compute the XD diagnostics. These are now explained in a little more detail, together with some of the possible pitfalls. Reference is made in this section to the XD Fortran namelists variables, rather than to the XD data block keywords. Use should be made of the data pre-processor, which will help to eradicate inconsistent input data, before running the diagnostic program.

When requesting XD diagnostics, the user should ensure that the `LEVELTYPE` keyword in the COMMON data block is set to `IS`, as the calculations are only valid in their current form for data on pressure surfaces. The XD calculations are carried out on fields that have previously been interpolated to pressure surfaces. The `DIAGNOSTICS` keyword in the COMMON data block must be set to `XD` to enable the XD diagnostic calculations.

The XD diagnostics compute any required zonal means internally to the XD routines; these are determined arithmetically (ie. without reference to land-air masks). However, if masks have been used to determine the interpolated fields, then the effects of such masks will permeate through to the XD zonal-mean fields.

Referring to the XD data block, the user then has a choice of instantaneous and/or time-averaged output; these are enabled by setting `LDXDIN` and `LDXDTD` to `.TRUE.` respectively. Note that the TEM fields are only available if time-averages have been selected since SUMO needs to determine the rate of change of certain quantities in the TEM equations.

Output may be in the form of printed output and /or UTFs or DRS or NetCDF files. Logical switches `LDXDZM` and `LDXDLM` need to be set to `.TRUE.` to generate the zonal-mean fields (described in the previous section) or the level-mean of these fields, respectively. Note that DRS and NetCDF output is not available for the level-mean fields.

Arrays `NXDPR` and `NXDPL` specify which of the fields in Table 10.1 are required; the user must also assign a value to `NXD` – the total number of different fields requested. Printed arrays are output to channel `NCXD` while array `CINTXD` defines the contour interval to be passed to the UTF (if appropriate) for each of the fields specified by `NXDPL`.

11 References

Andrews, D.G., Holton, J.R. and Leovy, C.R. (1987) Middle Atmosphere Dynamics. Academic Press. 489pp.

Blackburn, M., Carver, G. and Thuburn, J. (1992) UGAMP Transfer File (UTF) Format Description: version 1.4. *UGAMP Internal Report No. 21*, 19pp.

Blackburn, M., Gray, L., Shine, K.P., and Thuburn, J. (1989) Middle atmosphere extension of the ECMWF model: progress to date. *UGAMP Technical Report No. 5*, 15pp.

Blackmon, M.L., Wallace, J.M., Lau, N.-C. and Mullen, S.L. (1977) An observational study of the Northern Hemisphere wintertime circulation. *J. Atmos. Sci.*, **34**, 1040–1053.

Bolton, D (1980) The computation of equivalent potential temperature. *Mon. Wea. Rev.*, **108**, 1046–1053.

Edmon, H.J., Jr., Hoskins, B.J. and McIntyre, M.E. (1980) Eliassen-Palm cross sections for the troposphere. *J. Atmos. Sci.*, **37**, 2600–2616.

Eliassen, A. and Palm, E. (1961) On the transfer of energy in stationary mountain waves. *Geofys. Publ.*, **22**, No. 3, 1–23.

Haseler, J. (1979) The forecast and analysis post-processing package. European Centre for Medium Range Weather Forecasts, Research Department, Technical Memorandum No. 2.

Hoskins, B.J. and D.J. Karoly (1981) The steady linear response of a spherical atmosphere to thermal and orographic forcing.

Lary, D.J. (1993) Initialising tracers in three-dimensional models using an equivalent PV latitude–θ coordinate system. (*Available from the author, Dept. of Chemistry, University of Cambridge, Lensfield Rd, Cambridge CB2 1EW, UK; email: david@atm.ch.cam.ac.uk*)

Lary, D.J., Chipperfield, M.P., Pyle, J.A., Norton, W.A. and Riishøjgaard, L.P. (1994) Three dimensional tracer initialisation and general diagnostics using equivalent PV latitude – potential temperature coordinates. (*To be submitted to* JGR *in 1994.*)

Lau, N.-C., White, G.H. and Jenne, R.L. (1981) Circulation statistics for the extratropical Northern Hemisphere based on NMC analyses. *NCAR Tech. Note Tn-171+STR*, NCAR, Boulder, Colorado, USA, 138pp.

Lucic, D. (1991) Adding column amounts to the FLUX program. *UGAMP Internal Report No. 18*, 5pp.

Machenhauer, B. and Daley, R. (1972) A baroclinic primitive equation model with a spectral representation in three dimensions. Report No. 4, University of Copenhagen, Institute of Theoretical Meteorology.

Sardeshmukh, P.D. and Hoskins, B.J. (1984) Spatial filtering on the sphere. *Mon. Wea. Rev.*, **112**, 2524–2529.

Sardeshmukh, P.D. and Hoskins, B.J. (1988) The generation of global rotational flow by steady idealized tropical divergence. *J. Atmos. Sci.*, **45**, 1228–1251.

Steenman-Clark, L. (1993) UGAMP programming standards. I. Fortran code. *UGAMP Internal Report No. 30*, 17pp.

White, G.H. (1983) The global circulation of the atmosphere December 1981 – November 1982 based on ECMWF analyses. *Technical Report, University of Reading*, 162pp.