JCMM

MDIAG:
a Fortran 90 program to compute diagnostics on pressure levels from Unified Model data on model levels

Peter Panagi

June 2000

INTERNAL REPORT NO.117

NWP Technical Report No.313

Joint Centre for Mesoscale Meteorology

MDIAG
A FORTRAN 90 PROGRAM TO COMPUTE DIAGNOSTICS ON PRESSURE LEVELS FROM UNIFIED MODEL DATA ON MODEL LEVELS

Peter Panagi
JCMM, Univ. Reading
PO Box 243
Reading RG6 6BB
0118 9318794
p.panagi@reading.ac.uk
ppanagi@meto.gov.uk

June 16, 2000

Abstract

We present the top-down design, coding and implementation of the suite of diagnostic sub-programs, MDIAG V7.0.

The code uses Unified Model data on model levels to compute a number of user-specified diagnostics, after interpolation onto pressure levels.

Contents

List of Figures

Chapter 1

INTRODUCTION

MDIAG is a suite of subroutines that enable the user to compute a number of diagnostics, on pressure levels, from Unified Model data on model levels. The program(s) can work at all horizontal and vertical resolutions, for data on a spherical grid (either true or rotated).

History: Originally, MDIAG was developed at JCMM, at the University of Reading, by Tim Hewson and Sid Clough. The objective was to readily compute a large number of diagnostics useful to their research. The input data were from the operational UM limited area model (LAM, at 60 km horizontal resolution) domain, on pressure levels, transformed onto a polar stereographic grid and averaged onto a coarser (100km) resolution. The required input data were temperature, relative humidity, horizontal winds and ω. Diagnostics were stored as ASCII files and could be viewed with the graphics tool DISP (Roberts, 1998), also developed at JCMM. The code was written in Fortran 77 and ran on VAX VMS machines. The code ran four times a day, on output from each of the 4 operational forecast runs, producing diagnostics for hardcopy and for the Met. Office internal network. Archives of the data, dating back as far as 1992, are held at JCMM (Panagi & Dicks, 1997).

The original Fortran 77 formulation was based around a large 3-dimensional array, which held all the diagnostics at each time-step. This was fine at the coarse 100km resolutions, where the domain was 61x38 horizontally and 10 levels vertically (1000mb to 100mb every 100mb). Each point in the domain was represented by a single ASCII character (one byte), which limited the range of values that a diagnostic could take, but nevertheless led to manageable file sizes even with a large (up to 70) number of diagnostics.

The code was then re-written (also in Fortran 77), optimised and ported to both VMS and HP unix systems, but now using UM data on model levels

(Panagi & Dicks, 1997). Also, importantly, the code was made to work on the UK Mesoscale domain, a rotated latitude-longitude grid, as the number of users working on Mesoscale research was increasing. Input data were now taken directly from the UM operational prognostic variables, horizontal winds, ω, specific humidity, cloud liquid water and cloud ice, as well as temperature and orographic height.

Diagnostic output could be viewed with the IDL-based graphics utility JPLOT (Panagi & Dicks, 1997), written and supported by Ed Dicks in the department of Meteorology at the University of Reading.
Later revisions were also made, with further optimisation and porting to SUN-OS operating systems, to support the UK academic community as a whole, and the addition of a number of user-requested diagnostics. Further detailed information on these versions is available on the WWW,

`http://www.met.reading.ac.uk/~panagi/uwern/mdiag`

This was the state of MDIAG as of the beginning of this project.

Limitations of the current configuration: The Unified Model has increasingly changed to higher vertical and horizontal resolutions and sometime in 2000 the operational UM will move to a semi-Lagrangian integration scheme. The resolution changes have precipitated the need for high-powered workstations with memory in excess of 1 Gb, to enable the post-processing. The current Global UM (used for forecasts) utilises a 432x325x31 grid (50km resolution). The arrays needed to process data of this size are too large to fit in even the largest of workstations.
The limited area domains of the UM (LAM, UK Mesoscale) could in fact fit into the computers main memory only with large amounts of primary and secondary (swap) memory.

To maintain code between the Met. Office's HP w/stations and the Universities' SUN w/stations required that the code to be written with memory allocated statically. This was because the Fortran 77 standard does not allow the dynamic allocation of heap memory - only stack memory can be allocated dynamically in a certain manner from a called subroutine, and this is also limited by manufacturer specific operating systems, usually to less than 100Mb. Even though the HP compilers allowed dynamic allocation as an extension, the SUN-OS compiler did not. The problems with memory allocation had been overcome to some extent by compiling with the **-Bstatic** (for SUN-OS) or **-K** (for HP-UX) option, which allocates the required amounts of memory at run-time from the heap, but MDIAG at higher resolution (e.g. global operational) was still too big to fit into, and run in, memory.

Fortunately, Fortan 90 allows the dynamic allocation of heap memory. Also, with the addition of much simpler-to-code array calculations, pointer arithmetic and parameter passing between routines, it was decided to re-write the code in Fortran 90 from scratch, with the emphasis on readability and flexibility and to use as many of the new features as possible to improve the efficiency and speed of MDIAG.

Chapter 2

THE DESIGN PROCESS

A questionnaire was sent out to existing MDIAG users based primarily at the University of Reading, to ascertain what features of the existing MDIAG should be retained, those which were little used, and any other additions that may be useful to include. A summary of the responses led to the following design methodology.

LEVEL 0: Statement of the problem.

To enable the computation of user-specified diagnostics on pressure surfaces from data obtained from the operational Unified Model on model-levels.

LEVEL 1: Diagnostics and format details.

1. The diagnostics are to be taken from a pre-determined set The set has evolved over the years to the current number of about 100, but we would like the ability to readily add new diagnostics and so the final formulation should be sufficiently flexible to handle this.

2. The input data will be on model levels and the output data will be on pressure levels.

3. The format of the input and output data is pp-format. This is to enable the data to be displayed easily with JPLOT. We may relax the criterion of solely pp-format, to allow say netCDF formats, subject to user requirements, but we must enable pp-format at least.

4. Continuity with the existing Fortran 77 version (V6.1) of MDIAG.

5. The programs should work on any lat/long grid.

6. The UM will move to a semi-Lagrangian grid sometime in 2000 (the new dynamics), and so the formulation may take this into consideration. **At this stage, we will focus on the existing *eta*-level model grid, and defer implementation of new dynamics diagnostics until a later date.**

   Additional user requests:

7. Interpolation of arbitrary model-level fields onto pressure surfaces.

8. Ability to specify the actual levels onto which the computations will be made, as well as the specification of the levels on which data will be saved - this allows for faster computation and smaller data files.

9. Simple pre-smoothing of input data to remove high frequency noise, prior to diagnostics computations.

<u>LEVEL 2:</u>What needs to be done to implement LEVEL 1

1. Produce a list of pre-coded diagnostics.

2. Re-map the required input UM data on model levels onto pressure levels. Also, perform any interpolations so that all data is on the same horizontal grid.

3. Compute the diagnostics on the specified pressure levels and output on specified pressure levels, in the required format.

Chapter 3

MDIAG STRUCTURE

MDIAG is written as a program of seven MODULES, each CONTAINing INTEGER and REAL variable declarations and SUBROUTINEs to perform the necessary computations. After much thought and experiment it was decided to make the basic variables required for nearly all the computations available through the MODULE REQUIRED_FIELDS. Modules have replaced Fortran 77 COMMON blocks as the method of statically passing variables and parameters to subroutines. An added advantage of USEing (*sic*) MODULES is that the passing of variables to subroutines can be checked at compile-time, minimising the number of erroneous calls.

All the arrays in this module are ALLOCATABLE in that their sizes are determine at run-time, after reading the necessary grid information from the input file header. Indeed, we make use of ALLOCATEable (*sic*) arrays throughout the program, as this feature enables the use of extremely large arrays (grids). Figure 3.1 depicts the MODULE structure and how modules USE other modules. (The MODULE MDIAG USEs the three MODULEs DIAGNOSTICS_TO_COMPUTE, SETUP and READ_AND_INTERPOLATE, the MODULE PP_ROUTINES is common to these three, and the MODULE REQUIRED_FIELDS USEs ALL_CONSTANTS and itself is USEd by DIAGNOSTICS_TO_COMPUTE.)

3.1 Computing sequence

3.1.1 Coarse top-down computing sequence

MDIAG works essentially as follows:

1. Open the input file and read in the grid information from the file header

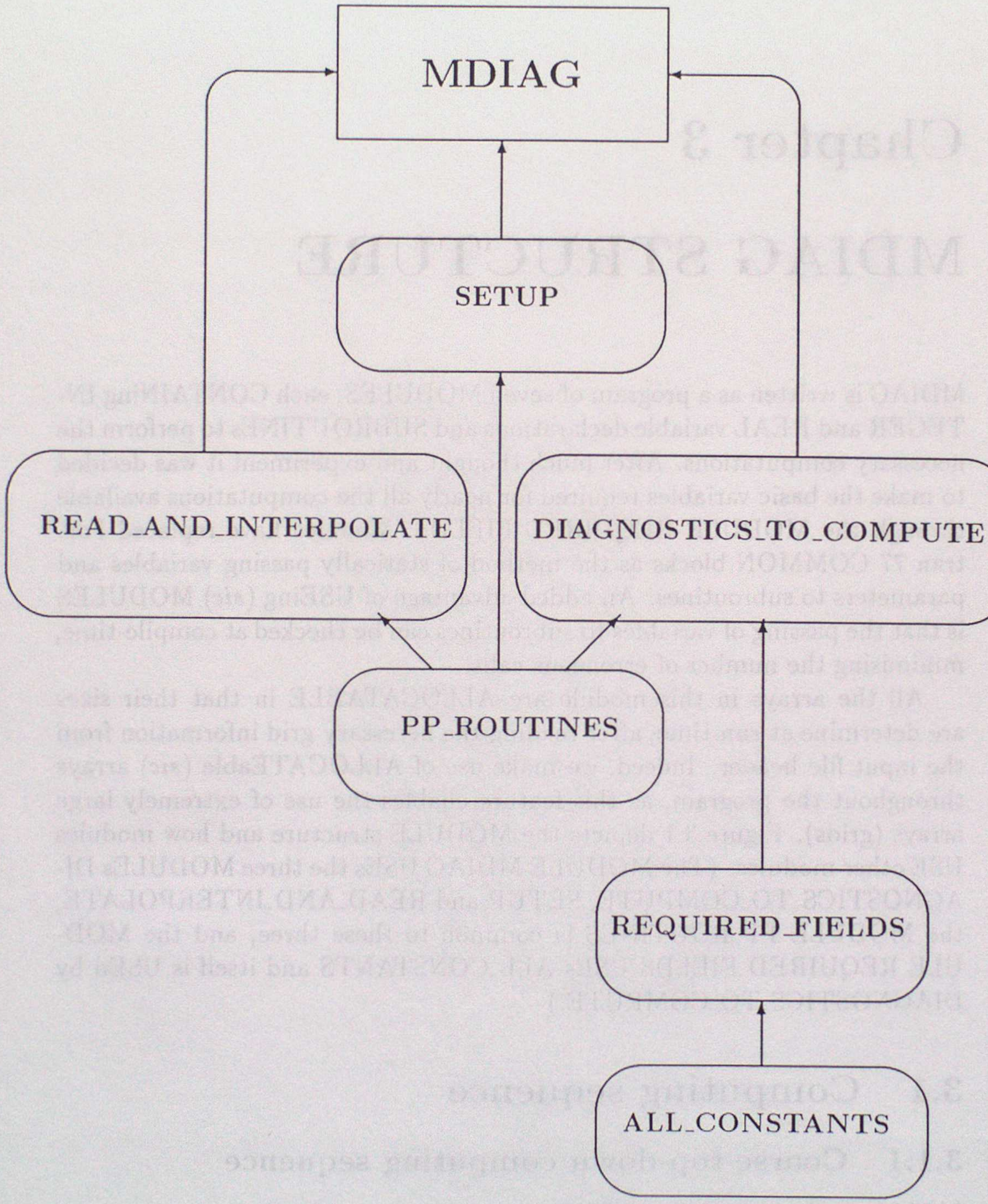

Figure 3.1: Schematic depicting the seven Fortran 90 MODULES in MDIAG and their hierarchial USE.

2. Open the file that contains information on the diagnostics to be computed as
3. LOOP-OVER-TIME:
   - read the basic prognostic variables required for the diagnostic calculations
   - interpolate these fields between model-levels and pressure levels as necessary
   - calculate the requested diagnostics and save to output file

Figure 3.2 is a coarse schematic of the top-down computing sequence

3.1.2 Detailed top-down computing sequence

Here we describe in more detail the computing sequence outlined in Figure 3.2.

1. Get the parameters of the domain (to dimension arrays dynamically) from the input pp-format file MDIAG_INPUT.PP. These will be
   - **NX, NY, NPP** which determine the horizontal domain (NX,NY) and the number of model η-levels (NPP)
   - A_k, $A_{k+1/2}$, B_k, $B_{k+1/2}$, values for each level k and half-level $k+1/2$, that are used to determine the pressure at each level (according to equation 4.2)
   - **LATITUDE, LONGITUDE** wrt the grid, **POLE_LAT, POLE_LON** if the grid is equatorial, to enable us to determine the true latitudes and longitudes.
2. Read the file **MDIAG.DIAGNOSTICS** (full listing in Appendix C). From this file, we determine the following parameters (in sequence)
   - The pressure levels onto which the computations are to be made e.g.

     ```
     1000,50,50
     ```

     would indicate that the diagnostics were to be calculated in the pressure range 1000 mb to 50 mb every 50 mb
   - The sub-grid onto which the data will be output, in the order top-left latitude and longitude and bottom-right latitude and longitude e.g.

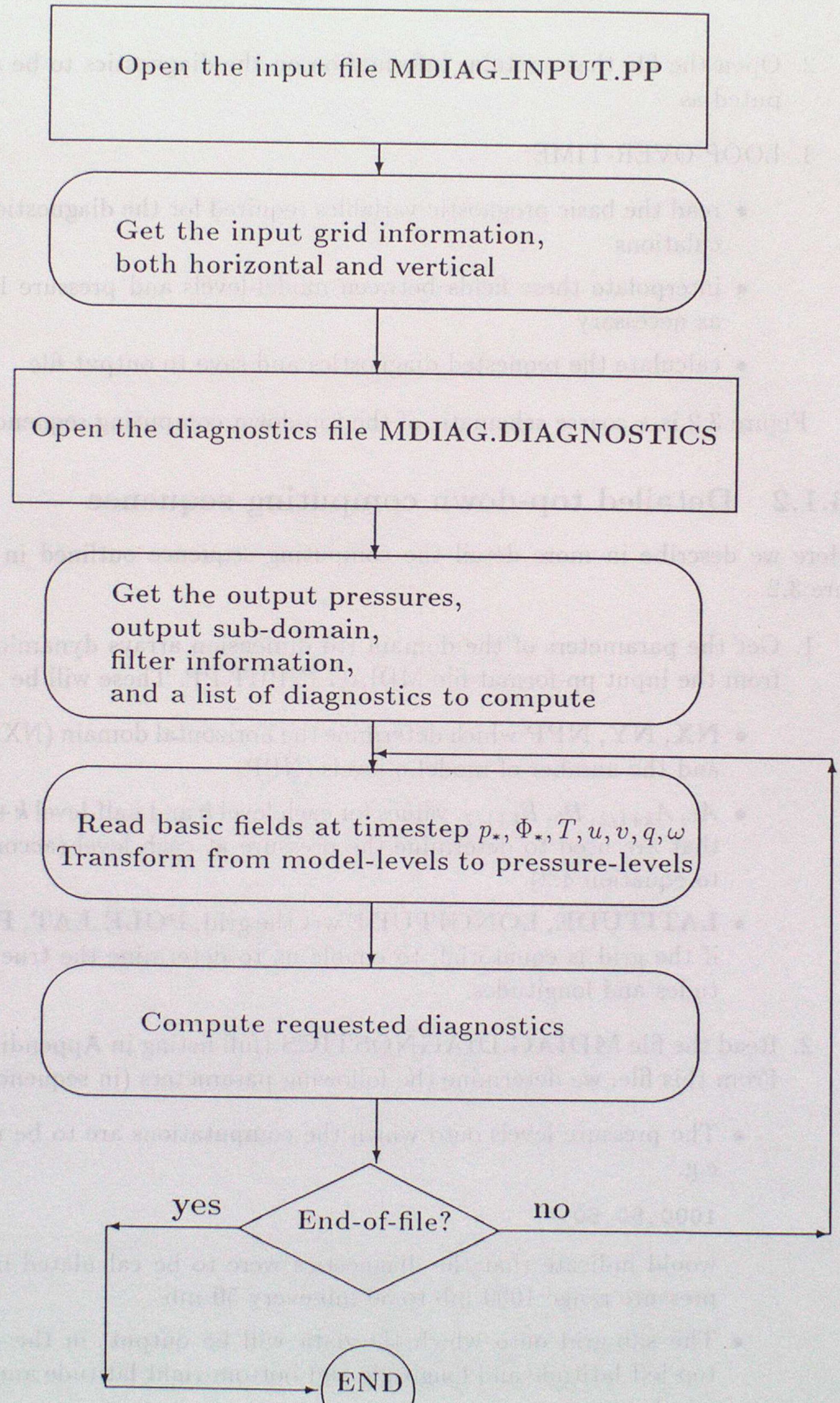

Figure 3.2: Coarse top-down computing sequence for MDIAG. This shows the main computations required to execute a successful run

```
40.0,-40.0,-40.0,40.0
```

indicates a box whose top left-hand corner starts at 40.0 deg latitude and −40.0 deg longitude (i.e. 320 deg) and bottom right-hand corner is at −40.0 deg latitude and 40.0 deg longitude. The program calculates the start and end points from information in the pp-file header. If the grid is an equatorial grid, such as the UK Mesoscale domain, MDIAG calculates the corner points by converting the coordinates of the output domain to an equatorial grid and then extracting the start and end points. **NB** It should be noted that there is no support at present for plotting data from sub-domains of rotated grids, within JPLOT.
If the sub-domain is coded as

```
0.0,0.0,0.0,0.0
```

it assumed that output is over the entire domain.

- The number of times that a linear bi-directional filter is to be applied to the basic variables (geopotential height, temperature, u-wind, v-wind, ω and specific humidity. This has the obvious effect of pre-smoothing the diagnostics calculations, which may or may not be desirable, depending on the user's science. Details are given under Section 5.5.
- The diagnostics to calculate. These are described in as much detail as possible under Section 4.

3. Compute the **CORIOLIS** parameter and $\cos\phi$ terms that will be used extensively later. In fact, as the CORIOLIS parameter goes to zero at the equator, we apply a cut-off at ±0.01 deg, computing the value at a fixed 0.01 deg within this range. This is required as the geostrophic wind calculations (equation 4.18) require division by the CORIOLIS parameter.
4. Save the LATITUDES and LONGITUDES if they have been requested through MDIAG.DIAGNOSTICS
5. Read in the prognostic variables, $\Phi_*, T, u, v, \omega, q$, as well as cloud liquid ice and cloud water contents (for continuity with the existing version of MDIAG) at one time, and interpolate between model-levels and requested pressure-levels according to Section 4
6. Compute, in addition, the potential temperature θ, equivalent potential temperature θ_e and wet-bulb potential temperature θ_w. These diagnostics occur repeatedly in the computations of other diagnostics, so we

compute them once and pass them to other routines *via* the MODULE REQUIRED_FIELDS. The θ_w computations are very CPU intensive, so the one-off computation at each time-step cuts down on the overall execution time.

7. Calculate each requested DIAGNOSTIC and save it to the output pp-format file

8. Repeat the procedure at the next time step until the end if the file is reached

If any prognostic variable is not present in the input file, the program continues but does not make any calculations using the missing variable. This is controlled by the INTEGER variable **IFAIL**, whose value is tested in each diagnostics subroutine prior to initiating a computation (IFAIL=1 indicates that a variable is missing).

3.1.3 Unusual behaviour - IMPORTANT:

There will obviously be problems near the equator in computing a quantity that requires division by the CORIOLIS parameter f, such as the geostrophic winds. Hence we make the following assumptions:

$$\|f\| \leq 0.01, \qquad f = 0.01$$

In addition, some of the frontal parameters (described later) divide by the components of the wind, so we take also make the following assumptions

$$\|u\| \leq SMALL, \qquad u = SMALL$$

where $SMALL$ is defined by the FORTRAN 90 command

$$\mathrm{SMALL} = \mathrm{EPSILON}(1.0)$$

and so is the smallest available real on a given platform.

3.2 Important common and control variables

Appendix D.2 lists the MODULE REQUIRED_FIELDS. This contains the variables that are common to all the subroutines as well as other control variables. Here we describe the important variables that control the flow of the program.

NUX	No. points in λ-direction on the staggered grid
NUY	No. points in ϕ-direction on the staggered grid
NX	No. points in λ-direction on the pressure grid
NY	No. points in ϕ-direction on the pressure grid
NPP	No. vertical model η-levels
NZ	No. vertical pressure levels onto which data will be interpolated
PRESSURE	The vertical (constant) pressure grid - dimensioned as NZ
DELTA_P	Constant pressure difference between levels
SAVEIT	Array of logicals that determine which diagnostics are required
DLAMBDA	$\Delta\lambda$ - constant spacing in longitude
DPHI	$\Delta\phi$ - constant spacing in latitude
CORIOLIS	Coriolis parameter at all latitudes
THELON	Array of true longitudes of all points
THELAT	Array of true latitudes of all points
EQLON	Array of rotated system longitudes of all points
EQLAT	Array of rotated system latitudes of all points
PHISTAR	Array of orographic height
GEOP_HT	Geopotential height on pressure surfaces
TEMPERATURE	Temperature on pressure surfaces
UWIND	u-component of the wind *wrt* the grid on pressure surfaces
VWIND	v-component of the wind *wrt* the grid on pressure surfaces
SPECIFIC_HUMIDITY	specific humidity q on pressure surfaces
OMEGA	$\omega = \frac{dp}{dt}$ on pressure surfaces
THETA	potential temperature θ on pressure surfaces
THETAE	equivalent potential temperature θ_e on pressure surfaces
THETAW	wet-bulb potential temperature θ_w on pressure surfaces

3.3 Input and output

MDIAG takes input data from any of the Unified Model latitude/longitude and rotated (equatorial) latitude/longitude domains - they are both spherical coordinates systems (transformations between equatorial and true lat/lon systems are given in Goddard (1998) eqs. 5.14,5.19.,5.20). They are equivalent systems for computational purposes, so that, as long as the input parameters are with respect to the grid, the metric terms that arise in the derivatives can be calculated in the same manner with the same piece of code.

NB The only exception(s) is the calculation of any position-dependent force(s), such as the **CORIOLIS** parameter, which must be calculated on the true lat/lon. grid.

Documentation on many aspects of the Unified Model can be found in The Met. Office (1998) (see the bibliography), as well as

http://www.met.reading.ac.uk/~panagi/umdocs

All the diagnostics that have been requested can be derived from the basic prognostics variables of the Unified Model, namely

p_*	Surface pressure	Pa
Φ_*	Orographic height	m
T	Temperature	K
u	u-wind	$m\,s^{-1}$
v	v-wind	$m\,s^{-1}$
ω	rate of change of pressure	$Pa\,s^{-1}$
q	specific humidity	$kg\,kg^{-1}$

The variables are held on model-levels (η), levels that are terrain following near the surface and almost constant pressure surfaces at higher altitudes. The input data are transformed onto a regular pressure grid, specified by the user and then the diagnostics computed. Both input and output formats of the data files is pp-file format (Appendix B), where the data are stored as Fortran unformatted data files, sequentially in time. Each field is stored 2-dimensionally (either η for input or pressure (mb) for output), as header followed by data. The header is composed of two arrays, INTEGER and REAL. As an example of FORTRAN 90 code to OPEN and read a pp-field, we would access the data as follows

```
INTEGER, DIMENSION(45)                  :: IPP
REAL,    DIMENSION(19)                  :: RPP
REAL,    DIMENSION(:,:), ALLOCATABLE :: DATA

OPEN(UNIT=1,FILE='INPUT.PP',FORM='UNFORMATTED',STATUS='OLD',ACTION='READ')

READ(1)IPP,RPP
NX = IPP(19)
NY = IPP(18)
ALLOCATE(DATA(NX,NY), STAT= ioerror)
IF(ioerror /= 0) EXIT
READ(1)DATA
```

This block of code dynamically allocates memory from the heap for the DATA array, so it can be used in principle for any resolution. The values which are represented by the different array elements of the INTEGER and REAL arrays is described in more detail in Appendix B.

3.4 UNIX scripts to compile and run MDIAG

MDIAG has been successfully built and tested under SUN-OS 5.7 and HP-10.20. The compilers for the SUN and HP platforms are different, producing different size executables.
For SUN-OS, we have used the Fujitsu **frt** compiler and for the HP the manufacturers own Fortran 90 compiler. Both compilers take different options, and care needs to be exercised if another compiler is used because of the modular structure of MDIAG. To compile, use the makefile and issue the command

make mdiag

which should result in an executable of the order of 500 Kb in size.
Below we list the makefile

```
FORT = f90

# In here go the OPTimisation parameters for the compiler
#   For the Fujitsu frt compiler OPT = -c -Am
#   For the HP native compiler   OPT = -c +O3
OPT = -c -Am

# Change the SUFFIXES inference to f90 and o(bject) files
# HP make does not recognise f90 as a SUFFIX whereas SUN does
.SUFFIXES:
.SUFFIXES: .f90 .o

.f90.o:
${FORT} ${OPT} $<

mdiag: mdiag.o
${FORT} -o mdiag *.o

clean:
\rm *.o

ALL_CONSTANTS.o:

REQUIRED_FIELDS.o: ALL_CONSTANTS.o

PP_ROUTINES.o: REQUIRED_FIELDS.o
```

```
SETUP.o: PP_ROUTINES.o

READ_AND_INTERPOLATE.o: PP_ROUTINES.o

DIAGNOSTICS_TO_COMPUTE.o: PP_ROUTINES.o

mdiag.o: SETUP.o READ_AND_INTERPOLATE.o DIAGNOSTICS_TO_COMPUTE.o
```

Chapter 4

DIAGNOSTIC FORMULAE AND CODES

In this section we detail, where possible, the derivation of each diagnostic, including the required input parameters.

Each diagnostics is accorded a unique code enabling plotting packages such as **jplot** written by Ed Dicks (Panagi & Dicks, 1997) to differentiate between different fields. We use **STASH** codes, as detailed in Appendix B, starting at 90001.

Computations are performed in SI units, but output is on pressure surfaces measured in mb and **NOT** Pa**scals** - this is for continuity with the existing Fortran 77 version of MDIAG (V6.1).

We assume standard notation for constants, which are all available through the MODULE ALL_CONSTANTS (Appendix D.1).

90001: Geopotential height Z - $[m]$

SUBROUTINE GeopHtModelToPressure

Required parameters: orography Z_* (STASH=33), surface pressure p_* (STASH=1), temperature on model-levels $T(\eta)$ (STASH=4004), specific humidity on model-levels $q(\eta)$ (STASH=4010), A, B coefficients at full model levels (η_k) and model half-levels $(\eta_{k-\frac{1}{2}})$.

The height Z_i, at a pressure p_i, <u>within</u> a layer k, bounded at $k-1/2, k+1/2$, is given by Goddard (1998, eq. 3.6),

$$\begin{aligned} Z_i \;=\; Z_* + \frac{c_p}{g}\Big\{ & \textstyle\sum_{m=1}^{k-1} (\theta_v)_m \left(\pi_{m+\frac{1}{2}} - \pi_{m-\frac{1}{2}}\right) + \\ & (\theta_v)_k \left[\pi_{k-\frac{1}{2}} - \pi_k\right] - \\ & \tfrac{1}{2}\left(\frac{\partial\theta}{\partial\pi}\right)_k \left[\pi_i\left(\pi_i - 2\pi_k\right) - \pi_{k-\frac{1}{2}}\left(\pi_{k-\frac{1}{2}} - 2\pi_k\right)\right] \Big\} \end{aligned} \tag{4.1}$$

where Z_* is the height above sea-level at the surface, and the **EXNER** pressure is defined as

$$\pi = \left(\frac{p}{100000}\right)^{\kappa}$$

so that

$$\theta = \frac{\pi}{T}$$

Further,

$$\pi_k = \frac{\pi_{k+\frac{1}{2}} p_{k+\frac{1}{2}} - \pi_{k-\frac{1}{2}} p_{k-\frac{1}{2}}}{(\kappa+1)\left(p_{k+\frac{1}{2}} - p_{k-\frac{1}{2}}\right)}$$

and the pressure at half-levels $k+\frac{1}{2}$ is given in terms of the surface pressure p_* and A and B coefficients according to

$$p_{k+\frac{1}{2}} = \left(A_{k+\frac{1}{2}} - A_{k-\frac{1}{2}}\right) + \left(\left(A_{k+\frac{1}{2}} - A_{k-\frac{1}{2}}\right) p_* \qquad (4.2)$$

In addition, we have the derivative

$$\left(\frac{\partial\theta}{\partial\pi}\right)_k = \frac{1}{\pi_k}\left[\frac{T_{kp} - T_{km}}{\pi_{kp} - \pi_{km}} - \theta_k\right], km = max(1, k-1), kp = min(top, k+1)$$

and the virtual potential temperature θ_v, in terms of the temperature T and specific humidity q is given by

$$\theta_v = \frac{T\left(1 + \left(\frac{1}{\epsilon} - 1\right) q\right)}{\pi}$$

Where the required pressure is would be below ground, we assume a constant lapse rate below ground of

$$\gamma = 0.0065 K\ m^{-1}$$

and use the following

$$Z_i = Z_* + \left(\frac{T_s}{\gamma}\right)\left[1 - \left(\frac{p_i}{p_*}\right)^{\frac{\gamma R}{g}}\right] \qquad (4.3)$$

where the surface temperature is calculated according

$$T_s = T_5 \left(\frac{p_*}{p_5}\right)^{\frac{\gamma R}{g}}$$

and the subscript 5 refers to the 5th level (when there are 4 levels in the boundary layer).

Once we have Z_i at pressure p_i, we can interpolate onto a regular pressure grid using a logarithmic variation in pressure according to **equation 5.2**.

90002: Temperature T - $[K]$

SUBROUTINE TemperatureModelToPressure

Required parameters: surface pressure p_* (STASH=1), temperature on model-levels $T(\eta)$ (STASH=4004), specific humidity on model-levels $q(\eta)$ (STASH=4010), A, B coefficients at full model levels (η_k) and model half-levels $(\eta_{k-\frac{1}{2}})$.

The temperature T_i at pressure p_i is calculated according to whether p_i falls in the top or bottom half of layer k:

$$T_i = \begin{cases} T_k + \dfrac{(T_{k-1}-T_k)\theta_k(\pi_i-\pi_k)}{\theta_k\left(\pi_{k-\frac{1}{2}}-\pi_k\right)+\theta_{k-1}\left(\pi_{k-1}-\pi_{k-\frac{1}{2}}\right)} & \pi_{k-\frac{1}{2}} \geq \pi_i \geq \pi_k \\ T_k + \dfrac{(T_{k+1}-T_k)\theta_k(\pi_k-\pi_i)}{\theta_k\left(\pi_k-\pi_{k+\frac{1}{2}}\right)+\theta_{k+1}\left(\pi_{k-\frac{1}{2}}-\pi_{k+1}\right)} & \pi_k \geq \pi_i \geq \pi_{k+\frac{1}{2}} \end{cases} \tag{4.4}$$

where $\theta\,\pi$ potential temperature and exner pressure have been previously defined for the geopotential height (90001). Extrapolation above the top level is

$$T_i = T_{top} + \frac{(T_{top}-T_{top-1})\,\theta_{top}\,(\pi_{top}-\pi_i)}{\theta_{top}\left(\pi_{top-\frac{1}{2}}-\pi_{top}\right)+\theta_{top-1}\left(\pi_{top-1}-\pi_{top-\frac{1}{2}}\right)} \qquad \pi_i \leq \pi_{top} \tag{4.5}$$

and between the bottom level and the surface as follows:

$$T_i = \begin{cases} T_1 + \dfrac{(T_1-T_2)\theta_1(\pi_i-\pi_1)}{\theta_2\left(\pi_{\frac{3}{2}}-\pi_2\right)+\theta_1\left(\pi_1-\pi_{\frac{3}{2}}\right)} & \pi_* \leq \pi_i > \pi_1 \\ T_r\left[\frac{p_i}{p_r}\right]^{\frac{\gamma R}{g}} & p_i > p_* \end{cases} \tag{4.6}$$

90003: u-wind u - $[m\,s^{-1}]$

SUBROUTINE InterpolateInLogP

Required parameters: u-wind on full (η) model-levels (STASH=2), surface pressure (STASH=1) and A, B coefficients on full (η) model-levels.

If the data is on the Arakawa B-grid (winds are staggered *wrt* horizontal pressure grid), the data is initialy re-mapped onto the pressure grid using a 4-point bi-linear interpolation as follows

$$u_{ij} = \frac{1}{4}\left(u_{ij} + u_{i-1,j} + u_{i-1,j-1} + u_{i,j-1}\right) \tag{4.7}$$

The winds are then mapped from η-levels to constant pressure levels using the logarithmic pressure dependence of **equation 5.2**.

90004: v-wind v - $[m\,s^{-1}]$

SUBROUTINE InterpolateInLogP

Required parameters: v-wind on full (η) model-levels (STASH=2), surface pressure (STASH=1) and A, B coefficients on full (η) model-levels.

If the data is on the Arakawa B-grid (winds are staggered *wrt* horizontal pressure grid), the data is initialy re-mapped onto the pressure grid using a 4-point bi-linear interpolation as follows

$$v_{ij} = \frac{1}{4}\left(v_{ij} + v_{i-1,j} + v_{i-1,j-1} + v_{i,j-1}\right) \tag{4.8}$$

The winds are then mapped from η-levels to constant pressure levels using the logarithmic pressure dependence of **equation 5.2**.

90005: Relative humidity *wrt* ice U_{ice} - [%]

SUBROUTINE relativeHumidity

Required parameters: Temperature $T(p)$(90002), specific humidity $q(p)$ (90077), mixing ratio $r(p)$ (90009) and saturation mixing ratio $r_s(p)$ (computed according to the tables of Landolt–Bornstein (1987)).

$$U_{ice} = 100\frac{r}{r_s} \tag{4.9}$$

90006: Potential temperature θ - [K]

SUBROUTINE potentialTemperature

Required parameters: Temperature $T(p)$ (90002).

$$\theta = \left(\frac{100000}{p}\right)^{\frac{R}{c_p}} \tag{4.10}$$

90007: Equivalent potential temperature θ_e - $[K]$

SUBROUTINE equivalentPotentialTemperature

Required parameters: Temperature $T(p)$ (90002), relative humidity U (90005), mixing ratio r (90009).

The formula is taken from Bolton (1980), eq. 43,

$$\theta_e = \theta_m exp\left[\left(\frac{3.376}{T_L} - 0.00254\right) r\,(1.0 + 0.00081r)\right] \tag{4.11}$$

where the moist potential temperature is given by

$$\theta_m = T\left(\frac{100000}{p}\right)^{0.2854(1.0-0.0028r)}$$

and the lifting condensation level

$$T_L = \frac{1}{\frac{1}{T-55} - \frac{ln\left(\frac{U}{100}\right)}{2840}} + 55$$

and the mixing ratio r (90009) is in units of $g\,kg^{-1}$ (i.e. multiplied by 1000).

90008: Density ρ - $[kg\,m^{-3}]$

SUBROUTINE density

Required parameters: Temperature $T(p)$ (90002), specific humidity $q(p)$ (90077).

$$\rho = \frac{p}{RT_v} \tag{4.12}$$

where the virtual temperature T_v is related to the temperature via

$$T_v = T\left(1.0 + \frac{1-\epsilon}{\epsilon}q\right)$$

90009: Humidity mixing ratio r - $[kg\,kg^{-1}]$

SUBROUTINE mixingRatio

Required parameters: specific humidity $q(p)$ (90077)

$$r = \frac{q}{1-q} \tag{4.13}$$

90010: Saturation equivalent potential temperature θ_{es}- $[K]$

SUBROUTINE equivalentPotentialTemperature
Required parameters: Temperature $T(p)$ (90002), mixing ratio r (90009).

Uses the same routine as θ_e (90007) with the relative humidity $U = 100$ %, namely

$$\theta_{es} = \theta_e(U = 100\%) \tag{4.14}$$

90011: $\frac{du}{dp}$ - $[m\,s^{-1}\,Pa^{-1}]$

SUBROUTINE firstDerivatives
Required parameters: u-wind $u(p)$ (90003).

90012: $\frac{dv}{dp}$ - $[m\,s^{-1}\,Pa^{-1}]$

SUBROUTINE firstDerivatives
Required parameters: v-wind $v(p)$ (90004).

90013: $\frac{d\theta}{dp}$ - $[K\,Pa^{-1}]$

SUBROUTINE firstDerivatives
Required parameters: potential temperature θ (90006).

90014: $\frac{d\theta_e}{dp}$ - $[K\,Pa^{-1}]$

SUBROUTINE firstDerivatives
Required parameters: equivalent potential temperature θ_e (90007).

90015: Divergence of the wind $\nabla \bullet V$ - $[s^{-1}]$

SUBROUTINE horizontalDivergence
Required parameters: v-wind $u(p)$ (90004), v-wind $v(p)$ (90004).

$$\nabla \bullet \mathrm{V} = \frac{1}{a\cos\phi}\left(\frac{\partial u}{\partial \lambda} + \frac{\partial(v\cos\phi)}{\partial \phi}\right) \tag{4.15}$$

90016: Dry Ertel potential vorticity Q_d - $[K\,kg^{-1}\,m^2\,s^{-1}]$

SUBROUTINE potentialVorticity

Required parameters: potential temperature θ (90006), absolute vorticity ξ (90024).

The Ertel potential vorticity is defined as (e.g Holton, 1992)

$$Q = -\frac{1}{\rho}\zeta \bullet \nabla\theta$$

Expanding and transforming to pressure coordinates we have,

$$Q_d = -g\left[-\frac{1}{a\cos\phi}\frac{\partial v}{\partial p}\frac{\partial \theta}{\partial \lambda} + \frac{1}{a}\frac{\partial u}{\partial p}\frac{\partial \theta}{\partial \phi} + \frac{\partial \theta}{\partial p}\xi\right] \tag{4.16}$$

90017: Moist Ertel potential vorticity Q_m - $[K\,kg^{-1}\,m^2\,s^{-1}]$

SUBROUTINE potentialVorticity

Required parameters: equivalent potential temperature θ_e (90007), absolute vorticity ξ (90024).

As with the dry Ertel potential vorticity **90016**, we have

$$Q_m = -g\left[-\frac{1}{a\cos\phi}\frac{\partial v}{\partial p}\frac{\partial \theta_e}{\partial \lambda} + \frac{1}{a}\frac{\partial u}{\partial p}\frac{\partial \theta_e}{\partial \phi} + \frac{\partial \theta_e}{\partial p}\xi\right] \tag{4.17}$$

900018: Geostrophic u-wind u_g - $[m\,s^{-1}]$

SUBROUTINE geostrophicWinds

Required parameters: Geopotential height $Z(p)$ (90001).

In isobaric coordinates, the geostrophic wind V_g, in terms of the geopotential Φ, can be written as

$$fV_g = g\hat{\mathbf{r}} \times \nabla\Phi$$

Expanding the curl in component form using equation5.6, and using the relationship between geopotential Φ and geopotential height Z for a mean gravitational acceleration at the Earth's surface g,

$$Z = \frac{\Phi}{g}$$

we get u-component of the geostrophic wind,

$$u_g = -\frac{g}{f}\frac{1}{a}\frac{\partial Z}{\partial \phi} \tag{4.18}$$

900019: Geostrophic v-wind v_g - $[m\,s^{-1}]$

SUBROUTINE geostrophicWinds

Required parameters: Geopotential height $Z(p)$ (90001).

As with the geostrophic u-wind u_g, we expand for the v-component, giving,

$$v_g = \frac{g}{f}\frac{1}{a\cos\phi}\frac{\partial Z}{\partial\lambda} \tag{4.19}$$

90020: Wind strength V - $[m\,s^{-1}]$

SUBROUTINE vectorMagnitude

Required parameters: u-wind $u(p)$ (90003), v-wind $v(p)$ (90004).

$$V = \sqrt{u^2+v^2} \tag{4.20}$$

90021: Q-vector Q_1 - $[K\,m^{-1}\,s^{-1}]$

SUBROUTINE qvectors

Required parameters: geostrophic winds u_g (90018), v_g (90019), temperature T (90002).

According to Holton (1992, eq. 6.36),

$$Q_1 = -\frac{R}{p}\frac{\partial \mathbf{V_g}}{\partial x} \bullet \nabla T$$

In spherical coordinates, we obtain

$$Q_1 = -\frac{R}{p}\frac{1}{a\cos\phi}\frac{\partial \mathbf{V_g}}{\partial\lambda} \bullet \nabla T \tag{4.21}$$

The routine qvectors uses the **SUBROUTINE advection (equation 5.8** to compute the dot product

90022: Q-vector Q_2 - $[K\,m^{-1}\,s^{-1}]$

SUBROUTINE qvectors

Required parameters: geostrophic winds u_g (90018), v_g (90019), temperature T (90002).

According to Holton (1992, eq. 6.36),

$$Q_2 = -\frac{R}{p}\frac{\partial \mathbf{V_g}}{\partial y} \bullet \nabla T$$

In spherical coordinates, we obtain

$$Q_2 = -\frac{R}{p}\frac{1}{a}\frac{\partial \mathrm{V_g}}{\partial \phi} \bullet \nabla T \tag{4.22}$$

The routine qvectors uses the **SUBROUTINE advection (equation 5.8** to compute the dot product

90023: Divergence of the Q-vectors $\nabla \bullet Q$ - $[K\, m^{-2}\, s^{-1}]$

SUBROUTINE qvectors

Required parameters: Q-vectors Q_1 (90021) and Q_2 (90022)

$$\nabla \bullet Q = \nabla \bullet (Q_1, Q_2) \tag{4.23}$$

90024: Absolute vorticity ξ - $[s^{-1}]$

SUBROUTINE vorticity

Required parameters: relative vorticity ζ (90038), Coriolis parameter f,

$$\xi = \zeta + f \tag{4.24}$$

90025: Vertical velocity w - $[m\, s^{-1}]$

SUBROUTINE omegaToW

Required parameters: ω (90087), density ρ (90008)

Assuming the hydrostatic relation (**equation 5.1**), we transform from omega to vertical velocity using

$$w = -\frac{\omega}{\rho g} \tag{4.25}$$

90028: $\frac{du_g}{dp}$ - $[m\, s^{-1}\, Pa^{-1}]$

SUBROUTINE geostrophicWinds

Required parameters: geostrophic u-wind u_g (90018)

90029: $\frac{dv_g}{dp}$ - $[m\, s^{-1}\, Pa^{-1}]$

SUBROUTINE geostrophicWinds

Required parameters: geostrophic v-wind v_g (90019)

90031: Magnitude of the geostrophic deformation D_g - $[s^{-1}]$

SUBROUTINE geostrophicDeformation

Required parameters: geostrophic winds u_g (90018) and v_g (90019).

According to Keyser, Reeder & Reed (1988, eq. 1.2), the geostrophic deformation (we use $\|D_g\|$ rather than E) is given by

$$\|D_g\| = \sqrt{E_{st}^2 + E_{sh}^2}$$

where

$$E_{st}^2$$

is the stretching deformation ($\hat{\lambda}$-component of the geostrophic deformation, 90032), and

$$E_{st}^2$$

is the shearing deformation ($\hat{\phi}$-component of the geostrophic deformation, 90033)

$$\|D_g\| = \sqrt{\left(\frac{1}{a\cos\phi}\frac{\partial u_g}{\partial\lambda} - \frac{1}{a}\frac{\partial v_g}{\partial\phi}\right)^2 + \left(\frac{1}{a\cos\phi}\frac{\partial v_g}{\partial\lambda} + \frac{1}{a}\frac{\partial u_g}{\partial\phi}\right)^2} \tag{4.26}$$

90032: $\hat{\lambda}$-component of the geostrophic deformation $(D_g)_\lambda$ - $[s^{-1}]$

SUBROUTINE geostrophicDeformation

Required parameters: geostrophic winds u_g (90018) and v_g (90019).

$$(D_g)_\lambda = \|D_g\| \cos\left\{\frac{1}{2}\arctan\left(\frac{\frac{1}{a\cos\phi}\frac{\partial v_g}{\partial\lambda} + \frac{1}{a}\frac{\partial u_g}{\partial\phi}}{\frac{1}{a\cos\phi}\frac{\partial u_g}{\partial\lambda} - \frac{1}{a}\frac{\partial v_g}{\partial\phi}}\right)\right\} \tag{4.27}$$

90033: $\hat{\phi}$-component of the geostrophic deformation $(D_g)_\phi$ - $[s^{-1}]$

SUBROUTINE geostrophicDeformation

Required parameters: geostrophic winds u_g (90018) and v_g (90019).

According to Keyser, Reeder & Reed (1988, eq. 1.2), the geostrophic deformation is given by

$$(D_g)_\lambda = \|D_g\| \sin\left\{\frac{1}{2}\arctan\left(\frac{\frac{1}{a\cos\phi}\frac{\partial v_g}{\partial\lambda} + \frac{1}{a}\frac{\partial u_g}{\partial\phi}}{\frac{1}{a\cos\phi}\frac{\partial u_g}{\partial\lambda} - \frac{1}{a}\frac{\partial v_g}{\partial\phi}}\right)\right\} \tag{4.28}$$

90034: Magnitude $\|\nabla\theta_w\|$ - $[K\,m^{-1}]$

SUBROUTINE gradThetaORThetaW

Required parameters: wet-bulb potential temperature θ_w (90041)

According to **equation 5.4**

$$\|\nabla\theta_w\| = \sqrt{\left(\frac{1}{a\cos\phi}\frac{\partial\theta_w}{\partial\lambda}\right)^2 + \left(\frac{1}{a}\frac{\partial\theta_w}{\partial\phi}\right)^2} \tag{4.29}$$

90035: $\hat{\lambda}$-component of $\nabla\theta_w$ - $[K\,m^{-1}]$

SUBROUTINE gradThetaORThetaW

Required parameters: wet-bulb potential temperature θ_w (90041)

According to **equation 5.4**

$$(\nabla\theta_w)_\lambda = \frac{1}{a\cos\phi}\frac{\partial\theta_w}{\partial\lambda} \tag{4.30}$$

90036: $\hat{\phi}$-component of $\nabla\theta_w$ - $[K\,m^{-1}]$

SUBROUTINE gradThetaORThetaW

Required parameters: wet-bulb potential temperature θ_w (90041)

According to **equation 5.4**

$$(\nabla\theta_w)_\phi = \frac{1}{a}\frac{\partial\theta_w}{\partial\phi} \tag{4.31}$$

90037: Goestrophic deformation *wrt* θ_w, $D_g^{\theta_w}$ - $[s^{-1}]$

SUBROUTINE deformationThetaW

Required parameters: geostrophic winds u_g (90018), v_g (90019), wet-bulb potential temperature θ_w (90041), geostrophic deformation $\|D_g\|$ (90031).

According to Keyser, Reeder & Reed (1988eq. 1.1),

$$D_g^{\theta_w} = -\frac{1}{2}\|D_g\| \cos\left\{2\left[\frac{1}{2}\arctan\left(\frac{\frac{1}{a\cos\phi}\frac{\partial v_g}{\partial\lambda}+\frac{1}{a}\frac{\partial u_g}{\partial\phi}}{\frac{1}{a\cos\phi}\frac{\partial u_g}{\partial\lambda}-\frac{1}{a}\frac{\partial v_g}{\partial\phi}}\right)\right] - \arctan\left(\cos\phi\frac{\left(\frac{\partial\theta_w}{\partial\phi}\right)}{\left(\frac{\partial\theta_w}{\partial\lambda}\right)}\right)\right\} \tag{4.32}$$

90038: Relative vorticity ζ - $[s^{-1}]$

SUBROUTINE vorticity

Required parameters: u-wind $u(p)$ (90003), v-wind $v(p)$ (90004).

According to **equation 5.6**, the vorticity is

$$\hat{\mathbf{r}} \bullet \nabla \times \mathbf{V} = \frac{1}{a\cos\phi}\left[\frac{\partial v}{\partial\lambda} - \frac{\partial\left(u\cos\phi\right)}{\partial\phi}\right] \tag{4.33}$$

90039: Geostrophic relative vorticity ζ_g - $[s^{-1}]$

SUBROUTINE vorticity

Required parameters: geostrophic winds $u_g(p)$ (90018), $v_g(p)$ (90019).

According to **equation 5.6**, the vorticity is

$$\hat{\mathbf{r}} \bullet \nabla \times \mathbf{V_g} = \frac{1}{a\cos\phi}\left[\frac{\partial v_g}{\partial\lambda} - \frac{\partial\left(u_g\cos\phi\right)}{\partial\phi}\right] \tag{4.34}$$

90040: - [degrees from North deg]

SUBROUTINE windsToNSEW

Required parameters: u-wind $u(p)$ (90003), v-wind $v(p)$ (90004).

If the winds are on a rotated lat/long grid, they are rotated to a true lat/long grid, and the direction computed so that 0 deg winds are true southerly, and 90 deg winds are true westerly

$$u \geq 0, v \geq 0 \quad : \quad 0\,\text{deg} \geq dir^n \leq 90\,\text{deg} \tag{4.35}$$
$$u \geq 0, v \leq 0 \quad : \quad 90\,\text{deg} \geq dir^n \leq 180\,\text{deg} \tag{4.36}$$
$$u \leq 0, v \leq 0 \quad : \quad 180\,\text{deg} \geq dir^n \leq 270\,\text{deg} \tag{4.37}$$
$$u \leq 0, v \geq 0 \quad : \quad 270\,\text{deg} \geq dir^n \leq 360\,\text{deg} \tag{4.38}$$

90041: Wet-bulb potential temperature θ_w - $[K]$

SUBROUTINE wetBulbPotentialTemperature

Required parameters: Temperature T (90002), specific humidity q (90077)

This routine has been adapted from the Unified Model Fortran-77 source code as well as the algorithm for computing θ_w.

The temperature variation of the latent heat of vapourisation (or condensation) at a temperature T is

$$L_V(T) = 2.501 \times 10^6 - 2.34 \times 10^3 \,(T - 273.15)$$

Start by calculating the function $G(T_0)$ at the actual temperature T at pressure p which we define as T_0,

$$G(T_0) = qL(T_0) + T_0 \left(c_p^D + qc_p^V\right)$$

and setting the wet-bulb potential temperature initialy to $\theta_w = T_0$.
Now iterate the following system of equations to find $\theta_w(p)$

$$\begin{aligned}
G_s(\theta_w) &= r_s L(\theta_w) + \theta_w \left(c_p^D + qc_p^V\right) \\
\frac{dG}{d\theta_w} &= \frac{r_s M_V L_V^2}{R_* \theta_w^2} + c_p^D + qc_p^V \\
\theta_{w_{i+1}} &= \theta_w - \frac{(G(\theta_w) - G(T_0))}{\frac{dG}{d\theta_w}} \\
L(\theta_{w_{i+1}}) &= 2.501 \times 10^6 - 2.34 \times 10^3 \left(\theta_{w_{i+1}} - 273.15\right)
\end{aligned}$$

until the following condition is satisfied

$$\| \left(G_s(\theta_{w_{i+1}}) - G(T_0)\right) \| \leq 1.0$$

This loop should converge rapidly, so we set the maximum number of loop iterations to 10. Values of the constants c_p^D, c_p^V, R_* are given in Appendix D.1. Further, r_s is the saturation mixing ratio (90010) *wrt* water and quantities with the subscript D refer to dry air and the subscript V water vapour. Once we have $\theta_w(p)$, we descend down the saturated adiabat to $1000\,hPa$ solving sequentially,

$$\begin{aligned}
d\theta_w &= \frac{\frac{\epsilon r_s L_V}{p} + \frac{R_* \theta_w}{M_D}}{pc_p^D + \frac{\epsilon r_s M_V L_V^2}{R_* \theta_w^2}} dp \\
\theta_w &= \theta_w + d\theta_w dp
\end{aligned}$$

$$
\begin{aligned}
p &\rightarrow p + dp \\
L_V(d\theta_w) &= 2.501 \times 10^6 - 2.34 \times 10^3 \left(\theta_w - 273.15\right) \\
d(\theta_w)_{L_V(d\theta_w)} &= \frac{\frac{\epsilon r_s L_V(d\theta_w)}{p} + \frac{R_* \theta_w^i}{M_D}}{p c_p^D + \frac{r_s p M_V L_V^2}{R_* (\theta_w^i)^2}} \\
\theta_w(p + dp) &= \frac{1}{2}\left(d(\theta_w)_{L_V(d\theta_w)} + d\theta_w\right) dp + \theta_w(p) \\
L_V(\theta_w(d + dp)) &= 2.501 \times 10^6 - 2.34 \times 10^3 \left(\theta_w(p + dp) - 273.15\right)
\end{aligned}
$$

where dp is the decrement in pressure

90042: Ageostrophic u-wind u_a - $[m\,s^{-1}]$

SUBROUTINE ageostrophicWinds

Required parameters: u-wind u (90003), geostrophic u-wind u_g (90018)

$$u_a = u - u_g \tag{4.39}$$

90043: Ageostrophic v-wind v_a - $[m\,s^{-1}]$

SUBROUTINE ageostrophicWinds

Required parameters: v-wind v (90004), geostrophic u-wind v_g (90019)

$$v_a = v - v_g \tag{4.40}$$

90044: Magnitude $\|\nabla\,\theta\|$ - $[K\,m^{-1}]$

SUBROUTINE gradThetaORThetaW

Required parameters: potential temperature θ (90041)

According to **equation 5.4**

$$\|\nabla\,\theta\| = \sqrt{\left(\frac{1}{a\cos\phi}\frac{\partial\theta}{\partial\lambda}\right)^2 + \left(\frac{1}{a}\frac{\partial\theta}{\partial\phi}\right)^2} \tag{4.41}$$

90045: $\hat{\lambda}$-component of $\nabla\,\theta$ - $[K\,m^{-1}]$

SUBROUTINE gradThetaORThetaW

Required parameters: potential temperature θ (90041)

According to **equation 5.4**

$$(\nabla\,\theta)_\lambda = \frac{1}{a\cos\phi}\frac{\partial\theta}{\partial\lambda} \tag{4.42}$$

90046: $\hat{\phi}$-component of $\nabla\,\theta$ - $[K\,m^{-1}]$

SUBROUTINE gradThetaORThetaW

Required parameters: potential temperature θ (90041)

According to **equation 5.4**

$$(\nabla\,\theta)_\phi = \frac{1}{a}\frac{\partial\theta}{\partial\phi} \tag{4.43}$$

90053: Relative humidity wrt water U - [%]

SUBROUTINE relativeHumidity

Required parameters: Temperature $T(p)$(90002), specific humidity $q(p)$ (90077), mixing ratio $r_(p)$ (90009) and saturation mixing ratio $r_s(p)$ (computed according to the tables of Landolt–Bornstein (1987)).

$$U_{water} = 100\frac{r}{r_s} \tag{4.44}$$

The following diagnostics, **90061-90074**, relate to the objective determination of fronts, and whose formulation has been driven by the work of Hewson (1998). Details of most of the diagnostics can be gleaned from his paper.

90061: Frontal zones $\nabla^2\theta_w^{FZ}$ - $[K\,m^{-2}]$

SUBROUTINE frontalZones

Required parameters: wet-bulb potential temperature θ_w (90041)

These are defined according to Hewson (1998, Appendix 2). We start by defining the following the components

$$(\nabla\|\nabla\theta_w\|)_\lambda = \left(\frac{\partial\|\nabla\theta_w\|}{\partial\lambda}\right)$$

$$(\nabla\|\nabla\theta_w\|)_\phi = \left(\frac{\partial\|\nabla\theta_w\|}{\partial\phi}\right)$$

in a similar manner to 90045, 90046. Then,

$$
\begin{aligned}
A_{i,j} &= \frac{((\nabla\|\nabla\theta_w\|)_\lambda)^2 - \left((\nabla\|\nabla\theta_w\|)_\phi\right)^2 -}{\|\nabla\theta_w\|} \\
B_{i,j} &= \frac{(\nabla\|\nabla\theta_w\|)_\lambda\,(\nabla\|\nabla\theta_w\|)_\phi}{\|\nabla\theta_w\|}
\end{aligned}
$$

$$
\begin{aligned}
VT_{i,j} &= 2.0\,(A_{i-1,j} + A_{i,j-1} + A_{i,j} + A_{i+1,j} + A_{i,j+1}) \\
UT_{i,j} &= B_{i-1,j} + B_{i,j-1} + B_{i,j} + B_{i+1,j} + B_{i,j+1}
\end{aligned}
$$

$$
\begin{aligned}
SX &= \cos\left(\frac{1}{2}\arctan\left(\frac{VT}{UT}\right)\right) \\
SY &= \sin\left(\frac{1}{2}\arctan\left(\frac{VT}{UT}\right)\right)
\end{aligned}
$$

$$
\begin{aligned}
UC_{i,j} &= \left[\left(\frac{\partial\|\nabla\theta_w\|}{\partial\lambda}\right)_{i+1,j} SX_{i,j} + \left(\frac{\partial\|\nabla\theta_w\|}{\partial\phi}\right)_{i+1,j} SY_{i,j}\right] SX_{i,j} \\
UA_{i,j} &= \left[\left(\frac{\partial\|\nabla\theta_w\|}{\partial\lambda}\right)_{i-1,j} SX_{i,j} + \left(\frac{\partial\|\nabla\theta_w\|}{\partial\phi}\right)_{i-1,j} SY_{i,j}\right] SX_{i,j} \\
VB_{i,j} &= \left[\left(\frac{\partial\|\nabla\theta_w\|}{\partial\lambda}\right)_{i,j+1} SX_{i,j} + \left(\frac{\partial\|\nabla\theta_w\|}{\partial\phi}\right)_{i,j+1} SY_{i,j}\right] SY_{i,j} \\
VD_{i,j} &= \left[\left(\frac{\partial\|\nabla\theta_w\|}{\partial\lambda}\right)_{i,j-1} SX_{i,j} + \left(\frac{\partial\|\nabla\theta_w\|}{\partial\phi}\right)_{i,j-1} SY_{i,j}\right] SY_{i,j}
\end{aligned}
$$

giving

$$
\nabla^2\theta_w^{FZ} = -\frac{1}{2}\left[(UC - UA) + (VD - VB)\right] \qquad (4.45)
$$

90062: Equivalent $\nabla^2\theta_w^E$ - $[K\,m^{-2}]$

SUBROUTINE equivalentGradsq

Required parameters: wet-bulb potential temperature θ_w (90041)

According to Renard & Clarke (1965), define a thermal front parameter according to

$$\frac{(\nabla\|\nabla\,\theta_w\|)\bullet\nabla\,\theta_w}{\|\nabla\,\theta_w}$$

then, expanding in spherical coords.,

$$\nabla^2\theta_w^E = -\left\{\frac{\frac{1}{a\cos\phi}\frac{\partial\|\nabla\,\theta_w\|}{\partial\lambda}(\nabla\,\theta_w)_\lambda + \frac{1}{a}\frac{\partial\|\nabla\,\theta_w\|}{\partial\phi}(\nabla\,\theta_w)_\phi}{\|\nabla\,\theta_w\|}\right\} \tag{4.46}$$

90063: Frontal $\nabla^2\,\theta_w$ - $[K\,m^{-2}]$

SUBROUTINE frontalGradsq

Required parameters: wet-bulb potential temperature θ_w (90041)

Frontal parameter defined by Kirk (1965)

$$-\nabla^2\,\theta_w \tag{4.47}$$

90064: "Local" $\nabla\,\theta_w^L$ - $[K\,m^{-2}]$

SUBROUTINE localGrad

Required parameters: wet-bulb potential temperature θ_w (90041

According to the definitions of Hewson (1998, eq. 11), one of the criteria that defines a "type 3 front" is that the thermal gradient in the baroclinic zone adjacent to the front is greater than some prescribed value,

$$\nabla\,\theta_w^L = \|\nabla\,\theta_w\| + \frac{1}{\sqrt{2}}\|\nabla\,\|\nabla\,\theta_w\|\| \tag{4.48}$$

90065: Frontal speed of θ_w- $[m\,s^{-1}]$

SUBROUTINE frontalSpeed

Required parameters: wet-bulb potential temperature θ_w(90041) , winds u (90003) and v (90004)

According to Hewson (1998, eq. 13), define the speed of a front by

$$\frac{\mathbf{V}\bullet\nabla\|\nabla\,\theta_w\,\|}{\|\nabla\,\|\nabla\,\theta_w\|\|} \tag{4.49}$$

90066: θ_w-advection - $[K\,s^{-1}]$

SUBROUTINE frontalAdvection

Required parameters: wet-bulb potential temperature θ_w(90041) , winds u (90003) and v (90004)

$$-\mathbf{V} \bullet \nabla\theta_w \tag{4.50}$$

90067: Geostrophic advection of θ_w - $[K\,s^{-1}]$

SUBROUTINE frontalAdvection

Required parameters: wet-bulb potential temperature θ_w(90041) , geostrophic winds u_g (90018) and v_g (90019)

According to Hewson (1998, eq. 12), cold/warm fronts are defined according to whether the geostrophic wind implied by the local isobaric pattern is tending to advect them to warmer/colder air

$$-\mathbf{V_g} \bullet \nabla\theta_w \tag{4.51}$$

90068: Frontal zones $\nabla^2\theta^{FZ}$ - $[Km^{-2}]$

SUBROUTINE frontalZones

Required parameters: potential temperature θ (90006)

These are defined according to Hewson (1998, Appendix 2).

As with formulation of the θ_w frontal zones diagnostic **90061**, we have

$$\nabla^2\theta^{FZ} = -\frac{1}{2}\left[(UC - UA) + (VD - VB)\right] \tag{4.52}$$

where UC, UA, VD, VB are *wrt* θ.

90069: Equivalent $\nabla^2\theta^E$ - $[K\,m^{-2}]$

SUBROUTINE equivalentGradsq

Required parameters: potential temperature θ (90006)

According to Renard & Clarke (1965), define a thermal front parameter according to

$$\frac{(\nabla\|\nabla\theta\|) \bullet \nabla\theta}{\|\nabla\theta}$$

then, expanding in spherical coords.,

$$\nabla^2\theta^E = -\left\{\frac{\frac{1}{a\cos\phi}\frac{\partial\|\nabla\theta\|}{\partial\lambda}(\nabla\theta)_\lambda + \frac{1}{a}\frac{\partial\|\nabla\theta\|}{\partial\phi}(\nabla\theta)_\phi}{\|\nabla\theta\|}\right\} \tag{4.53}$$

90070: Frontal $\nabla^2\theta$ - $[K\,m^{-2}]$

SUBROUTINE frontalGradsq

Required parameters:potential temperature θ (90006)

Frontal parameter defined by Kirk (1965)

$$-\nabla^2\theta \tag{4.54}$$

90071: "Local" $\nabla\theta^L$ - $[K\,m^{-2}]$

SUBROUTINE localGrad

Required parameters: potential temperature θ (90006)

According to the definitions of Hewson (1998, eq. 11), one of the criteria that defines a "type 3 front" is that the thermal gradient in the baroclinic zone adjacent to the front is greater than some prescribed value,

$$\nabla\theta^L = \|\nabla\theta\| + \frac{1}{\sqrt{2}}\|\nabla\|\nabla\theta\|\| \tag{4.55}$$

90072: Frontal speed of θ- $[m\,s^{-1}]$

SUBROUTINE frontalSpeed

Required parameters: potential temperature θ(90006), winds u (90003) and v (90004)

According to Hewson (1998, eq. 13), define the speed of a front by

$$\frac{\mathbf{V}\bullet\nabla\|\nabla\theta\|}{\|\nabla\|\nabla\theta\|\|} \tag{4.56}$$

90073: θ-advection - $[K\,s^{-1}]$

SUBROUTINE frontalAdvection

Required parameters: potential temperature θ (90006), winds u (90003) and v (90004)

$$-\mathbf{V} \bullet \nabla \theta \tag{4.57}$$

90074: Geostrophic advection of θ - $[K\, s^{-1}]$

SUBROUTINE frontalAdvection

Required parameters: potential temperature θ (90006), geostrophic winds u_g (90018) and v_g (90019)

According to Hewson (1998, eq. 12), cold/warm fronts are defined according to whether the geostrophic wind implied by the local isobaric pattern is tending to advect them to warmer/colder air

$$-\mathbf{V_g} \bullet \nabla \theta \tag{4.58}$$

90075: Shear relative vorticity ζ^{sh} - $[s^{-1}]$

SUBROUTINE shearRelativeVorticity

Required parameters: winds u (90003) and v (90004)

Begin by defining the following components of the wind

$$\begin{aligned} u_x &= \frac{u}{\sqrt{u^2+v^2}} \\ u_y &= \frac{v}{\sqrt{u^2+v^2}} \end{aligned}$$

Then, further define

$$\begin{aligned} U_0 &= \left[(u_x)_{i,j}\, u_{i+1,j} + (u_x)_{i,j}\, v_{i+1,j}\right] (u_y)_{i,j} \\ U_1 &= \left[(u_x)_{i,j}\, u_{i-1,j} + (u_x)_{i,j}\, v_{i-1,j}\right] (u_y)_{i,j} \\ U_2 &= \left[(u_x)_{i,j}\, u_{i,j+1} + (u_x)_{i,j}\, v_{i,j+1}\right] (u_x)_{i,j} \\ U_3 &= \left[(u_x)_{i,j}\, u_{i,j-1} + (u_x)_{i,j}\, v_{i,j-1}\right] (u_x)_{i,j} \end{aligned}$$

arriving at the shear relative vorticity

$$\zeta^{sh} = \frac{U_0 - U_1}{a \cos \phi} - \frac{U_2 - U3}{a} \tag{4.59}$$

90076: Jet angle α^{jet}- [deg]

SUBROUTINE jetAngle

Required parameters: Shear relative vorticity ζ^{sh} (90075), winds u (90003) and v (90004).

Begin by defining the following components of the wind

Define the components and magnitude of the derivative of the shear relative vorticity ζ^{sh} by

$$\begin{aligned}
\left(\nabla\zeta^{sh}\right)_\lambda &= \frac{1}{a\cos\phi}\frac{\partial\zeta^{sh}}{\partial\lambda} \\
\left(\nabla\zeta^{sh}\right)_\phi &= \frac{1}{a}\frac{\partial\zeta^{sh}}{\partial\phi} \\
\|\nabla\zeta^{sh}\| &= \sqrt{((\nabla\zeta^{sh})_\lambda)^2 + \left((\nabla\zeta^{sh})_\phi\right)^2}
\end{aligned}$$

Then, the jet angle is defined as

$$\alpha^{jet} = \arccos\left[\frac{1}{\mathbf{V}\|\nabla\zeta^{sh}\|}\left(u\left(\nabla\zeta^{sh}\right)_\lambda + v\left(\nabla\zeta^{sh}\right)_\phi\right)\right] \tag{4.60}$$

where

$$\mathbf{V} = \sqrt{u^2+v^2}$$

90077: Specific humidity $q(p)$ - $[kg\,kg{-}1]$

SUBROUTINE interpolateInLogP

Required parameters: specific humidity on full (η) model-levels (STASH=4010), surface pressure (STASH=1) and A, B coefficients on full (η) model-levels.

Interpolation between model levels and constant pressure levels is via **equation 5.2**

90080: Frontal waves - $[m^{-3}\,K^{-2}\,s^{-1}]$

SUBROUTINE frontalWave

Required parameters: wet-bulb potential temperature θ_w (90041), geostrophic winds u_g (90018) and v_g (90019)

According to Hewson (1997 equation M3),

$$(\nabla\|\nabla\theta_w\|\times\hat{\mathrm{r}})\bullet\nabla(-\mathbf{V}_\mathrm{g}\bullet\nabla\theta_w) \tag{4.61}$$

90081-84: Frontogenesis F-vectors, tilt and function

SUBROUTINE frontogenesisFunction
Required parameters: ω (90087), winds u (90003) and v (90004), temperature T (90002), potential temperature θ (90006), density ρ (90008).

According to Rotunno et al (1994), the following diagnostics have been defined

90081: Frontogenesis F_1- vector - $[K\, m^{-1}\, s^{-1}]$

$$F_1 = -\frac{R}{p}\frac{1}{a\cos\phi}\frac{\partial u}{\partial\lambda}\bullet\nabla T \tag{4.62}$$

90082: Frontogenesis F_2- vector - $[K\, m^{-1}\, s^{-1}]$

$$F_2 = -\frac{R}{pa}\frac{\partial v}{\partial\phi}\bullet\nabla T \tag{4.63}$$

90083: Frontogenesis tilt F_T - $[K^2\, m^{-2}\, s^{-1}]$

$$\nabla w\bullet\nabla\theta\frac{1}{\rho g}\frac{\partial\theta}{\partial p} \tag{4.64}$$

90084: Frontogenesis function $\mathcal{F}$ - $[K^2\, m^{-2}\, s^{-1}]$

$$\mathcal{F} = (F_1, F_2)\bullet\nabla\theta + \nabla w\bullet\nabla\theta\frac{1}{\rho g}\frac{\partial\theta}{\partial p} \tag{4.65}$$

90085: True westerly u-wind - $[m\, s^{-1}]$

SUBROUTINE windsToNSEW
Required parameters: winds u, v, latitude and longitude of the rotated pole.

When the winds (90003, 90004) are wrt an equatorial grid, such as the UK Mesoscale, they can be rotated so that they represent true southerly and westerly winds. We state the transformation formulae below, but the details can be found in Goddard (1998) eqs. 5.14, 5.19., 5.20.

The true westerly u is obtained from the relation

$$u^t = c_1 u + c_2 v \tag{4.66}$$

where the coefficients c_1, c_2 are given by

$$\begin{aligned} c_1 &= -\sin(\lambda - \lambda_0)\sin\lambda'\sin\Phi_p && + \cos(\lambda - \lambda_0)\cos\lambda' \\ c_2 &= \sin\Phi\sin\lambda'\cos(\lambda - \lambda_0)\sin\Phi_p && - \sin\Phi\cos\lambda'\sin(\lambda - \lambda_0) \\ & && - \sin\lambda'\cos\Phi\cos\Phi_p \end{aligned}$$

Primed quantities refer to coordinates in the equatorial (rotated) system, unprimed λ, ϕ to true longitude and latitude Φ_p is the latitude of the rotated pole, λ_0 the longitude of the rotated pole. Values of the rotated grid are obtained from the pp-file header (Appendix B).

90086: True southerly v-wind - $[m\, s^{-1}]$

SUBROUTINE windsToNSEW

Required parameters: winds u, v, latitude and longitude of the rotated pole.

As with diagnostic 90085, the true southerly wind v^t is calculated from,

$$v^t = c_1 v - c_2 u \tag{4.67}$$

90087: ω - $[Pa\, s^{-1}]$

SUBROUTINE interpolateInLogP

Required parameters: ω on full (η) model-levels (STASH=12201), surface pressure (STASH=1) and A, B coefficients on full (η) model-levels.

Interpolation between model levels and constant pressure levels is via **equation 5.2**

90088: Dry static stability N_d^2 - $[s^{-2}]$

SUBROUTINE staticStability

Required parameters: density ρ (90008), potential temperature θ (90006)

The dry adiabatic static stability is defined as

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

Assuming the hydrostatic relation, and converting between height z and pressure p (equation 5.1), we obtain,

$$N_d^2 = -\left(\frac{\rho\, g^2}{\theta}\right)\frac{\partial\theta}{\partial p} \tag{4.68}$$

90089: Moist static stability N_m^2 - $[s^{-2}]$

SUBROUTINE staticStability

Required parameters: density ρ (90008), potential temperature θ (90006), Temperature T (90002), specific humidity q (90077)

According to Durran & Klemp (1982), the moist static stability is defined by

$$N_m^2 = \frac{g}{T}\left(\frac{\partial T}{\partial z} + \Gamma_m\right)\left(1 + \frac{Lr_s}{RT}\right) - \frac{g}{1+q}\frac{\partial q}{\partial z}\Gamma_m = \frac{g}{c_p}\left[\frac{1+\frac{Lr_s}{RT}}{1+\frac{L^2 r_s}{R_v T^2 c_p}}\right]$$

where r_s is the saturation mixing ratio (90010), L the latent heat of vapourisation, R, R_v, c_p are the dry and moist air gas constants and specific heat capacity

Assuming the hydrostatic relation, and converting between height z and pressure p (equation 5.1), we obtain,

$$N_m^2 = \frac{g}{T}\left(-\rho g\frac{\partial T}{\partial p} + \Gamma_m\right)\left(1 + \frac{Lr_s}{RT}\right) - \frac{\rho g^2}{1+q}\frac{\partial q}{\partial p} \tag{4.69}$$

90090: Richardson number $\mathcal{R}i$ - [dimensionless]

SUBROUTINE staticStability

Required parameters: density ρ (90008), potential temperature θ (90006), horizontal winds u and v.

The Richardson number is the ratio of the dry adiabatic static stability to the square of the winds shear, namely

$$\mathcal{R}i = \frac{N_d^2}{\left(\frac{\partial V}{\partial z}\right)^2}$$

where

$$V = \sqrt{u^2 + v^2}$$

Assuming the hydrostatic relation, and converting between height z and pressure p (equation 5.1, we obtain,

$$\mathcal{R}i = \frac{N_d^2}{\left(\rho\, g\, \frac{\partial V}{\partial p}\right)^2} \tag{4.70}$$

90091: Moist Richardson number $\mathcal{R}i_m$ - [dimensionless]

SUBROUTINE staticStability

Required parameters: density ρ (90008), potential temperature θ (90006), horizontal winds u and v.

The moist Richardson number is the ratio of the moist adiabatic static stability to the square of the winds shear, namely
where

$$V = \sqrt{u^2 + v^2}$$

$$\mathcal{R}i_m = \frac{N_m^2}{\left(\frac{\partial V}{\partial z}\right)^2}$$

Assuming the hydrostatic relation, and converting between height z and pressure p (equation 5.1, we obtain,

$$\mathcal{R}i_m = \frac{N_m^2}{\left(\rho\, g\, \frac{\partial V}{\partial p}\right)^2} \tag{4.71}$$

90092: Simple mean pseudo water-vapour relative humidity - [%]

SUBROUTINE simplePseudoWV

Required parameters: Temperature $T(p)$(90002), specific humidity $q(p)$ (90077), mixing ratio $r_(p)$ (90009) and saturation mixing ratio $r_s(p)$ (computed according to the tables of Landolt-Bornstein (1987)).

This parameter is defined as an average relative humidity between $30000hPa \leq p \leq 60000hPa$, namely

$$\bar{U} = \frac{1}{N} \sum_{p=60000}^{p=30000} U \tag{4.72}$$

where N is the number of points that contribute within the pressure range.

Chapter 5

NUMERICAL APPROXIMATIONS

5.1 Height to pressure transformations

We assume the hydrostatic approximation, namely

$$\frac{dp}{dz} = -\rho g$$

giving the transformation between height and pressure as

$$\frac{d}{dp} = -\rho g \frac{d}{dz} \tag{5.1}$$

5.2 Interpolation between pressure-surfaces

SUBROUTINE InterpolateInLogP

Vertical interpolation between pressure surfaces is assumed to be logarithmic in pressure, according to

$$F_k = \begin{cases} F_{top} & p_i < p_{top} \\ \alpha F_j + (1-\alpha) F_{j-1} & p_j < p_i < p_{j-1} \\ F_1 & p_i > p_1 \end{cases} \tag{5.2}$$

where the interpolation coefficients α are calculated from Goddard (1998, eqs. 3.3,3.4), namely

$$\alpha = \frac{\ln\left[\frac{p_i}{p_{j-1}}\right]}{\ln\left[\frac{p_j}{p_{j-1}}\right]} \quad j-1 < i < j \tag{5.3}$$

5.3 Vector identities in spherical coordinates

It is implicitly assumed that the vector operators act **HORIZONTALLY ONLY**.
Gradient (**SUBROUTINE grad**)

$$\nabla\psi = \frac{1}{a\cos\phi}\frac{\partial\psi}{\partial\lambda}\hat{\lambda} + \frac{1}{a}\frac{\partial\psi}{\partial\phi}\hat{\phi} \tag{5.4}$$

Divergence (**SUBROUTINE horizontalDivergence**)

$$\nabla \bullet \mathbf{V} = \frac{1}{a\cos\phi}\left(\frac{\partial u}{\partial\lambda} + \frac{\partial(v\cos\phi)}{\partial\phi}\right) \tag{5.5}$$

Curl (vorticity) (**SUBROUTINE kdotcurl**)

$$\hat{\mathbf{r}} \bullet (\nabla \times \mathbf{V}) = \frac{1}{a\cos\phi}\left(\frac{\partial v}{\partial\lambda} - \frac{\partial(u\cos\phi)}{\partial\phi}\right) \tag{5.6}$$

∇^2 (**SUBROUTINE gradSq**)

$$\nabla^2\psi = \frac{1}{a^2\cos^2\phi}\left(\frac{\partial^2 d\psi}{\partial\lambda^2} + \cos\phi\frac{\partial}{\partial\phi}\left(\cos\phi\frac{\partial\psi}{\partial\phi}\right)\right) \tag{5.7}$$

Advection (**SUBROUTINE advection**)

$$\mathbf{V} \bullet \nabla\psi = \frac{u}{a\cos\phi}\frac{\partial\psi}{\partial\lambda} + \frac{v}{a}\frac{\partial\psi}{\partial\phi} \tag{5.8}$$

NB the vector quantities are based on horizontal derivatives only, so scalar quantities are only really pseudo-scalars - the magnitude does not transform exactly between orthogonal coordinate systems.

5.4 Finite difference schemes (Abramowitz & Stegun, 1972)

Computation of the first and second order derivatives in section 5.3 is through 3- and 5-point finite differences, with forward and backward differences at the boundaries

First order derivative:

Centre difference:

$$\left(\frac{\partial}{\partial p}, \frac{\partial}{\partial \phi}, \frac{\partial}{\partial \lambda}\right) u_0 = \frac{1}{2h}\left(-u_{-1} + u_1\right) \tag{5.9}$$

forward difference

$$\left(\frac{\partial}{\partial p}, \frac{\partial}{\partial \phi}, \frac{\partial}{\partial \lambda}\right) u_0 = \frac{1}{2h}\left(-3u_0 + 4u_1 - u_2\right) \tag{5.10}$$

backward difference

$$\left(\frac{\partial}{\partial p}, \frac{\partial}{\partial \phi}, \frac{\partial}{\partial \lambda}\right) u_0 = \frac{1}{2h}\left(u_0 - 4u_{-1} + 3u_{-2}\right) \tag{5.11}$$

Second order derivatives:
centre difference

$$\left(\frac{\partial^2}{\partial p^2}, \frac{\partial^2}{\partial \phi^2}, \frac{\partial^2}{\partial \lambda^2}\right) u_0 = \frac{1}{12h^2}\left(-u_{-2} + 16u_{-1} - 30u_0 + 16u_1 + 11u_2\right) \tag{5.12}$$

forward difference

$$\left(\frac{\partial^2}{\partial p^2}, \frac{\partial^2}{\partial \phi^2}, \frac{\partial^2}{\partial \lambda^2}\right) u_0 = \frac{1}{12h^2}\left(-u_{-3} + 4u_{-2} + 6u_{-1} - 20u_0 + 11u_1\right) \tag{5.13}$$

backward difference

$$\left(\frac{\partial^2}{\partial p^2}, \frac{\partial^2}{\partial \phi^2}, \frac{\partial^2}{\partial \lambda^2}\right) u_0 = \frac{1}{12h^2}\left(35u_0 - 104u_1 + 114u_2 - 56u_3 + u_4\right) \tag{5.14}$$

5.5 Bi-linear interpolation

It is possible to pre-smooth the basic variables, prior to calculation of the diagnostics with a bi-linear 2-dimensional filter as follows

$$u_{i,j} = \frac{1}{8}\left(u_{i-1,j-1} + u_{i-1,j+1} + 4u_{i,j} + u_{i+1,j+1} + u_{i+1,j-1}\right) \tag{5.15}$$

Repeated applications of this filter approximates (eventually) to a Gaussian.

Appendix A

DIAGNOSTICS IN PREPARATION

A.1 QG OMEGA EQUATION INVERSION

At present, the quasi-geostrophic omega-equation inversion is only available as a self-contained program, run separately from MDIAG. This is because it uses the NAG PDE solver D03UBF (NAG 1993), which is only generally available in Fortran-77. I hope to be able to write an elliptic PDE solver in the near future so that it can be integrated into the main MDIAG Fortran-90 code.

The ω-equation with $2\nabla.Q$ and diabatic ($\frac{1}{\rho\theta}\nabla^2 S$, see below) forcing can be written as

$$\sigma\nabla_h^2\omega + f_0^2\frac{\partial^2\omega}{\partial p^2} = -2\nabla.Q - \frac{1}{\rho\theta}\nabla_h^2 S \tag{A.1}$$

where the two components of the Q-vector are given by

$$Q_1 = -\frac{R}{p}\frac{1}{a cos\phi}\frac{\partial \mathbf{V_g}}{\partial\lambda}\bullet\nabla T \tag{A.2}$$

$$Q_2 = -\frac{R}{p}\frac{1}{a}\frac{\partial \mathbf{V_g}}{\partial\phi}\bullet\nabla T \tag{A.3}$$

the Earth's radius $a = 6371000m$ and $\mathbf{V_g}$ the geostrophic wind.

The diabatic heating term is given by

$$\nabla_h^2 S = \frac{L}{c_p}\nabla_h^2\left(-\omega\frac{\partial r_s}{\partial p}\right) \tag{A.4}$$

L is the latent heat of either vapourisation or sublimation

$$L_v = 2.5008 \times 10^6 JKg^{-1} \quad T > 273.15K \tag{A.5}$$

$$L_s = 2.839 \times 10^6 JKg^{-1} \quad T < 273.15K \tag{A.6}$$

and r_s is the saturation mixing ratio (90010)

The mean static stability $\bar{\sigma}$ is given by

$$\bar{\sigma} = -\frac{1}{\bar{\rho}\bar{\theta}}\left(\frac{\bar{\partial\theta}}{\partial p}\right) \tag{A.7}$$

and the mean quantities $\bar{\rho}, \bar{\theta}, \left(\frac{\bar{\partial\theta}}{\partial p}\right)$ are averaged over the entire λ, ϕ domain (N points), at each pressure level p, *viz*

$$\bar{\rho} = \frac{1}{N}\sum_{\lambda,\phi} \rho\,(\lambda, \phi, p) \tag{A.8}$$

$$\bar{\theta} = \frac{1}{N}\sum_{\lambda,\phi} \theta\,(\lambda, \phi, p) \tag{A.9}$$

$$\left(\frac{\bar{\partial\theta}}{\partial p}\right)_p = \frac{1}{N}\sum_{\lambda,\phi} \frac{\partial\theta(\lambda, \phi, p)}{\partial p} \tag{A.10}$$

$$N = n_\phi \times n_\lambda \tag{A.11}$$

In spherical coordinates, we can write

$$\nabla_h^2 \omega = \frac{1}{a^2 cos^2\phi}\left[\frac{\partial^2\omega}{\partial\lambda^2} + cos\phi\frac{\partial}{\partial\phi}\left(cos\phi\frac{\partial\omega}{\partial\phi}\right)\right] \tag{A.12}$$

To solve this elliptic partial differential (Poisson) equation, with appropriate boundary conditions, we use the NAG 3-D inverter D03UBF. The matrix equation form can be written, in matrix-vector form as

$$M\omega = A \tag{A.13}$$

where A is(are) the forcing term(s). The seven-diagonal form is

$$\begin{aligned} a_{ijk}\omega_{ij,k-1} &+ b_{ijk}\omega_{i,j-1,k} + c_{ijk}\omega_{i-1,jk} \\ &+ d_{ijk}\omega_{ijk} \quad\;\; + e_{ijk}\omega_{i+1,jk} \\ &+ f_{ijk}\omega_{i,j+1,k} + g_{ijk}\omega_{ij,k+1} \\ &= A_{ijk} \end{aligned} \tag{A.14}$$

We can code this up using 3-point finite differences as follows :

$$
\begin{aligned}
a: \quad \omega_{ij,k-1} &= \frac{f_0^2}{(\Delta p)^2} \\
b: \quad \omega_{i,j-1,k} &= \frac{\bar{\sigma}}{2a^2 cos\phi_j (\Delta\phi)^2}\left[cos\phi_j + cos\phi_{j-1}\right] \\
c: \quad \omega_{i-1,jk} &= \frac{\bar{\sigma}}{a^2 cos^2\phi_j (\Delta\phi)^2} \\
d: \quad \omega_{ijk} &= \frac{-2\bar{\sigma}}{a^2 cos^2\phi_j (\Delta\phi)^2} - \frac{\bar{\sigma}}{2a^2 cos\phi_j (\Delta\phi)^2}\left[cos\phi_{j+1} + 2cos\phi_j + cos\phi_{j-1}\right] - 2\frac{f_0^2}{(\Delta p)^2} \\
e: \quad \omega_{i+1,jk} &= \frac{\bar{\sigma}}{a^2 cos^2\phi_j (\Delta\phi)^2} \\
f: \quad \omega_{i,j+1,k} &= \frac{\bar{\sigma}}{2a^2 cos\phi_j (\Delta\phi)^2}\left[cos\phi_j + cos\phi_{j-1}\right] \\
g: \quad \omega_{ij,k+1} &= \frac{f_0^2}{(\Delta p)^2}
\end{aligned}
\tag{A.15}
$$

The boundary conditions used in the solution are

$$\omega_{ij,k=1} = 0 \tag{A.16}$$

$$\omega_{ij,k=TOP} = 0 \tag{A.17}$$

which may not necessarily be good at say 100mb (the top layer in the LAM).

In the Unified Model, $\Delta\lambda, \Delta\phi$ take the following values on the true lat.lon. global and rotated lat/long equatorial UK Mesoscale grid,

$$
\begin{aligned}
\Delta\lambda, \Delta\phi &= 0.8333^c, -0.5556^c \quad Global\, 432 \times 325 \\
\Delta\lambda, \Delta\phi &= 0.11^c, 0.11^c \quad UKMesoscale\, 146 \times 182
\end{aligned}
\tag{A.18}
$$

The NAG routine DO3UBF needs some tuning to achieve a balance between convergence and time spent iterating to convergence. We limit the number of iterations to a maximum of **100** and set the convergence criteria so that both the **maximum residual and residual change** are **below 1%**. Another important parameter which controls the speed of convergence is the acceleration parameter **APARAM**, which, after much experimentation, we set as **APARAM=15.0D0**.

NB All NAG parameters **MUST BE** in either **INTEGER or DOUBLE PRECISION**.

Appendix B

PP-FORMAT FILE HEADERS

Format of the pp-format file header

Listed below are details of the format of the 64-word header required for any field in 'PP-format'.

```
 1  LBYR      Year  (eg 1986 or 86) \
 2  LBMON     Month           (1-12)  |  Validity time of field,
 3  LBDAT     Day of month    (1-31)  |          - or -
 4  LBHR      Hour            (0-23)  |  Start of averaging period (for time
 5  LBMIN     Minute          (0-59)  |  mean fields).
 6  LBDAY     Day number of run       /

 7  LBYRD     Year  (eg 1986 or 86) \
 8  LBMOND    Month           (1-12)  |  Data time (for forecast fields),
 9  LBDATD    Day of month    (1-31)  |          - or -
10  LBHRD     Hour            (0-23)  |  End of averaging period (for time
11  LBMIND    Minute          (0-59)  |  mean fields).
12  LBDAYD    Day number of run       /

13  LBTIM     Time indicator.  This indicates what the times in words 1-12
              represent.  Referring to the times represented by words 1-6 and
              7-12 as 'T1' and 'T2' respectively, LBTIM is coded as
              (100*IA + 10*IB + IC) where:

                IA = 0  except for time mean fields in which case IA
                        is the time interval in hours between the
                        individual fields from which the mean was
                        computed (IA may be left as zero for
```

time-means to indicate that the time interval is unspecified).

IB = 0 if only the validity time (T1) is valid.
= 1 if the field is a forecast from T2 valid at T1.
= 2 if the field is a time mean between T1 and T2, or represents a sequence of times between T1 and T2.
= 3 if the field is a time mean from T1 to T2 for each year from LBYR to LBYRD.
= 4 if the field is a difference between fields valid at T1 and T2 (in sense T2-T1).
= 5 if the field is a mean daily cycle between T2 and T1

IC = 0 if 'model time' is used for T1 and T2 (i.e. only day number, hour and minute are set).
= 1 if the 'real' (i.e. Gregorian) calendar is used for T1 and T2.
= 2 if the '360-day' year calendar (i.e. 12 30-day months) is used for T1 and T2. (This is used in Met.O.20 for some model runs.)
= 3 if 'model time' is used for T1 and T2 (i.e. only day number, hour and minute are valid; year, month and day in month are to be ignored if set).

'IC' corresponds to the parameter MCAL in COMCON. If 'IC' is 1 or 2, coding of the 'day numbers' (words 6 and 12) is optional: code as 0 if not used.

14 LBFT Forecast period (hours).

15 LBLREC Length of data record in words (including any 'extra data').

16 LBCODE Grid code. This indicates the type of grid and is coded as:

1 Regular latitude/longitude grid.
2 Regular lat/long grid boxes (grid points are box centres).
3 Polar stereographic grid.
4 Spectral coefficients.
7 Mercator grid.

8 Plane polar grid.
9 Plane Cartesian grid.

For grids with non-standard polar axis, add 100 to the above numbers.

For cross sections, code as (10000 + 100*IX + IY) where IX and IY are codes for the x- and y-axes from the list below. Cross section fields indicated in this way must contain x- and y-coordinate vectors in the extra data. When LBCODE is coded as 30000 + 100*IX + IY, with IX and IY from the same list, the axes are given the same interpretation as for a normal cross section, but coordinate vectors need not be supplied. In this case, however, the field is not regarded as a cross section by PP cross-section routines.

Axis codes are as follows:

0 Sigma (or eta, for hybrid coordinate data).
1 Pressure (mb).
2 Height above sea level (km).
3 Eta (U.M. hybrid coordinates) only.
4 Depth below sea level (m).
5 Model level.
6 Theta.
7 Sigma only.
10 Latitude (degrees N).
11 Longitude (degrees E).
12 (Horizontal) distance (km).
13 Site number (set of parellel rows or columns eg timeseries)
20 Time (days). (Gregorian calendar (not 360 day year) if distinction applicable)
21 Time (months).
22 Time (years).
23 Time (model days with 360 day model calendar)
31 Logarithm to base 10 of pressure in mb.
40 Pseudolevel
99 Other.

17 LBHEM Hemisphere indicator. For geographical grids (lat/long, polar

stereographic, Mercator or 'Kurihara' grids), this is coded as:

- 0 Global field (i.e. covering the WHOLE globe).
- 1 Northern hemisphere polar stereographic grid, or other geographic grid covering the WHOLE northern hemisphere.
- 2 Southern hemisphere polar stereographic grid, or other geographic grid covering the WHOLE southern hemisphere.
- 3 Limited area of globe without 'wrap-around' (i.e. NOT covering the full 360-degree longitude range).
- 4 Limited area of globe with 'wrap-around' (i.e. covering the full 360-degree longitude range).

For SPECTRAL COEFFICIENTS, code 0, 1 or 2 as appropriate.
For CROSS-SECTION FIELDS, a value of 3 should be coded.
For GRAPH FIELDS, a value of 5 should be coded.
For LATERAL BOUNDARY DATA a value of 99 should be coded.

18 LBROW Number of rows in field.
(For spectral coefficients, code the 'n' truncation level.)

19 LBNPT Number of grid points in each row.
(For staggered grids, code values appropriate for the longest row. For spectral coefficients, code the 'm' truncation level.)

20 LBEXT Length of 'extra data' (x- and y-vectors for cross sections, and field title if any) in words.

21 LBPACK Packing method indicator (for fields in packed format.)

- 0 Field not packed
- 1 Field packed using WGDOS archive method
- 2 (Reserved for GRIB code data)
- 2000 unpacked Cray data
- 2001 packed Cray data

22 LBREL Header release number. (Set to 2 for format described here.)

23 LBFC Field code. This indicates what the data in the field represents Some of the more common field codes are listed below.

1 Height field	73 Relative vorticity field

8	Pressure field	74	Divergence field
16	Temperature field	88	Relative humidity field
40	Vertical velocity (dp/dt)	90	Total Precipitation
56	Westerly wind component	95	Specific humidity field
57	Southerly wind component		

24 LBCFC Second field code. This is only used for a field which is a combination of two field types; for example, a meridional temperature flux field (a product of V and T). Coding is as for the field code above.

25 LBPROC Processing code. This indicates what processing has been done to the basic field. It should be 0 if no processing has been done: otherwise add together the relevant numbers from the list below:

1	Difference from another experiment.
2	Difference from zonal (or other spatial) mean.
4	Difference from time mean.
8	X-derivative (d/dx).
16	Y-derivative (d/dy).
32	Time derivative (d/dt).
64	Zonal mean (or spatially smoothed) field.
128	Time mean field.
256	Product of two fields.
512	Square root of a field.
1024	Difference between fields at levels BLEV and BRLEV.
2048	Mean over layer between levels BLEV and BRLEV.
4096	Minimum value of field during time period.
8192	Maximum value of field during time period.
16384	Magnitude of a vector, not specifically wind speed
32768	log10 of a field.
65536	Variance of a field.
131072	Mean over an ensemble of parallel runs.

(For details of BLEV and BRLEV, see words 32 and 33.)

26 LBVC Vertical co-ordinate type. The 'vertical co-ordinate' is the one which has the same value at every grid point; e.g. 'pressure' for a 500mb height field or 'longitude' for a zonal mean cross section. The co-ordinate type is coded using the

table of field codes as for word 23. Numbers 126-139 are used for special levels. The most common vertical co-ordinate types are:

1	Height (m)	8	Pressure (mb)
9	Hybrid co-ordinates	10	Sigma (=p/p*)
128	Mean sea level	129	Surface
130	Tropopause level	131	Maximum wind level
132	Freezing level	142	Upper hybrid level
143	Lower hybrid level	176	Latitude (deg)
177	Longitude (deg)		

27 LBRVC Vertical co-ordinate type for reference level. Used only when a reference level is applicable as in thickness fields or layer mean fields. Coded as above.
Set to 0 by the U.M. if no reference level.

28 LBEXP Experiment number (optional parameter for user's reference only).
In Operational use this is set to the ITAB number:

1 Global main run data
2 Global update run data
3 Limited-area main run data
4 ECMWF data
5 Washington data
6 Limited-area update run data
7 Global wave data
8 European wave data
9 UK Mesoscale model main runs
(Formerly limited-area SIGMA/ETA data (obsolete))
10 Paris data

21 Bosnia Mesoscale model runs
22 Gulf Mesoscale model runs
23 Stratospheric model runs
24 Short term Mesoscale models
25 FOAM model runs

29 LBEGIN (For fields on direct access datasets only) Address of start of field in direct access dataset.

30	LBNREC	(For fields on direct access datasets only) Number of records occupied by field on direct access dataset.
31	LBPROJ	(For Met Office fields file use) Fields file projection number.
32	LBTYP	(For Met Office fields file use) Fields file field type code.
33	LBLEV	(For Met Office fields file use) Fields file level code. 7777 = multi-level field in lateral boundary data.
34-37	LBRSVD(4)	Reserved for future PP-package use.
38	LBSRCE	In Met Office Unified Model, set to 1111 to indicate items 39-43 are in use as below. Otherwise, spare for user's use.
39	LBUSER(1)	In Met. Office Unified Model, Indicator for real/integer/logical/land points only/ timeseries 1=real data for all output fields; fields of INTEGER/LOGICAL type will not be available until further development work on the diagnostic system is undertaken, as FIELDCOS does not handle them.
40	LBUSER(2)	In Met. Office Unified Model, Start address in DATA
41	LBUSER(3)	In Met. Office Unified Model, For timeseries - number of sampling periods
42	LBUSER(4)	In Met. Office Unified Model, the STASH code
43	LBUSER(5)	In Met. Office Unified Model, Timeseries indicator 1=data is timeseries 0=not (STASH pseudo dimension)
44	LBUSER(6)	No present use - spare for user's use, as are items 39-43 outside the Met. Office Unified Model. See also UM Documentation Paper F3.
45	LBUSER(7)	As 44. Note: there is presently an ambiguity in the Met.Office Unified Model about the type of item 45. Dumps and PP fields within the model define the first 45 items as integer and items 46 onwards as real. The integer array LBUSER is equivalenced to a real array BUSER.

46-49 BRSVD(4) Reserved for future PP-package use.

50 BDATUM Constant value subtracted from each value in field. This is usually zero but would be 273.15 for a temperature field in degrees Celsius.

51 BACC (Packed fields only) Packing accuracy.

52 BLEV Level. This is the value of the vertical co-ordinate LBVC (word 26) appropriate for the field; e.g. '500.0' for a 500 mb height field. For hybrid levels code the 'B'-value of the level. BLEV should be zero if the vertical co-ordinate type is in the range 128 to 139.

53 BRLEV Reference level. This is the value of the vertical co-ordinate LBRVC (word 27) appropriate for the field. It is used when a second level is relevant as for example with thickness fields when the 'reference level' should be the one nearest the ground. Code as for BLEV.

54 BHLEV (Hybrid levels) 'A'-value of level.

55 BHRLEV (Hybrid levels) 'A'-value of reference level.

56 BPLAT Real latitude of 'pseudo' N pole of projection. Code as '90.0' for fields on grid with normal polar axis.

57 BPLON Real longitude of 'pseudo' N pole of projection. Code as '0.0' for fields on grid with normal polar axis.

```
58  BGOR   \
59  BZY     |  These five parameters define the grid for the field.
60  BDY     |  The coding depends on what type of grid the field is on.
61  BZX     |  Details for various types follow:
62  BDX    /
```

(i) Latitude/longitude grids.
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

BGOR Not used - set to zero.

BZY Latitude of 'zeroth' row (i.e. an imaginary row one grid length before the first row) in degrees (north positive).

BDY Latitude interval between rows in degrees (negative if rows are north to south).
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

```
BZX    Longitude of 'zeroth' point in row (i.e. an imaginary
       point one grid length before the first point) in degrees
       (east positive).
BDX    Longitude spacing of points in each row in degrees
       (negative if points run from east to west).

(ii) Polar stereographic grids.
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
BGOR   Grid orientation. This is the longitude in degrees of the
       meridian which would be vertical with north at the top on
       a chart drawn for the grid.
BZY    Reference latitude in degrees (see BDY).
BDY    Grid length in metres at reference latitude.
                     \   These co-ordinates are in grid lengths
BZX    'X' co-ordinate |  (not necessarily whole numbers and not
       of the pole.    |  necessarily representing a point
BDX    'Y' co-ordinate |  within the area covered by the grid)
       of the pole.    |  counting the bottom left grid point of
                     /   a chart as the point (1.0, 1.0).
(iii) Mercator grids.
~~~~~~~~~~~~~~~~~~~~~   For Mercator grids, the coding is the
same as in (i) except that BZY and BDY are values of:

    (earth's radius) * log((1+sin(latitude))/cos(latitude))

instead of latitude. (The earth's radius is in metres and 'log'
refers to the natural logarithm.)

(iv) Cross sections.
~~~~~~~~~~~~~~~~~~~~    For cross-section grids, coding is as in
(i) above except that BZY and BDY are values of the 'y'
co-ordinate and BZX and BDX are values of the 'x' co-ordinate.
(Note that for zonal mean fields or cross sections along a
meridian, BZX and BZY are longitudes.)  If the cross section is
on irregularly spaced levels, BZY and BDY should both be zero.

(v) Graph fields.
~~~~~~~~~~~~~~~~~       For graph fields, coding is as in (i) above
except that BZX and BDX are values of the 'x' co-ordinate, and
BDY and BZY are both coded as 1.0. If points on the graphs are
not evenly spaced in the 'x' co-ordinate, BZX and BDX should
```

both be coded as zero.

(vi) Spectral coefficients.

BGOR	Not used - coded as zero.
BZY	Not used - coded as zero.
BDY	Not used - coded as zero.
BZX	Reference longitude - usually 0.0. This is the meridian which is treated as the origin of longitude for the coefficients.
BDX	Not used - coded as zero.

63 BMDI Value used in the field to indicate missing data points. If the field contains no missing data, code a value of -1.0E30.

64 BMKS MKS scaling factor, equal to the size of the unit in which the field is expressed divided by the corresponding mks unit. (e.g. 10.0 for height in dam, 100.0 for pressure in mb, 0.01 for relative humidity in % etc.).

Appendix C

MDIAG.DIAGNOSTICS

```
# MDIAG.DIAGNOSTICS.FULL
#
# The following are the bottom, top and delta-p pressure levels on which
#   the computations are to be made
1000,50,50
#
# Below we specify the lat/lon of the top left and bottom right of the sub-grid
#  on which we wish to calculate
#  - code as 0,0,0,0 (the default) to use the whole domain
0.0,0.0,0.0,0.0
#
# The following integer value is the number of times we wish to apply a
#  linear 2D filter (average); this filter is applied to the 6 basic variables
#  GEOP_HT, TEMPERATURE, SPECIFIC_HUMIDITY, UWIND, VWIND, OMEGA
#  - set the default to zero (no averaging)
0
#
# Now we list the DIAGNOSTICS we require,
# Lines beginning with # are not yet available
90001  Geopotential Height                          m
90002  Temperature                                  K
90003  U component of Wind (wrt Model grid)         m s-1
90004  V component of Wind (wrt Model grid)         m s-1
90005  Relative Humidity (ice)                      %
90006  Potential Temperature                        K
90007  Equivalent Potential Temperature             K
90008  Density                                      Kg m-3
90009  Humidity Mixing Ratio                        DIMENSIONLESS
```

```
90010  Sat Equ Pot Temp                          K
90011  du/dp                                     m s-1 Pa-1
90012  dv/dp                                     m s-1 Pa-1
90013  dTheta/dp                                 K Pa-1
90014  dTheta_e/dp                               K Pa-1
90015  Div V                                     s-1
90016  Dry PV                                    K Kg-1 m s-2
90017  Moist PV                                  K Kg-1 m s-2
90018  Geostrophic U Wind (wrt Model grid)       m s-1
90019  Geostrophic V Wind (wrt Model grid)       m s-1
90020  Wind strength                             m s-1
90021  Q1                                        K m-1 s-1
90022  Q2                                        K m-1 s-1
90023  Div Q                                     K m-2 s-1
90024  Absolute Vorticty                         s-1
90025  Vertical Velocity                         m s-1
90028  dug/dp                                    m s-1 Pa-1
90029  dvg/dp                                    m s-1 Pa-1
90031  Mag Geo Defmn                             s-1
90032  u cpt  Gedef axis (wrt Model grid)        s-1
90033  v cpt  Gedef axis (wrt Model grid)        s-1
90034  Mag Grad(ThW)                             K m-1
90035  u cpt  Grad(ThW)  (wrt Model grid)        K m-1
90036  v cpt  Grad(ThW)  (wrt Model grid)        K m-1
90037  Geo Defmn wrt Theta_W                     s-1
90038  Relative Vorticity                        s-1
90039  Geostrophic Relative Vorticity            s-1
90040  Wind direction                            Degrees
90041  Theta-W                                   K
90042  Ageostrophic U Wind (wrt Model grid)      m s-1
90043  Ageostrophic V Wind (wrt Model grid)      m s-1
90044  Mag Grad(Th)                              K m-1
90045  u cpt  Grad(Th)    (wrt Model grid)       K m-1
90046  v cpt  Grad(Th)    (wrt Model grid)       K m-1
90053  Relative  Humidity (water)                %
90061  Frontal Zones (ThW)                       u
90062  Front -^DelSqd (ThW)                      K m-2
90063  Front -DelSqd (ThW)                       K m-2
90064  Front |grad (ThW)|                        K m-1
90065  ThW-Frnt Spd                              m s-1
90066  ThW-advection                             K s-1
```

90067	Geost ThW-advn	K s-1
90068	Frontal Zones (Th)	u
90069	Front -^DelSqd(Th)	K m-2
90070	Front -DelSqd(Th)	K m-2
90071	Front |grad(Th)|	K m-2
90072	Th-Frnt Spd	m s-1
90073	Th-advection	K s-1
90074	Geost Th-advn	K s-1
90075	Shear relative vorticity	
90076	Jet angle	
90077	Specific Humidity	Kg Kg-1
90080	Frontal waves	K^2 m-3 s-1
90081	F1 vector	K m-1 s-1
90082	F2 vector	K m-1 s-1
90083	Frontogenesis tilt	K^2 m-2 s-1
90084	Frontogenesis function	K^2 m-2 s-1
90085	u-wind WESTERLY	m s-1
90086	v-wind SOUTHERLY	m s-1
90087	omega	mb s-1
90088	dry static stability (N^2)	s-2
90089	moist static stability (N^2_m)	s-2
90090	Dry Richardson No.	DIMENSIONLESS
90091	Moist Richardson No.	DIMENSIONLESS
90096	Cloud liquid water	kg kg-1
90097	Cloud ice	kg kg-1
90101	LAT/LON on both the rotated and true grids	

Appendix D

IMPORTANT FORTRAN-90 SOURCE

D.1 ALL_CONSTANTS.f90

```
MODULE ALL_CONSTANTS
! Thermodynamic, meteorological and UM constants available to all routines

 IMPLICIT NONE

 SAVE

! Filenames and LUN's are hardwired
 CHARACTER (LEN = 17), PARAMETER :: diagnostic_file = 'MDIAG.DIAGNOSTICS'
 CHARACTER (LEN = 12), PARAMETER :: latlon_file     = 'LATLON.ASCII'
 CHARACTER (LEN = 14), PARAMETER :: input_file      = 'MDIAG_INPUT.PP'
 CHARACTER (LEN = 15), PARAMETER :: output_file     = 'MDIAG_OUTPUT.PP'
 CHARACTER (LEN = 12), PARAMETER :: orography_file  = 'OROGRAPHY.PP'
 INTEGER,              PARAMETER :: LUN_DIAGNOSTICS = 11
 INTEGER,              PARAMETER :: LUN_LATLON      = 12
 INTEGER,              PARAMETER :: LUN_MDIAGIN     = 13
 INTEGER,              PARAMETER :: LUN_MDIAGOUT    = 14
 INTEGER,              PARAMETER :: LUN_OROGRAPHY   = 15


! MDIAG STASH diagnostics - MINIMUM and MAXIMUM values are used to dimension
!  the array SAVEIT in MODULE DIAGNOSTICS_TO_COMPUTE
 INTEGER,              PARAMETER :: STASH_MIN   = 90001
```

```
INTEGER,              PARAMETER :: STASH_MAX    = 90200


! Dry air constants
REAL, PARAMETER :: R_D                = 287.05          ! Gas constant dry air J/kg/K
REAL, PARAMETER :: CP_D               = 1005.0          ! Isobaric spec. heat cap J/kg/I
REAL, PARAMETER :: CV_D               = 719.0           ! Isochoric spec. heat cap J/kg,
REAL, PARAMETER :: GAMMA_D            = CP_D/CV_D       ! Ratio of specific heats
REAL, PARAMETER :: KAPPA_D            = R_D/CP_D        ! Ratio of specific heats
REAL, PARAMETER :: LAPSE              = 0.0065          ! Near surface lapse rate K/km
REAL, PARAMETER :: LAPSE_TROP         = 0.002           ! Tropopause lapse rate K/km
REAL, PARAMETER :: G                  = 9.80665         ! Surface gravity m/s^2
REAL, PARAMETER :: CPD_OVER_G         = CP_D/G
REAL, PARAMETER :: LAPSE_R_OVER_G = LAPSE*R_D/G
REAL, PARAMETER :: M_D                = 0.02896         ! Mean mol. wt dry air Kg/mol

! Moist (water vapour) constants
REAL, PARAMETER :: M_V                = 0.018015        ! Mean mol. wt water Kg/mol
REAL, PARAMETER :: EPSYLON            = M_V/M_D
REAL, PARAMETER :: C_VIRTUAL          = 1.0/EPSYLON-1.0
REAL, PARAMETER :: RSTAR              = 8.314           ! Universal gas constant
REAL, PARAMETER :: R_V                = 461.51          ! Gas constant for water J/kg/K
REAL, PARAMETER :: CP_V               = 1850.0          ! Isobaric spec. heat cap. J/kg
REAL, PARAMETER :: CV_V               = 1390.0          ! Isochoric spec. heat cap. J/k
REAL, PARAMETER :: GAMMA_V            = CP_V/CV_V       ! Ratio of spec. heats
REAL, PARAMETER :: KAPPA_V            = R_V/CP_V        ! Ratio of spec. heats

! Other thermodynamic constants
REAL, PARAMETER :: PREF               = 100000.                  ! Reference pressure Pa
REAL, PARAMETER :: LH_FUSION          = 3.34E5                   ! J/kg
REAL, PARAMETER :: LH_VAPOUR          = 2.501E6                  ! J/kg
REAL, PARAMETER :: LH_SUBLIM          = LH_FUSION + LH_VAPOUR ! J/kg
REAL, PARAMETER :: TZERO              = 273.15                   !

! Other useful constants
REAL, PARAMETER :: PI                 = 3.141592653589
REAL, PARAMETER :: DEG_TO_RADIANS = PI/180.0
REAL, PARAMETER :: RADIANS_TO_DEG = 180.0/PI
REAL, PARAMETER :: EARTH_RADIUS       = 6371000.0       ! m
REAL, PARAMETER :: EARTH_OMEGA        = 7.292E-5        ! /s
REAL, PARAMETER :: SMALL              = EPSILON(1.0)
```

```
! UM constants
 REAL, PARAMETER :: MISSING_DATA   = -32768.0*32768.0

END MODULE ALL_CONSTANTS
```

D.2 REQUIRED_FIELDS

```
MODULE REQUIRED_FIELDS
! Array dimensions, parameters and other variables are determined at run-time
!  using SUBROUTINE getGridParameters
! The arrays are all ALLOCATEable whose sizes are determined at run-time
! We can now set the (regular) PRESSURE grid onto which we want to interpolate.
! A list of the required diagnostics is saved as a LOGICAL SAVEIT
!  array, and the entire array is initialised to .FALSE.
! The AK, AKH, BK and BKH coeffients required for the height to pressure
!  interpolations are now found from the pp-file and are set up at run-time
! The number of pp-levels are determined from the i/p pp-file also at run-time.

  USE ALL_CONSTANTS

  IMPLICIT NONE

  SAVE

! Whether the wind data are on a staggerred grid or not
   LOGICAL                                       :: STAGGERRED = .FALSE.

! Number of x,y points of the input wind data on model-levels
   INTEGER                                       :: NUX = 0
   INTEGER                                       :: NUY = 0

! Number of x,y points of the remaining input data on model-levels
   INTEGER                                       :: NX = 0
   INTEGER                                       :: NY = 0

! Number of model vertical (eta) levels
   INTEGER                                       :: NPP = 0

! Number of output (pressure) levels
   INTEGER                                       :: NZ = 0

! Number of iterations of a linear 2-d smoothing filter to be applied to the
!  basic fields (phi, T, u, v, w, q) - the resultant diagnostics will be
!  smoother in apperance.
   INTEGER                                       :: FILTER = 0
```

```
! Constant latitude and longitude increments
  REAL                                              :: DPHI = 0.0
  REAL                                              :: DLAMBDA = 0.0

! Array of pressures and difference in pressure between levels
  REAL, DIMENSION(:),      ALLOCATABLE              :: PRESSURE
  REAL                                              :: DELTA_P = 0.0

! Arrays of equatorial (for rotated grids)  and real latitude and longitude
  REAL, DIMENSION(:,:),    ALLOCATABLE, TARGET      :: EQLAT
  REAL, DIMENSION(:,:),    ALLOCATABLE, TARGET      :: EQLON
  REAL, DIMENSION(:,:),    ALLOCATABLE, TARGET      :: THELAT
  REAL, DIMENSION(:,:),    ALLOCATABLE, TARGET      :: THELON

! COSPHI is required for the horizontal derivatives
  REAL, DIMENSION(:,:),    ALLOCATABLE, TARGET      :: COSPHI

! Coriolis parameter as a function of lat/lon
  REAL, DIMENSION(:,:),    ALLOCATABLE, TARGET      :: CORIOLIS

! A, B full and half-level coefficients required for calculation of temperature
!  and geopotential height on pressure levels
  REAL, DIMENSION(:),      ALLOCATABLE              :: AK
  REAL, DIMENSION(:),      ALLOCATABLE              :: BK
  REAL, DIMENSION(:),      ALLOCATABLE              :: AKH
  REAL, DIMENSION(:),      ALLOCATABLE              :: BKH

! lat/lon sub-grid specification - _I and _J refer to the array indices to which
!  _LAT and _LON correspond
!   TL top left corner
!   BR bottom right corner
  REAL                                              :: TL_LAT = 0.0
  REAL                                              :: TL_LON = 0.0
  REAL                                              :: BR_LAT = 0.0
  REAL                                              :: BR_LON = 0.0
  INTEGER                                           :: TL_I = 0
  INTEGER                                           :: TL_J = 0
  INTEGER                                           :: BR_I = 0
  INTEGER                                           :: BR_J = 0

! The array that contains the diagnostics to be computed - elements
```

```
!  are set according to the file MDIAG.DIAGNOSTICS
   LOGICAL, DIMENSION(STASH_MIN:STASH_MAX)       :: SAVEIT = .FALSE.

! The following basic fields on pressure-levels are required and accessed
!  by nearly all subroutines.
   REAL, DIMENSION(:,:),     ALLOCATABLE, TARGET  :: PHISTAR
   REAL, DIMENSION(:,:,:),   ALLOCATABLE, TARGET  :: GEOP_HT
   REAL, DIMENSION(:,:,:),   ALLOCATABLE, TARGET  :: TEMPERATURE
   REAL, DIMENSION(:,:,:),   ALLOCATABLE, TARGET  :: UWIND
   REAL, DIMENSION(:,:,:),   ALLOCATABLE, TARGET  :: VWIND
   REAL, DIMENSION(:,:,:),   ALLOCATABLE, TARGET  :: SPECIFIC_HUMIDITY
   REAL, DIMENSION(:,:,:),   ALLOCATABLE, TARGET  :: OMEGA
   REAL, DIMENSION(:,:,:),   ALLOCATABLE, TARGET  :: THETA
   REAL, DIMENSION(:,:,:),   ALLOCATABLE, TARGET  :: THETAE
   REAL, DIMENSION(:,:,:),   ALLOCATABLE, TARGET  :: THETAW

! FOUNDPHI (for the orography) is required and read in only once, since it
!  is constant
   LOGICAL                                        :: FOUNDPHI = .FALSE.

! A return code for the success (0) or failure (1) of a subroutine
   INTEGER                                        :: IFAIL = 0

  CONTAINS

    SUBROUTINE deallocateRequiredFields

      IF(ALLOCATED(PHISTAR))             DEALLOCATE(PHISTAR)
      IF(ALLOCATED(GEOP_HT))             DEALLOCATE(GEOP_HT)
      IF(ALLOCATED(TEMPERATURE))         DEALLOCATE(TEMPERATURE)
      IF(ALLOCATED(UWIND))               DEALLOCATE(UWIND)
      IF(ALLOCATED(VWIND))               DEALLOCATE(VWIND)
      IF(ALLOCATED(SPECIFIC_HUMIDITY))   DEALLOCATE(SPECIFIC_HUMIDITY)
      IF(ALLOCATED(OMEGA))               DEALLOCATE(OMEGA)
      IF(ALLOCATED(THETA))               DEALLOCATE(THETA)
      IF(ALLOCATED(THETAE))              DEALLOCATE(THETAE)
      IF(ALLOCATED(THETAW))              DEALLOCATE(THETAW)
      IF(ALLOCATED(EQLAT))               DEALLOCATE(EQLAT)
      IF(ALLOCATED(EQLON))               DEALLOCATE(EQLON)
      IF(ALLOCATED(THELAT))              DEALLOCATE(THELAT)
      IF(ALLOCATED(THELON))              DEALLOCATE(THELON)
```

```
    IF(ALLOCATED(COSPHI))           DEALLOCATE(COSPHI)
    IF(ALLOCATED(CORIOLIS))         DEALLOCATE(CORIOLIS)
    IF(ALLOCATED(PRESSURE))         DEALLOCATE(PRESSURE)

  END SUBROUTINE deallocateRequiredFields

END MODULE REQUIRED_FIELDS
```

ACKNOWLEDGEMENTS Thanks to all those who responded to my questionnaire and additionally to Morwenna Griffiths and Michael Ziemianski for very useful discussions on the omega-equation inversion. Also thanks to Danny Chapman for IDL code that computes the moist Richardson number and static stability.

Bibliography

[1] Abramowitz M, Stegun I.A.: 1972, "Handbook of mathematical Functions", publ. Dover

[2] Bolton D., 1980: Mon. Weather Review, **108**, 1046

[3] Cullen, M.J.P., Davies T., Mawson, M.: 1993, "Unified model documentation paper **10**"

[4] Durran D.R., Klemp J.B.: 1982, J. Atmos. Sci. **39**, 2152

[5] Goddard D.M.: 1998, "Unified model documentation paper **S1**"

[6] Hewson T.D.: 1997, *Meteorolog. Appl.*, **4**, 311

[7] Hewson T.D.: 1998, *Meteorolog. Appl.*, **5**, 37

[8] Holton J.R.: 1992, "An introduction to dynamic meteorology" publ. Academic Press

[9] Keyser D., Reeder M.J., Reed R.J.: 1988, *Mon. Weather Rev.* **116**, 72

[10] Kirk T.H.: 1965, *Meteorolog. Mag.*, **94**, 351

[11] Landolt-Bornstein: 1987, "Numerical data and functional relationships in Science and technology, Group V/Vol 4B, Physical and Chemical properties of air" 25

[12] NAG: 1993, "Fortran Library Manual, Mark 16"

[13] Panagi P.M., Dicks E.M.: 1997, "Met. Office Unified Model data, Diagnostics, graphics programs and other observational data available from the JCMM through the aegis of the Universities Weather Research Network UWERN", JCMM internal report **69**

[14] Renard R.J., Clarke L.C.: 1965, *Mon. Weather Rev.*, **93**, 547

[15] Roberts N.: 1998, "How to use the display program DISP" JCMM internal report **91**

[16] Rotunno R, Skamarock W.C., Snyder C.: 1994 *J. Atmos. Sci.* ,**51**, 3379

[17] The Met. Office: 1998, "The Unified Model User Guide"