Technical description of the air mass transformation model

W.M. Remmers

technical reports TR-88

technische rapporten TR-88

de bilt, 1986

TECHNICAL DESCRIPTION OF THE AIR MASS TRANSFORMATION MODEL.

Author : W.M. Remmers, KNMI, The Netherlands.

Table of contents:

1. INTRODUCTION.

The complete air mass transformation model (AMT-model) consist of two large modules:

+ the trajectory-calculation model
+ the atmospheric boundary layer model.

The first program uses the ECMWF-output to calculate backward/forward trajectories. It can be run on analysed or forecasted windfields, thus giving socalled analysed or forecasted trajectories.
The second program uses the ECMWF-trajectories and data from radiosoundings in the surroundings of the trajectory-starting points to construct an initial profile for the potential temperature and the specific humidity. The boundary layer model advects this profile from the source-area to the receptor-point (De Bilt or any other station).

The programming language was ALGOL. Since ALGOL is not widely spread over the world and because there is an increasing interest for boundary-layer models, it was decided to translate the AMT-model into FORTRAN ´77. The old ALGOL program had three mainroutines:

1.	Decodetemps	-reads the selected radiosonde-data outof large so called Dcm/Decotemp files,where all information of all radiosoundings at a specific date & time was recorded in a decoded (=compressed) form.
2.	Analysetemps	-this module takes all selected radio-soundings in the source-area together by using distance depending weighing factors.
3.	ABM	-this module describes the evolution of the Boundary Layer along the trajectories.

The trajectory program and the decode-procedure being totally dependent of the specific KNMI-situation, we focussed our attention at the second and third procedure.This Technical Report will give a complete description of these FORTRAN 77 programs (Version V FORTRAN 1).

2. NOTATIONS.

The general setup of the program is in accordance to the suggested coding conventions as proposed for the HIRLAM-project (ref. 1). Emphasis is laid on the naming conventions of the different types of variables. Throughout the program the first and sometimes also the second character of the name follows from the next table:

STATUS \ TYPE	INTEGER	REAL	LOGICAL	CHARACTER
Local	I	Z	LO	YO
Dummy argument	K	P(NOT PP)	LA	YA
Loop control	J(NOT JP)	-	-	-
Parameter	JP	PP	LP	YP
Global or common	M,N	A-G,O,Q-X	L (NOT LO,LA,LP)	Y (NOT YO,YA,YP)

A FORTRAN 77-variable name is a sequence of six characters. This system clearly obscures the identification of physical variables within the program. To raise this problem we have constructed some identification tables for the larger subroutines .

The larger subroutines have the following declarative statement to encourage the programmer to use this scheme:

```
IMPLICIT LOGICAL(L),CHARACTER*8(Y)
```

To improve the legibility of the program the use of common blocks is restricted to hold physical constants such as RGASS(gasconstant) and G(gravityconstant). Subroutine in- and output is done completely by the list of dummy arguments following the declaration. Output dummy arguments will be underlined in this technical report. Thus the

```
  INTEGER FUNCTION MAXMUM(PH,K,KIN)
  REAL PH(*)
  INTEGER K,KIN
```

has the REAL array PH,the INTEGERS K and KIN as input. This function outputs (by its function name) the index of the maximum value found in the array PH , so MAXMUM is underlined.

The AMT-model decribed in this report version runs on the KNMI-Burroughs computer. Although some constructs (data-cards/ filename-conventions) are machine dependent, they are still included for clarity. Typical Burroughs/KNMI/ECMWF-constructs are indicated by a vertical line at the left side of the text, f.e.:

```
| IDEP=INT(BKUPD(REAL(ISHIFT),12.))
  ISUF=IMT
| WRITE(ZHNAME,'(A,I2,A,I8,A)') 'FORETRAJ/',ISUF,'/',ISHIFT,
|*                              'ON TEMP.'
  CALL QUEST(---------------)
```

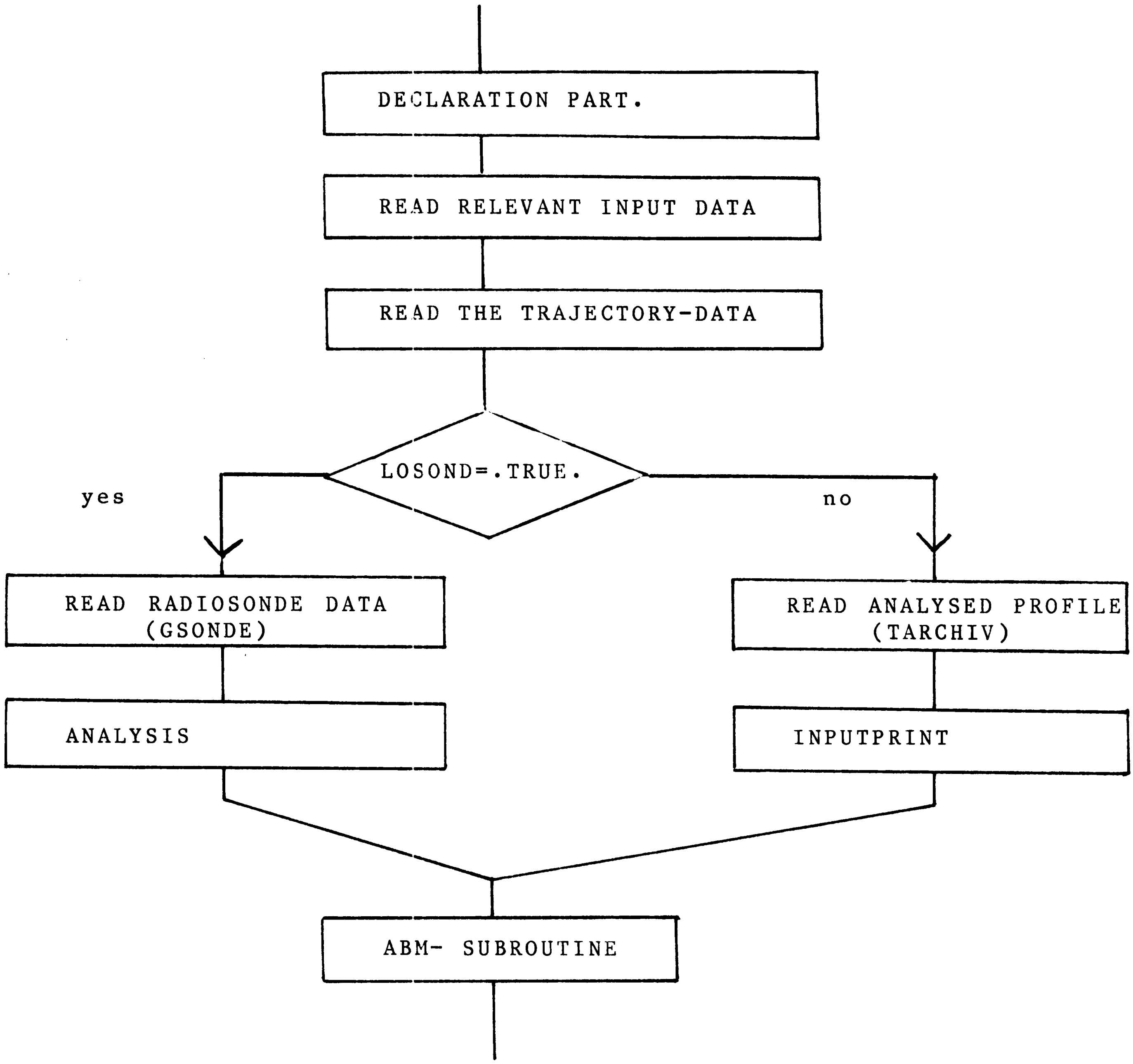
Although the 'DOCTOR standard' (ref. 1) encourages the practice of building in as much explanatory text as possible,one should not go into extremes: the program length would become unpracically large , so only condensed information and extra comment is included in the source text. The reader should look for the corresponding passage in this Technical Report.

We left out all greek and other special symbols in this text so that this file can be printed out on any printer (matrix/daisy wheel) . The only special symbol in this file is "#", which stands for : "is proportional to".

3. DESCRIPTION OF THE MAIN-PROGRAM.

The MAIN-program takes care that all manipulations and subroutines are carried out in the correct order. First it reads some datacards that give programmatic and physically relevant information (time for calculations,timespan AMT-model,cloudcover and seawater-temperatures and the logical LOSOND). Normally the program starts to read both the trajectory- and radiosonde data (GSONDE). In the ANALYS-subroutine the starting locations of the different trajectories are needed to determine the weighing factors of the radiosoundings. At last ABM is called: this subroutine advects the initial profile from the source-area to the receptor point.
If LOSOND=.FALSE. the ANALYSIS-subroutine for the relevant run was called executed before, so the program looks for an archived profile file and reads it in TARCHIV.

3.1 FLOW DIAGRAM OF THE MAIN PROGRAM.



END.

3.2. DESCRIPTION and BASIC CONCEPTS of the MAIN-program.

The MAIN-program starts with the file-declarations: (lines 1210-1640) : (un)even numbers are used for output(input) devices. Also the library function BACKUPDATE is declared. This function adds a number of hours to a specific date/time variable .
In the declaration section of the mainprogram (lines 4000000-4008050) we meet the following important variables:

ISHIFT= the date/time variable in the form YYYYMMDDHH
IMT = timespan of the ABM-run in hours (12,24,36)
IDEP = departure time of the trajectories and the ABM-calculations.
ZPDB = expected arrival pressure at the endpoint of the groundtrajectory (condensed form for Pressure at De Bilt) (see also ref 2 p 410: Treatment of Surface Pressure).

Further the program expects for every two hours of calculation data on cloudcover or seawater-temperatures.

ZCLOUD= array of total cloud cover along the trajectories in octa´s. These data are not used when the groundtrajectory is above sea.
ZWLIST= array of seawatertemperatures along the groundtrajectory. "99" means above land.

LOSOND= logical to control program-flow.
LOSOND=.TRUE. -read the sonde-data of stations in the surroundings of the starting points of the trajectories, construct an initial profile and do the ABM-calculations.
LOSOND=.FALSE. -read the initial profile from disk and do the ABM-calculations.

FORETRAJ/IMT/ISHIFT=name of the trajectory-file used in this program.

ZTRLNG,ZTRLAT,ZTRP,= ZTRVX,ZTRVY,ZTRVW (0:JPLEN4,JPLEN2) arrays for storing the trajectory information:
The first index (JJ) is the timeblock-indication:
JJ=0 00 hrs relative (=model) time
JJ=1 02 hrs relative time.
The second index (II) indicates the different presssure levels for the trajectories:
II=0 groundtrajectory
II=1 1000 millibar trajectory
II=2 925 millibar trajectory, etc.
In ZTRLNG(JJ,II) and ZTRLAT(JJ,II) the geographical length and latitude of the II-trajectory is stored. ZTRP contains the pressure, ZTRVX,ZTRVY,ZTRVW give the large-scale velocities (VX & VY in meters/second,

	VW in millibar/minute).
ZTRAJ(JPLEN2,0:JPLEN1)	array for storing the 'static' trajectory information : (II refers to level) ZTRAJ(II,0)-end pressure level ZTRAJ(II,1)-starting ,, ,, ZTRAJ(II,2)- ,, longitude ZTRAJ(II,3)- ,, latitude.
ZTEMPS(JPLEN3,JPLEN1)	array for holding the radiosonde information: (JI refers to stationnumber) ZTEMPS(JI,1)- identification number: 06260= De Bilt ZTEMPS(JI,2)- logitude of station JI ZTEMPS(JI,3)- latitude of station JI
ZCHAR(JPLEN1,0:JPLEN5, JPLEN1)	ZCHAR(I,J,K) contains the J-th characteristic level of observation K from radiosonde station I. J=0 : surface level. K=1 : pressure, K=2 : potential temperature, K=3 : specific humidity.
IDEV2,IDEV4,etc	decive number variables. By using these variables it is easy to redirect output/input to or from other devices.

The execution order of the MAIN is as follows:
- -the programs first assigns the IDEV-variables
- -the datacards are read ,and the information is written on the printer
- -the trajectory filename is constructed (line 4518000-4519000)
- -an investigation is carried out in QUEST to see whether the specified trajectory-file exist on device IDEV9
- -the trajectory-file is read and for each traj-level II the ZTRAJ(II,*) is filled
- -to get the analysed profile either the ANALYS-subroutine is done or the program reads in this profile from disk in TARCHIV
- -then the ABM routine is called
- -the program jumps back and starts to read new datacards until all runs have been executed.

3.3 DESCRIPTION OF THE SUBROUTINES FOR THE MAIN-PROGRAM.

```
SUBROUTINE CHNAME(KDEV,YASTR)
CHARACTER YASTR*(*)
INTEGER KDEV
```

- lines 80000-88000
- links a device-number with a filename by the CHANGE-statement. CHNAME= mnemonic of ´change name´.

```
SUBROUTINE QUEST(KDEVD,KDEV6,YANAME,LAPRES)
INTEGER KDEVD,KDEV6
LOGICAL LAPRES
CHARACTER YANAME*(*)
```

- lines 100000-112000
- first invokes CHNAME and then inquires whether the file with the name YANAME exist on the device KDEVD.

```
 SUBROUTINE GSONDE(KDATE,KMT,PTEMPS,PCHAR,KNRO,KDEV6,KDEV9,
*                 KLEN1,KLEN3,KLEN5,LAPRES)
 INTEGER KDATE,KMT,KNRO,KDEV6,KDEV9,KLEN1,KLEN3,KLEN5
 REAL PTEMPS(KLEN3,KLEN1),PCHAR(KLEN3,0:KLEN5,KLEN3)
 LOGICAL LAPRES
```

- lines 300000-323000
- this subroutine investigates first whether the file ´SONDE/´ & KMT & KDATE & ´ ON TEST´ exists. If the file is present, a date-check is performed. If this test is successfull, the radio-sonde data are read in in a formatted form. For each station the PTEMPS and the PCHAR have to be known. If the number of characteristic levels is less then (KLEN5+1) the PCHAR-array is filled with default value´s. In the file the last level of any station is followed by a series of three zero´s.

```
 SUBROUTINE TARCHV(KDATE,PSFCDT,PSGNPT,KMAX,KMT,KDEV9,KDEV6,
*                KLEN1,KLEN3,KLEN5,LAPRES)
 INTEGER KDATE,KMAX,KMT,KDEV9,KDEV6,KLEN1,KLEN3,KLEN5
 REAL PSFCDT(KLEN1),PSGNPT(0:KLEN3*KLEN5,KLEN1)
 LOGICAL LAPRES
```

-(<< read temp in archives, lines 1500000-1519000) the initial profile that was constructed before is read in if the file is present on device number KDEV4.

4.THE ANALYSIS SUBROUTINE.

```
      SUBROUTINE ANALYS(KNRO,KSHIFT,PTRAJ,PTEMPS,PCHAR,PSFCDT,PSGNPT,
     *                  KMT,KDEV4,KDEV6,KDEV8,KLEN1,KLEN2,KLEN3,
     *                  KLEN5,KMAX)
      IMPLICIT            LOGICAL(L),CHARACTER*8(Y)
      INTEGER KSHIFT,KNRO,KDEV4,KDEV6,KDEV8
      INTEGER  KMT,KLEN1,KLEN2,KLEN3,KLEN5,KMAX
      REAL PSFCDT(KLEN1),PSGNPT(0:KLEN3*KLEN5,KLEN1),
     *PTRAJ(KLEN2,0:KLEN1),
C                 (JI,0): END LEVEL OF      TRAJECTORY JI
C                 (JI,1): STARTING LEVEL        ,,      ,,
C                 (JI,2): STARTING LONGITUDE    ,,      ,,
C                 (JI,3): STARTING LATITUDE     ,,      ,,
     *PTEMPS(KLEN3,KLEN1),
C                 (JI,1): IDENTIFICATION OF PTEMPS JI
C                 (JI,2): LONGITUDE IN DEGREES EAST
C                 (JI,3): LATITUDE IN DEGREES NORTH
C                 PTEMPS(KLEN3,KLEN1)
     *PCHAR(KLEN3,0:KLEN5,KLEN1)
C                 (I,J,K): CHARACTERISTIC LEVEL J(LEQ 30) OF
C                 PTEMPS I CONTAINING OBSERVATION K(LEQ  3)
C                 J=0 FOR SURFACE LEVEL
C                 PCHAR(KLEN3,0:KLEN5,1:K) WITH K=NR OF RELEVANT
C                 OBSERVATIONS
```

-(analysis, lines 1432008-1450000)

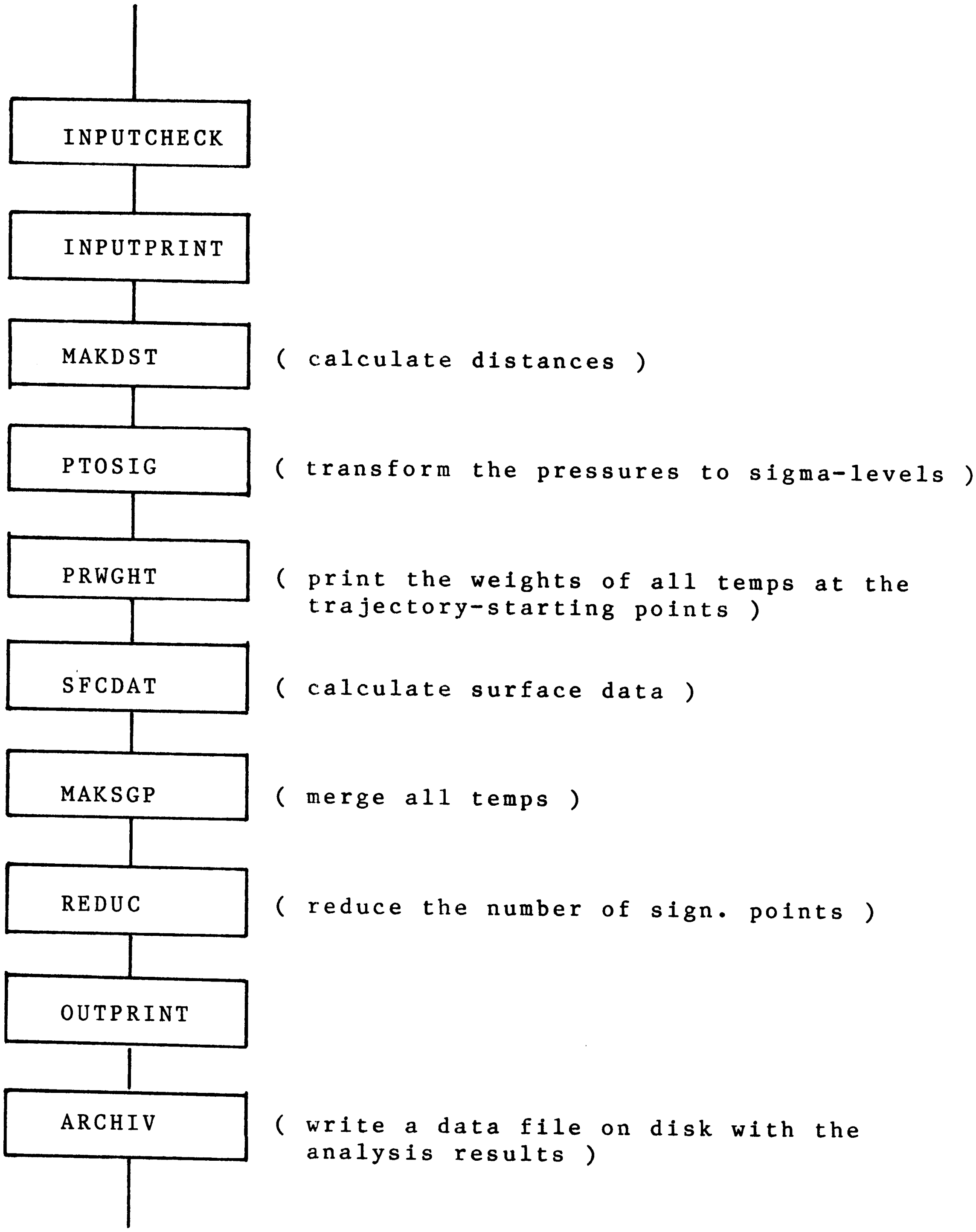
This subroutine (first MAIN-subroutine) calls all other subroutines that are needed to construct an initial profile for the ABM-calculations.

The central subroutine is MAKSGP (<< make significant points). This subroutine merges all pressure levels of the selected radiosoundings (so called characteristic points) into one new array called PSGNPT (<< significant points). Values for potential temperature and specific humidity are obtained by a weighed interpolation from the radiosoundings. The number of "raw" significant points is reduced (in REDUC) if two points coincide nearly or if one significant points deviates too little from an interpolation line drawn through its neighbouring points.

Some printing is done to follow the analysis:

-INPUTPRINT to print the radiosounding-data
-PRWGHT   to print a table of distances and weights
-OUTPRT   to print out the resulting initial profile
            for the ABM-calculations.

4.1. FLOW-DIAGRAM OF THE ANALYS-SUBROUTINE.



4.2 DESCRIPTION OF SUBROUTINES AND FUNCTIONS FOR THE ANALYSIS.

```
 SUBROUTINE CHECK(KDATE,KNRO,PTEMPS,PCHAR,KDEV8,KLEN1,KLEN2,
*                    KLEN3,KLEN5)
 INTEGER KNRO,KDATE,KDEV8,KLEN1,KLEN2,KLEN3,KLEN5
 REAL PTEMPS(KLEN3,KLEN1),PCHAR(KLEN3,0:KLEN5,KLEN1)
```

-this subroutine checks whether the radiosonde-data were given correct order , i.e.: first the groundlevel observations and then the lower pressure data. If a wrong pressure level is detected , the entire characteristic level is thrown away and the following levels are shifted "upwardsly".

```
 REAL FUNCTION ZLNEAR(PX,PA,PFA,PB,PFB)
 REAL PX,PA,PB,PFA,PFB
```

-The value attributed to ZLNEAR (<<linear) will be the linearly interpolated or extrapolated value of some function f from its function value PFA in PA and PBA in FB in the interpolation point PX ; if PFA or PFB equals 99999 then ZLNEAR will return the value 99999.

```
 SUBROUTINE INPRT(KDATE,PTRAJ,KNRO,PTEMPS,PCHAR,KDEV6,KLEN1,
*                          KLEN2,KLEN3,KLEN5,KMT)
 INTEGER KNRO,KDATE,KDEV6,KLEN1,KLEN2,KLEN3,KLEN5,KMT
 REAL PTRAJ(KLEN2,0:KLEN1),PTEMPS(KLEN3,KLEN1),
*         PCHAR(KLEN3,0:KLEN5,KLEN1)
```

-this subroutine prints out all data regarding the radio-sondes and the trajectories.

```
 REAL FUNCTION  DISTNC(PLON1,PLAT1,PLON2,PLAT2)
 REAL PLON1,PLON2,PLAT1,PLAT2
```

-The distance between two points with specified LON-(gitudes) and LAT(titudes) is calculated by using the sherical triangle formula.

```
 SUBROUTINE MAKDST(PTRAJ,KNRO,PTEMPS,PDSTNC,PWGTS,KLEN1,KLEN2,
*                                      KLEN3)
 REAL PTRAJ(KLEN2,0:KLEN1),
*     PTEMPS(KLEN3,KLEN1),
*     PDSTNC(KLEN2,KLEN3),
*     PWGTS(KLEN2,KLEN3)
```

-Distances between trajectory starting points and the temps are calculated. The weight of a specific radio-sonde to a specific level (in millibar) is inversely quadratic with the distance to the starting point of the trajectory on the specific pressure level (<<MAKE DISTANCES).

```
 SUBROUTINE PTOSIG(PTRAJ,KNRO,PCHAR,PWGTS,KLEN1,KLEN2,KLEN3,KLEN5)
 INTEGER KNRO,KLEN1,KLEN2,KLEN3,KLEN5
 REAL PTRAJ(1:KLEN2,0:KLEN1),PCHAR(KLEN3,0:KLEN5,KLEN1),
*                                  PWGTS(KLEN2,KLEN3)
```

-All pressure levels of the different stations are expressed in sigma*1000 form. A starting pressure (ZSP) for the lowest trajectory starting point is obtained by a weighed interpolation of the surface levels. All trajectory starting pressures are normalised by this pressure.(<< PRESSURE to SIGMA)

```
SUBROUTINE WEGHTS(PZPRES,PWHTS,PTRAJ,KNRO,PWGTS,KLEN1,KLEN2,KLEN3)
INTEGER KNRO,KLEN1,KLEN2,KLEN3
REAL PWHTS(KLEN3),PTRAJ(KLEN2,0:KLEN1),PWGTS(KLEN2,KLEN3)
REAL PZPRES
```

-In MAKDST distance-weighing factors for the different characteristic levels were calculated. If a pressure level is between other char. levels of a radiosonde , the station-weighing factor can be obtained by interpolation. The subroutine first looks for the index LOW-INDEX & HIGHINDEX with PTRAJ(HIGHINDEX,1) <= PZPRESS and PTRAJ(LOWINDEX,1) >= PZPRESS.

```
 SUBROUTINE PRWGHT(PTRAJ,KNRO,PTEMPS,PDSTNC,PWGTS,KDEV6,KLEN1,
*                      KLEN2,KLEN3)
 INTEGER KNRO,KDEV6,KLEN1,KLEN2,KLEN3
 REAL PTRAJ(KLEN2,0:KLEN1),PTEMPS(KLEN3,KLEN1),PDSTNC(KLEN2,KLEN3),
*          PWGTS(KLEN2,KLEN3)
```

-This subroutine prints a table of distances (between trajectory starting points and radiosonde locations) and prints a list of weighing-factors at standard levels.(<< PRINT WEIGHTS)

```
INTEGER FUNCTION MAXMUM(PH,K,KIN)
REAL PH(*)
INTEGER K,KIN
```

-MAXMUM will become equal to the lowest index (K <= INDEX <= KIN) for which holds PH(INDEX) >= PH(J) for all J with K <= J <= KIN.

```
 SUBROUTINE SFCDAT(PTRAJ,KNRO,PCHAR,PWGTS,KL,PSFCDT,KLEN1,KLEN2,
*                          KLEN3,KLEN5)
 INTEGER KL,KNRO,KLEN1,KLEN2,KLEN3,KLEN5
 REAL PCHAR(KLEN3,0:KLEN5,KLEN1),PTRAJ(KLEN2,0:KLEN1),
*          PWGTS(KLEN2,KLEN3),PSFCDT(KL)
```

-the array PSFCDAT(<< surfacedata) is filled with the surface values for pressure ,pot. temperature and specific humidity by interpolation from ¬ radiosonde data.

```
 REAL FUNCTION ZINTER(KL,KI,PCHAR,PSIGMA,KINR,KNRO,KLEN1,
*                                         KLEN3,KLEN5)
 INTEGER KL,KI,KINR,KNRO,KLEN1,KLEN3,KLEN5
 REAL PSIGMA
 REAL PCHAR(KLEN3,0:KLEN5,KLEN1)
```

- -KI =radiosonde station index: don't confuse this number with the PTEMPS(KI,1)-number PTEMPS(KI,1)= 06260(=NL,De Bilt) / 06447(=BELGIUM), etc . (1 <= KI <= 10)
- -KL =observable: KL=2 potential temperature
  KL=3 specific humidity

-the function ZINTER gets the interpolated value for the observable KL from observations made at station KI. The KINR-integer variable is used to indicate the

highest characteristic value below PSIGMA, so normally
PCHAR(KI,KINR+1,1) < PSIGMA < PCHAR(KI,KINR,1).
One special case has to be met: if KINR=0 the array-element PCHAR(KI,0,1) was untransformed by PTOSIG : by definition this value is 1000.

```
 SUBROUTINE MAKSGP(PTRAJ,KNRO,PWGTS,KL,PCHAR,PSGNPT,KLEN1,KLEN2,
*                 KLEN3,KLEN5,KMAX)
 INTEGER KNRO,KL,KLEN1,KLEN2,KLEN3,KLEN5,KMAX
 REAL PTRAJ(KLEN2,0:KLEN1),PWGTS(KLEN2,KLEN3),PSGNPT(0:KLEN3*KLEN5,
*           KLEN1),PCHAR(KLEN3,0:KLEN5,KLEN1)
```

-lines 1260000-1270000

-in this subroutine the sigma-coordinates of all observed characteristic points are merged in decreasing pressure order in a new array of "raw" significant points.The corresponding value for the pot.temperature and the specific humidity are computed by weighed interpolation. The first significant point is at sigma 1000 level. Then there is a loop-construction(IMAX-times) where the ZHN-array contains the higher (in kilometer from ground) sigma-levels of all radio-sonde stations that so far have <u>not</u> been used as a significant level for the PSGNPT-array. If an array-element of ZHN is 0 (which means that all levels of a station have been used) , no further interpolation between characteristic levels can be done anymore, and the subroutines stops (LOGOON-[logical GO-ON] =.FALSE.). The MAXMUM-function is invoked to determine the highest sigma-level in the array ZHN. For this level the value for THETA and Q are determined by interpolation ,ICOUNT giving the level in the target-array PSGNPT. So if one radiosonde station has only a few characteristic levels, the lowest pressure level determines the height to which the initial profile will extend.

Suppose we have the following situation, where only the sigma levels (PCHAR(I,J,1)) are given in this table:

STATION	I	II	III	PSGNPT
J=0	1012	1005	1000	1000
1	980	990	965	990
2	700	980	900	980
3	500	970	850	970
4	300	960	600	960

WARNING: always check the radio-sonde data before running the AMT-model: a bad or corrupted temp will give an unpredictable result for the ABM-calculations.
(MAKSGP << make siginificant points)

```
SUBROUTINE REDUC(PSGNPT,KNRO,PTOL,PLOBS,KMAX,KLEN1,KLEN3,KLEN5)
INTEGER KNRO,PLOBS,KMAX,KLEN1,KLEN3,KLEN5
REAL PSGNPT(0:KLEN3*KLEN5,KLEN1),PTOL(KLEN1)
*                       KLEN3,KLEN5,KMT)
```

-(lines 1270000-1361000)

The reduction of the number of significant points takes two phases: in the first reduction coincidence of sign. points is assumed if [D= delta]

```
    D(pressure)       is less than PTOL(1)= 5       millibar
AND D(pot.temp.)      is less than PTOL(2)= 0.3     ´ C
AND D(sp humidity)    is less than PTOL(3)= 2 E-4   gr/Kg.
```

If there is such a level, this level is kicked out of PSGNPT and the number of significant points (KMAX) is reduced by one.

Phase two look for those sign. points which differ more than PTOL (for THETA & Q) with respect to the value obtained by linear interpolation from their neighbouring points. This is done iteratively by reducing the interval-range until no outstanding values are found in the interval. Because there is an (implicit) relation between THETA and Q, these reductions are intermixed to save computer time. Every time an outrange-value is found, the original interval is split up into two parts and the program goes one level deeper (KLEVEL= level, ISPLIT= right-hand side index of the new interval). The routine toggles forth and back through each interval until the uppermost level (KLEVEL=0) is also treated completely.

After the temperature and humidity check all significant points (LOTAKE=.TRUE.) are put into the lowest PSGNPT-array index positions. The last level (highest level above ground in kilometers) is marked by a zero-pressure.

(REDUC << reduce the number of significant points)

```
SUBROUTINE OUTPRT(KDATE,PSFCDT,PSGNPT,LACRUD,KAMX,KDEV6,KLEN1,
*                  KLEN3,KLEN5,KMT)
LOGICAL LACRUD
INTEGER KDATE,KMAX,KDEV6,KLEN1,KLEN3,KLEN5
REAL PSFCDT(KLEN1),PSGNPT(0:KLEN3*KLEN5,KLEN1)
```

-(output print subroutine lines 138600-1412000).

In this subroutine all significant points and the ground level observables are printed out.

```
SUBROUTINE ARCHIV(KDATE,PSFCDT,PSGNPT,KMAX,KDEV4,KLEN1,KLEN3,
*                     KLEN5,KMT)
INTEGER KDATE,KMAX,KDEV4,KLEN1,KLEN3,KLEN5,KMT
REAL PSFCDT(KLEN1),PSGNPT(0:KLEN3*KLEN5,KLEN1)
```

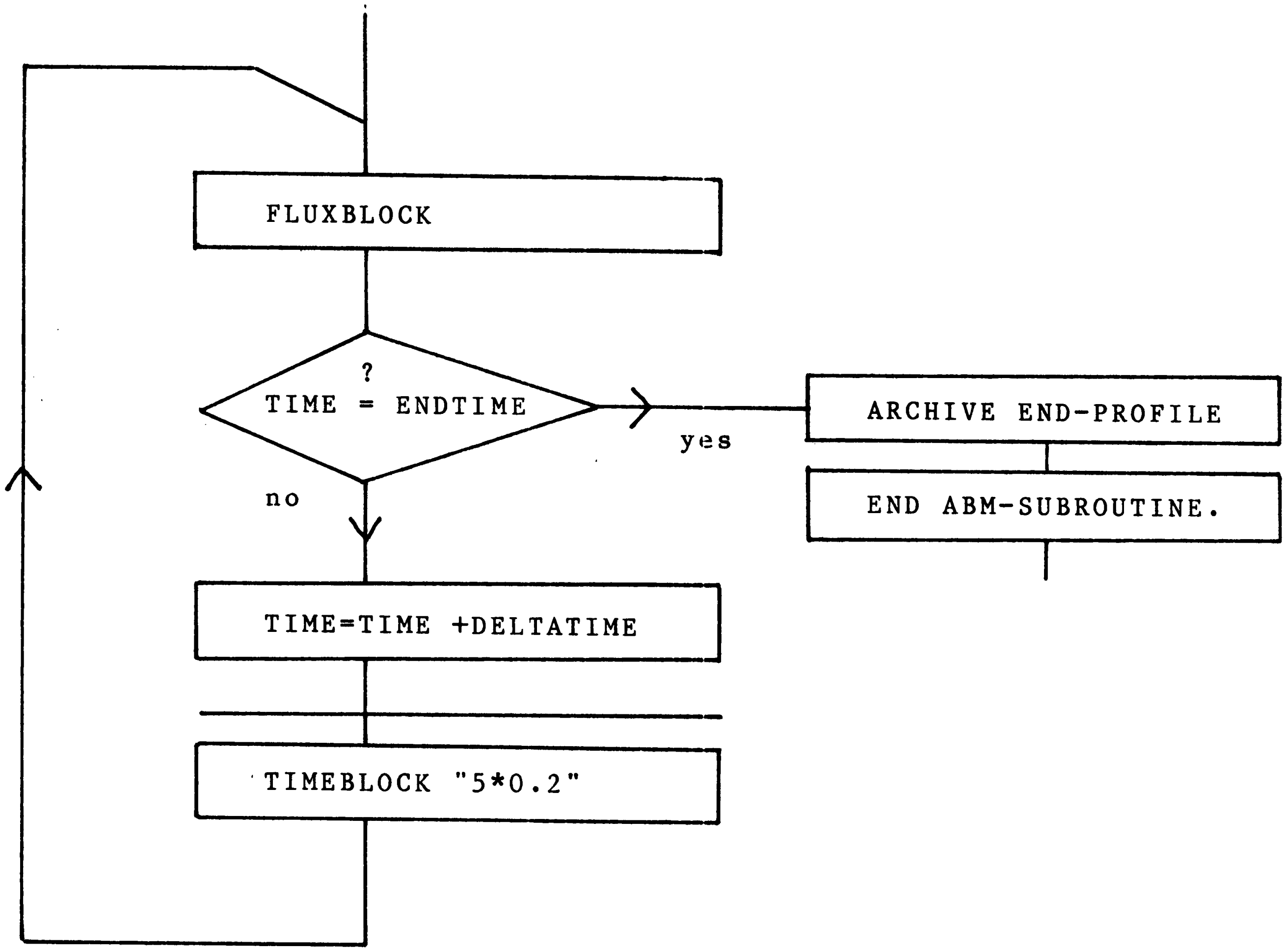
-this routine write a file ´FTTEMP/´ & KMT & MONTH & SHIFT & ´ ON TEMP.´ with all significant points and surface data on DISK in unformatted form.

5. THE ATMOSPHERIC BOUDARY LAYER MODEL.

```
      SUBROUTINE ABM(PSFCDT,PSGNPT,PTRLNG,PTRLAT,PTRP,
     *    PTRVX,PTRVY,PTRVW,KSHIFT,PPDB,KMT,PCLOUD,PWLIST,KDEV2,
     *    KDEV10,KLEN1,KLEN2,KLEN3,KLEN4,KLEN5,KMAX)
      IMPLICIT            LOGICAL(L),CHARACTER*8(Y)
      INTEGER KSHIFT,KMT,KDEV2,KDEV10,KLEN1,KLEN2,KLEN3,KLEN4,KLEN5,
     *        KMAX
      REAL PPDB
      REAL PSFCDT(KLEN1),PCLOUD(0:KLEN4),PSGNPT(0:KLEN3*KLEN5,KLEN1),
     *  PTRLNG(0:KLEN4,KLEN2),PTRLAT(0:KLEN4,KLEN2),PTRP(0:KLEN4,KLEN2),
     *     PTRVX(0:KLEN4,KLEN2),PTRVY(0:KLEN4,KLEN2),
     *     PTRVW(0:KLEN4,KLEN2),PWLIST(0:KLEN4)
      -(lines 3000000-3100000)
```

The first part of the ABM-subroutine is initialization, then the analysed profile is copied from the two-dimensional array PSGNPT into the three one-dimensional arrays ZHP (pressure),ZHTETA (potential temperature) and ZHQ (specific humidity). Also the mean ABL-potential temperature and specific humidity is computed (ZHTTAM and ZHQM).
At night we assume that the ABL is stable in the first timestep; this assumption can save one iteration in the "Flux block".
The subroutine ABM has two major blocks:
-the Flux block and
-the Timeblock "5*0.2".
At time HHMM these fluxes are computed, then the time changes to HHMM+10 minutes (=+deltatime), and the new profile for THETA en Q is computed. The program goes further with the calculation of the new fluxes (forward time difference scheme).

5.1. FLOW DIAGRAM ABM-SUBROUTINE



5.2. NUMBERING THE SIGNIFICANT POINTS

-the following index conventions must be adhered to in this program

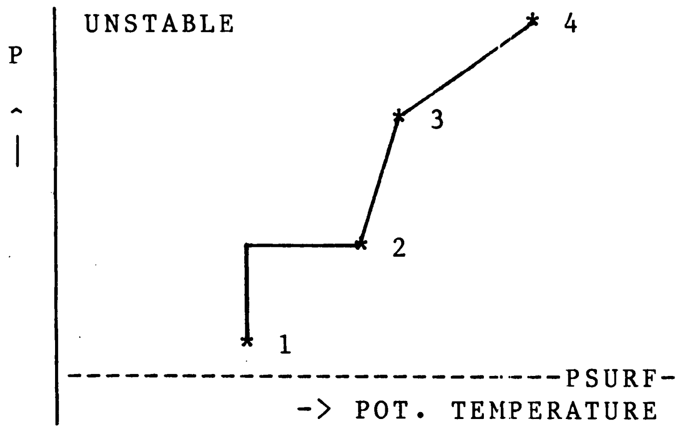


fig 1: UNSTABLE model profile

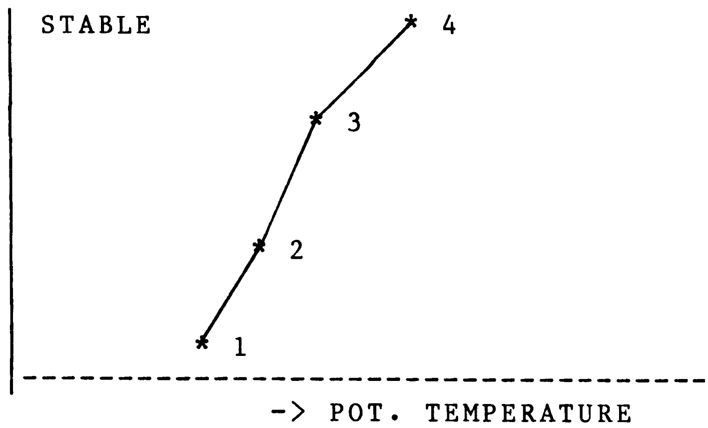


fig 2: STABLE modelprofile

-the surface pressure (ZPSURF) in general will not be equal to the pressure at the lowest significant point (ZHP(1)) ; see also ref. 2 appendix A). Note that there is a "hidden" significant point in the unstable case (between the numbered points 1 and 2).

5.3. THE FLUXBLOCK

-(lines 3015000-3031200)

The calculation order in the FLUXBLOCK is as follows:

- the Coriolis-parameter(ZHF)
- the surface pressure(ZPSURF)
  this value is obtained by linear interpolation between the pressure at the lowest trajectory starting location (PSFCDT(1)) and the expected/observed pressure at the endpoint of this trajectory(PZPDB).
- the absolute temperature at the surface(ZHTABS)
- the daynumber(ZNRDAY),the hour & minute indicators
- the sine of the sun´s height(ZSINZ)
- the netto radiation
- the dragcoefficient
  this program has a stability dependent ZCDRAG if the model is above land and a fixed value for oversea situations. The correctionfactors (2.0/1.50 for land,1.25 for sea) are tuned-up values that give optimum results for the ECMWF-trajectories. The reason for these factor is that the flux-parametrisations were based on a Vg (geostrophic wind) and the lowest horizontal ECMWF-wind is influenced by the surface in the ECMWF-model.
  This rather crude method of computing the friction velocity will be replaced in future by more sophisticated methods. If a stability-change takes place the lines 3017400-3023800 will be done twice.
- the fluxes over water(FLXWAT) or over land(FLXLA)
  All fluxes in the entire program are given is WATT*(M-2).

```
ZTETAF  = "H " = sensible heat
ZQFLUX  = "LE" = latent heat flux
ZVIRFX         = virtual heat flux
                 ZVIRFX= "H" + 0.61* Cp * Tabs * "LE" /L
```

- the mean ABL-temperature(ZHTM)
- the ABL-height(ZHABL)
- the friction velocity(PUSTER)
- the Obukhov-length and the ZRATIO
- the new stability which depends only of the sign of the virtual flux: ZVIRFX < 0 gives INWSTB=1 (stable)
  ZVIRFX > 0 gives INWSTB=2 (unstable)
- the limit value of the stable boundary layer height

Now if the end-time is reached, the program prints out the the fluxes for this endtime : note that nothing is "done" with these fluxes.

At the start with a stable ABM a new point is created at the stable limit ABL-height. This is done because we need a guess for the ABL-height. The actual height was determined by the history of the air mass involved. This history beeing unknown, we assume that the actual height is of the same order as the limit value. If the humidity in the second point is larger than ZHQ(1), the profile is smoothed (i.e. : the discontinuity is set to zero, while conserving the total humidity content.

In case of stability change, the CHANGE-subroutine (and for completeness the STAP-subroutine) is called,. The fluxblock ends with some printing.

5.4. THE TIMEBLOCK "5*0.2".

-(lines 3035295-3056500)
Before entering the timeblock the following actions are taken:

- the time is incremented with 10 minutes. This time should be regarded as the time when the timeblock has finished all calculations.
- the minute-indicator IMINU is recomputed
- every hour the "shift" is updated
- the fractional time between trajectory timesteps is computed
- the velocities and geographical positions of all significant points are computed by the INTER-subroutine
- the ZWTIME/ZCLAND time is kept
- and the timestep is changed when a transition stable-unstable takes place.

In the timeblock the calculation order is as follows:

- first the entrainment velocity is computed (ENTRN1/2)
- in the stable case an extra point is then created if the new ABL height decreases after a period of growth and the slope between the second and third significant point is very flat (point 2´ in fig. 4). Remember: THETA(2) follows the slope of the T-profile between the points 2 & 3. Now if no extra point is created, the area under point 2 is filled up, which is highly unrealistic (see fig.3).

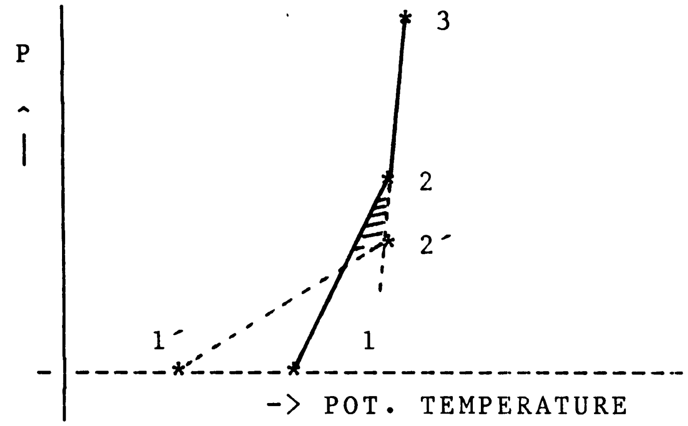


fig 3: the shaded area is filled up: unrealistic.

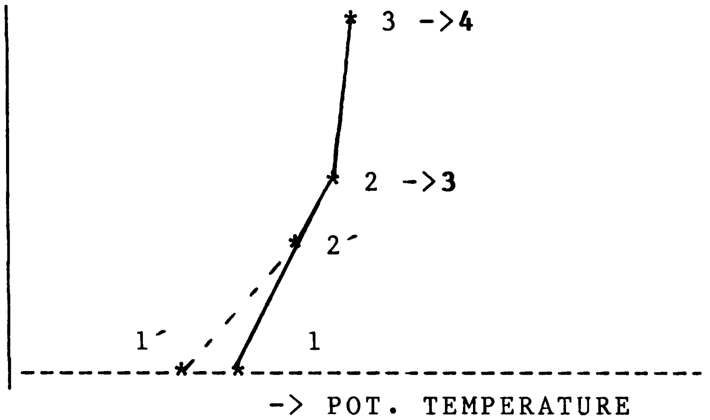


fig 4: an extra point was created below the old second point.

- the new pressure for the second point is computed (ZH2P(2)). If this points lies above the third point the timestep is changed to TIME2.
- if case of a large inversion (remember: point 3 & 4 were in the unstable case the two points at the discontinuity!), the ABL can not surpass this barrier and the entrainment velocity becomes zero. Case 2 handles all actions when the ABL meets the third significant point.
- then we arrive at the central equations for the development of the ABL. The new mean temperature is determined by the old heat content of the ABL, the entrainment influence, the fluxterm and the new height of the ABL. The physical units in line 3044800 for the denominator are [Celsius] * [millibar], hence a factor of 0.6*g/Cp appears

to convert the fluxterm ZTETAF to the correct units. The handling of the humidity is exactly similar. The mean ABL-temperature does not decrease if the relative humidity is over 100%.

- the lowest significant point temperature is computed
  - the mean ABL-temperature ZH2TTM should not exceed ZH2TTA(1), so if this occurs, the heat content is redistributed (the discontinuity at the top of the unstable boundary layer disappears). The same action is taken if the mean ABL-humidity is lower than ZH2Q(2).
  - in the stable cases a linear profile for the temperature and the humidity is assumed (note: the humidity is later in the program (lines 3050200-3050900) mixed over the ABL.
- the third significant point is annihilated if the second point ran into it
- the old pressure values are copied into the array for the new pressures, with the important exception of the second point
- the humidity is mixed over the stable ABL; the IMIXER-variable gives the history of the program.Normally IMIXER=3 except in the case that was handled in lines 3037300-3039500.

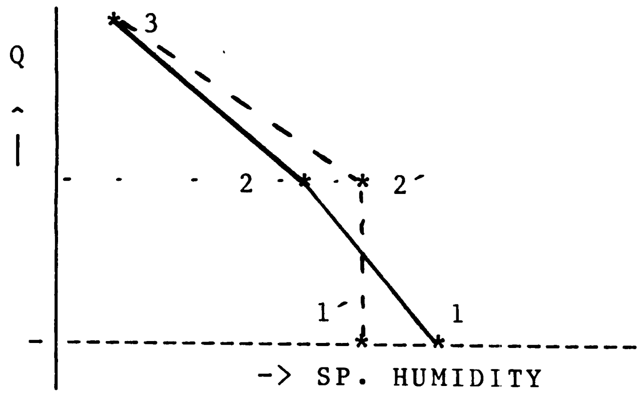


fig 4a: humidity mixing over the stable ABL.
Solid line : old profile.
Dashed line : new profile.

- ZDTETA & ZDTEDT are cumputed
- the large scale velocities are dealt with: if due to very wild l.s.-velocities a significant point comes lower than 300 millibar, this and the following levels are put aside.
- at last the new values for the physical variables are copied into the old arrays

If all five small timesteps or one 10 minute timestep is run through the timeblock ends , and every printmodus the profile is printed out. The program goes now further with computing the fluxes.
The ABM-subroutine ends with writing a file on disk with the final temperature/humidity profile.

5.5. VARIABLE TABLE FOR THE ABM-SUBROUTINE.

-for the description of the dummy arguments : see MAIN-program.

-INTEGER PARAMETERS:

JPIVET	length of the significant points arrays

-INTEGER VARABLES:

ITBLOK	trajectory timeblock
IMAX	number of sign. points
IHOUR	hour indication
IMINU	minute indication
INWSTB	new stability
IOLSTB	old stability
IMIXER	mix index for the humidity (see timeblock)
ILINE	line-index for the lineprinter
ISTART	relative starttime in minutes.
ITEND	relative endtime in minutes.
ITIME1	normal timestep.
ITIME2	small timestep
IPRINT	printmodus.

-REAL VARIABLES:

ZTIME	relative time in minutes.
ZTFRAK	fractional time between trajectory ITBLOK.
ZDTIME	actual timestep
ZRDOLD	old netto radiation
ZSINLD	old sun´s sine.
ZCRTYD	transitiontime storer.
ZHTTAM	mean potential temperature in the ABL.
ZH2TTM	new ZHTTAM
ZHQM	mean specific humidity in the ABL.
ZH2QM	new ZHQM
ZHTABS	absolute surface-temperature.
ZTETAF	sensible heat.
ZQFLUX	latent heat.
ZVIRFX	virtual flux.
ZVRFXN	virtual flux at night.
ZCL	cloudcover
ZMECH	mechanical turbulence in the unstable case.
ZBUOY	buoyance turbulence in the unstable case.
ZRATIO	# w*/u*
ZUSTER	friction velocity.
ZPSURF	surface pressure.
ZHPENT	entrainmentvelocity * timestep
ZHGEO	ABL-wind
ZHF	coriolis-parameter
ZVENTR	entrainment velocity.
ZCDRAG	dragcoefficient.
ZHTM	mean ABL-temperature.
ZVEOLD	old entrainment velocity.
ZCT	transition time storer.
ZDTETA	difference in pot.temp. between first and second significant point.
ZDTEDT	ZDTETA differentiated w.r.t. time.
ZOBUKHW	Obukhov-length.
ZWTIME	seawater time
ZCLAND	cloudcover * landtime.

for each significant	point the index I of
ZHLAT(1:JPIVET)	lattitude (east > 0)
ZHLNG(1:JPIVET)	longtitude
ZHVX(1:JPIVET)	west-east windspeed. (m/s)
ZHVY(1:JPIVET)	south-north windspeed. (m/s)
ZHVW(1:JPIVET)	vertical windspeed. (millibar/minute)
ZHP(1:JPIVET)	pressure level.
ZH2P(1:JPIVET)	new pressure level.
ZHTETA(1:JPIVET)	potential temperature.
ZH2TTA(1:JPIVET)	new potential temperature.
ZHQ(1:JPIVET)	specific humidity.
ZH2Q(1:JPIVET)	new specific humidity.

-LOGICAL VARIABLES:	if .TRUE. then
LOWATR	the model is above water.
LOQUIT	the program leaves Timeblock 5* 0.2
LOSTOP	the program leaves the " calculation of the new stability" loop .
LOFIRST	the first timestep is processed.
LOANIL	an annihilation of a sign.point must be executed.
LOCHNG	the stability changes.
LOSDT	the timestep is 0.2 * ITIME2.

5.6 DESCRIPTION OF PROCEDURES AND FUNCTIONS FOR THE ABM-CALCULATIONS.

```
SUBROUTINE ENTRN1(PVENTR,PLIMIT,PDHP,PDTETA,PDTEDT)
REAL PVENTR,PLIMIT,PDHP,PDTETA,PDTEDT
```

-for the stable cases (INWSTB=1) the entrainment velocity is computed from formula 14 in ref 2. The program text needs some explanation: the entrainment velocity is proportional to the height difference between the actual height and the limit value:

```
Oe # (hlim-h)=((PZSURF-PLIMIT)-(PZSURF-(PHP(1)+PHP(2))=
                          PHP(1)-PHP(2)- PLIMIT=
                               PDHP   -PLIMIT.
Oe                            -entrainment velocity
hlim  =PZSURF-PLIMIT          -limit value for the ABL-height in
                               the stable case.
h    =PHP(2)-PHP(1)+PZSURF  -actual ABL-height.
```

The PZSURF-PHP(1) term should be regarded as a correction term (see also Appendix A in ref 2)

We thus have Oe=(PDHP-PLIMIT)*(PDeeThEtaDeeTee)/(PDeltaThETA)=(PDHP-PLIMIT)*PDTEDT/PDTETA.

We insert a factor of 60 in the above formula because we need the entrainment velocity in millibar/minute.

```
 SUBROUTINE ENTRN2(PVENTR,PBUOY,PMECH,PUSTER,PRHO,PHTABS,
*                  PDHP,PDHQ,PDHTTA,PVIRFX)
 REAL  PVENTR,PBUOY,PMECH,PUSTER,PRHO,PHTABS,PDHP,PDHQ,
*      PDHTTA,PVIRFX
```

-for the unstable case (INWSTB=2) formula 10 from ref. 2 is used:

Oe= (c* $(\overline{\theta_v' \omega_0'})$-A*(USTAR**3)*g*TSURF*(RHO**2)/h)/(Delta THETA virtual)

where $(\overline{\theta_v' \omega_0'})$=(-g/Cp)*PVIRFX, with PVIRFX is the virtual flux in WATT*(M-2):

PVIRFX=PTETAF+0.61*Cp/L*(PQFLUX)

PTETAF=vertical surface flux density of sensible heat (H)

PQFLUX=vertical surface flux density of moisture.

Delta THETA vitual is the difference in virtual temperature of the lowest two model points.

Because h (actual ABL-height) is given in millibar at the calling moment, ´ 100´ appears at line 2084000

In S.I. the dimensions of the entrainment velocity would be PASCAL/SECOND : we use here millibar/minute, hence a factor 0.6 appaers in the formula. Futhermore we limit the entrainment velocity to -2 millibar/mimute.

REAL FUNCTION RHO(PTEMP,PZPRES)
REAL PZPRES,PTEMP
-this function computes the density RHO as a function of the temperature (PTEMP,in KELVIN) and the pressure (PZPRES,in N*(M-2)).

REAL FUNCTION ES(PTEMP)
REAL PTEMP
-this function calculates the saturation water vapour pressures as a function of the temperature(PTEMP). PTEMP in Celsius, ES in millibar.

REAL FUNCTION QS(PTEMP,PZPRES)
REAL PTEMP,PZPRES
-calculation of the saturation specific humidity as a function of the pressure (PZPRES) and the temperature (PTEMP). PTEMP in Celsius,PZPRES in millibar, QS in Kg/Kg.

REAL FUNCTION S(PTEMP)
REAL PTEMP
-calculation of the slope of the specific humidity curve (d ES/ d T). S in millibar /C. This formula is a direct differentation of the formula for ES(PTEMP).

REAL FUNCTION ZLV(PTEMP)
REAL PTEMP
-calculation of the latent heat (L) in first order approximation.

REAL FUNCTION GAMGS(PTEMP,PZPRES)
REAL PTEMP,PZPRES
-calculates the slope of the specific saturation humidity temperature curve (ZSQ=0.621/PZPRES*S(PTEMP) divided by GAMMA=Cp/L. This function is needed for the partition of H and LE in the fluxscheme above land (see formula B9 and B10 from ref. 2).

REAL FUNCTION TINVRT(PHTETA,PMBAR)
REAL PHTETA,PMBAR
-inverts the potential temperatur (PTHETA) to a real temperature.

SUBROUTINE DAYNR(KSHIFT,PNRDAY)
INTEGER KSHIFT
REAL PNRDAY
-calculates the daynumber at the calling moment. The daynumber (PNRDAY) is given in grads: 360 grads p 1 year).

REAL FUNCTION SINUSZ(PHLAT,PHLNG,PTIME,PDAYNR,PBOWEN)
REAL PHLAT,PHLNG,PTIME,PDAYNR,PBOWEN
-(lines 2176000-2196000)
calculates the sine of the sun's height as a function of the location on earth (PHLAT & PHLNG=lattitude and longitude) ,the time of the day (PTIME) and the day-number (PNRDAY). Also the correction factor for the Bowen ratio (PBOWEN, in ref. 2 formula B11 the symbol

B´) is given back as a function of the maximum sun´s height.

```
REAL FUNCTION RADNET(PSINZ,PCL,PTEMP,PDAYNR,LAWATR)
REAL PSINZ,PCL,PTEMP,PDAYNR
LOGICAL LAWATR
-the netto radiation is computed, the formulas following
 Appendix A.2 in ref.2 . ZKSTER is the incoming shortwave
 radiation.

SUBROUTINE FLXWAT(PHABS,PHWATR,PHQ,PHQS,PZSURF,PHGEO,PVIRFX,
*                 PVRFXN,PTETAF,PQFLUX)
REAL PHABS,PHWATR,PHQ,PHQS,PZSURF,PHGEO,PVIRFX,PVRFXN,
*                PTETAF,PQFLUX
-(lines 2234000-2252000)
 calculates the surface fluxes above water in WATT*(M-2).
  PTETAF= sensible heat.
  PQFLUX= latent heat.
  PVIRFX= virtual flux.
  PVRFXN= vitual flux during night.
  ZDVIR = the difference in virtual temperature between the
          lowest two boundary-layer points.

SUBROUTINE FLXLA(PBETA2,PCD,PCL,KCASE,PRNET,PHABS,PZSURF,
*                PHGEO,PTETAF,PQFLUX,PVIRFX,PVRFXN)
INTEGER KCASE
REAL PBETA2,PCD,PCL,PRNET,PHABS,PZSURF,PHGEO,
*    PTETAF,PQFLUX,PVIRFX,PVRFXN
-calculates the fluxes above land for three cases :
 ( formulas from ref. 2)
  night: ZFLUX1 =sensible heat:  according to B12 & B13.
         ZQFLX1 =latent   heat:  0.
  day  : ZFLUX2 =sensible heat   according to B10.
         ZQFLX2 =latent   heat   according to B09.
 During the transitionperiod (after sunrise & sunset)
 the maximum of both parametrizations is taken.


C          END OF FLUX CALCULATIONS


SUBROUTINE  CHANGE(PHA,PHB,PHC,PHMB,PHMC,PHTS,KMAX,KNWSTB,
*                  PDTETA,PDTEDT,KMIXER,PZSURF)
INTEGER KMAX,KNWSTB,KMIXER
REAL PHMB,PHMC,PHTS,PDTETA,PDTEDT,PZSURF
REAL PHA(*),PHB(*),PHC(*)
-(lines 2298000-2378200)
 This subroutine does the administration if there is a
 transition in the stability from unstable to stable
 (lines 2308000-2342000) or vice versa (lines 2344000-
 2377000).
 PHA  - array of pressure          -values of the significant points.
 PHB  - array of pot.temperature -values of the significant points.
 PHC  - array of specific humidity    -values of the significant poin

 UNSTABLE to STABLE.
 -in this case two points are created: one point  at 3
  millibar above  the first point (at height # 25
```

meters,this point is the starting height for the ABL) and another point at the discontinuity of the unstable profile, which is needed to incorporate the heat/humidity content around the hidden significant point . The new mean temperature (PHMB) and specific humidity (PHMC) is computed ,together with the new surface temperature (PHTS) and PDTETA and PDTEDT (cooling values for the nocturnal boundary layer in formula 14 of ref. 2). (x/y = old/**new** significant point in sketch, sketch not on scale).

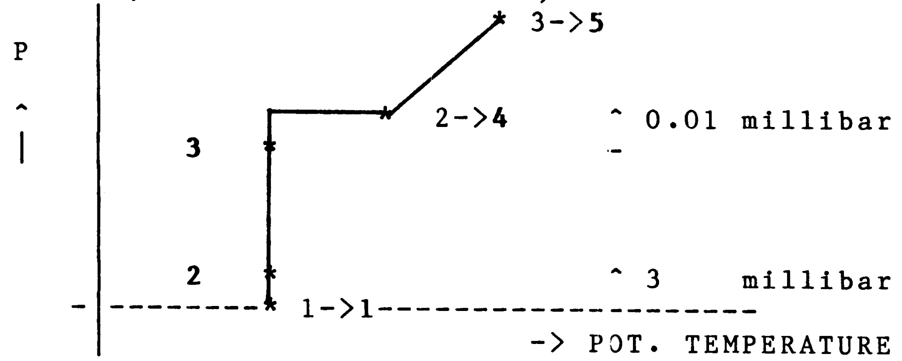


fig 5.

STABLE to UNSTABLE.

-normally PHB(1) is smaller than PHB(2) , and then one point is created at 3 millibar above the surface. The new values for this second point are obtained by interpolation.

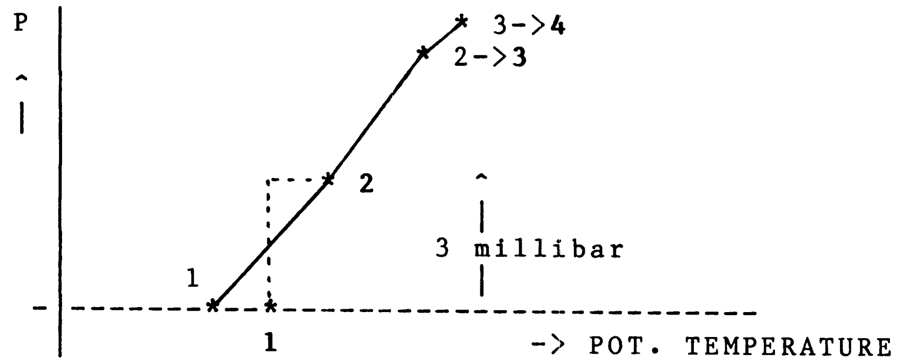


fig 6.

-sometimes a superadiabatic profile is encountered when the model starts (warm summerday for example). This situation does not suit our model assumptions, because the model "knows" only the stable or the unstable case. Therefore we spread out the heat-content of the lower three points over the lowest two points, thus annilhilating one point . If the heat content of the superadiabatic piece is large, the mean-ABL temperature could become larger than PHB(2). In this cases (which will be very seldomly encountered) this schema is not energy conserving. The same applies to the mixing-ratio profile.

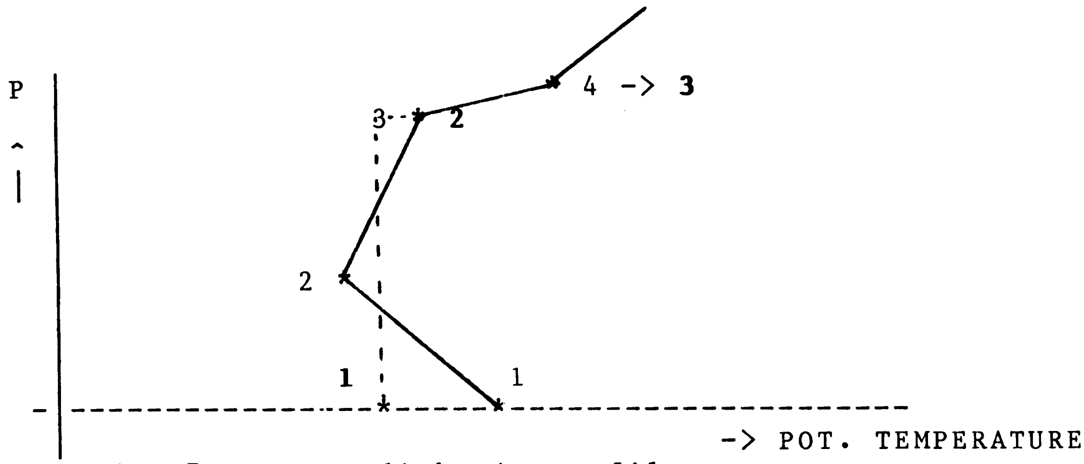


fig 7.: superadiabatic profile

```
SUBROUTINE OBUKHL(PUSTER,PVIRFX,PVRFXN,
*                  PHGRSL,PHTABS,PZSURF,POBUKHW,PRATIO)
REAL PUSTER,PVIRFX,PVRFXN,PHGRSL,PHTABS,PZSURF,POBKHW,
*        PRATIO
```

-this subroutine calculates the Obukhov-length. A limit value of 0.5 WATT*(M-2) for the virtual fluxes is taken to avoid computational problems. Note that in the transition period PVIRFX can be different from PVRFXN. Also the PRATIO is calculated (PRATIO # w*/u*).

```
SUBROUTINE HLIMIT(PUSTER,PZSURF,PHTABS,PHF,POBKHW,PLIMIT,KDEV6)
INTEGER KDEV6
REAL PUSTER,PZSURF,PHTABS,PHF,POBKHW,PLIMIT
```

- in this subroutine the endlimit for the stable boundary layer height is computed : PLIMIT in millibar. The formula is the interpolation formula 15 in ref. 2. If this equation is worked out to an expression for hlim, the neutral solution is not include in the equation, therefore this case in treated seperately. Line 2382000 converts this limit (meters) by the hydrostatic equation to millibars. **N.B.**: this pressure value is taken relative to the groundpressure PZSURF. So (PZSURF - PLIMIT) is the end value for the stable boundary layer height. An extra factor 0.5 appears in these formulas because we use in the model a linear temperature profile at night (see ref. 2 ,page 398,2.3).

```
SUBROUTINE BLOCK(PHELP,KBTIME,K,PHLAT,PHLNG,PHVX,PHVY,PHVW,PTRP,
*       PTRLAT,PTRLNG,PTRVX,PTRVY,PTRVW,KLEN2,KLEN4)
INTEGER KBTIME,K,KLEN2,KLEN4
REAL PHLAT,PHLNG,PHVX,PHVY,PHVW,PHELP
REAL PTRP(0:KLEN4,KLEN2),PTRVX(0:KLEN4,KLEN2),PTRVY(0:KLEN4,
*      KLEN2),PTRVW(0:KLEN4,KLEN2),PTRLNG(0:KLEN4,KLEN2),
*      PTRLAT(0:KLEN4,KLEN2)
```

-(lines 2390000-2400000)

This is just a help-routine for the interpolation in the subroutine STAP. Interpolation is done between the trajectory levels K and K+1 at a time KBTIME (Block time) for the latitude (PHLAT), the longitude (PHLNG), the horizontal wind speeds (PHVX and PHVY) and the vertical wind speed (PHVW).

NOTE that this is an interpolation in the vertical direction.

```
 SUBROUTINE STAP(PTRP,PTRVX,PTRVY,PTRVW,
*          PTRLNG,PTRLAT,KTBLOK,KMT,PTIME,KTEND,PTFRAK,PHP,
*          PHLAT,PHLNG,PHVX,PHVY,PHVW,KMAX,KLEN2,KLEN4)
 REAL PTFRAK,PTIME
 INTEGER KTEND,KTBLOK,KMT,KMAX,KLEN2,KLEN4
 REAL PTRP(0:KLEN4,KLEN2),PTRVX(0:KLEN4,KLEN2),PTRVY(0:KLEN4,
*     KLEN2),PTRVW(0:KLEN4,KLEN2),PTRLNG(0:KLEN4,KLEN2),
*     PTRLAT(0:KLEN4,KLEN2),PHP(*),PHLAT(*),
*     PHLNG(*),PHVX(*),PHVY(*),PHVW(*)
```

- this routine merely invokes the BLOCK-routine twice: once at time ITBLOK and once for time ITBLOK+1.(this is two hours later in this model version). Remember that the ABM-model keeps track of the modeltime by the- ITBLOCK-time in combination with the (block-)timefraction (PTFRAK).

  F.E.: Run starting at 2400 hrs. At 0200 hrs ITBLOK will become 1, and the PTFRAK will become zero, like it was at 1200 hrs.

  Lines 2480900-2486000 give the time interpolation between the datasets of ITBLOK and ITBLOK+1.

```
 SUBROUTINE INTER(PTRP,PTRVX,PTRVY,PTRVW,PTRLNG,PTRLAT,KTBLOK,
*    KMT,PTIME,KTEND,PTFRAK,PHP,PHLAT,PHLNG,PHVX,PHVY,PHVW,KMAX,
*    PHGEO,PWATER,PWLIST,PCLOUD,PCL,KLEN2,KLEN4)
 REAL PTRP(0:KLEN4,KLEN2),PTRVX(0:KLEN4,KLEN2),PTRVY(0:KLEN4,
*     KLEN2),PTRVW(0:KLEN4,KLEN2),PTRLNG(0:KLEN4,KLEN2),
*     PTRLAT(0:KLEN4,KLEN2),PHP(*),PHLAT(*),PHLNG(*),
*     PHVX(*),PHVY(*),PHVW(*),
*     PWLIST(0:KLEN4),PCLOUD(0:KLEN4)
 REAL PTFRAK,PCL,PHGEO,PWATER,PTIME
```

-(lines 2500000-2548000)

this routine takes care that all interpolation variables get their values:

- -for the trajectory-information interpolation STAP is called,
- -for the surface-level the "quasi"-geostrophic wind is calculated (see also the remarks in this Report concerning the socalled drag-coefficients).
- -the seawatertemperatures are interpolated form the seawaterlist (PWLIST) that was read in the MAIN-program. Note that seawatertemperatures were given every two hours and that the model has an overland situation if OR the PWLIST(KTBLOK)=99 or the PWLIST(KTBLOK+1)=99 !!!.
- -finally the cloudcover is interpolated in a transparent way.

```
 SUBROUTINE PAGE(KLINE,KDEV6)
 INTEGER KLINE,KDEV6
```

-takes care that the lineprinter skips to the next page after 60 lines were printed.

```
 SUBROUTINE FLXPRT(KSHIFT,KMINU,PVENTR,PTETAF,PQFLUX,PVIRFX,
*           LAWATR,PHTABS,PHGEO,PSINZ,PGRSL,PLIMIT,PWATER,
*           PZSURF,PCD,POBKHW,KNWSTB,PRATIO,
*           PHTM,PCL,PNTRAD,KDEV6,KLINE)
 REAL PVENTR,PTETAF,PQFLUX,PVIRFX,
*          PHTABS,PHGEO,PSINZ,PGRSL,PLIMIT,PWATER,
*          PZSURF,PCD,POBKHW,PRATIO,
*          PHTM,PCL,PNTRAD
 INTEGER KSHIFT,KMINU,KNWSTB,KLINE
 LOGICAL LAWATR
 -(lines 2600000-2640000)
  this procedure prints out the fluxes and some additional
  information:
  SHIFT       - actual forecast time in the form YYYYMMDDHH.
  MINU        - minutes  from SHIFT-'HH' time.
  VENTR       - entrainment velocity (millibar/minute)
  H           - sensible heat    |
  LE          - latent heat      |- WATT*(M-2)
  VFLX        - virtual flux     |
  BOWEN       - bowen ratio H/LE* 100 %.
  QRNET       - netto radiation above land (above water : 99)
  QSAT        - saturation  specific humidity for the surface level
                (*1000).
  HGEO        - quasi  geostrophic  wind for the calculation  of
                the surface fluxes.
  SINZ        - sine of the sun's height.
  HABL        - height of the boundary layer.
  HLIM        - limit  height of the stable boundary layer ( 999
                if unstable situation)
  TW          - seawater temperature ( 99 = land )
  N           - relative cloudiness ( 0 <= N <=1 )
  PSURF       - surface pressure in millibar.
  CDRAG       - dragcoefficient  (the  numerical  values   are
                tuned up for the ECMWF-trajectories ).
  OBUKHOWL    - Obukhov length in meters
  RATIO       - # w*/u* ( see OBUKHL-subroutine.)
  NEWSTAB     - new stability ( 1= stable, 2= unstable).

 SUBROUTINE PROFIL(PWTIME,KTEND,KSTART,PTIME,KMAX,PZSURF,
*                PHP,PHQ,PHTETA,PHLAT,PHLNG,PHVX,PHVY,PHVW,PCLAND,
*                LAFRST,KSHIFT,KMINU,KDEV6,KLINE,KMT)
 INTEGER KTEND,KSTART,KMAX,KSHIFT,KMINU,KLINE,KMT
 REAL PZSURF,PCLAND,PTIME,PWTIME
 LOGICAL LAFRST
 REAL  PHP(*),PHQ(*),PHTETA(*),PHLAT(*),PHLNG(*),
*                PHVX(*),PHVY(*),PHVW(*)
 -(lines 2641000-2667500)
  this  subroutine  prints  out the array  of  significant
  points . For each significant point we have :
  PRESSURE      - pressure in millibar (with correction  for
                  the possible difference PZSURF-PHP(1).
  TETA          - potential temperature.
  TEMP          - real temperature.
  SPEC  HUM     - specific   humidity  (mixing  ratio)
                  (Kg H2O/Kg air).
  RV            - relative humidity.
  LAT           - lattitude of the significant point.
  LNG           - longitude of the significant point.
```

```
VX           - windspeed in horizontal direction.
                 ( > 0 for westwind, in meters/second)
VY           - windspeed in horizontal direction.
                 ( > 0 for southwind, in meters/second)
VW           - windspeed in vertical direction.
                 ( > 0 is downward movement, in millibar/minute)
SEA.PCT      - percentage  of time the model was so far above sea.
LANDCL.PCT   - land*cloudcover percentage .

C     END OF PRINTING PROCEDURES

      SUBROUTINE FYSDAT
      COMMON/BLOCK1/AC1L,AC1W,AC2L,AC2W,ACC1,ACC2,ACC3,ACC4,
     *       ASWINB,APLTRG,ASTEFA,ASOLAR,AALBDO,
     *       /BLOCK2/ACNST1,ACNST2,ACLOUD,
     *             ABETA1,ASOILF,ARADDV,
     *        /BLOCK3/ANEUT,ACIL,
     *        /BLOCK4/AKARM,
     *       /BLOCK5/ACONST,
     *       /BLOCK6/ACELS,
     *       /BLOCK7/API,RGASS,G,CP,ALLATH
      -(lines 2900000-2949000)
       in this subroutine all model constants & parameters are
       put in the appropiate common blocks.
```

6. REFERENCES

1. Cats,G,1986: Coding conventions for the HIRLAM-project KNMI ,FM-86-7,KNMI,De Bilt,The Netherlands [**Unpublished memorandum**]
2. Reiff,J et al,1984: An air mass transformation model for short-range weather forecasting. Mon.Wea.Rev. ,112,393-412.