European Centre
for Medium Range
Weather Forecasts

Documentation of the E.C.M.W.F. Spectral Model

Internal Report No 19
Research Dept.

October 1978

Centre Européen pour les Prévisions Météorologiques
à Moyen Terme

Europäisches Zentrum für mittelfristige Wettervorhersage

DOCUMENTATION OF THE E.C.M.W.F. SPECTRAL MODEL

by

A. P. M. Baede
(with an Appendix by M. Jarraud)

European Centre for Medium Range Weather Forecasts, Bracknell

Internal Report No. 19

RESEARCH DEPARTMENT

July 1978

No. 612

N O T E:

This paper has not been published and should be regarded as an Internal Report from ECMWF Research Department.

Permission to quote from it should be obtained from the Deputy Director, Head of Research, at ECMWF.

Contents

Contents (contd.) page

Contents (contd.) page

Contents (contd.)

ERRATUM

References made to Internal Report 21 (to be published) should read Technical Report 15.

A B S T R A C T

This report contains the documentation of the adiabatic, triangular, semi-implicit version of the ECMWF's Spectral Model. Its mathematical formulation is described separately in ECMWF Internal Report 21 (to be published).

This program was developed on the CYBER-175 but later implementation on the CRAY-1 was envisaged, in particular with respect to vectorisation of the code. Nevertheless, some machine dependent features are present in the program.

The overall structure of the program is very similar to that of the grid point model, documented in ECMWF Internal Report No. 9. To avoid duplication that report is often referred to. As in the grid point model, the Olympus programming system is used, including its coding conventions, except in some routines which were obtained from other sources.

-.-.-.-

Introduction

The program documentation, presented in this report, follows roughly the same lines as the documentation of the grid point model (Haseler and Burridge, 1977, further referred to as HB). This is possible because the general structure of both programs is very similar. Both models use the Olympus programming system, described by Roberts, 1974.

In chapter 1 a general outline of the program structure is described, followed in chapter 2 by a detailed description of the data structure. Chapter 3 describes the I/O system, including a detailed documentation of the sub-routines which control the I/O. Chapter 4 is devoted to the dynamical computations. This is done in five sections, the first devoted to the initialization, the second to the adiabatic computations in the first scan, followed by a special section on subroutine DYN which controls all grid point computations in the first scan. Here the relevant equations are presented, followed by a detailed documentation of the subroutines involved. Section 4 of chapter 4 describes the computation in spectral space, in particular the completion of the timestep and the horizontal diffusion. Finally, in the last section the computations in the second scan are documented.

Chapters 5 and 6 describe the use and creation of the Start Data Set and the Initial Data Set. The next chapter describes how to run the job and gives examples of card decks, creating the data sets, and running the job. Chapters 8, 9 and 10 present details on the namelists used in the program, contain tables for all common blocks and give a comprehensive list of all subroutines, followed by some diagrams.

In two appendices, we give information on some auxiliary routines, obtained from other sources.

After completion of this documentation, the program was modified by M. Jarraud, mainly to allow for other types of truncation and for hemispheric as well as global integrations. These modifications are documented in a third appendix.

CHAPTER 1 - PROGRAM STRUCTURE

The structure of the program is very similar to that of the grid point model (Internal Report No. 9). This could be achieved by considering a spectral model as a special type of grid point model, in which the computations in spectral space are only an intermediate stage in the course of a timestep. All data files (i.e. work files and history files) therefore contain grid point information only.

A simplified flow diagram of the model is presented in diagram 1. A comparison with diagram 1 in HB shows that the structure of the programs is identical down to STEPON.

For a description of this part, we refer to Roberts, 1974 and HB. Below we present a general outline.

In diagram 1, those subroutines which have not been changed, or have been changed only in minor details which do not influence the flow diagram, are underlined by solid lines. For the flow diagram of these routines, see HB. Subroutines which have been changed substantially are underlined by a dashed line. Their flow diagrams are presented here in chapter 10. New subroutines are not underlined. Apart from the main program, these are all routines controlled by STEPON, and contain the actual model's computations.

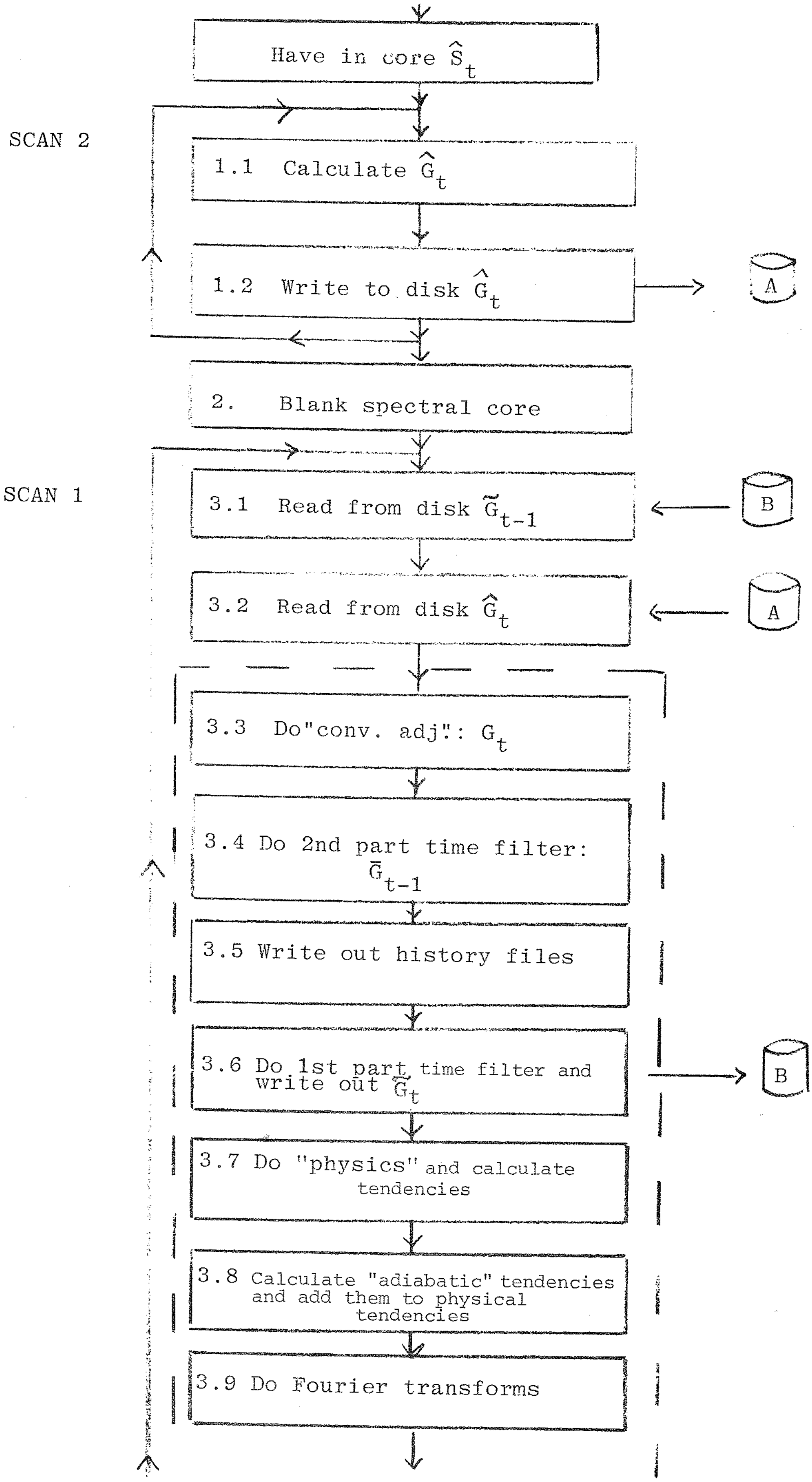
The general outline of the program is as follows: the Start Data Set (SDS) is read to locate the input data. This SDS contains a record for the initial data file and for each history file produced in the course of the integration. By specifying whether an initial run or a restart run is required, the appropriate record is selected from the SDS and the initial data can be located.

From the selected initial data work files are set up and written to disk. This happens in subroutines INITAL in the case of an initial run, or else in RESUME. The structure of these workfiles is described in detail in chapter 2. There are 3 such work files: one for input of T-1 data, one for output of T-1 data and one for I/O of T data (T-1 and T refer to the time level in the timestep). Each file contains 1 record for each latitude line of the Gaussian grid. At any time sufficient data to perform computations at one latitude line and 2 time levels are available in memory. Special I/O routines take care of the data transport between core and work files. This is described in chapter 3.

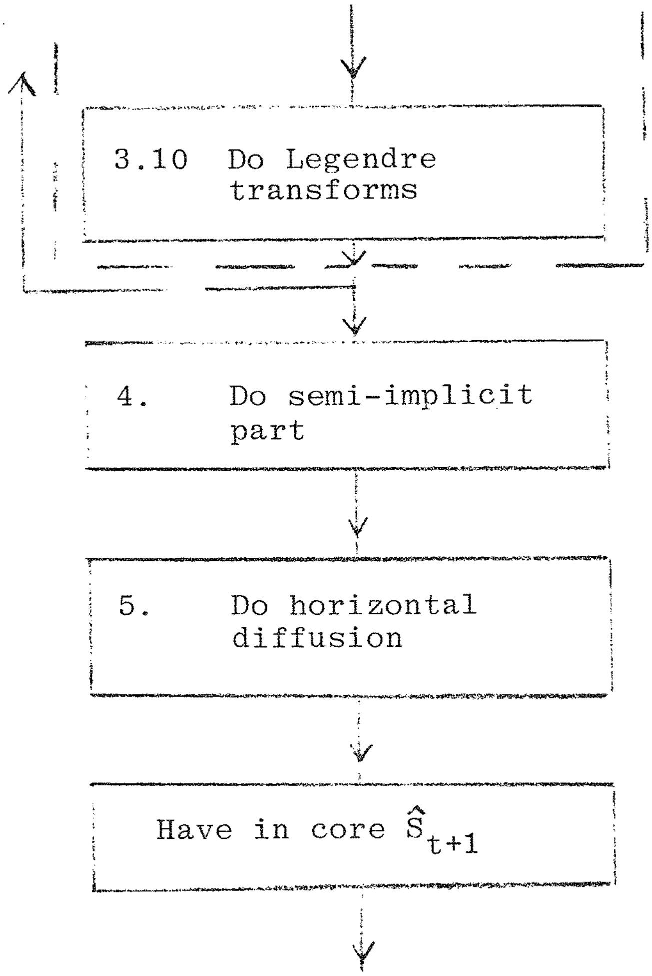
Fig.1.1 outlines the flow of a timestep and corresponds to the part of diagram 1, controlled by subroutine STEPON. A description of the details of Fig.1.1 is given in Baede and Jarraud, 1978.

Suffice it here to stress that the computation of one timestep takes place in two North to South scans through the Gaussian latitudes, controlled by subroutines STARTN + SCAN1 and SCAN2. This is in contrast to the grid point model where only one scan per timestep is performed.

From Fig.1.1 it is immediately clear why 2 T-1 work files and only 1 T work file is required. In SCAN1 T-1 data are both read and written, therefore requiring separate files which at the end of the timestep can be swapped. In contrast T data are written in SCAN2 and read in SCAN1, so that no conflict can arise and one file is sufficient.



(see next page)



Symbols

S_t : set of spectral coefficients of prognostic quantities at time t

G_t : set of grid point values of prognostic quantities at time t and at one line of latitude

$\hat{x}$: a quantity that has not been subjected to time smoothing or convective adjustment

x : a quantity that has been adjusted convectively but has not been time smoothed

$\tilde{x}$: half time smoothed quantity

$\bar{x}$: fully time smoothed quantity

Fig. 1.1 Flow diagram of a timestep

The part in the dashed box in SCAN 1 is controlled by subroutine LINEMS and contains the adiabatic part of the calculation (controlled by DYN), the physics (controlled by PHYS) and the convective adjustment (controlled by CONVAD).

At specified times history files are written from LINEMS and corresponding records added to the start Data Set.

CHAPTER 2 - DATA STRUCTURE

2.1 Spectral data

Fields of complex spectral expansion coefficients of the following variables are kept in memory:

ζ vorticity (sec^{-1})

D divergence (sec^{-1})

T' $T-T_o$ in which T is the temperature (K) and T_o is a specified reference temperature (K) which is only dependent on the vertical coordinate σ

q humidity mixing ratio (kg/kg)

$\ln p_*$ where p_* is the surface pressure (bar)

These fields are in COMMON block COMSPE and are not subject to any I/O.

The spectral fields are stored in one-dimensional arrays. At present the truncation is triangular only. The order of the coefficients in these arrays is shown in Fig.2.1.1. This diagonal arrangement, rather than the more common column-wise arrangement, was chosen for reasons of vectorization. For the three dimensional variables ζ, D, T', q the coefficients of all levels are stored in one array, the levels being arranged from the top of the atmosphere to the bottom (see paragraph 2.3).

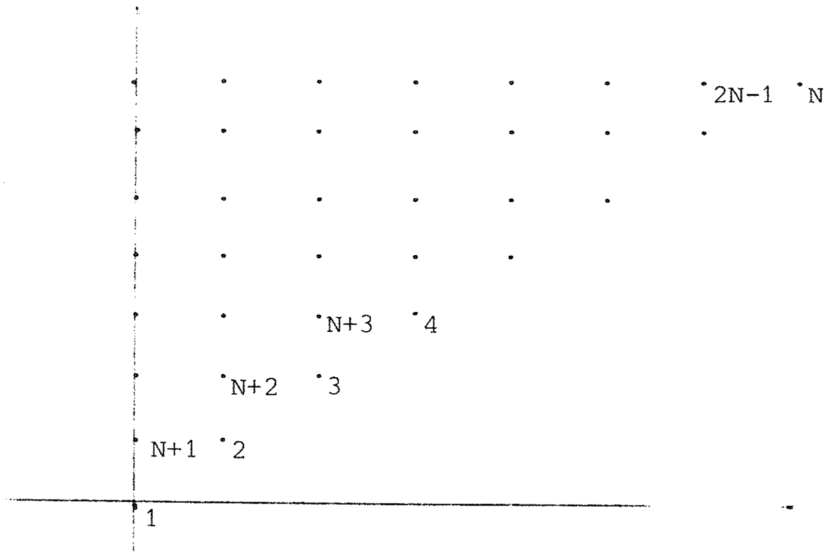


Fig. 2.1.1 Order of spectral coefficients in triangular truncation

2.2 The gaussian grid

Part of the calculation is performed in grid point space on a so-called gaussian (or transform) grid, which is regular in the east-west direction and slightly irregular in the north-south direction. The poles do not belong to this grid.

All prognostic variables are kept in the same points of the grid (no staggering).

Throughout this documentation, a "row" of data will mean the data of all, regularly spaced, longitude points at one gaussian latitude, the first point being situated at the Greenwich meridian, and from there proceeding in an easterly direction.

At any time one row of data for each time level will be available in memory for computations.

Such a row contains the following variables.
Prognostic variables:

ζ	vorticity (sec^{-1})
D	divergence (sec^{-1})
T'	temperature deviation (K)
q	mixing ratio (kg/kg)
$\ell n p_*$	p_* being the surface pressure (bars)

Derived variables:

U	$U = u \cos\phi$, u being the E-W velocity component and ϕ being the gaussian latitude
V	$V = v \cos\phi$, v being the N-S velocity component
$\frac{\partial \ell n p_*}{a\partial\lambda}$	E-W derivative of $\ell n p_*$ (available only at one time level)
$(1-\mu^2)\frac{\partial \ell n p_*}{a d\mu}$	N-S derivative of $\ell n p_*$ (with $\mu = \sin\phi$)

Constants:

ϕ_*	surface geopotential (m^2/sec^2)

2.3 Vertical distribution of variables

The vertical distribution of variables is identical to that of the grid point model. All prognostic variables are held on full σ-levels, whereas $\dot{\sigma}$ is calculated at half-levels. The distribution is shown schematically in Fig. 2.3.1

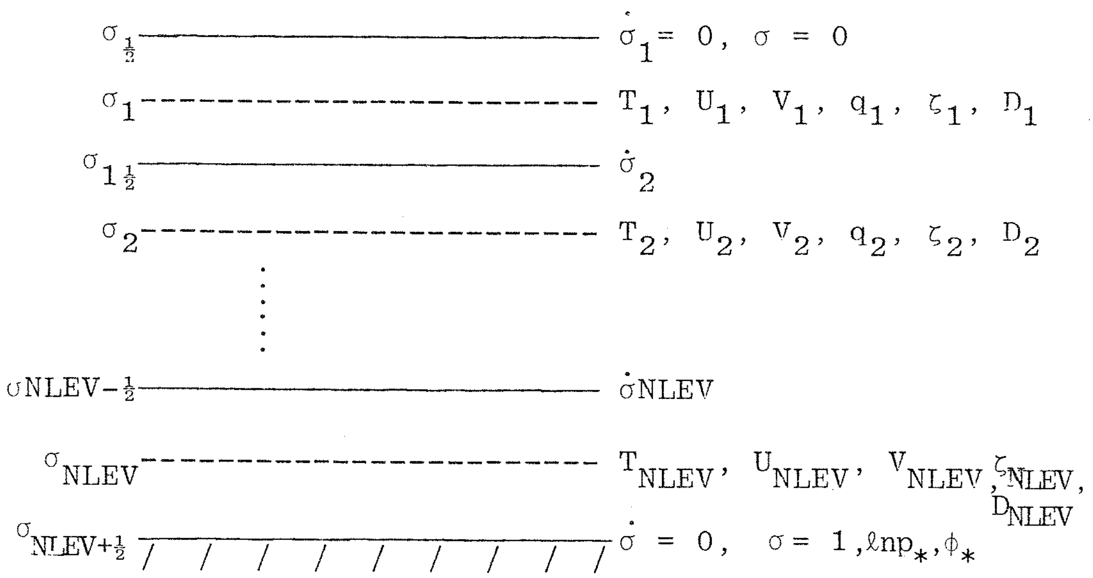


Fig. 2.3.1 Vertical distribution of variables

2.4 Grid point data storage

Grid point data are stored in three different ways:

1. Temporary work files on channel numbers NWKIN, NWKOUT and NWKIO.

2. Permanent files on channels NDATA, NM1A and NM1B.
   On NDATA the initial data set is found, and subsequent history files are written.
   To NM1A and NM1B data files are written of the time-steps following the write-up timesteps, in order to allow a restart from the history files.
   The file handling for a restart run is discussed below in paragraph 3.

3. The work buffers in memory.

Each of these three ways of data storage will be discussed in the next paragraphs.

2.5 Temporary work files

As can be seen in Fig. 1.1 and as discussed in Chapter 1, there are two types of workfiles: A and B.

- workfile A is used to transfer data from the second gaussian loop of a timestep to the first gaussian loop of the next timestep. Because the read and write are done in separate loops, no conflict can arise and one such file A on disk is sufficient.

- workfile B is used to transfer data from the first gaussian loop of a timestep to the first gaussian loop of the next timestep. Both read and write take place in the same loop, and therefore two files B are required on disk, one for input and one for output. These two files are swapped each timestep.

These workfiles are random access data sets, read and written using routines which can proceed in parallel with CPU-processing. There is 1 record for each latitude row, ordered from north to south. Each row contains data at one time level only, organised in the following order:

File A:

$$\phi_*,\ U_1,\ U_2,\ldots,U_{NLEV},V_1,\ldots,V_{NLEV},\ell np_*,T'_1,\ldots,T'_{NLEV},$$
$$q_1,\ldots,\ q_{NLEV},\zeta_1,\ldots,\zeta_{NLEV},D_1,\ldots,D_{NLEV},\ \frac{\partial\,\ell np_*}{a\,\partial\lambda},(1-\mu^2)\frac{\partial\,\ell np_*}{a\partial\mu}$$

where $x_k = x_{k,1},x_{k,2},\ldots,x_{k,NLON}$

i.e. variable x at level k for all longitude points at the given latitude. At the end of each group of NLON points, two extra words are allowed for the Fast Fourier Transform.

The vertical levels are stored from top (k=1) to bottom (k=NLEV).

File B:

has the same structure as file A, but the two derivatives of ℓnp_* at the end are omitted.

The following channel numbers are used for these work files:

NWKIO	:	work file A
NWKIN	:	work file B input
NWKOUT	:	work file B output

2.6 Permanent files

The permanent files are organised as sequential data sets. The common blocks COMBAS, COMHKP and COMMAP are stored in the first three records, followed by 1 record for each row of data, stored from North to South.

The structure of each row on channel NDATA (history file) is identical to that of file B. The structure of each row on channels NM1A and NM1B is identical to that of file A (see 2.5).

2.7 Work buffers in memory

In contrast to the grid point model, the dynamics of the spectral model require only 1 row of data at two time levels at any time in core. This requires two buffers in core, one containing one row of data from file A and one row from file B. A third buffer is required for the output to file B. In order to allow the I/O to proceed

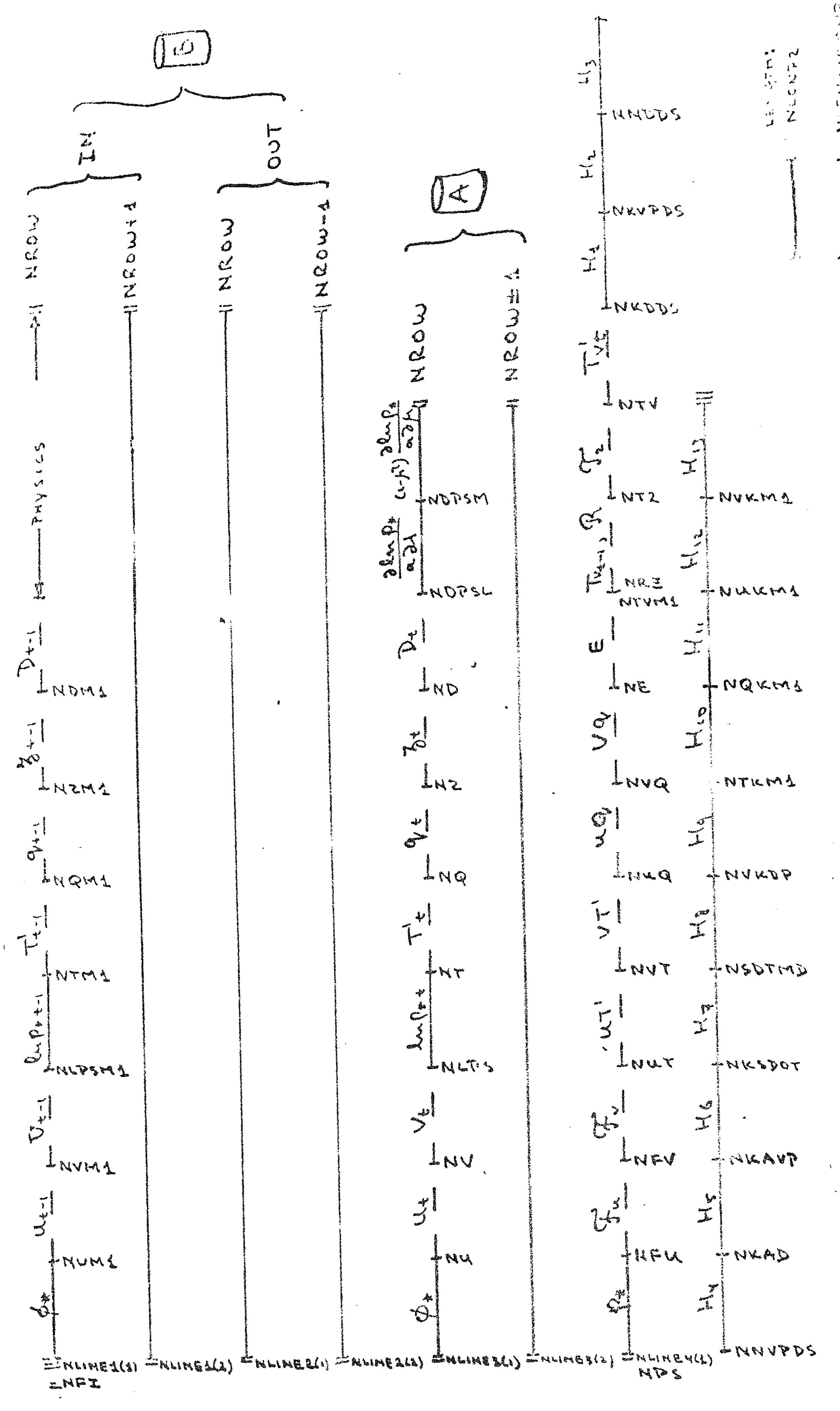


Fig. 2.7.1 Layout of blank common

in parallel with the computations, all buffers require a duplicate.

Therefore there are 4 buffers in core with the structure and length of a row of data of file B, and 2 buffers with the structure and length of a row of data of file A.

All these buffers are stored contiguously in blank common, followed by extra work space. The complete layout, including the pointers, is shown in Fig. 2.7.1 (see also: section 4.6 page 57).

2.8 Other disk files

One more work file is kept on disk on channel NTPLEG. This file contains the Associate Legendre Polynomials $P_{m,n}(\mu_j)$ and its derivatives $(1-\mu_j^2)\frac{d}{d\mu}P_{m,n}(\mu_j)$ for each gaussian latitude μ_j. The workfile is a sequential workfile, each record containing the data for one line of latitude, going from North to South. For the format of the fields see Fig. 4.1.1. Details of the production of the file are described in Chapter 4, pp.27-29.
The records are written and read with unformated WRITE and READ.

CHAPTER 3 - INPUT/OUTPUT

3.1 I/O between workfiles and memory

At the start of the first gaussian scan (SCAN1), the input work file B on channel NWKIN contains data for time T-1, and workfile A on channel NWKIO contains data for time T. During the scan from North to South, these data are read in row by row and a new output buffer, containing data at time T is generated and written to output workfile B on channel NWKOUT. At the end of the first scan the workfiles B are swapped so that the old output file B becomes the new input file B and vice versa.

In the course of the second scan (SCAN2) an output buffer is generated and written to workfile A on channel NWKIO.

The layout of the buffers in memory is presented in detail in Fig. 2.7.1 and is repeated here schematically in Fig. 3.1, together with the row number the buffer corresponds with, and its length. In SCAN1 computations are proceeding on the basis of lines 1 and 5, producing new values at line 3. In parallel with these computations, new values for the next row are read in to lines 2 and 6, and values generated at the previous row are written out from line 4. In SCAN2 grid point values are generated at line 5, whilst values generated at the previous line are written out from line 6 in parallel. When computations and I/O at row NROW have been completed lines 1-2, 3-4 and 5-6 are swapped.

displacement		ref	I/O	row	length
	B				
NLINE1(1)	|———————|	1	IN	NROW	NBFLNB
NLINE1(2)	|———————|	2	IN	NROW+1	NBFLNB
NLINE2(1)	|———————|	3	OUT	NROW	NBFLNB
NLINE2(2)	|———————|	4	OUT	NROW+1	NBFLNB
	A				
NLINE3(1)	|———————|	5	IN/OUT	NROW	NBFLNA
NLINE3(2)	|———————|	6	IN/OUT	NROW±1	NBFLNA

Fig. 3.1

3.2 Subroutines for I/O between work files and memory

As can be seen from diagram 1, the timestepping is handled by subroutine STEPON. From this routine, three subroutines are called which deal with the I/O between workfiles and memory: STARTN and SCAN1 deal with the I/O of the first scan and SCAN2 deals with the I/O of the second scan. In the following we treat these subroutines in a systematic way, referring to the flow diagrams at the end of this book in Chapter 10.

<2.1> STEPON (see diagram 6) controls timestep

<1.1> spectral fields are blanked at beginning of each timestep

<1.2> STARTN is called to start the I/O at the northernmost row of the first scan of each timestep

<1.3> SCAN1 is called to control the I/O at the subsequent rows of the first scan. At the end of the scan the sequential file NTPLEG, which contains the Legendre polynomials and derivatives for each row, must be rewound

<2.1> TSTEP is called to finalise the time extrapolation in spectral space, followed by a call to HORDIF which performs the linear horizontal diffusion

<3.1> SCAN2 is called to control the I/O of the second scan

<2.7> STARTN (see diagram 7) starts I/O of first scan at northernmost row

<1.1>-<1.4> The displacements of the buffers in blank common (see Fig. 3.1) are defined

<2.1> The northernmost row of data is read from workfiles A and B

<2.2> The second row is read from A and B.

<2.3> LINEMS is called to control the physics - and dynamics computations and the row counter NROW is incremented

<2.8> SCAN1 (see diagram 8) controls first scan, except 1st row

<1.1> The displacements of the buffers in Blank Common are swapped

<1.2> Initiate write of previous row to work file B

<1.35> In case present row is last row, wait for completion of reading

<1.4> Initiate read of next row from workfiles A and B

ECMWF LIBRARY

<1.5> Call LINEMS in order to control the computation on present row. Increment row number.

<2.12> SCAN2 (see diagram 9) controls second scan.

<1.1> GRCALC is called to perform Legendre transform from spectral space. Next MRFFT2 is called to perform the Fourier transform. This last transform is performed in steps of 64 transforms (for vectorization)

<1.2> The displacements of the buffers in blank common for file A are swapped

<1.3> The output of the gridpoint values at the present row to workfile A is initiated

<1.4> If present row is last row then wait for completion of last write and return. Otherwise increment row number and return.

3.3 I/O from and to permanent files

The system, described in this section, is designed to handle its permanent files automatically. The user only attaches the SDS, which contains enough information for the program to request, attach, catalog or change any other necessary permanent files. This system makes use, however, of subroutines, implemented on the CYBER-175, and not available at present on the CRAY-1. For a description of the Start Data Set we refer to Chapter 5.

3.4 Subroutines handling I/O from and to permanent files

The I/O from and to the permanent files is handled by three subroutines: INITAL and RESUME, where the initial data file is read in case of an initial run or a restart run respectively, and LINEMS, where history files are written and those files which are required for a proper restart.

Automatic file handling takes place by means of a set of subroutines, performing the functions, usually performed through the corresponding NOS/BE Job Control Language commands. We refer for documentation to the NOS/BE manual and the documentation of these routines (N. Storer, 1976). Let us summarise here some characteristics:

CALL MOUNT (IFAIL, NDMTSN, NDMTVS)

CALL DSMOUNT (IFAIL, NDMTSN, NDMTVS)

These subroutines have not been implemented and are at present replaced by dummy functions. If the user wishes so, he can mount and dismount his private disk through Job Control Language

IFAIL : return error code (see documentation)
NDMTSN: private disk set name
NDMTVS: private disk VSN

CALL ATTACH (IFAIL, NDATA, NDTFN,NDTACY, NDTAPN)

Attaches existing permanent file logically to job.

IFAIL : return error code (see documentation)
NDATA: logical unit number
NDTFN(4): Hollerith array containing file name, right filled with blanks
NDTACY and NDTAPW: see call CATALOG below

CALL REQUEST (IFAIL, NDATA, NDREQ)

Requests permanent file space for file on channel NDATA.

IFAIL : return error code (see documentation)
NDATA: logical unit number
NDREQ: pass word for request (e.g.: *PF. for public disk, *SN=DSETnn for private disk)

CALL ALTER (IFAIL, NM1A)
Permits a sequential file to be overwritten from current position
IFAIL : return error code (see documentation)
NM1A : logical unit number

CALL EXTEND (IFAIL, NM1A)
Permits permanent modification of a permanent file
IFAIL : return error code (see documentation)
NM1A : logical unit number

CALL CATALOG (IFAIL, NM1A, NM1AFN, NM1ACY, NM1APW)
Makes file on NM1A permanent
IFAIL : return error code (see documentation)
NM1A : logical unit number
NM1AFN(4): Hollerith array containing file name
NM1ACY: cycle number (if=0, then next available cycle number is selected
NM1APW(4): Hollerith array containing pass word (e.g.: "ID=EWAB3")

CALL RETURN (IFAIL, NDATA)
Detaches file logically from job
IFAIL : return error code (see documentation)
NDATA : logical unit number

The following subroutines handle the I/O from and to permanent files

<1.6> <u>INITAL (diagram 3)</u>. Initialises initial run

<1.2> COMBAS variables which may have been changed in MODIFY and which therefore differ from the values in COMBAS on the initial data file, are stored temporarily in local variables.

<2.1> if NLMNT=.TRUE. a private disk is mounted. As remarked above this has, however, not yet been implemented.

<2.2> The file, containing the initial data set is attached.

<2.5> CALL DATCOM to read common blocks COMBAS, COMHKP and COMMAP from initial data set.

<2.6> COMBAS variables, saved in <1.2> are put back in COMBAS

<2.7> CALL DATINI (diagram 5) to initialise some constants

<3. > Files NWKIO and NWKIN are opened for random access I/O and NWKOUT is opened and structured. (See Burridge and Haseler, 1976).

<4. > Permanent file space for the T+1 data files, required for restart, is requested on channels NM1A and NM1B. The file names are generated by subroutine FILENM.

<4.2> NM1A is catalogued, using the same password as for the history files.
ENDFILE NM1A and REWIND NM1A are the minimum necessary operations for creating an empty permanent file.

<4.5>- <4.7> idem for NM1B.

<5.1> Data is read from initial data set into blank common

<5.15> The reference temperature profile T_o is subtracted

<5.2> That part of the initial data to be written to file B is copied to end of the data, read in <5.1>. Subroutine COPYBC copies data in blank common.

<5.3>- <5.4> Work files A and B are filled with data. The first timestep is a forward step so the data in A and B are the same, except for the derivatives of ℓnp_*, which are read in to A but not B.

<6. > Files are returned and private disk dismounted (not implemented)

<1.7> RESUME (diagram 4) Initialises restart run

<1. >- <3. > see INITAL

<4. > The T+1 data file, required for restart is attached.

<4.3>-4.4> In order to check if the proper timestep has been attached, COMBAS is read in <4.3> and the timestep value in COMBAS is compared with NSTEP in <4.4>.

<5.1> Data is read from history file in to blank common.

<5.15> Refererence temperature profile T_o is subtracted.

<5.2> Data are read from T+1 data file in to blank common.

<5.3>-<5.4> Work files A and B are filled with data. A restart timestep is necessarily a leapfrog timestep and therefore the data in A and B are different.

<6. > Files are returned and private disk dismounted.

<2.13> LINEMS (diagram 10) output to permanent files.
Note: here LINEMS is documented, in so far as it

controls the I/O to permanent files. The documentation of its other function is found at pp. 31-32.

In order to restart properly, two files have to be saved at regular intervals, one containing the data of file A (time T) on channel NM1A, and the other one containing the data of file B (time T-1) on channel NDATA. The latter moreover serves as a history file. For this reason a new permanent file is created on channel NDATA for each write-up. The file on NM1A is swapped with a file on NM1B at the next write-up and overwritten alternately.

<2. > Write up if NSTEP.EQ.NWRITE+1 or if SWITCH1 is set.

<2.05> if NLMNT= .TRUE. a private disk is mounted, otherwise public disk is used.
<u>This has not yet been implemented</u>. At present the private disk should be mounted by Job Control Language.

<2.1> The information concerning files on NM1A and NM1B is swapped.

<2.15> file NM1A is attached

<2.2> The file NM1A is rewound and altered in order to allow overwriting this permanent file.

<2.3> Before writing the history file, the reference temperatures T_o are added.

<2.35> Row NROW is written to NM1A and NDATA and the status of the files is checked.
Note: the temperatures on the history file are real temperatures $T=T_o+T'$, but the temperatures on NM1A are temperature deviations T'.

<2.4> After writing the history file, the reference temperatures T_o are subtracted again.

<2.5> When the last row has been written, generate a file name for the history file (CALL FILENM) and catalog and return the history file

<2.6> Extend file NM1A, i.e. make the overwriting of this file permanent and return it.

<2.7> CALL SDS in order to add a record to the start data set, containing enough information to enable the program to restart from this timestep

<2.85> if NLMNT=.TRUE. the private disk is dismounted by CALL DSMOUNT.
This has not yet been implemented.

CHAPTER 4 - THE DYNAMICS

Four parts of the code are relevant to the dynamical computations: the initialization, the computations in the first scan (the bulk of the computations), the computations in spectral space and, finally, the computations in the second scan. In the following four paragraphs these four parts will be discussed in detail, each subroutine being treated in a subsection of these paragraphs. For the mathematical formulation of the model, see Baede and Jarraud (1978). Here we only repeat the applied formulae. The last section 4.6 contains the names and the meaning of the pointers in blank common.

4.1 Initialization

Initialization is the set-up of constants, relevant to the integration. Part of this is done in the creation of the data set and is communicated to the integration through common blocks COMBAS, COMHKP and COMMAP. This is discussed in Chapter 5, on the creation of a data set. The rest of the initialization is performed during the run, in subroutines MODIFY and DATINI. MODIFY allows the modification of some basic control data, specific for each run. DATINI initialises a large number of common block variables, which have not been initialised before. This subroutine is called from both INITAL and RESUME (see paragraph 3.4 and diagrams 3-5). In this paragraph we present a documentation of the subroutines MODIFY and DATINI.

<0.2> MODIFY (diagram HB). Modification of basic control data.

<1.1> Namelist REST is read, which determines whether the present run is an initial or a restart run. This changes the COMBAS variables NLRES and NREC (see 9.2).

<1.2> In case of a restart, the last record of the start data set is read if NREC=1; otherwise the corresponding record IREC=NREC is read

<1.3> SDS is called to read (ICALL=2) the appropriate record from the start data set

<1.4> In case of an initial run (NLRES=.FALSE.) the record IREC=NREC is read from the start data set.

<u><1.10> DATINI</u> (diagram 5). Initialization of common variables.

<1. > A number of housekeeping parameters in COMHKP, COMIOC and COMDBC are preset. Moreover some local constants are defined.

<2. > Namelist NEWRUN is read and on the basis of these values some default values are overwritten.

<3.1> An isothermal reference atmosphere (T=300K) is defined for the semi-implicit scheme. This reference profile may be changed via namelist SEIMP.

<3.2> The constant matrix $\underline{\underline{\tau}}$ is computed here, the elements of which are given by

$$\tau_{k,j} = \frac{1}{2\Delta\sigma_k} \Big[\underbrace{(\bar{T}_{k+1} - \bar{T}_k)}_{(1)} \Big\{ \sigma_{k+\frac{1}{2}} . \Delta\sigma_j - \underbrace{\left| \begin{matrix} 0 \; (k < j) \\ \Delta\sigma_j (k \geqslant j) \end{matrix} \right.}_{(2)} \Big\} \Big]$$

$$+ \frac{1}{2\Delta\sigma_k} \Big[\underbrace{(\bar{T}_k - \bar{T}_{k-1})}_{(3)} \Big\{ \sigma_{k-\frac{1}{2}} . \Delta\sigma_j - \underbrace{\left| \begin{matrix} 0 \; (k-1 < j) \\ \Delta\sigma_j (k-1 \geqslant j) \end{matrix} \right.}_{(4)} \Big\} \Big|$$

$$+ \underbrace{K . \bar{T}_k . A_{kj}}_{(5)}$$

First τ_{11} is computed, followed in 321-loop by τ_{1j}.

In the 322-loop the terms (1) and (3) are added up for τ_{kj} ($k \geq 2$, j=1,NLEV) and finally in 323-loop terms (2), (4) and (5) are added. Note that we make use of the fact that matrix A_{kj} is a lower triangular matrix. Note further the use of factor IJL to meet the special conditions in term (4).

<3.3> Two quantities used in subroutine STATS are computed: $\sum_{\ell=1}^{NLEV} T_o(\ell)\Delta\sigma_\ell$ and $\sum_{\ell=1}^{NLEV} T_o^2(\ell).\Delta\sigma_\ell$

<4. > The NLEV*NLEV matrices $\underline{\underline{A}}_n^{-1} = \left[\frac{1}{c_n} I + R(\underline{\underline{G}}\ \underline{\underline{\tau}} + \underline{T_o}.\Delta\sigma|)\Delta t^2\right]^{-1}$

are computed here for each n, and stored in a large 1-dimensional array BM1 as follows

BM1 |(NLEV*NLEV) (NLEV*NLEV). . . .(NLEV*NLEV)|
n=o n=1 n=NMAX-1

<4.1> Compute $\underline{\underline{B}} = R.(G.\underline{\underline{\tau}} + T_o.\Delta\sigma|)$

and store in array AQ in COMIMP and in local array ZB

<4.2> Compute the eigenvalues of this matrix, which are the squares of the NLEV normal mode gravity wave speeds. Write these gravity wave speeds to output.

<4.3> Multiply by Δt^2 (431-loop) and add $\frac{1}{c_n} I$ (432-loop). Invert this matrix $\underline{\underline{A}}_n$ and store the result in array BM1 in COMIMP.

<5. > Constants related to the Legendre transform are computed in this section. The values $\sin\phi_j = \mu_j$ of the gaussian latitudes are stored in a local array

ZB for one hemisphere only. The counter I indicates the hemisphere (Northern hemisphere: I=0; SH:I=1). So the sections 5.1-5.4 are done for the NH first and then repeated for the SH.

<5.1> CALL PHCS: computes the values of the Associated Legendre Polynomials (A.L.P) $P_{m,n}(\mu_j)$ and their derivatives $(\mu_j^2-1)\,\frac{d}{d\mu}\,P_{m,n}(\mu_j)$ for gaussian latitude μ_j = ZB(JN). (The A.L.P. are required on an extended triangular field (see Fig.4.1.1). The output of PHCS however is an extended rhomboidal field ZALP in column-wise storage.

CALL REORD1: reorders the extended rhomboidal field with column-wise storage into an extended triangular field with diagonal-wise storage (see Fig.4.1.2).

<5.2> This is repeated for the derivatives of the Associated Legendre Polynomials (D.A.L.P.) on a normal triangular field.

See appendix 1 and 2 for a documentation of subroutines PHCS, REORD1, REORD2.

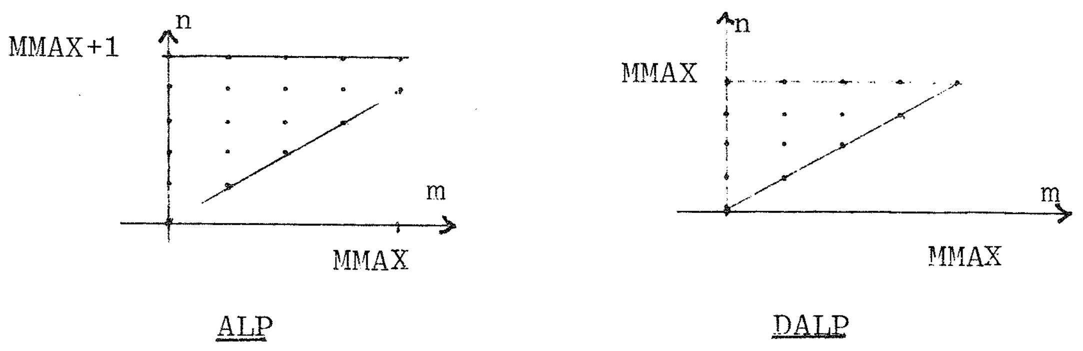


Fig. 4.1.1 Triangular (right) and extended triangular (left) truncation

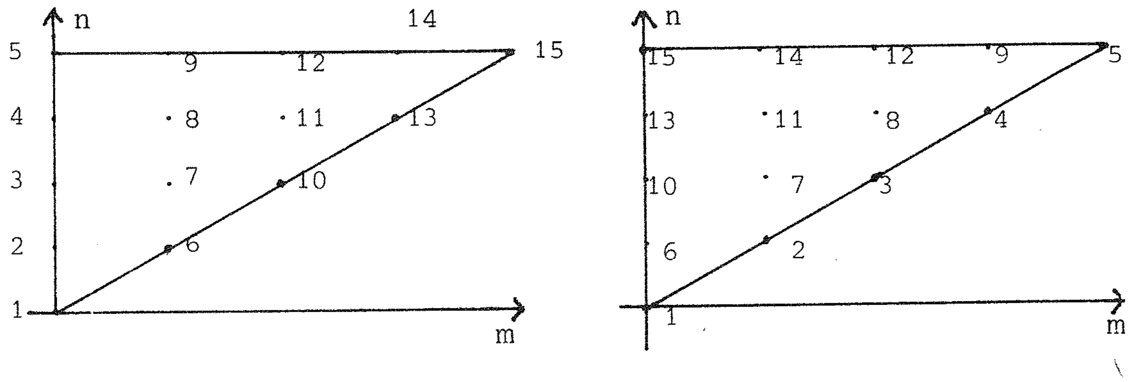


Fig. 4.1.2 Column-wise (left) and diagonal-wise (right) storage

<5.3> The following normalization of the spherical harmonics is used:

$$\int_{0}^{2\pi}\int_{-1}^{+1} Y_{m_1,n_1}(\lambda,\mu) . Y^{*}_{m_2,n_2}(\lambda,\mu)\; d\mu d\lambda = 2\pi\; \delta_{m_1,m_2}\; \delta_{n_1,n_2}$$

with $Y_{m,n}(\lambda,\mu) = P_{m,n}(\mu).e^{im\lambda}$

In order to get this normalization the A.L.P's must be multiplied by $\sqrt{2}$ (see app. 2.3). Moreover, $(1-\mu_j^2)\frac{d}{d\mu}\, P_{m,n}(\mu_j)$ is required rather than the computed $(\mu_j^2-1)\frac{d}{d\mu}\, P_{m,n}(\mu_j)$.

<5.4> The fields ALP and DALP are written to a sequential file on channel NTPLEG.

<5.5> I is incremented and the computations 5.1-5.4 are repeated for the southern hemisphere.

<6.1> CALL RFTSET to set up trigonometric tables in common block COMFFT for the Fast Fourier Transform. For a documentation of RFTSET, see the comments in the source listing of this subroutine.

<6.2> The FFT subroutine used at present is a vectorised one. The number of transforms is therefore divided in groups of NCRAY=64 and a remainder. The first group of transforms of length 4*NLEV+1 is the group of time level t-1 values of the prognostic quantities:

$\ln p_*(t-1)$, $T'(t-1)$, $q(t-1)$, $z(t-1)$, $D(t-1)$

The second group of length 8*NLEV is the group of non-linear quantities:

$\mathcal{F}_u$, $\mathcal{F}_v$, UT', VT', UQ, VQ, E, R

The third group of length 6*NLEV+3 is the group of time level t values in the second scan:

$U(t)$, $V(t)$, $\ln p_*(t)$, $T'(t)$, $Q(t)$, $\zeta(t)$, $D(t)$, $\frac{\partial \ln p_*}{a\partial\lambda}(t)$ and

$$(1-\mu^2)\frac{\partial \ln p_*}{a\partial\mu}(t)$$

4.2 Control of adiabatic computations in the first scan

All dynamical computations in the first gaussian loop are controlled by subroutine LINEMS, called from subroutine SCAN1 (see diagram 1). LINEMS calls POINTS to set up the pointers to blank common, organises the time filter TIMESM, writes history and restart data to permanent files and finally calls PHYS and DYN to compute the non-adiabatic and adiabatic contributions to the tendencies. All these routines will now be documented in this order.

Subroutine DYN and all routines called from DYN are discussed in section 4.3.

<2.13> LINEMS (diagram 10) controls the dynamics in first scan. The part of this subroutine, which controls the I/O to permanent files was documented in section 3.4. Here we shall limit ourselves to the part, relevant to the dynamics, i.e. sections <0.7>, <1. >, <3. >, <4. > and <5.>.

<0.1> CALL POINTS to set pointers to blank common (see below).

<0.2> Common block COMLEG is filled with data, relevant to the present line of latitude.

<1. > The organisation of this section is identical to that of the grid point model. We refer to pp.18-19 of HB. The only differences are:

<1.5> This is the place to perform the convective adjustment of the temperature and humidity profiles. At present CONVAD is a dummy routine. With respect to the first steps of an initial and restart run, CONVAD follows the same rules as the second part of the timefilter, i.e. in case of an initial run, no adjustment takes place during the first three timesteps and in case of a restart run no adjustment takes place during the first timestep (see Fig.4.1.3 on p.19 of HB).

<1.6> In contrast to the grid point model, both parts of the time smoothing are done by the same routine TIMESM (see below). In this section the second part of the time filter replaces the partially time smoothed values $\tilde{G}_{t-1}$ in blank common by the fully time smoothed values $\bar{G}_{t-1}$.

<3. > The first part of the time filter produces partially smoothed grid point values $\tilde{G}_t$ and stores them in the output buffer for file B, to be written out to disk later on.

<3.1> If no 1st part of the time smoothing takes place, values are copied straight to the output buffer Subroutine COPYBC transfers data within blank common and makes use on the CYBER of the CDC feature MOVLEV. For other machines, COPYBC has to be adapted.

<4. > CALL PHYS is at present a dummy routine.

<5. > CALL DYN see below.

<u><2.9> POINTS</u> (Initialises displacements of grid point variables in blank common)

Fig. 2.7.1 presents the lay-out of blank common, including the names of the pointers. The meaning of the names is found in section 4.6.

<u><2.17> TIMESM</u> (KDISA, KDISB, KDISC, PA, PB, KNO)

This routine forms a linear combination of two fields in blank common with length KNO, starting at KDISA+1 and KDISB+1, and stores the result in blank common starting at KDISC+1.

<u><2.20> PHYS</u> (calls physics routines)

This routine is empty at present. It only initialises the velocity tendency fields NFU and NFV.

4.3 Subroutine DYN; the adiabatic computation

Subroutine DYN calls all subroutines required to do the grid point calculations in the first scan, and the subsequent transformation back to spectral space. Before presenting a detailed documentation of these subroutines we give the equations, referring to Baede and Jarraud (1978) for a detailed description.

4.3.1 The equations in spectral, finite difference form

In the following suffixes j, k and l indicate the vertical levels , suffixes m,n indicate the spectral indices. All important terms are numbered and have a letter L, G, I attached to it. G indicates that this term is evaluated in grid point space; L indicates that this term is calculated during the Legendre-transform back to spectral space; and I indicates that this term is added during the implicit part of the calculation in spectral space.

Vorticity equation:

$$(\zeta_k)_{m,n}(t+\Delta t) = (\zeta_k)_{m,n}(t-\Delta t)+2\Delta t.(Z_k)_{m,n} \qquad (1.\ )$$

$$(Z_k)_{m,n} = \left[\frac{1}{1-\mu^2}\ \frac{\partial}{\partial\lambda}\ (\mathcal{F}_v)_k - \frac{\partial}{\partial\mu}(\mathcal{F}_u)_k\right]_{m,n} \qquad (1.1)\text{L}$$

$$\left.\begin{aligned}
(\mathcal{F}_v)_k &= \overset{(a)}{-U_k\cdot\zeta_k} + \frac{1}{2\Delta\sigma_k}\left[\overset{(b)}{\dot\sigma_{k+\frac12}(V_{k+1}-V_k)}+\overset{(c)}{\dot\sigma_{k-\frac12}(V_k-V_{k-1})}\right] \\
&\quad \overset{(d)}{-R(T_v')_k\ (1-\mu^2)\ \frac{\partial\,\ell n p_*}{a\,\partial\mu}} \\
(\mathcal{F}_u)_k &= \overset{(a)}{V_k\cdot\zeta_k} - \frac{1}{2\Delta\sigma_k}\left[\overset{(b)}{\dot\sigma_{k+\frac12}(U_{k+1}-U_k)}+\overset{(c)}{\dot\sigma_{k-\frac12}(U_k-U_{k-1})}\right] \\
&\quad \overset{(d)}{-R(T_v')_k\ \frac{\partial\,\ell n p_*}{a\partial\lambda}}
\end{aligned}\right\} \qquad (1.2)\text{G}$$

Temperature equation

$$(T'_k)_{m,n}(t+\Delta t) = (T'_k)_{m,n}(t-\Delta t)+2\Delta t.(\mathcal{J}1_k + \mathcal{J}2_k)_{m,n} \quad (2.\ )$$

$$-2\Delta t.\sum_{\ell=1}^{N} \tau_{k\ell}(\bar{D}^t_\ell)_{m,n}$$

$$(\mathcal{J}1_k)_{m,n} = \left[-\frac{1}{1-\mu^2}\frac{\partial}{a\partial\lambda} A_k - \frac{\partial}{a\partial\mu} B_k \right]_{m,n} \quad (2.1)L$$

$$A_k = U_k.T'_k \quad (2.2)G$$

$$B_k = V_k.T'_k \quad (2.3)G$$

$$\mathcal{J}2_k = \overset{(a)}{D_k T'_k} - \frac{1}{2\Delta\sigma_k}\left[\overset{(b)}{\dot{\sigma}_{k+\frac{1}{2}}(T'_{k+1}-T'_k)}+\overset{(c)}{\dot{\sigma}_{k-\frac{1}{2}}(T'_k-T'_{k-1})}\right]$$

$$- \frac{1}{2\Delta\sigma_k}\left[(\bar{T}_{k+1}-\bar{T}_k)\ \overset{(d)}{\left(\sigma_{k+\frac{1}{2}}.\sum_{j=1}^{N}\Delta\sigma_j\vec{V}_j.\nabla\ell np_* - \sum_{j=1}^{k}\Delta\sigma_j.\vec{V}_j.\nabla\ell np_*\right)}\right]$$

$$- \frac{1}{2\Delta\sigma_k}\left[(\bar{T}_k-\bar{T}_{k-1})\ \overset{(e)}{\left(\sigma_{k-\frac{1}{2}}.\sum_{j=1}^{N}\Delta\sigma_j\vec{V}_j.\nabla\ell np_* - \sum_{j=1}^{k-1}\Delta\sigma_j.\vec{V}_j.\nabla\ell np_*\right)}\right]$$

$$\overset{(f)}{- K.\bar{T}_k.\sum_{j=1}^{N} A_{kj}.\vec{V}_j\ \nabla\ell np_*}$$

$$\overset{(g)}{- K.T'_{vk}.\sum_{j=1}^{N} A_{kj}(\vec{V}_j\ \nabla\ell np_*+D_j)}$$

$$\overset{(h)}{+ K.T_v.\vec{V}_k.\nabla\ell np_*} \quad (2.4)G$$

$$\sum_{\ell=1}^{N} \tau_{k\ell}(\overline{D_\ell})^t_{m,n} \quad (2.5)I$$

Humidity equation

$$(q_k)_{m,n}(t+\Delta t) = (q_k)_{m,n}(t-\Delta t) + 2\Delta t(Q1_k + Q2_k)_{m,n} \qquad (3.\ )$$

$$(Q1_k)_{m,n} = \left[-\frac{1}{1-\mu^2} \frac{\partial}{a\partial\lambda} F_k - \frac{\partial}{a\partial\mu} G_k \right]_{m,n} \qquad (3.1)\text{L}$$

$$(Q2)_k = \underset{(a)}{D_k \cdot q_k} - \frac{1}{2\Delta\sigma_k} \left[\underset{(b)}{\dot{\sigma}_{k+\frac{1}{2}}(q_{k+1}-q_k)} + \underset{(c)}{\dot{\sigma}_{k-\frac{1}{2}}(q_k-q_{k-1})} \right] \qquad (3.2)\text{G}$$

$$F_k = U_k \cdot q_k \qquad (3.3)\text{G}$$

$$G_k = V_k \cdot q_k \qquad (3.4)\text{G}$$

Continuity equation

$$(\ell n p_*)_{m,n}(t+\Delta t) = (\ell n p_*)_{m,n}(t-\Delta t) + 2\Delta t . \mathcal{P}_{m,n} - 2\Delta t . \sum_\ell \Delta\sigma_\ell . (\bar{D}^t_\ell)_{m,n} \qquad (4.\ )$$

$$\mathcal{P} = -\sum_{j=1}^{N} \Delta\sigma_j \; V_j . \nabla \ell n p_* \qquad (4.1)\text{G}$$

$$\sum_\ell \Delta\sigma_\ell . (\bar{D}^t_\ell)_{m,n} \qquad (4.2)\text{I}$$

Helmholtz equation

$$(\bar{D}^t_k)_{m,n} = \sum_{\ell=1}^{NLEV} (A_n^{-1})_{k,\ell} . \left[\frac{a^2}{n(n+1)} . (D_\ell)_{m,n}(t-\Delta t) + \Delta t \right.$$

$$\left. \left\{ \frac{a^2}{n(n+1)} (\mathcal{L}_\ell)_{m,n} + (\mathcal{R}_\ell)_{m,n} + \Delta t . R\left(\sum_{i=\ell}^{NLEV} B_{i\ell} \tilde{\mathcal{T}}1_i \right)_{m,n} \right\} \right] \qquad (5.\ )$$

$$(\mathcal{L}_\ell)_{m,n} = \left[\frac{1}{1-\mu^2} \frac{\partial}{a\partial\lambda}(\mathcal{F}u)_\ell + \frac{\partial}{a\partial\mu}(\mathcal{F}v)_\ell + \frac{a^2}{n(n+1)}(E_\ell) \right]_{m,n} \qquad (5.1)\text{L}$$

$$E_\ell = \frac{U_\ell^2+V_\ell^2}{2(1-\mu^2)} \qquad (5.2)G$$

$$\mathcal{R}_k = \phi_* + R \sum_\ell G_{k\ell} \left[(T_v(t-\Delta t))_\ell + \Delta t \mathcal{T}2_\ell \right] + R\bar{T}_k \cdot \ell n p_*(t-\Delta t) + \Delta t R \bar{T}_k \mathcal{P} \qquad (5.3)G$$

$$\Delta t R\left(\sum_{i=\ell}^{NLEV} B_{i\ell} \mathcal{T}1_i\right)_{m,n} \qquad (5.4)L$$

Divergence equation

$$(D_k)_{m,n}(t+\Delta t) = 2(\bar{D}_k^t)_{m,n} - (D_k)_{m,n}(t-\Delta t) \qquad (6.1)I$$

We now present a documentation of the different routines which compute the different terms in these equations.

4.3.2 The adiabatic subroutines, called from DYN

<2.16> GRMULT Non-linear calculations in grid point space.
In this subroutine all terms are computed which are labelled G in para. 4.3.1.

<1. > Some common block constants are transferred to local constants. One local constant is defined: ZVIR, being the ratio of the molecular weights of water and dry air, and used in the computation of the virtual temperatures.
If no virtual temperatures are required, set ZVIR=1.

<2. > The 13 auxiliary fields at the end of blank common (see Fig. 2.7.1) are set equal to 0.

<3.1> The following auxiliary quantities are computed here:

pointer	term
NVKDP	$\vec{V}_k . \vec{\nabla} \ell n p_*$
NNDDS	$\sum_{j=1}^{NLEV} D_j . \Delta\sigma_j$
NNVPDS	$\sum_{j=1}^{NLEV} (\vec{V}_j . \nabla \ell n p_*) . \Delta\sigma_j$

<3.2> The virtual temperatures at time levels t and t-1 are computed using the expression:

$$T_v = T \cdot \frac{0.622+q}{0.622(1+q)}$$

Note that field NTV contains $T_v'(t)$, i.e. the deviation from the reference temperature; field NTVM1 however contains $T_v(t-1)$, i.e. the complete virtual temperature.

<4. > The expressions for $\mathcal{F}u$, $\mathcal{F}v$, $\mathcal{T}2$ and Q (see para. 4.3.1) contain terms which, once they are computed, can be used at the next vertical level. For example, terms (d) of $\mathcal{T}2$ at vertical level k is identical to term (e) at level k+1. We take advantage of this situation by storing such fields in auxiliary fields and using them at the next vertical level. For this reason the computation is split in two parts.
It is important in this context to remember that the vertical scheme is subject to the following boundary conditions:

$$\dot{\sigma}_{\frac{1}{2}} = \dot{\sigma}_{NLEV+\frac{1}{2}} = 0$$

<4.05> For each level k the fields NKAD and NKAVP are initialized to zero.

<4.1> In this section some vertical sums up to level k are computed. First for each level j the term $\vec{V}_j.\nabla \ell np_*$ is computed (pointer NVKDP). Then the following sums are computed

pointer	term
NKAD	$\sum_{j=1}^{\ell} A_{kj}.D_j$
NKAVP	$\sum_{j=1}^{k} A_{kj}.(\vec{V}_j.\nabla \ell np_*)$

It is clear that at the end of the vertical summation field NVKDP contains: $\vec{V}_k.\nabla \ell np_*$.

In fetching matrix element A_{kj} we use pointer IJK2. Here we make use of the fact that A is a lower triangular matrix.

<4.2> Some straightforward non-linear terms are computed here. We list here the pointers and the reference to the terms in para. 4.3.1

pointer	term
NUT	(2.2)G
NVT	(2.3)G
NUQ	(3.2)G
NVQ	(3.3)G
NE	(5.2)G

Moreover, the first part of the terms $\mathcal{T}_2$, (2,4) and Q_2, (3.2)G are computed here as discussed above:

$\mathcal{F}u$: (a) + (c) + (d)
$\mathcal{F}v$: (a) + (c) + (d)
$\mathcal{T}2$: (a) + (c) + (e) + (f) + (g) + (h)
Q 2 : (a) + (c)

In computing these terms we use the auxiliary fields NKSDOT, containing $\dot{\sigma}_{k-\frac{1}{2}}$, and NSDTMD, containing term (e) of ϒ2. For k=1 these fields contain zeros, in agreement with the vertical boundary conditions.

Note further that the humidity tendency is added immediately to the q(t-1) because nowhere its value is required explicitly.

<5. > Computation of the second part of the tendencies. Again in accordance with the vertical boundary condition, this part can be skipped for level NLEV.

<5.1> Temperature, humidity and the velocity components at present level k are stored in auxiliary fields.

<5.2> The following terms are computed at level k:

pointer	term
NKDDS	$\sum_{j=1}^{k} D_j \cdot \Delta\sigma_j$
NKVPDS	$\sum_{j=1}^{k} (\vec{V}_j \cdot \vec{\nabla} \ell n p_*) \Delta\sigma_j$
NSDTMD	$\sigma_{k+1} \cdot \sum_{j=1}^{N} (\vec{V}_j \cdot \vec{\nabla} \ell n p_*) \Delta\sigma_j - \sum_{j=1}^{k} (\vec{V}_j \cdot \vec{\nabla} \ell n p_*) \Delta\sigma_j$
NKSDOT	$\dot{\sigma}_{k+\frac{1}{2}}$

In the computation of the next level in <4.2> these terms are used again for k → k-1

<5.3> Now we can add the second part of the tendencies:

$\mathcal{F}u$: (b)

$\mathcal{F}v$: (b)

$\mathcal{T}2$: (b) + (d)

Q 2 : (b)

<6.0> Computation of term $\mathcal{R}$(5.3)G in divergence equation. Note that for economy reasons the pointers NR and NTVM1 are the same, so that the field containing $T'_v(t-\Delta t)$ is overwritten here by $\mathcal{R}$.

$\mathcal{R}_k$ is computed for each k. Level-pointer is ILEV. The vertical sums in $\mathcal{R}_k$ are done in an internal loop, with level-pointer ILEV2.

Note that we use the fact that matrix G is upper-triangular.

<7.0> The parts of the tendencies computed in GRMULT are added to the previous time step values. Remember that this happened already to the mixing ratio q so only ℓnp_* and T are left.

<7.1> Add $2\Delta t.\mathcal{P}$ to $\ell np_*(t-\Delta t)$

<7.2> Add $2\Delta t\,\mathcal{T}2$ to $T(t-\Delta t)$

<8.0> Compute p_* from ℓnp_*.

<u><5.10> STATS</u> accumulation of statistics on latitude lines.
In order to save core, it was decided to compute statistics in grid point space, i.e. accumulate them during the scan through the gaussian grid. In that case no spectral fields for the kinetic energy and the surface pressure are required. Moreover both methods give identical results for quadratic terms,

which implies that only the computed kinetic energy is different. Experiments have shown that this is not a serious problem. The following statistics are computed:

RMS-vorticity

$$\sqrt{\overline{\zeta^2}} = \sqrt{\frac{1}{2NLON} \cdot \sum_j w_j \left[\sum_\ell \left(\sum_i \zeta_i \cdot \zeta_i \right) \Delta\sigma_\ell \right]}$$

Here and in the following $\sum_i$ denotes a summation over the NLON points on a gaussian latitude line; $\sum_\ell$ denotes a summation over the vertical levels; and $\sum_j$ denotes a summation over the gaussian latitude lines with w_j being the gaussian weight.

RMS-divergence

$$\sqrt{\overline{D^2}} = \sqrt{\frac{1}{2NLON} \sum_j w_j \left[\sum_\ell \left(\sum_i D_i \cdot D_i \right) \Delta\sigma_\ell \right]}$$

RMS-temperature

$$\sqrt{\overline{T^2}} = \sqrt{\sum_\ell T_o^2 \Delta\sigma_\ell + \frac{1}{2NLON} \sum_j w_j \left[2\sum_\ell \left(T_o \sum_i T_i'\right) \Delta\sigma_\ell + \sum_\ell \left(\sum_i T_i'^2 \right) \Delta\sigma_\ell \right]}$$

Kinetic energy

$$\bar{E} = \frac{1}{2NLON} \cdot \frac{1}{g} \sum_j w_j \left[\sum_\ell \left(\sum_i p_{*_i} E_i \right) \Delta\sigma_\ell \right]$$

Potential + internal energy

$$\overline{P+I} = \frac{1}{2NLON} \cdot \frac{1}{g} \sum_j w_j \left[\sum_i (p_{*_i} \phi_{*_i}) + C_p \left(\sum_i p_{*_i} \right) * \left(\sum_\ell T_o \Delta\sigma_\ell \right) + C_p \sum_\ell \left(\sum_i p_{*_i} T_i' \right) \Delta\sigma_\ell \right]$$

Mean Sea level pressure

$$PMSL = \frac{1}{2NLON} \sum_j w_j \left(\sum_i p_{*_i} \right)$$

Average humidity

$$\bar{Q} = \frac{1}{2NLON} \cdot \frac{1}{g} \sum_j w_j \left[\sum_\ell \left(\sum_i p_{*_i} \cdot q_i\right) \Delta\sigma_\ell\right]$$

The coding of this routine is straightforward and presents no problems.

Subroutine MRFFT2 Fast Fourier Transform, written by C. Temperton, ECMWF (1978). (See also appendix 2.6).
For a documentation we refer to the comments preceding the source code of this routine.

In DYN the following fields are fourier transformed. We give the pointers and the contents of the fields at the moment of calling MRFFT2:

NLPSM1	$\ell n p_*(t-\Delta t)+2\Delta t.\mathcal{P}$	contiguous in blank common
NTM1	$T'(t-\Delta t)+2\Delta t \mathcal{T}_2$	
NQM1	$q(t-\Delta t) + 2\Delta t.Q_2$	
NZM1	$\zeta(t-\Delta t)$	
NDM1	$D(t-\Delta t)$	
NFU	$\mathcal{F}u$	contiguous in blank common
NFV	$\mathcal{F}v$	
NUT	$U.T'$	
NVT	$V.T'$	
NUQ	$U.Q$	
NVQ	$V.Q$	
NE	E	
NR	$\mathcal{R}$	

The last set of 8 contiguous fields is used in LEG to compute the remainder of the adiabatic tendencies and the RHS of the Helmholtz equation.

<2.17> LEG computation of the contribution of each gaussian latitude to the Legendre transform to spectral space.

<1.> set same local constants.

<1.1> Multiply ALP and DALP by the gaussian weight

$$ALP(j) = w_j . P_{m,n}(\mu_j)$$

$$DALP(j) = w_j(1-\mu_j^2) \frac{d}{d\mu} P_{m,n}(\mu_j)$$

<2. > Sections <2. > and <3.> are done in one big loop over the levels.

The following general remarks should be made:

1. The spectral fields are stored diagonally as shown in Fig. 4.1.2.(page 29). The inner vector loops are loops over each diagonal and are therefore of a decreasing length in the present triangular truncation.

2. The following pointers and constants are used in the loops:

   ILEVS: points to a vertical level of a spectral field
   ILEVG: id for a grid point field
   IALP: points to the relevant diagonal in field ALP
   IDALP: idem in field DALP
   ISPEC: idem in prognostic spectral fields
   IMAX: length of a diagonal

The following do-variables are used:

JL: loop over levels
JN: loop over diagonals
JM: loop over points on one diagonal

<2.1> For each level the above pointers are set.

<2.2> For each diagonal the pointers are set and the length IMAX of the diagonal is computed

<2.3> Add to the spectral field of ℓnp_* the following gaussian contribution: (suffix m to a term in square brackets indicates the wave m of the fourier transform of that term at latitude μ_j)

$$w_j \cdot [\ell np_*(t-\Delta t) + 2\Delta t \mathcal{T}_2]_m \; P_{m,n}(\mu_j)$$

Note that no complex arithmetic is used. The real and imaginary part are computed separately.

<2.4> Add to spectral temperature field

$$w_j \; [T'(t-\Delta t)+2\Delta t \, \mathcal{T}_2]_m \; P_{m,n}(\mu_j)$$

$$+ \; w_j \; \{\frac{-im}{1-\mu_j^2} \; [UT']_m \; P_{m,n}(\mu_j)+[VT']_m \; \frac{d}{d\mu}P_{m,n}(\mu_j)\}$$

The second term is term $\mathcal{T}1$, (2.1)L in para.4.3.1.

<2.5> Add to spectral humidity field

$$w_j \cdot [q(t-\Delta t)+2\Delta t Q_2]_m \; P_{m,n}(\mu_j)$$

$$+ \; w_j \{\frac{-im}{1-\mu_j^2} \; [Uq]_m \; P_{m,n}(\mu_j)+[Vq]_m \; \frac{d}{d\mu} \; P_{m,n}(\mu_j)\}$$

The second term is Q_1, (3.1)L.

<2.6> Add to spectral vorticity field:

$$w_j [\zeta \; (t-\Delta t) \;]_m \; \cdot \; P_{m,n}(\mu_j)$$

$$+ \; w_j \; \{ \; \frac{im}{1-\mu_j^2} \; [\mathcal{F}_v]_m \; P_{m,n}(\mu_j)+[\mathcal{F}_u]_m \cdot \frac{d}{d\mu} \; P_{m,n}(\mu_j)\}$$

The second term is Z, (1.1)L.

<2.7> Add to spectral divergence field:

$$w_j \left[D(t-\Delta t) \right]_m \cdot P_{m,n}(\mu_j)$$

<3. > Computation of the RMS of the Helmholtz equation.

We rewrite eq.(5) in para. (4.3.1) as follows:

$$(D_k^{-t})_{m,n} = \sum_{\ell=1}^{NLEV} (A_n^{-1})_{k,\ell} \left[\frac{a^2}{n(n+1)} (D_\ell)_{m,n}(t-\Delta t) + (I_\ell)_{m,n} \right]$$

with $(I_\ell)_{m,n} = \Delta t \{ \frac{a^2}{n(n+1)} (\mathcal{D}_\ell)_{m,n} + (\mathcal{R}_\ell)_{m,n} + \Delta t R (\sum_{i=\ell}^{NLEV} B_{i\ell} \mathcal{T}1_i)_{m,n} \}$

In this section the gaussian contribution to $(I_k)_{m,n}$ is computed:

$$(I_k)_{m,n} = \frac{\Delta t}{a(1-\mu_j^2)} w_j' \{ a(1-\mu_j^2) \cdot [\mathcal{R}_k]_m + \frac{a^2}{n(n+1)} im \quad [\mathcal{F} u_k]_m - \Delta t . im . R \sum_\ell G_{k\ell} [(UT')_\ell]_m$$

$$+ a[E]_m \} \cdot P_{m,n}(\mu_j) - \frac{\Delta t}{a(1-\mu_j^2)} \cdot w_j \cdot \{ \frac{a^2}{n(n+1)} \cdot [\mathcal{F} v_k] -$$

$$\Delta t . R \sum_\ell G_{k\ell} [(VT')_\ell]_m \} \cdot (1-\mu_j^2) \frac{d}{d\mu} P_{m,n}(\mu_j)$$

<3.1> Compute the terms:

$$R \sum_{\ell=k}^{NLEV} G_{k\ell} \ [(UT')_\ell]_m$$

and $R \sum_{\ell=K}^{NLEV} G_{k\ell} \ [(VT')_\ell]_m$

and store these in fields NUT and NVT, i.e. overwrite the original values $[UT]_m$ and $[VT]_m$ which were stored there.

<3.2> Compute the gaussian contributions to the above expression for $(I_k)_{m,n}$. The same pointers as in section <2. > of this routine are used. Real and imaginary parts are computed separately.

4.4 Computations in spectral space: completion of a timestep

After having finished the first gaussian loop, the timestep may be completed in spectral space, because the remaining computations are purely linear: completion of the semi-implicit timestep: subroutine TSTEP, and the horizontal diffusion: subroutine HORDIF.

<2.18> TSTEP Solution of the Helmholtz equation and completion of timestep by adding the semi-implicit part.

<1. > Set some local constants.

<2. > The RHS of the Helmholtz equation has been computed except for the term $\frac{a^2}{n(n+1)} (D_\ell)_{m,n}(t-\Delta t)$. This term is added here.

<3. > In this section the Helmholtz equation (5) is solved for $(\bar{D}^t)_{m,n}$.

Matrices $\underline{\underline{A}}_n^{-1}$ are NLEV*NLEV matrices for each n, stored sequentially for n=0,1,...,NMAX in one-dimensional field BM1.

The term in square brackets

$$\left[\frac{a^2}{n(n+1)} (D_\ell)_{m,n}(t-\Delta t)+\Delta t.\left\{ \frac{a^2}{n(n+1)} (\mathcal{D}_\ell)_{m,n} + (\mathcal{R}_\ell)_{m,n} + \Delta t.R \left(\sum_{i=\ell}^{NLEV} B_{i\ell} \mathcal{T}1_i \right)_{m,n} \right\} \right]$$

is stored in field RH in the same way as other spectral fields: i.e. diagonally for each level. Therefore the solution for spectral component (m,n) at level K is

$$(\bar{D}_k^t)_{m,n} = \sum_{\ell=1}^{NLEV} (A_n^{-1})_{k,\ell} \, (RH_\ell)_{m,n}$$

The following strategy is followed:

The outer loop is a loop over the diagonals of the spectral fields (380-loop).

The next inner loop is a loop over the levels k (369-loop) then for each diagonal and level k, the summation over the levels ℓ is done (368-loop).

Finally this is done for all spectral components on the selected diagonal (365-loop). This is the vector-loop.

The result for each diagonal is temporarily stored in arrays ZR (real part) and ZI (imaginary part) and after completion of the computation on a diagonal, put back in field RH.

At the end of the computations, field RH contains the solution $(\bar{D}^t)_{m,n}$.

<3.1> The pointer ISPEC, pointing at the beginning of a diagonal, is initialised.

<3.2> Auxiliary fields ZR and ZI are set equal to 0.

<3.3> The beginning of a diagonal (ISPEC) and its length (IMAX) are computed.

The pointer ILEVZ, pointing at the levels in auxiliary field ZR and ZI, is initialised.

<3.4> Pointer IBM points at the beginning of the proper row k of matrix $(A_n^{-1})_{k,\ell}$ for the value of n of the first element of the current diagonal:
i.e. part (JN-1)*INL2 points at the beginning of A_n^{-1} for the proper value of n

part (JL-1)*NLEV points at the proper row k within A_n^{-1}

Furthermore, pointer ILEVK, which points at the proper level ℓ in field RH is initialized to zero.

<3.5> Pointer IBM1 points at the element $(A_n^{-1})_{k,\ell}$ for the value of n of the first element of the current diagonal.

Pointer IRH points at the beginning of the current diagonal for level ℓ of field RH.

<3.6> For all spectral coefficients on the present diagonal the value of $(A_n^{-1})_{k,\ell} \cdot (RH_\ell)_{m,n}$

is computed and added.

Because stepping up along the diagonal, increments both m and n by 1 for each step, the pointer IBM1 to field A_n^{-1} must be incremented by NLEV*NLEV=INL2
BM1(IBM1+(JM-1)*INL2)

<3.7> The solution for all levels k and for the current diagonal JN is stored back to its proper place in RH.

<4. > $(\bar{D}^t)_{m,n}$ has now been computed so the semi-implicit parts can be added to complete the timestep.

<4.2> Add to $(\ell n p_*)_{m,n}$:

$$-2\Delta t.\sum_{\ell=1}^{NLEV} \Delta\sigma_\ell \;.\; (\bar{D}^t_\ell)_{m,n} \qquad (4.2)I$$

<4.3> Add to $(T'_k)_{m,n}$:

$$-2\Delta t \sum_{\ell=1}^{NLEV} \tau_{k\ell}(\bar{D}^t_\ell)_{m,n} \qquad (2.5)I$$

<4.4> Compute new divergence

$$(D_k)_{m,n}(t+\Delta t) = 2(\bar{D}^t_k)_{m,n} - (D_k)_{m,n}(t-\Delta t) \qquad (6.1)I$$

<5.7> After the first forward timestep, the timestep length is multiplied by 2. Because matrix A_n is a function of Δt, it has to be computed and inverted again (see documentation of DATINI). Because field AQ contains the factor Δt^2 it is multiplied here by 4, corresponding to a doubling of Δt. The further computations are identical to those in DATINI.

<u><2.19> HORDIF</u> Does linear diffusion on the σ-surfaces of ζ, D, T' and q.

The following diffusion terms are added to the prognostic equations:

$$\zeta : \quad k.(\nabla^4\zeta + \frac{4\zeta}{a^4})$$

$$D : \quad k.(\nabla^4 D + \frac{4D}{a^4})$$

T': $k.\nabla^4 T$

q : $k.\nabla^4 q$

The diffusion operators work on the newly computed ($t+\Delta t$)-values.

In spectral space this corresponds with multiplication of the new values with C:

$$T'_{m,n}, q_{m,n}: \; C = \frac{1}{1+2\Delta t.k \, \dfrac{n^2(n+1)^2}{a^4}}$$

$$\zeta_{m,n}, D_{m,n}: \begin{cases} C = \dfrac{1}{1+2\Delta t.k \, \dfrac{n^2(n+1)^2-4}{a^4}} & \text{for } n \geq 1 \\ C = 1 & \text{for } n = 0 \end{cases}$$

<1.1> Set some local constants.

<1.2> Compute the above constants for each n and store them in arrays: ZDIFTQ for T and q, and ZDIFZD for ζ and D.

<2. > Multiply all spectral components at all levels by C. The vector-loop is as usual the loop over the coefficients on a diagonal.

4.5 Computation of gridpoint values in second scan

We now have available the following spectral fields at timestep ($t+\Delta t$):

$$\zeta_{m,n}, \; D_{m,n}, \; T'_{m,n}, \; q_{m,n}, \; (\ell n p_*)_{m,n}$$

In the second gaussian loop we shall compute from these for each gaussian latitude the following gridpoint values:

$$\zeta,\ D,\ T',\ q,\ U,\ V,\ \ell n p_*,\ \frac{\partial \ell n p_*}{a \partial \lambda},\ (1-\mu^2)\frac{\partial \ell n p_*}{\partial \mu}$$

These data for each gaussian latitude will then be written to disk, in the way described in para. 3.2.

The I/O is controlled by subroutine SCAN2 (page 18), which also calls subroutine GRCALC for Legendre-part, and subroutine MRFFT2 for the Fourier-part of the transformation to gridpoint space.

Before presenting the documentation of GRCALC we present first the mathematical expressions, used in the transformation:

$$\{\zeta, D, T', q, \ell n p_*\}(\mu_j,\lambda_i) = \sum_{m=-M}^{+M} \sum_{n=|m|}^{+M} \{\zeta_{m,n}, D_{m,n}, T'_{m,n}, q_{m,n}, (\ell n p_*)_{m,n}\}. P_{m,n}(\mu_j)e^{im\lambda_i}$$

ECMWF LIBRARY

$$\{U,V\}(\mu_j,\lambda_i) = \sum_{m=-M}^{+M} \sum_{n=|m|}^{+M+1}{}' \{U_{m,n},V_{m,n}\}P_{m,n}(\mu_j)e^{im\lambda_i} \quad \text{(see footnote)}$$

(Note upper limit of second summation !)

with

$$\left.\begin{aligned} U_{m,n} &= -\varepsilon'_{m,n}\,\zeta_{m,n-1} - \frac{a}{n(n+1)}.im.D_{m,n} + \varepsilon'_{m,n+1}\,\zeta_{m,n+1} \\ V_{m,n} &= \varepsilon'_{m,n}\,D_{m,n-1} - \frac{a}{n(n+1)}.im.\zeta_{m,n} - \varepsilon'_{m,n+1}\,D_{m,n+1} \end{aligned}\right\} \qquad (4.5.1)$$

with $\varepsilon'_{m,n} = \frac{a}{n}\left(\frac{n^2-m^2}{4n^2-1}\right)^{\frac{1}{2}}$

Clearly $U_{m,n}$ and $V_{m,n}$ are defined on an extended triangular truncation. However, terms in these expressions which do not exist within the normal triangular truncation of $\zeta_{m,n}$ and $D_{m,n}$ are set equal to 0.

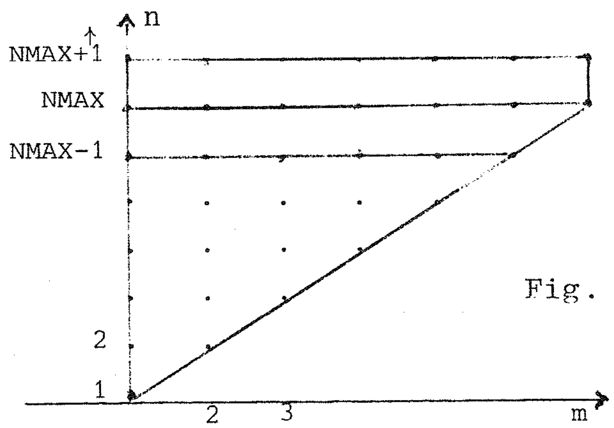


Fig. 4.5.1

Therefore on row NMAX the formulae for $U_{m,n}$ and $V_{m,n}$ are:

$$U_{m,NMAX} = -\varepsilon'_{m,NMAX}\,\zeta_{m,NMAX-1} - \frac{a}{NMAX(NMAX+1)}\ im\ D_{m,NMAX}$$

$$V_{m,NMAX} = \varepsilon'_{m,NMAX}\,D_{m,NMAX-1} - \frac{a}{NMAX(NMAX+1)}\ .im\,\zeta_{m,NMAX}$$

(4.5.2)

Footnote: For a more efficient approach see Appendix 3.

and on row NMAX+1

$$U_{m,NMAX+1} = -\varepsilon'_{m,NMAX+1} \, \zeta_{m,NMAX}$$

$$V_{m,NMAX+1} = \varepsilon'_{m,NMAX+1} \, D_{m,NMAX} \qquad (4.5.3)$$

Finally we have:

$$\frac{\partial \, \ell np_*}{a\partial \lambda} (\mu_j, \lambda_i) = \frac{1}{a} \sum_{m=-M}^{+M} im \sum_{n=|m|}^{M} (\ell np_*)_{m,n} \cdot P_{m,n}(\mu_j) e^{im\lambda_i}$$

$$(1-\mu_j^2) \frac{\partial \, \ell np_*}{a\partial \mu} (\mu_j, \lambda_i) = \frac{1}{a} \sum_{m=-M}^{+M} \sum_{n=|m|}^{M} (\ell np_*)_{m,n} \cdot (1-\mu_j^2) \, P_{m,n}(\mu_j) \, . e^{im\lambda_i}$$

All double sums are split according to:

$$X(\lambda_i, \mu_j) = \sum_{m=-M}^{+M} X^m(\mu_j) e^{im\lambda_i} \qquad \text{(Fourier transform)}$$

with

$$X^m(\mu_j) = \sum_{n=|m|}^{M} X_{m,n} \, P_{m,n}(\mu_j) \qquad \text{(Legendre transform)}$$

or corresponding forms for the derivatives of ℓnp_*.

<2.10> <u>GRCALC</u> 2) Computes the Legendre-transforms from spectral space to gridpoint space.

In order to remove differences in the naming of same arrays, an equivalence statement is included.

---

2) See, however, Appendix 3.

<0.1> Pointers to blank common are computed.

<0.2> The Legendre polynomials and their derivatives for the current latitude line are read from disk. Some constants in COMLEG, relevant to the Legendre transform are set.

<1.1> The part of blank common where the results of GRCALC will be stored, is initialised to 0.

<1.2> Local pointers to the gridpoint fields in blank common are defined. Because the pointers are incremented in steps of 2, separate pointers for the real and imaginary part are defined.

<2. > The Legendre-transforms of the NLEV-level variables T', q, ζ, D, U and V are computed in this section.

The pointer-system in this subroutine is distinctly different from that in the routines LEG and TSTEP.

<2.1> Pointers, set here, point to the following fields

pointer	points at
INDP	diagonal of ALP and EPS=EPL=$\varepsilon'_{m,n}$ ($\varepsilon'_{m,n}$ is stored in an extended triangular field, like ALP)
INDD	diagonal of DALP
IENEXT	Next diagonal of EPS (to fetch values $\varepsilon'_{m,n+1}$ for computation of U and V.
INDEZ	position of (0,1) component of vorticity at current level.

INDR	first element minus 1 at current level of spectral fields ζ, D, T', q
IPREVR, IPREVI	first element minus 1 of previous diagonal. This is to fetch spectral component $\zeta_{m,n-1}$ and $D_{m,n-1}$ in calculation of $U_{m,n}$ and $V_{m,n}$. There are separate pointers for real and imaginary part. These two values are here arbitrarily set to 0. For the first diagonal they are undefined. Later on in <2.5> they will be computed.
INEXTR, INEXTI	first element minus 1 of next diagonal, to fetch $\zeta_{m,n+1}$ and $D_{m,n+1}$. Again there are separate pointers for real and imaginary part.

The following variables are initialized here

IMMAX	length of diagonal (complex words)
IMMAX2	length of diagonal (real words)
IMMXSM	length of diagonal minus 1.

<2.2> Start loop over diagonals

<2.3> Vector loop over spectral components on diagonal. Sum up the Legendre transforms for each m of ζ, D, T and q.

<2.4> Do the same for U and V taking into account the special expressions for the top rows of the extended truncation of U and V.

First subtract the earth-vorticity from $\zeta_{0,1}$, then do the computation on diagonals for all spectral components on rows 1 to NMAX-1 (see Fig. 4.5.1). On these rows the complete equations (eq. 4.5.1) are valid. This is the 241-loop. Realise, however, that diagonal JN=NMAX has no point on rows 1 to NMAX-1.

In section 242 the contributions of row NMAX are added (eq.4.5.2) and finally the contribution of row NMAX+1 (eq.4.5.3). The first diagonal JN=1 has no points on row NMAX+1.

After this, the earth vorticity is added again to the vorticity field.

In this procedure we have missed the left uppermost point of the extended triangular truncation. The contribution of that point will be added in <2.57>.

<2.5> Pointers and variables for the next diagonal are updated.

The first diagonal JN=1 is exceptional in the following aspects:

INDP=INDP-1: because of the shape of the extended triangular truncation, INDP for the next diagonal must be diminished by 1.

IPREVR=INDR
IPREVI=INDI } : IPREVR and IPREVI were temporarily set to zero, although they had no meaning for JN=1. Here they are properly defined for the next diagonal.

<2.57> Here the upper left point of the extended triangular truncation is added.

<2.6> After the previous paragraph the computation for one level has finished. In this section pointers are updated for the next level.

<3. > Computations for:

$$\ell np_*, \frac{\partial \ell np_*}{a\partial \lambda}, (1-\mu^2) \frac{\partial \ell np_*}{a\partial \mu}$$

3.3 Note here that the Legendre-transform of $\frac{\partial \ell np_*}{a\partial \lambda}$ is directly multiplied by i.

3.5 The Legendre-transform of $\frac{\partial \ell np_*}{a\partial \lambda}$ has been multiplied by i already and is here multiplied by m.
Moreover, both derivatives are divided by the radius of the earth a.
The factor $(1-\mu_j^2)$ in the μ-derivative has come into the computation through DALP which is $(1-\mu_j^2) \frac{d}{d\mu} P_{m,n}(\mu_j)$

4.6 Pointers to blank common

Pointers to blank common are computed in subroutine POINTS, relative to the displacements NLINE1(1), NLINE3(1) and NLINE4(1).

The pointers always point to the first element of the field minus 1. Thus, for example, the first element in the field containing the vorticity at timestep t-1 will be: B(NZM1+1).

The list below presents the names of the pointers and the field to which they point and the formula defining the quantity in that field.

pointer	points to field	formula
NF1	ϕ_*	
NUM1	U_{t-1}	
NVM1	V_{t-1}	
NLPSM1	$\ell n p_{*t-1}$	
NTM1	T'_{t-1}	
NQM1	q_{t-1}	
NZM1	ζ_{t-1}	
NDM1	D_{t-1}	
NU	U_t	
NV	V_t	
NLPS	$\ell n p_{*t}$	
NT	T'_t	
NQ	q_t	
NZ	ζ_t	
ND	D_t	
NDPSL	$\frac{\partial \ell n p_*}{a \partial \lambda}$	
NDPSM	$(1-\mu^2)\frac{\partial \ell n p_*}{a \partial \mu}$	
NPS	p_*	
NFU	$\mathcal{F}_u$	(1.2)G
NFV	$\mathcal{F}_v$	(1.2)G
NUT	U.T'	
NVT	V.T'	
NUQ	U.q	
NVQ	V.q	
NE	E	(5.2)G
NTVM1	$T_{v\ t-1}$	
NR	$\mathcal{R}$	(5.3)G
NT2	$\mathcal{T}_2$	(2.4)G

pointer	points to field	formula
NTV	T'_{vt}	
NKDDS	H_1	$\sum_{j=1}^{k} D_j . \Delta\sigma_j$
NKVPDS	H_2	$\sum_{j=1}^{k} (\vec{V}_j . \vec{\nabla} \ell n p_*) \Delta\sigma_j$
NNDDS	H_3	$\sum_{j=1}^{NLEV} D_j . \Delta\sigma_j$
NNVPDS	H_4	$\sum_{j=1}^{NLEV} (\vec{V}_j . \vec{\nabla} \ell n p_*) \Delta\sigma_j$
NKAD	H_5	$\sum_{j=1}^{k} A_{kj} . D_j$
NKAVP	H_6	$\sum_{j=1}^{k} A_{kj} (\vec{V}_j . \vec{\nabla} \ell n p_*)$
NKSDOT	H_7	$\dot{\sigma}_{k+\frac{1}{2}}$
NSDTMD	H_8	$\sigma_{k+\frac{1}{2}} . \sum_{j=1}^{NLEV} (\vec{V}_j . \vec{\nabla} \ell n p_*) \Delta\sigma_j$
		$- \sum_{j=1}^{k} (\vec{V}_j . \vec{\nabla} \ell n p_*) \Delta\sigma_j$
NVKDP	H_9	$\vec{V}_k . \nabla \ell n p_*$
NTKM1	H_{10}	T'_{k-1}
NQKM1	H_{11}	q_{k-1}
NUKM1	H_{12}	U_{k-1}
NUKM1	H_{13}	V_{k-1}

CHAPTER 5 - THE START DATA SET (SDS)

5.1 Common block COMSDS

The use and structure of the SDS is identical in the gridpoint and spectral model. Therefore we refer to HB Chapter 7 for a description of the SDS. In details the content of the common block COMSDS differs however. In particular constants, related to the filtering near the pole have been removed. A list of COMSDS variables is found in section 9.13.

5.2 Creation of an SDS; subroutine MAKESD

An SDS is created by:

```
CALL MAKESD (KIN,KOUT,KSDS)
with  KIN : input channel
      KOUT: print output channel
      KSDS: channel to which SDS is written
```

<U.28> MAKESD Creates start data set.

<1.1> Default values of COMSDS-variables are set according to the list in section 9.13.

<1.2> Namelist STARTD is read and printed, to change default values.

<1.3> File name and pass words of initial data file are read from channel KIN

For example:

```
  ID=EWAB3,SN=DSET15
  ↑
  1
T21DATASET
↑
1
```

<1.4> If a private disk is to be mounted (NLMNT=.TRUE.) then request parameter set and volume name are read from channel KIN.
For example:

```
  DSET15  PA003Y
  ↑       ↑
  1       11

*SN=DSET15
↑
1
```

<1.5> If the job is run on public disk (NLMNT=.FALSE.), then private disk parameters are blanked and request-parameter is set accordingly.

<1.6> COMSDS is written to channel KSDS.

CHAPTER 6 - THE INITIAL DATA SET (IDS)

6.1 Structure of the IDS

The IDS contains three common blocks COMBAS, COMHKP and COMMAP as the first three records, followed by 1 record for each latitude row of data, ordered from north to south. Two subroutines have been provided, one to initialise the constants in COMMAP, the second one to create the IDS.

6.2 Common block COMMAP and subroutine MAPFAC

Common block COMMAP is initialised by calling:
CALL MAPFAC
A list of values of COMMAP is presented in section 9.4.

<1.11> MAPFAC Sets up constants in COMMAP.

Via DATA-statements some constants, related to the vertical scheme are input:

ZSIG : values of σ at full levels of 9-layer GFDL-model

ZSIGH: values of σ at half levels of 9-layer GFDL-model, starting at top $\sigma_{\frac{1}{2}}=0$ and ending at bottom $\sigma_{9\frac{1}{2}}=1$.

G : integration matrix of hydrostatic equation. The present matrix contains 9*9=81 values, organised in rows from top to bottom. This matrix is based on the following method of integration of the hydrostatic equation

$$\phi_k = R. \sum_{\ell=k}^{NLEV} G_{k\ell} T_\ell + \phi^*$$

with

$$\phi_k = \tfrac{1}{2}(\phi_{k+\frac{1}{2}} + \phi_{k-\frac{1}{2}})$$

and $$\phi_{k+\frac{1}{2}} = R \sum_{\ell=k+1}^{NLEV} T_\ell \cdot \ell n \frac{\sigma_{\ell+\frac{1}{2}}}{\sigma_{\ell-\frac{1}{2}}} + \phi_*$$

Here ϕ_k, T_k, σ_k are geopotential, temperature and σ at level k, and ϕ_* is the geopotential height of the orography.

This leads to:

$$\phi_k = \phi_* + \tfrac{1}{2} RT_k \; \ell n \frac{\sigma_{k+\frac{1}{2}}}{\sigma_{k-\frac{1}{2}}} + R \sum_{\ell=k+1}^{NLEV} T_\ell \; \ell n \frac{\sigma_{\ell+\frac{1}{2}}}{\sigma_{\ell-\frac{1}{2}}}$$

and therefore:

$$G_{k\ell} = 0 \qquad \ell < k$$

$$G_{kk} = \tfrac{1}{2} \; \ell n \frac{\sigma_{k+\frac{1}{2}}}{\sigma_{k-\frac{1}{2}}} \qquad k \neq 1$$

$$= \ell n \frac{\sigma_{1\frac{1}{2}}}{\sigma_1} \qquad k = 1$$

$$G_{k\ell} = \ell n \frac{\sigma_{\ell+\frac{1}{2}}}{\sigma_{\ell-\frac{1}{2}}} \qquad \ell > k$$

Matrix G can be replaced by any other upper-triangular matrix via namelist HYDRO.

The matrix G, corresponding to model described in Hoskins and Simmons, 1975, with 9 GFDL-levels, is available on a permanent file:

```
ATTACH, TAPE9, G, ID=EWAW3.
```

It can be read by inserting in <1.6> of MAPFAC the statement:

```
READ(9)G
```

<1.1> Some physical constants are initialized.

EZ is the absolute vorticity of the earth. The meaning of the other constants is obvious.

<1.2> Initialization of some σ-level dependent arrays. Note that array element SIGKPH(K) contains $\sigma_{k+\frac{1}{2}}$, rather than $\sigma_{k-\frac{1}{2}}$ as is the case in the gridpoint model.

<1.3> The following arrays, which depend on spectral counter n are initialised here:

SQ : $\dfrac{n(n+1)}{a^2}$

RSQ : $\dfrac{a^2}{n(n+1)}$

XM : m

Note that loopcounter j runs from 1 to NMAX for n = 0, NMAX-1.

<1.4> The following arrays, depending on spectral counter (m,n) are initialised here.

EPL : $\dfrac{a}{n} \left[\dfrac{(n-m)\ (n+m)}{(2n+1)(2n-1)} \right]^{\frac{1}{2}}$ (extended triangular)

DEL : $a \cdot \dfrac{m}{n(n+1)}$ (normal triangular)

Both arrays are first computed in rhomboidal truncation with column-wise storage and then reordered to triangular truncation with diagonal storage.

<1.5> Some gaussian latitude dependent arrays are filled here.

CALL GAUAW computes the gaussian weights w_j and latitudes $\mu_j = \sin \phi_j$ and returns them in arrays ZSI and ZW.

Because of the symmetry between both hemispheres only N-hemisphere values are stored of the following quantities:

SIT	$\sin \phi_j = \mu_j$
W	w_j
CS	$(\cos\phi_j)^2 = 1-\mu_j^2$
ALAT	ϕ_j in degrees

<1.6> Namelist HYDRO, containing if desirable the matrix G, is read and printed. Then matrix A is computed so as to produce an energy conserving vertical scheme:

$$A_{\ell k} = G_{k\ell} \cdot \frac{\Delta\sigma_k}{\Delta\sigma_\ell}$$

(See Baede and Jarraud, 1978)

<2. > COMMAP is printed on channel NPRINT

6.3 Creation of the IDS; subroutine MAKEDT

The IDS is created by

CALL MAKEDT (KMAP,KDTIN,KRD,KWRITE,KDTOUT)

where

KMAP = channel number of map factor data set (because subroutine MAPFAC is called from MAKEDT, this channel number is unused)

KDTIN = channel number of input grid point data

KRD = card input channel

KWRITE = printer output channel

KDTOUT = channel to which the IDS is written.

The user must supply on channel KDTIN a data set containing $\phi_*, U_1, U_2, \ldots, U_{NLEV}, V_1, \ldots, V_{NLEV}, \ell n p_*, T_1, \ldots, T_{NLEV},$

$$q_1,\ldots,q_{NLEV},\zeta_1,\ldots,\zeta_{NLEV},D_1,\ldots,D_{NLEV},\frac{\partial \ell n p_*}{a\partial\lambda},(1-\mu^2)\frac{\partial \ell n p_*}{a\partial\mu}$$

where $X_k = X_{k,1}, \ldots, X_{k,NLON}$

for each latitude row. There must be 1 record for each latitude row, ordered sequentially from North to South. At the end of each group of NLON points two extra words must be allowed. The data set must be written with unformatted Fortran WRITE statement.

The subroutine initialises common blocks COMBAS and COMHKP (see chapter 9 for a listing of these common blocks). It calls MAPFAC to initialise COMMAP and finally writes common blocks + gridpoint data to channel KDTOUT.

<u><U.29> MAKEDT</u> creates initial data set on channel KDTOUT.

<1.1> COMBAS is initialised according to the list in chapter 9.2.

<1.2> COMHKP is initialised according to the list in chapter 9.12.

<1.3> Via namelist INIDAT some COMHKP-constants may be reset.

<1.4> MAPFAC is called to initialise COMMAP (see list in section 9.4).

<1.5> The three common blocks are written to channel KDTOUT.

<2. > For each line of latitude the gridpoint data are read from KDTIN, as specified above.

<2.2> Using buffered I/O the data are written to KDTOUT.

<2.3> After the last write, the status of KDTOUT is checked before returning control to the calling program.

CHAPTER 7 - RUNNING THE MODEL

7.1 Source and object libraries

On the CYBER-175 almost all subroutines are found on the following program library.

SPECTRSOURCE,ID=EWAB3, CY=1, MR=1.

This file has the proper OLDPL-format and can serve as input to the UPDATE system of the CYBER. At present this file contains the T21 9 layer model. All common blocks are in COMDECK's.

Corresponding to this program library, there is an object library on file.

SPECTROBJ,ID=EWAB3,CY=1, MR=1

The random-access I/O routines and the permanent file handling routines are to be found in the following object library.

ECMWF, ID=EWP3, CY=1, MR=1.

Dummy versions of the routines MOUNT and DSMOUNT, are to be found in the following object library:

GEMINILIB,ID=EWJC3, CY=2, MR=1.

7.2 Creating the data sets SDS and IDS

The following card deck creates the initial data set and the start data set for a T21 run on a private disk. The original gridpoint data are on T21GAUSSDATA.

```
EWAB3,STPAK.     ***    CREATES T21DATASET AND T21SDS ***
MOUNT,SN=DSET15,VSN=PA003Y.
REQUEST,TAPE7,*SN=DSET15.
REQUEST,TAPE8,*SN=DSET15.
FTN,I=0.
ATTACH,TAPE2,T21GAUSSDATA,ID=EWAB3.
ATTACH,LIB1,SPECTROBJ,ID=EWAB3,MR=1.
ATTACH,LIB2,ECMWF,ID=EWB3,MR=1.
LIBRARY,LIB1.
LDSET,LIB2.
LGO.
CATALOG,TAPE7,T21DATASET,ID=EWAB3.
CATALOG,TAPE8,T21SDS,ID=EWAB3.
7/8/9
      PROGRAM SET(TAPE1,TAPE2,INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
     *   TAPE7,TAPE8,TAPE9)
      CALL MAKEDT(1,2,5,6,7)
      CALL MAKESD(5,6,8)
      END
7/8/9
 $INTDAT NOREC=32,NLON=64,NLEV=9,MMAX=22,NMAX=22,NSPEC=253,
 $
 $HYDRO
 $
 $STARTD NWITME(1)=0,NSTOP=5,DTIME=2400,
      NLMNT=.TRUE.,
 $
T21DATASET
ID=EWAB3,SN=DSET15.
*SN=DSET15.
DSET15     PA003Y
6/7/8/9
```

This data set was set up for a 5 timestep run without writing of history files. Now we want to make a 10-day run on this same data set, with timesmoothing and diffusion, and write a history file after each 24 hours.

The following deck runs the job:

(See page 69.)

```
EWAB3,T10000,STBTG.    **** RUNNING T21 MODEL ***
FIN,OPT=2,L=0.
MOUNT,SN=DSET15,VSN=PA003Y.
ATTACH,TAPE30,T21SDS,ID=EWAB3,SN=DSET15.
ATTACH,LIB1,SPECTROBJ,ID=EWAB3.
ATTACH,LIB2,ECMWF,ID=EWP3,MR=1.
ATTACH,LIB3,GEMINILIB,ID=EWJC3,MR=1.
LIBRARY,LIB1.
LDSET,LIB=LIB2/LIB3.
LGO.  ***ONSWITCH TO TERMINATE***
COMMENT.*ONSWITCH TO TERMINATE***
AUDIT,ID=EWAB3,SN=DSET15.
7/8/9
      PROGRAM SPECTR(INPUT=400,TAPE5=INPUT,OUTPUT=400,TAPE6=OUTPUT
     *,TAPE7=OUTPUT
     *,TAPE10=0,TAPE11=0,TAPE12=0,TAPE13,TAPE20=0,TAPE21=65,TAPE22=65,
     *TAPE30)
      COMMON B(32000)
      CALL MASTER
      STOP
      END
7/8/9
 $REST
      NLRES=.FALSE.,
      NREC=1,
 $
T21 EPS=0.06,ZVIP=1,ORTG.G
14/6/78 A.BAEDE ECMWF


 $NEWRUN
      NWTIME(1)=36,72,108,144,180,216,252,288,324,360,
      NSTOP=362,
      NWPIR=1,
      EPS=0.06,
      DIF=6.0E+16,
 $
 $SETMP T0=229.304,209.45,218.147,237.6,256.647,268.71,277.454,
           283.131,285.666,
 $
6/7/8/9
```

CHAPTER 8 - NAMELISTS

8.1 Summary

Preset constants can be changed by the non-standard Fortran utility NAMELIST. There are altogether six such NAMELISTS, three for the creation of the datasets and three for running the program. Below we give a summary of these namelists, the subroutines where they are found and the constants which can be changed by them.

The following namelists are available

Namelist	In subroutine	Documented in section	on page
INIDAT	MAKEDT	8.2	70
HYDRO	MAPFAC	8.3	71
STARTD	MAKESD	8.4	71
REST	MODIFY	8.5	72
NEWRUN	DATINI	8.6	73
SEIMP	DATINI	8.7	74

8.2 Namelist INIDAT

This namelist has the following form:

```
NAMELIST/INIDAT/NOREC,NLON,NLEV,NCOM,
      NMAX,NMAX,NSPEC
```

It is defined in subroutine MAKEDT and allows some COMHKP-constants to be changed.

constant	type	Common block	where initially defined	initial value
NOREC	Int	COMHKP	MAKEDT	1
NLON	Int	COMHKP	MAKEDT	1
NLEV	Int	COMHKP	MAKEDT	1
NCOM	Int	COMHKP	MAKEDT	3
NMAX	Int	COMHKP	MAKEDT	1
NMAX	Int	COMHKP	MAKEDT	1
NSPEC	Int	COMHKP	MAKEDT	1

8.3 Namelist HYDRO

It has the following form

NAMELIST/HYDRO/G

It is defined in MAPFAC and allows the hydrostatic integration matrix G to be changed.

G is defined by

$$\phi_k = \phi_* + R \sum_{\ell=1}^{NLEV} G_{k\ell} \, T_\ell$$

It is initialised in MAPFAC by a DATA-statement containing the values of G as specified above in section 6.2.

constant	type	Common block	where initially defined	initial value
G	real array	COMMAP	MAPFAC	see section 6.2

8.4 Namelist STARTD

It has the following form

NAMELIST/STARTD/
NDATA, NWTIME, NSTOP, DTIME, EPS,
NLMNT, DIF

It is defined in MAKESD and allows some COMSDS-constants to be changed

constant	type	Common block	where initially defined	initial value
NDATA	Int	COMSDS	MAKESD	20
NWTIME	Int array	COMSDS	MAKESD	200*0
NSTOP	Int	COMSDS	MAKESD	0
DTIME	real	-	MAKESD	0.
EPS	real	COMSDS	MAKESD	0.
NLMNT	logical	COMSDS	MAKESD	.FALSE.
DIF	real	COMSDS	MAKESD	0.

8.5 Namelist REST

It has the following form

NAMELIST/REST/NLRES,NREC

It is defined in MODIFY and allows two COMBAS-constants to be changed, which identify whether the present run is an initial or a restart run and from which SDS-record the run starts.

constant	type	Common block	where initially defined	initial value
NLRES	logical	COMBAS	MAKEDT	.FALSE.
NREC	Int	COMBAS	MAKEDT	1

8.6 Namelist NEWRUN

It has the following form

```
NAMELIST/NEWRUN
* NLEDGE    NONLIN,  NOUT,    NPRINT,  NIN,     NPUNCH,
* NRUN,     MXDUMP,  NADUMP,  NPDUMP,  NVDUMP,  NWKIO,
* NWKIN,    NWKOUT,  MXBLDM,  NPBLDM,  NBDUMP,  NSTOP,
* NWTIME,   NWPTR,   DTIME,   EPS,     DIF,
* NLCHED,   NLHEAD,  NLOMT1,  NLOMT2,  NLOMT3,  NLREPT,
* NLSTAT,   NTPLEG
```

It is defined in DATINI and allows redefinition of some basic and housekeeping parameters.

constant	type	Common block	where initially defined	initial value	
NLEDGE	Int	COMBAS	MAKEDT, BASIC	30	
NONLIN	Int	COMBAS	MAKEDT, BASIC	1	
NOUT	Int	COMBAS	MAKEDT, BASIC	6	
NPRINT	Int	COMBAS	MAKEDT, BASIC	6	
NIN	Int	COMBAS	MAKEDT, BASIC	5	
NPUNCH	Int	COMBAS	MAKEDT, BASIC	7	
NRUN	Int	COMBAS	MAKEDT, BASIC	1	
MXDUMP	Int	COMDDP	BASIC	10	
NADUMP	integer array	COMDDP	BASIC	20 * 0	
NPDUMP	int.array	COMDDP	BASIC	20 * 0	
NVDUMP	int.array	COMDDP	BASIC	20 * 0	
NWKIO	Int	COMIOC	DATINI	12	
NWKIN	Int	COMIOC	DATINI	10	
NWKOUT	Int	COMIOC	DATINI	11	
MXBLDM	Int	COMDBC	DATINI	20	
NPBLDM	int.array	COMDBC	DATINI	20 * 0	
NBDUMP	int.array	COMDBC	DATINI	20 * 0	
NSTOP	Int	COMSDS	MAKESD	0	1)
NWTIME	int.array	COMSDS	MAKESD	20 * 0	1)
NWPTR	Int	COMSDS	MAKESD	-1	

contd.

constant	type	Common block	where initially defined	initial value	
DTIME	real	-	DATINI	0.	
EPS	real	COMSDS	MAKESD	0.	1)
DIF	real	COMSDS	MAKESD	0.	1)
NLCHED	logical	COMDDP	BASIC	.FALSE.	
NLHEAD	logical array	COMDDP	BASIC	9*.FALSE.	
NLOMT1	logical array	COMDDP	BASIC	50*.FALSE.	
NLOMT2	logical array	COMDDP	BASIC	100*.FALSE.	
NLOMT3	logical array	COMDDP	BASIC	50*.FALSE.	
NLREPT	logical	COMDDP	BASIC	.FALSE.	
NLSTAT	logical	COMSTA	DATINI	.TRUE.	
NTPLEG	Int	COMIOC	DATINI	13	

1) This value may have been redefined by namelist STARTD

8.7 Namelist SEIMP

It has the following form

NAMELIST/SEIMP/Tϕ

It is defined in DATINI and allows the reference temperature profile Tϕ to be changed.

constant	type	Common block	where initially defined	initial value
Tϕ	real array	COMIMP	DATINI	in each level 300.0

CHAPTER 9 - COMMON BLOCKS

9.1 Summary

The following common blocks are used in the model.

Olympus number	name	description	documented in section	on page
C1.1	COMBAS	basic system parameters	9.2	76-77
C1.9	COMDDP	development and diagnostic parameters	9.3	78-79
C3.1	COMMAP	map factors and physical constants	9.4	80-81
C3.3	COMSTA	statistics	9.5	82-83
C3.4	COMSPE	spectral fields	9.6	84
C3.5	COMLEG	Legendre transform variables	9.7	85
C3.6	COMIMP	Implicit timestep variables	9.8	86
C3.7	COMFFT	FFT-parameters and workfield	9.9	87
C4.1	COMIOC	Housekeeping parameters	9.10	88-89
C4.2	COMNDX	Random access file index arrays	9.11	90
C4.3	COMHKP	Data file description	9.12	91-94
C4.4	COMSDS	Start data set record	9.13	95-96
C4.5	COMGRD	Pointers to blank common	9.14	97
C5.1	COMDBC	Blank common dump parameters	9.15	98
C9.0	(blank)	Grid point fields	2.7	14

9.2 C1.1 COMBAS   Basic system parameters

variable	meaning	where defined	initial value	where redefined *)	new value
ALTIME	allocated job CPU-time (secs)	MASTER	CALL JOBTIM (ALTIME)	-	-
CPTIME	CPU-time used so far (secs)	BASIC	0.0	-	-
NLEDGE	channel number for start data set	BASIC	30	DATINI (P)	input to namelist NEWRUN
NLEND	.TRUE. if run is to be terminated	BASIC	.FALSE.	STEPON (P)	.TRUE. after last time step
NLRES	.TRUE. if run is a restart	BASIC	.FALSE.	MODIFY (P)	input to namelist REST
NONLIN	channel number for online I/O	BASIC	1		input to namelist NEWRUN
NOUT	current output channel	BASIC	NPRINT	DATINI (P)	input to namelist NEWRUN
NPRINT	channel number for printed output	BASIC	6	DATINI (P)	input to namelist NEWRUN
NREAD	channel number for card input	BASIC	5	DATINI (P)	input to namelist NEWRUN
NREC	current start data record number	BASIC	1	MODIFY (P)	input to namelist REST
NRESUM	resume from record on this channel	BASIC	NLEDGE	-	-
NSTEP	current step number	BASIC	0	(1) DATCOM (2) STEPON	(1) step number for restart (2) NSTEP=NSTEP+1

* (P) means : possibly
  (A) means : always

9.2 C1.1 (contd.)

variable	meaning	where defined	initial value	where redefined	new value
STIME	start time (secs)	MASTER	CALL SECOND (STIME)	-	-
LABEL1(5) LABEL2(5) LABEL3(5) LABEL4(5)	labels used to describe run	BASIC	blanks	LABRUN (A)	card input set by user
LABEL5(5) LABEL6(5)	labels available to programmer	BASIC	blanks	-	-
LABEL7(5) LABEL8(5)	labels reserved for system use	BASIC	blanks	-	-
NDIARY	channel for diary	BASIC	NPUNCH	-	-
NIN	current input channel	BASIC	NREAD	DATINI (P)	input to namelist NEWRUN
NPUNCH	channel for punched card output	BASIC	7	-	-
NRUN	maximum number of steps	BASIC	1	DATINI (P)	input to namelist NEWRUN

9.3 C1.9 COMDDP development and diagnostic parameters

variable	meaning	where defined	initial value	where redefined	new value
MAXDUM	maximum dimension of dump arrays	BASIC	20	-	-
MXDUMP	actual dimension of dump arrays	BASIC	10	DATINI (P)	input to namelist $NEWRUN
NADUMP(20)	codes for array dumps	BASIC	0	DATINI (P)	" "
NCLASS	most recent subroutine class reported	BASIC	0	-	" "
NPDUMP(20)	codes for dumping points	BASIC	0	DATINI (P)	input to namelist $NEWRUN
NPOINT	most recent point reported	BASIC	0	-	-
NSUB	most recent subroutine reported	BASIC	0	-	-
NVDUMP(20)	codes for scalar variable dumps	BASIC	0	DATINI (P)	input to namelist $NEWRUN
NLCHED	.TRUE. if class 0 report heads required	BASIC	.FALSE.	DATINI (P)	input to namelist $NEWRUN
NLHEAD(9)	.TRUE. if class 1-9 report heads required	BASIC	.FALSE.	"	" "

9.3 C1.9 COMDDP (contd.)

variable	meaning	where defined	initial value	where redefined	new value
NLOMT1(50)	.TRUE. if class 1 subroutine is to be omitted	BASIC	.FALSE.	DATINI (P)	input to namelist $NEWRUN
NLOMT2(100)	.TRUE. if class 2 subroutine is to be omitted	BASIC	.FALSE.	"	" "
NLOMT3(50)	.TRUE. if class 3 subroutine is to be omitted	BASIC	.FALSE.	"	" "
NLREPT	.TRUE. if any reports required	BASIC	.FALSE.	"	" "

9.4 C3.1 COMMAP (map factors and physical constants)

variable	meaning	value (and dimension)
AE	radius of the earth	6.371E+6 (m)
RA	1/radius of the earth	1/AE (m^{-1})
GA	acceleration of earth gravitational field	9.81 ($m.sec^{-2}$)
WW	angular velocity of the earth	7.292E-5 ($rad.sec^{-1}$)
EZ	absolute vorticity of the earth	WW/ $\sqrt{0.375}$ m^{-1}
CP	$C_p = R/\kappa$	R/AKAP
R	gas constant	287.04
AKAP	κ	0.2857143
RGA	1/acceleration of earth gravitational field	1/GA $sec^2 m^{-1}$
SIG(NLEV)	σ_k-values of full levels from top to bottom (σ_1 to σ_{NLEV}) NLEV= number of vertical levels	provided by user
SIGKPH (NLEV)	$\sigma_{k+\frac{1}{2}}$ -values at half levels from $\sigma_{1\frac{1}{2}}$ to $\sigma_{NLEV+\frac{1}{2}}$	provided by user
DSIGMA	$\sigma_{k+\frac{1}{2}} - \sigma_{k-\frac{1}{2}} = \Delta\sigma_k$	SIGKPH(J)-SIGKPH(J-1)
R2DSIG (NLEV)	$\frac{1}{2\Delta\sigma_k}$	0.5/DSIGMA(J)
SQ(NMAX)	$\frac{n(n+1)}{a^2}$, n being the meridional spectral index and a: the radius of the earth NMAX : highest meridional index number+1	(m^{-2})
RSQ(NMAX)	$\frac{a^2}{n(n+1)}$ (n > 0), 0 (n=0)	(m^2)
XM(MMAX)	m	

9.4 C3.1 (contd.)

variable	meaning	value (and dimension)
DEL(NM)	$a.\frac{m}{n(n+1)}$ m being the zonal spectral index defined on triangular truncation with $NM = \frac{NMAX.(NMAX+1)}{2}$	(m)
EPL(NMP)	$\frac{a}{n} \cdot \left[\frac{(n-m)\ (n+m)}{(2n+1)\ (2n-1)}\right]^{\frac{1}{2}}$ defined an extended triang-truncation with $NMP = \frac{NMAX.(NMAX+3)}{2}$	(m)
$SIT(\frac{NOREC}{2})$	$\mu_j = \sin\phi_j$, defined on northern hemisphere only from pole to equator NOREC being the number of lat. lines on the sphere ϕ_j = gaussian latitude	computed in subroutine GAUAW
$W(\frac{NOREC}{2})$	w_j, gaussian weights for one hemisphere only	"
$CS(\frac{NOREC}{2})$	$1-\mu_j^2 = \cos^2\phi_j$	
$RCS(\frac{NOREC}{2})$	$\frac{1}{1-\mu_j^2} = \frac{1}{\cos^2\phi_j}$	
$ALAT(\frac{NOREC}{2})$	ϕ_j in degrees	
G(NLEV*NLEV)	Integration matrix of hydrostatic equation	provided by user
A(NLEV*NLEV)	Integration matrix of conversion term	$A_{\ell k} = G_{k\ell} \cdot \frac{\Delta\sigma_k}{\Delta\sigma_\ell}$

9.5 C3.3 COMSTA (statistics parameters) see section 4.3.2, p. 40.

variable	meaning
RMSZ	RMS relative vorticity $\sqrt{\overline{\zeta^2}}$, the bar indicating an average over the whole atmosphere
RMSD	RMS divergence $\sqrt{\overline{D^2}}$
	$\overline{(\ )} = \int_{Sphere} \int_{\sigma} d\sigma \ d\lambda \ d\mu$
RMST	RMS temperature $\sqrt{\overline{T^2}}$
STPS	mean surface pressure $\bar{p}_*$
STQ	mean moisture content $\frac{1}{g}\ \overline{p_* q}$
STPE	mean potential + internal energy $\frac{1}{g}\ \overline{p_*(\phi_* + C_p . T)}$
STKE	mean kinetic energy $\frac{1}{g}\ \overline{p_* . \frac{u^2+v^2}{2}}$
STTE	mean total energy STPE+STKE
NLSTAT	if .TRUE., compute statistics NLSTAT is set .TRUE. in DATINI
VZ2(NROW)	zonal vector to collect squared vorticity during scan
D2(NROW)	id. for squared divergence
T2(NROW)	id. for squared temperature
PE(NROW)	id. for $p_* E_{kin}$

9.5 C3.3 (contd.)

variable	meaning
PT(NROW)	id. for p_*T
PQ(NROW)	id. for p_*q
TP(NROW)	id. for T'
PS(NROW)	id. for p_*
PFI(NROW)	id. for $p_*\phi_*$

9.6 C3.4 COMSPE (spectral fields)

variable	meaning
VZ(NLEV*NSPEC)	Spectral components of absolute vorticity. NLEV: number of vertical levels; NSPEC: number of spectral components in triangular truncation.
D (NLEV*NSPEC)	Spectral components of divergence
T(NLEV*NSPEC)	Spectral components of $T'=T-T_o$: the temperature deviation from the reference temperature T_o
Q(NLEV*NSPEC)	Spectral components of moisture mixing ratio
ALPS(NSPEC)	Spectral components of $\ell n p_*$, p_* being the surface pressure
RH(NLEV*NSPEC)	Spectral components of RHS of Helmholtz equation

9.7 C3.5 COMLEG Legendre transform quantities

The values of the variables in COMLEG are defined for each latitude in both gaussian scans in LINEMS and GRCALC

variable	meaning
ALP(NMP)	$P_{m,n}(\mu_j)$, Associated Legendre Polynomial at current latitude μ_j. Defined on extended triangular truncation with $NMP = \frac{NMAX.(NMAX+3)}{2}$
DALP(NM)	$(1-\mu_j^2)\frac{d}{d\mu}P_{m,n}(\mu_j)$, defined on triangular truncation with $NM = \frac{NMAX.(NMAX+1)}{2}$
CSJ	μ_j at present row: CS(NROW)
RCSJ	$1/\mu_j$ at present row: RCS(NROW)
WEIGHT	w_j at present row : W(NROW)

ECMWF LIBRARY

9.8. C3.6 COMIMP (Implicit timestep variables)

variable	meaning	where defined	initial value	where redefined	new value
TO(NLEV)	reference temperature profile	DATINI	300.0 in each vertical level	DATINI (P)	provide by user in namelist SEIMP
BM1(NLEV*NLEV*NMAX)	matrices $\underline{\underline{A}}_n^{-1}$	DATINI	depends on TO and vertical levels	TSTEP	After first forward timestep Δt is doubled and $\underline{\underline{A}}_n^{-1}$ are recomputed
TAU(NLEV*NLEV)	matrix $\underline{\underline{\tau}}$	DATINI	id	-	-
AQ(NLEV*NLEV)	matrix $\underline{\underline{G}}.\underline{\underline{\tau}} + \underline{T_o}.\underline{\Delta\sigma}$	DATINI	id	TSTEP	AQ=AQ*4. After first forward timestep
STODS	$\sum_{k=1}^{NLEV} T_o(k)\Delta\sigma_k$	DATINI	id	-	-
STO2DS	$\sum_{k=1}^{NLEV} T_o^2(k)\Delta\sigma_k$	DATINI	id	-	-

9.9 C3.7 COMFFT (FFT-parameters, functions and workfield)

variable	meaning	where defined	initial value
TRIG(NLON+1)	trigonometric functions	DATINI	CALL RFTSET(TRIG,NLON)
WORK(NCRAY*(NLON+2))	work field for subroutine MRFFT2	-	-
NTR1A	Number of groups of NCRAY transforms in LINEMS of previous timestep prognostic variables	DATINI	depends on NLEV and NCRAY
NRST1A	Corresponding remainder < NCRAY	id	id
NTR1B	Number of groups of NCRAY transforms in LINEMS of non-linear quantities	id	id
NRST1B	Corresponding remainder < NCRAY	id	id
NTR2	Number of groups of NCRAY transforms in SCAN2	id	id
NRST2	Corresponding remainder < NCRAY	id	id
NCRAY	Length of CRAY vector register	id	64

9.10 C4.1 COMIOC (House keeping parameters)

variable	meaning	where defined	initial value	where redefined	new value
NLINE1(2)	Displacement from the start of blank common of the two input B-buffers	STARTN	NLINE1(1) = 0 NLINE1(2)=NBFLNB	SCAN1 (A)	both are swapped
NLINE2(2)	Displacement from the start of blank common of the two output B-buffers	STARTN	NLINE2(1)=NBFLNB*2 NLINE2(2)=NBFLNB*3	SCAN1 (A)	both are swapped
NLINE3(2)	Displacement from the start of blank common of the two I/O A-buffers	STARTN	NLINE3(1)=NBFLNB*4 NLINE3(2)=NLINE3(1)+NBFLNA	1) SCAN1 (A) 2) SCAN2 (A)	both are swapped both are swapped
NWKIO	Channel number of I/O workfile A	DATINI	12	DATINI (P)	input from namelist NEWRUN
NWKIN	Channel number of input workfile B	DATINI	10	1) DATINI (P) 2) STEPON (A)	1) input from namelist NEWRUN 2) swap NWKIN and NWKOUT at end of timestep
NWKOUT	Channel number of output workfile B	DATINI	11	1) DATINI (P) 2) STEPON (A)	1) input from namelist NEWRUN 2) swap NWKIN and NWKOUT at end of timestep
NROW	Current row number	STARTN	1	STARTN SCAN1 STEPON SCAN2	} NROW=NROW+1 for first scan 1 for second scan NROW=NROW+1 for second scan

9.10 C4.1 COMIOC (contd.)

variable	meaning	where defined	initial value	where redefined	new value
NLNBUF	total length of I/O buffers + auxiliary fields in blank common	DATINI	NBFLNB*4+NBFLNA*2 +NBAUXL	–	–
MAXROW	number of latitude rows	INITAL (init.run) or RESUME (restart)	MAXROW=NOREC	–	–
NORS	(not used in spectral model)				
NSTART	step number at start of run	INITAL (init.run) or RESUME (restart)	NSTART=NSTEP	–	–
NWRITE	(stepnumber-1) of next write-up time (NB. it is the T-1data which is saved and this has step number NWRITE!)	SDS	NWTIME(NWPTR) where NWTIME is the array of write-up times, and NWPTR points to the next element of NWTIME	1) SDS (P)	1) NWRITE-NWTIME (NWPTR+1) at write-up times 2) NWRITE=-1 after last write-up time
				2) DATINI (P)	3) NWRITE-NWTIME (NWPTR) if NWPTR or NWTIME have been modified by namelist NEWRUN 4) NWRITE=-1 if NEWRUN input gives write-up time after last timestep
NTPLEG	channel number of file containing Legendre polynomials and derivatives	DATINI	13	DATINI (P)	Input from namelist NEWRUN

9.11 C4.2 COMNDX random access file index arrays

variable	meaning	where defined or redefined
NDXIN (200)	index array for input work file on channel NWKIN	file opened in INITAL (initial run) or RESUME (restart)
NDXOUT(200)	index array for output work file on channel NWKOUT	file opened in INITAL (initial run) or RESUME (restart)
NDXDTA(200)	index array for data file on channel NDATA	file opened in DATCOM and closed in INITAL (initial data for initial run), RESUME (initial data for restart run) or LINEMS (write-up time)
NDXM1A(200)	index array for first To+1 data file on channel NM1A	initial run - file opened and closed in INITAL restart - file opened in DATCOM and closed in RESUME write-up time - file opened in DATCOM and closed in LINEMS
NDXM1B(200)	index array for second To+1 data file on channel NM1B	initial run - file opened and closed in INITAL

9.12 C4.3 COMHKP Data file description

variable	meaning	where defined	initial value	where redefined *	new value
NOREC	number of latitude lines	DATCOM	common block COMHKP is read from the start of the initial/restart data set	–	–
NLON	number of longitude points	DATCOM	"	–	–
NLEV	number of vertical levels	DATCOM	"	–	–
LREC	Length of data for 1 time level	DATCOM	"	–	–
NCOM	number of common blocks at start of data	DATCOM	"	–	–
NM1A	channel number of first To+1 data file	DATCOM	dummy value (0) read in	1) DATINI(A) 2) LINEMS(P)	1) NM1A=21 2) swap NM1A and NM1B at write-up times, so first To+1 data file becomes second, and vice-versa
NM1ACY	cycle number of first To+1	DATCOM		LINEMS(P)	NM1ACY and NM1BCY swapped at write-up time
NM1AFN(4)	file name of first To+1 data file	DATCOM	1) initial run - dummy name (blanks) read in 2) restart - real name read in	1) INITAL (A) 3) LINEMS (P)	1) file name generated by call FILENM(NM1AFN,0) 3) swap NM1AFN(J) and NM1BFN(J) at write-up time

9.12 C4.3 (contd.)

variable	meaning	where defined	initial value	where redefined*	new value
NM1APW(11)	passwords of first To+1 data file (including ID= ...)	DATCOM	1) initial run - dummy passwords (blanks) read in 2) restart - real passwords read in	1) INITAL (A) 3) LINEMS (P)	1) NM1APW(J) = NDTPW(J), so that To+1 data file has same ID etc. as initial data 3) swap NM1APW(J) and NM1BPW(J) at write-up time
NM1B	channel number of second To+1 data file	DATCOM	dummy value 0 read in	1) DATINI (A) 2) LINEMS (P)	1) NM1B = 22 2) swap NM1A and NM1B at write-up times
NM1BCY	cycle number of second To+1 data file	DATCOM	read from COMHKP at start of initial/restart data set	LINEMS (P)	NM1ACY and NM1BCY swapped at write-up time
NM1BFN(4)	file name of second To+1 data file	DATCOM	1) initial run - dummy name (blanks) read in 2) restart - real name read in	1) INITAL (A) 3) LINEMS (P)	1) file name generated by CALL FØLENM(NM1BFN,0) 3) swap NM1AFN(J) and NM1BFN(J) at write-up time

9.12 C4.3 (contd.)

variable	meaning	where defined	initial value	where redefined *	new value
NM1BPW(11)	password of second To+1 data file	DATCOM	1) initial run - dummy passwords (blanks) read in	1) INITAL (A)	1) NM1BPW(J)=NDTPW(J) so that second To+1 data file has same ID etc. as initial data
			2) restart - real passwords read in	3) LINEMS (P)	3) swap NM1APW(J) and NM1BPW(J) at write-up times

9.12 C4.3 COMHKP (contd.)

variable	meaning	where defined	initial value	where redefined	new value
NLONP2	number of points along lat.line + 2	MAKEDT	NLON+2	-	-
MMAX	highest zonal wave number+1	MAKEDT	1	MAKEDT	by user via namelist INIDAT
NMAX	highest meridional wave number + 1	MAKEDT	1	MAKEDT	by user via namelist INIDAT
NSPEC	number of spectral coefficients within truncation	MAKEDT	1	MAKEDT	by user via namelist INIDAT
NSPECP	number of spectral coefficients within extended truncation	MAKEDT	NSPEC+MMAX	-	-
LDATA	total number of words of 1 record of initial data file	MAKEDT	LREC+NBPHYS+NBGRID	-	-
NBGRID	total number of words reserved for the two ℓnp* derivatives	MAKEDT	2*NLONP2	-	-
NBPHYS	total number of words reserved for physics-fields	MAKEDT	0	-	-
NBAUXL	total number of words in blank common reserved for auxiliary fields	MAKEDT	NLONP2* (10*NLEV+1) +NLON*13	-	-
NBFLNA	total number of words in blank common for one buffer for file A	MAKEDT	LREC+NBGRID	-	-
NBFLNB	total number of words in blank common for one buffer for file B	MAKEDT	LREC+NBPHYS	-	-

9.13 C4.4 COMSDS (Start data set record)

All values in this common block are preset in MAKESD. Most values can be reset either via namelist STARTD in MAKESD or in DATINI via namelist NEWRUN

variable	meaning	preset value	reset value
NRECRD	Start data set record number	1	–
NDATA	Channel number of initial/restart data	20	namelist STARTD
NDTCY	Cycle number of initial/restart data (NDTCY=0 means: highest available cycle number)	0	namelist STARTD
NWTIME(200)	Write-up time step numbers	200 * 0	namelists STARTD and NEWRUN
NWPTR	Pointers to current element of NWTIME	-1	(1) MAKESD: if (NWTIME(1)>0)NWPTR=1 (2) namelist NEWRUN
NSTOP	Step number of last step	0	(1) namelist STARTD. (2) If NSTOP >0 then NSTOP=NSTOP+1 (3) Namelist NEWRUN
TWODT	2 * timestep (secs)	-	TWODT=2 * DTIME, where DTIME=0 initially, but may be reset in STARTD and NEWRUN
EPS	time filter constant	0.	namelists STARTD and NEWRUN
DIF	horizontal diffusion constant	0.	namelists STARTD and NEWRUN
NSW	switch flag	0.	in STEPON: (1) NSW=1 (2) if (NWRITE+1.EQ.NSTEP)NSW=2

9.13 C4.4 COMSDS (contd.)

variable	meaning	preset value	reset variable
NDTFN(4)	name of initial/restart data file	blanks	input from channel KIN in format 4 A 10
NDTPW(11)	passwords of initial/restart data file	blanks	input from channel KIN in format 11 A 10. This must contain at least "ID=EWXXX." and may also contain SN, VSN, MR etc. information.
NDREQ(10)	request call parameters	"*PF."	if NLMNT=.TRUE., NDREQ must be provided via channel KIN in format 11 A 10
NDMTSN NDMTVS	set name for private disk VSN of private disk	blank blank	if NLMNT=.TRUE. both must be provided via channel KIN in format 2AW
NLMNT	if .TRUE. : private disk if .FALSE. : public disk	.FALSE.	namelist STARTD

9.14 C4.5 COMGRD (pointers to blank common)

The pointers to the grid point fields in blank common are all defined in subroutine POINTS, relative to the starting points NLINE1(1) etc.

Their meaning and definition can be found from Fig. 2.7.1 and table on pp.58-59.

9.15 C5.1 COMDBC (blank common dump parameters)

The facilities for dumping parts of blank common have not yet been implemented in the spectral model. The common block is provided, however, for later implementation.

variable	meaning	where defined	initial value	where redefined*	new value
NPBLIM(20)	codes for dumping points	DATINI	0	DATINI(P)	input to namelist $ NEWRUN
MXBLIM(20)	dimension of dump arrays	DATINI	20	DATINI(P)	input to namelist $ NEWRUN
NBDUMP(20)	codes for variable dumps	DATINI	0	DATINI(P)	input to namelist $ NEWRUN

CHAPTER 10 - SUBROUTINES

In this chapter we present a table of all subroutines used in the model with references to their source and documentation (section 10.1).

Some subroutines are identical to those used in the grid point model. We refer to the HB for diagrams of these routines. In section 10.2 we present diagrams of some subroutines which play an important role in the organisation of the model. Typical computational routines such as GRMULT, LEG and GRCALC are not presented here however in diagram, because we felt that a diagram would not add new information to that obtained from the written documentation in previous chapters.

A few new simple routines, especially written for the spectral model, are briefly documented in Appendix 1. Appendix 2 gives some information on mathematical routines, which we obtained from different sources.

10.1 Survey of all subroutines

	name	Olympus number	source[1)]	where[1)] documented	comments
A)	ALTER	–	E	NS	only on CDC/CYBER
	ARRAYS	5.3	S	–	not implemented
	ATTACH	–	E	NS	only on CDC/CYBER
	AUXVAL	–	S	–	dummy
B)	BASIC	0.1	S	HB(p.87)	
	BLDUMP	–	S	App.1	
	BLINES	U.3	S	–	
	BSSLZR	–	S	App.2	
C)	CATALOG	–	E	NS	only on CDC/CYBER
	CHECKR	U.32	S	–	
	CHKBF	3.4	S	–	
	CLEAR	–	S	–	dummy
	CLIST	5.2	S	–	not implemented
	COMHES	–	S	App.2	
	COMLR	–	S	App.2	
	CONVAD	–	S	–	dummy
	COPYBC	5.6	S	–	only on CDC/CYBER
	COPYI	U.24	S	–	
	COPYR	U.23	S	–	
	COTROL	0.3	S	HB(p.92)	
	CRASH	Z.1	S	–	not used
D)	DATA	–	S	–	dummy
	DATINI	1.10	S	pp. 26	
	DATCOM	1.9	S	HB(p.103)	
	DAYTIM	–	S	–	
	DMPBCM	5.4	S	–	not implemented
	DSMOUNT	–	G	–	not implemented; dummy
	DUMBL2	5.5	S	–	not implemented
	DUMCOM	U.27	S	–	not implemented
	DYN	2.15	S	pp. 33	
E)	ENDRUN	–	S	–	dummy
	EXPERT	–	S	–	dummy
	EXTEND	–	E	NS	only on CDC/CYBER

	name	Olympus number	source[1)]	where[1)] documented	comments
	FILENM	2.14	S	HB(p.104)	
G)	GAUAW	-	S	App.2	
	GRCALC	2.10	S	pp.	
	GRMULT	2.16	S	pp.	
H)	HARRAY	U.10	S	-	
	HORDIF	2.19	S	pp.	
	HVAR	U.6	S	-	
I)	IARRAY	U.9	S	-	
	INITAL	1.6	S	pp.	
	IVAR	U.5	S	-	
J)	JOBTIM	U.17	S	-	
L)	LABRUN	1.1	S	-	
	LARRAY	U.19	S	-	
	LEG	2.17	S	pp.	
	LINEMS	2.13	S	pp.	
	LVAR	U.7	S	-	
M)	MAKEDT	U.29	S	pp.	
	MAKESD	U.28	S	pp.	
	MAPFAC	1.11	S	pp.	
	MASTER	0.0	S	HB(p.86)	
	MESAGE	U.1	S	-	
	MINV	-	S	App.2	
	MODIFY	0.2	S	HB(p.88)	
	MOUNT	-	G	-	not implemented; dummy
	MRFFT2	-	S	App.2	
O)	OUTPUT	-	S	App.1	
P)	PAGE	U.2	S	-	
	PHCS	-	S	App.2	
	PHYS	2.20	S	-	dummy
	POINTS	2.9	S	pp.	
	PRESET	1.3	S	-	dummy
Q)	QREIG	-	S	App.2	
R)	RARRAY	U.8	S	-	
	RARRAY2	U.20	S	-	

name	Olympus number	source[1)]	where[1)] documented	comments
READBF	3.2	S	–	
READF	2.5	S	BC	only on CDC/CYBER
READR	U.30	S	–	
RECOVR	–	G	–	dummy; replaces CYBER system routine
REORD1	–	S	App.1	
REORD2	–	S	App.1	
REPTHD	U.11	S	–	
REQUEST	–	E	NS	only on CDC/CYBER
RESETH	U.16	S	–	
RESETI	U.15	S	–	
RESETL	U.18	S	–	
RESETR	U.14	S	–	
RESUME	1.7	S	pp.22	
RETURN	–	E	NS	only on CDC/CYBER
RFTSET	–	S	App.2	
RUNTIM	U.12	S	–	
RVAR	U.4	S	–	
S) SCALEI	U.21	S	–	
SCALER	U.22	S	–	
SCAN1	2.8	S	pp.17	
SCAN2	2.12	S	pp.18	
SDS	1.8	S	HB(p.89)	
SIGNI	U.25	S	–	
SIGNR	U.26	S	–	
STATS	5.10	S	pp.40	
START	–	S	–	dummy
STARTN	2.7	S	pp.17	
STEPON	2.1	S	pp.16	
T) TESEND	4.1	S	–	
TIMCPU	–	S	–	written in COMPASS
TIMESM	2.17	S	pp.32	
TRANSR	2.2	S	BC	only on CDC/CYBER
TRANSW	2.3	S	BC	only on CDC/CYBER
TSTEP	2.18	S	pp.46	
V) VPASS2	–	S	App.2	

	name	Olympus number	source[1]	where[1] documented	comments
W)	WRITEBF	3.3	S	-	
	WRITEF	2.4	S	BC	only on CDC/CYBER
	WRITER	U.31	S	-	

1. The following abbreviations are used:

E: ECMWF, ID=EWP3 object library

S: SPECTRSOURCE, ID=EWAB3 source library
SPECTROBJ, ID=EWAB3 object library

G: GEMINISOURCE, ID=EWJC3 source library
GEMINILIB, ID=EWJC3 object library

NS: N.Storer, 10 Jan. 1976, ECMWF internal documentation "New permanent file function subroutines"

HB: J. Haseler and D. Burridge, Documentation for the ECMWF grid point model, ECMWF Internal Report 9, May 1977.

BC: D. Burridge and J. Charlewood, Random Access I/O Routines which can proceed in parallel with CPU processing. ECMWF Internal Documentation, 7 June 1976.

If no further reference is given, page and appendix numbers refer to the present report.

10.2 Diagrams

Diagram 1 - Outline Flow Diagram

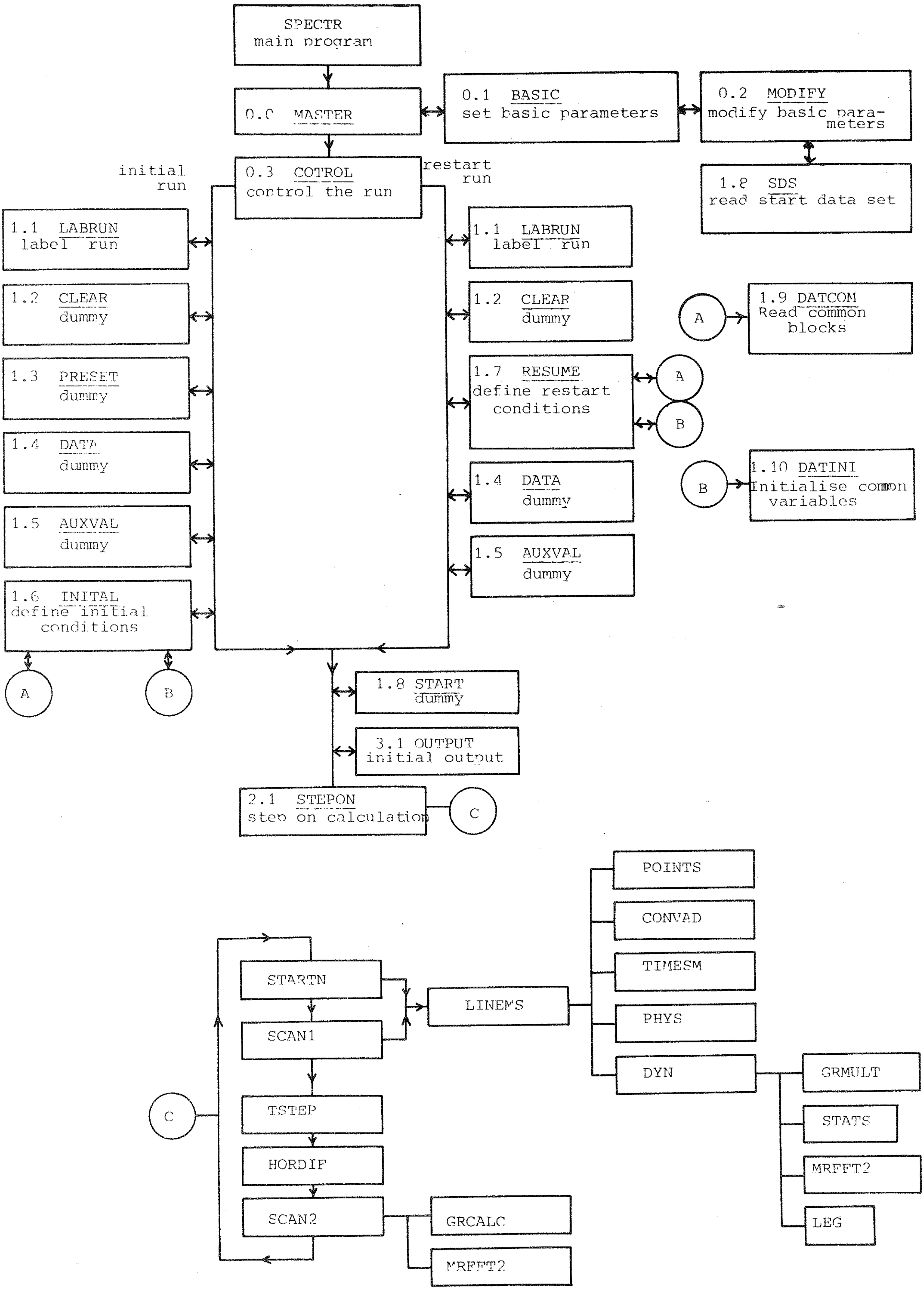
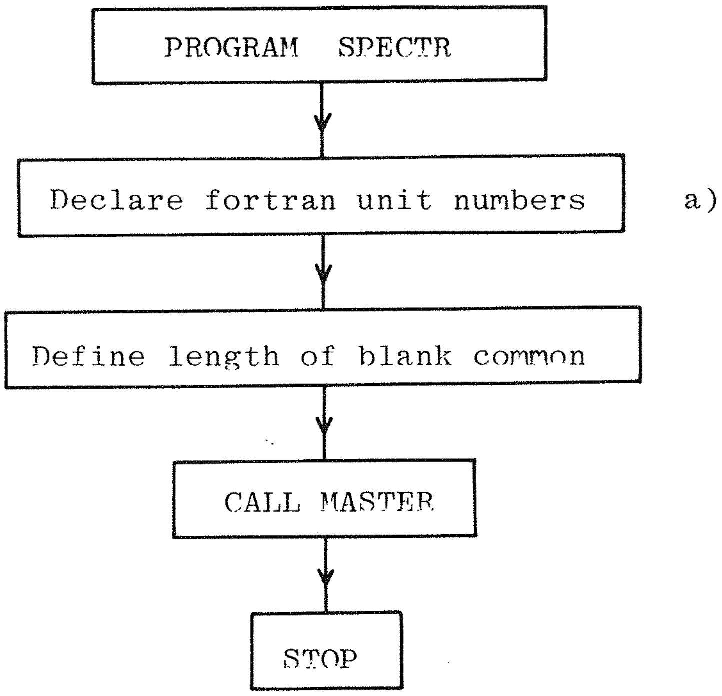


Diagram 2 - Main Program



a) Fortran unit numbers:

TAPE5 = INPUT

TAPE6 = OUTPUT

TAPE7 = OUTPUT (originally: punched card output)

TAPE10 }
TAPE11 } temporary work files
TAPE12 }

TAPE13 tamporary file for Legendre polynomials + derivatives

TAPE20 initial/restart data, then writeup data

TAPE21 }
TAPE22 } T+1 data files

TAPE30 start data set

Diagram 3   1.6   Subroutine INITAL

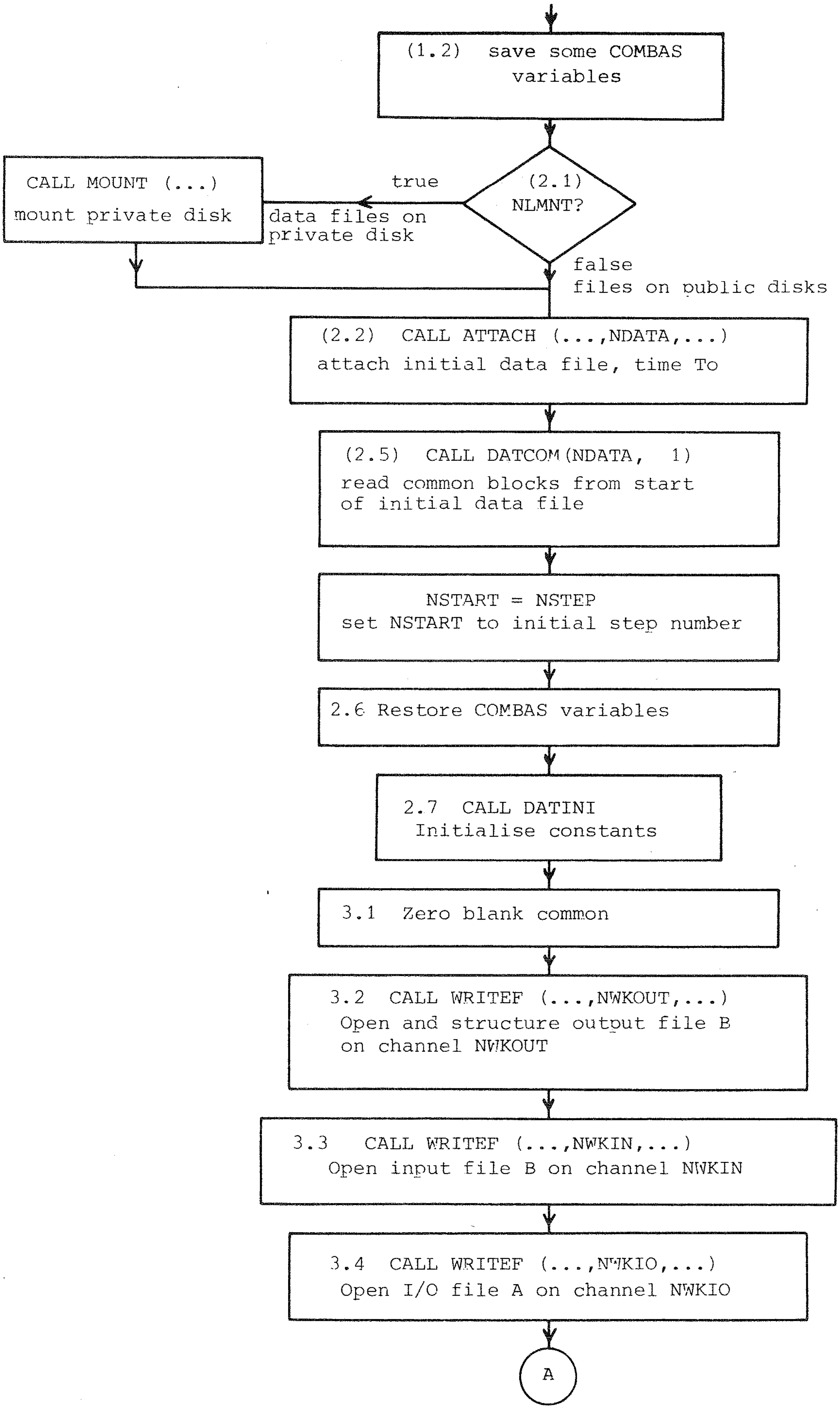
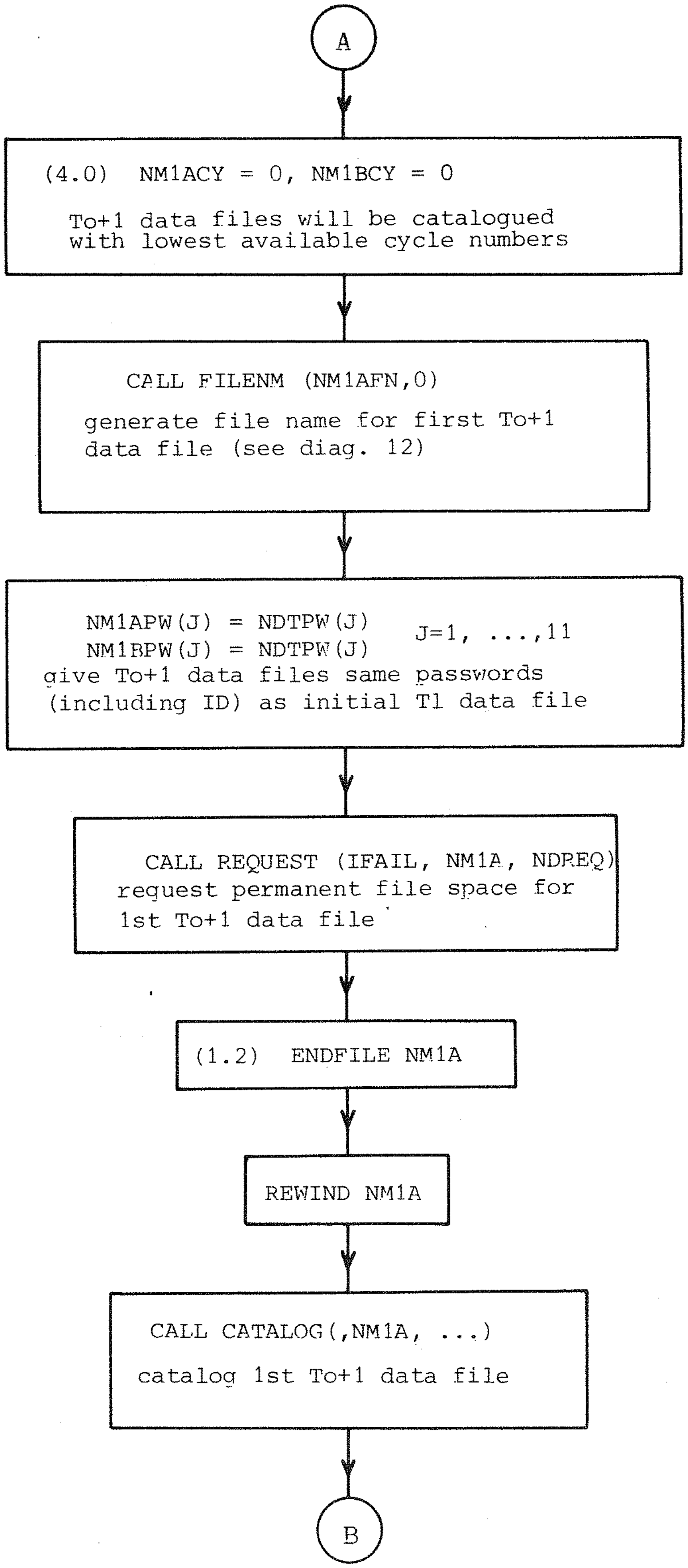


Diagram 3 (contd.1)



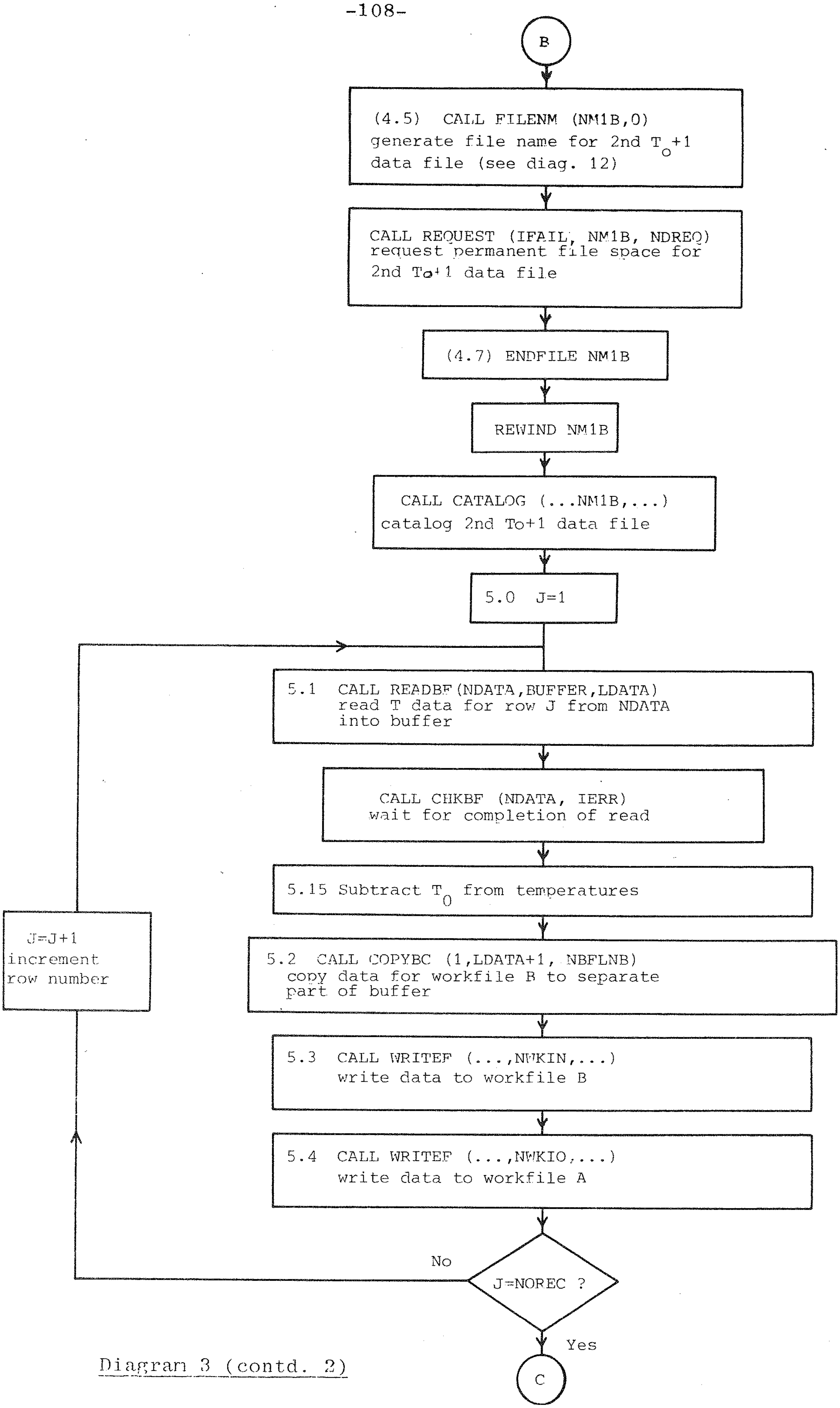
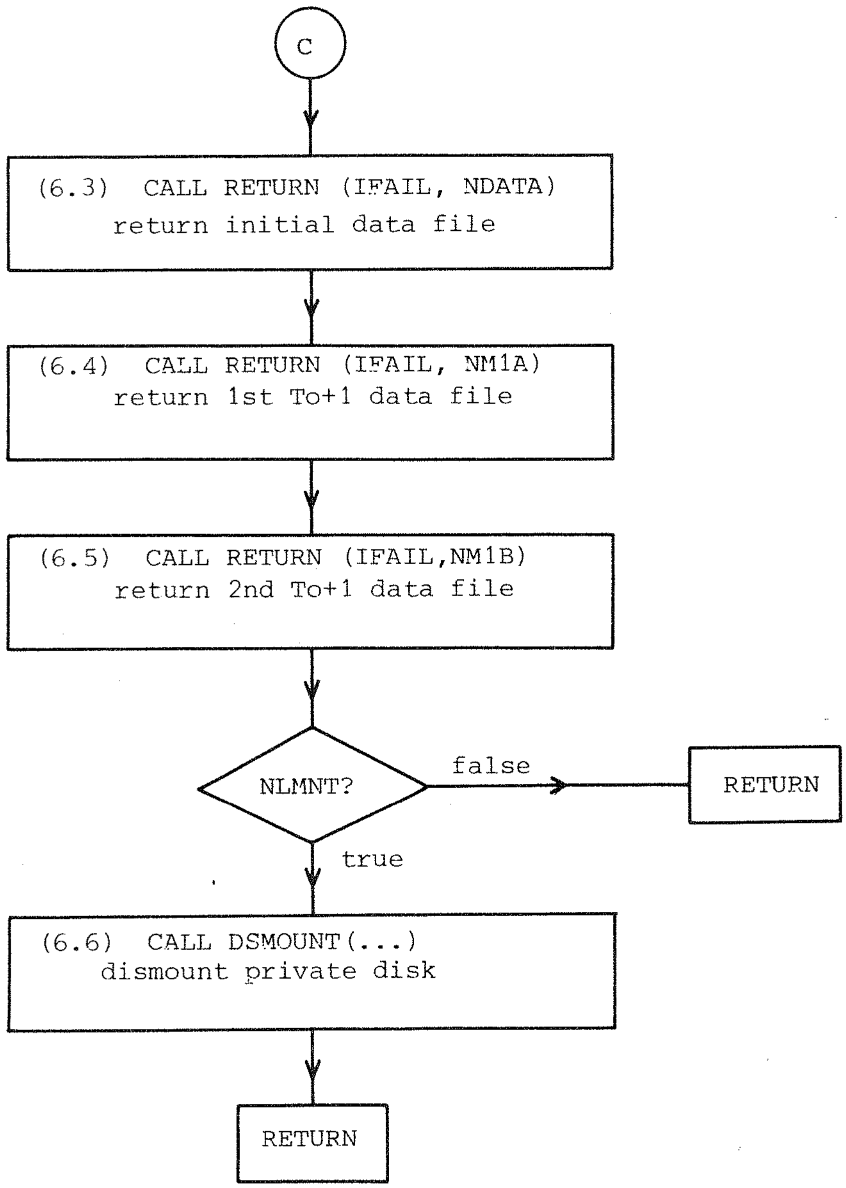


Diagram 3 (contd. 2)

Diagram 3 (contd. 3)



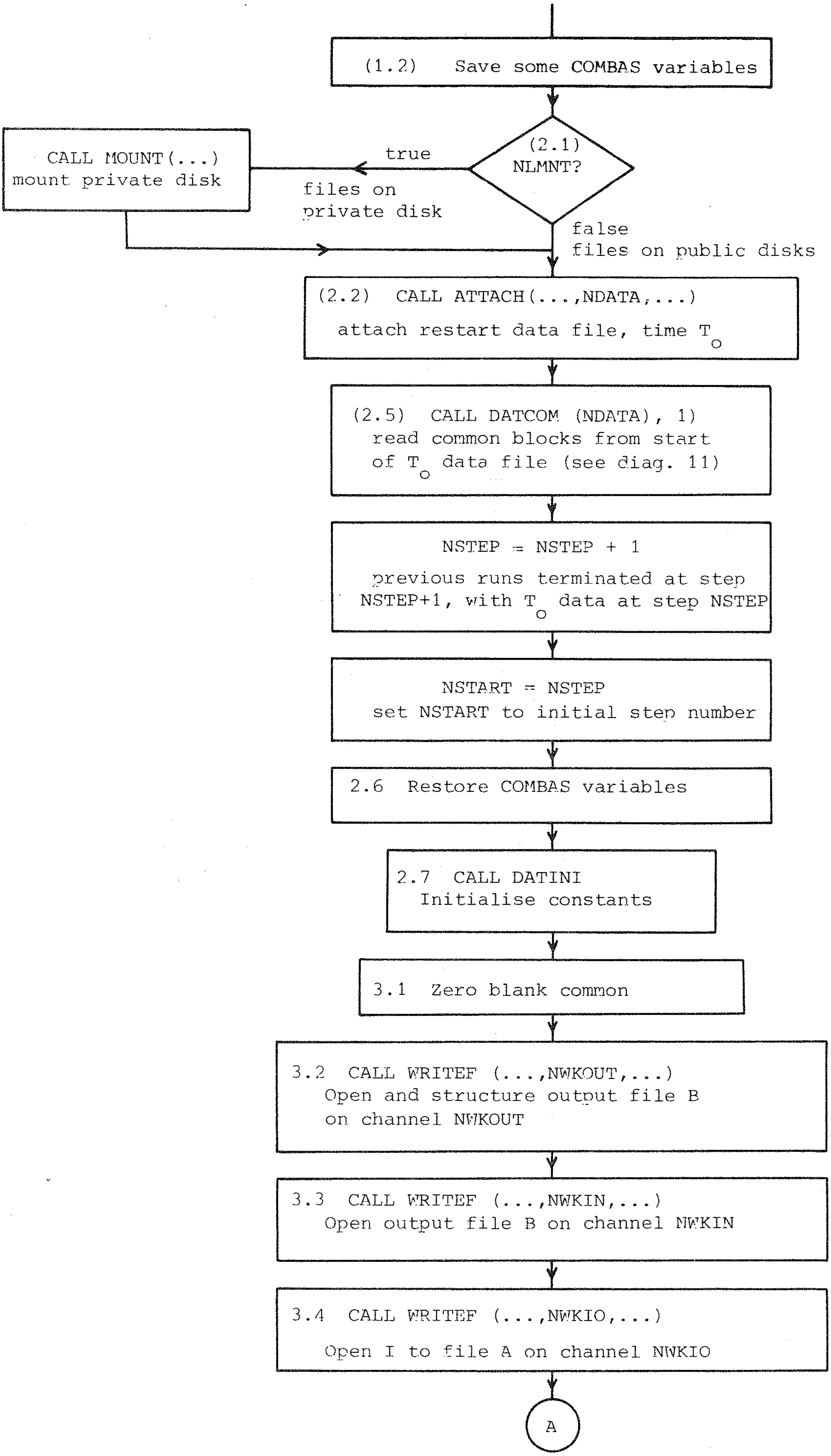
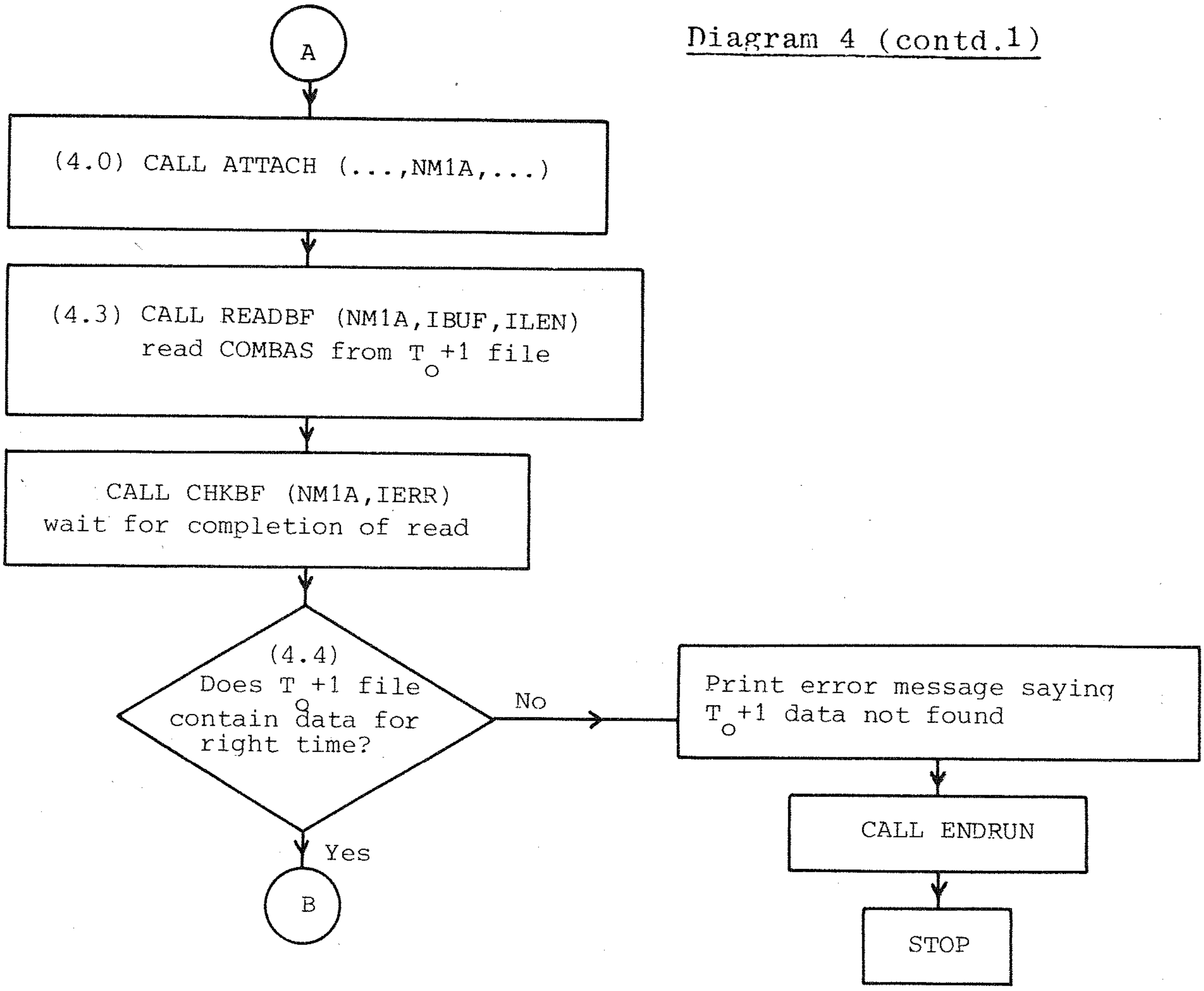
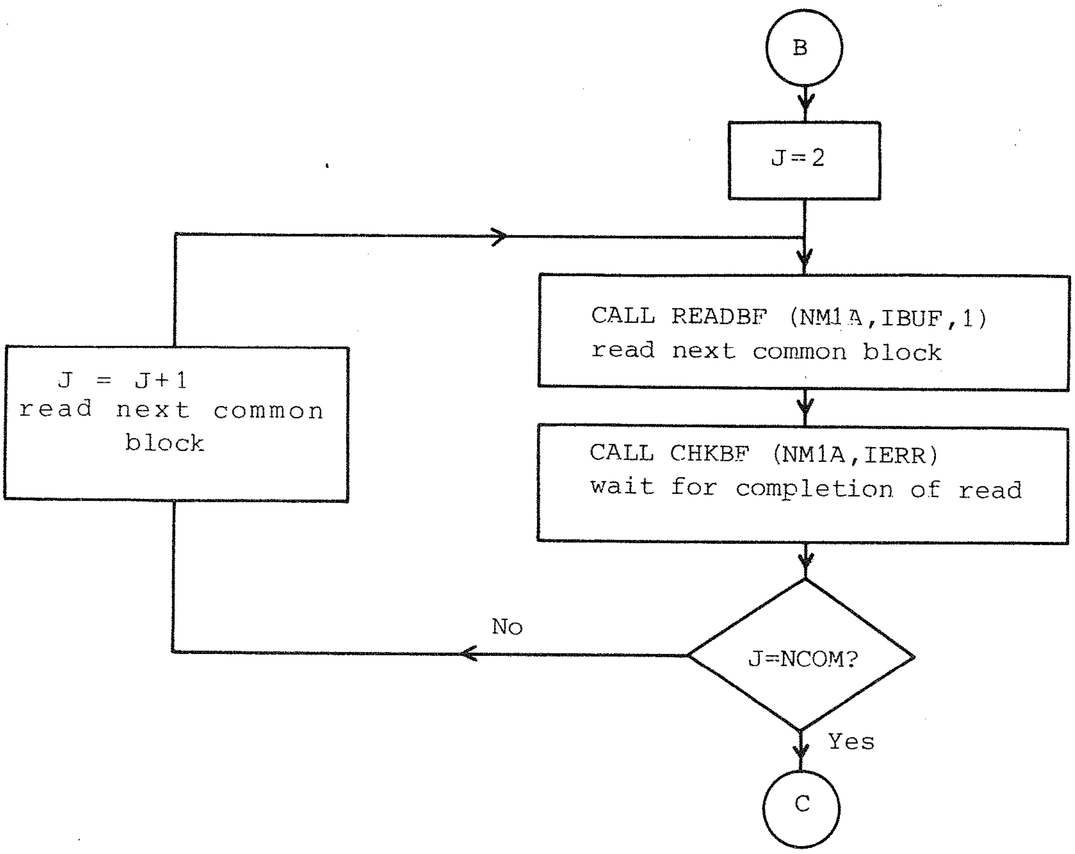


Diagram 4 <1.7> Subroutine RESUME

Diagram 4 (contd.1)





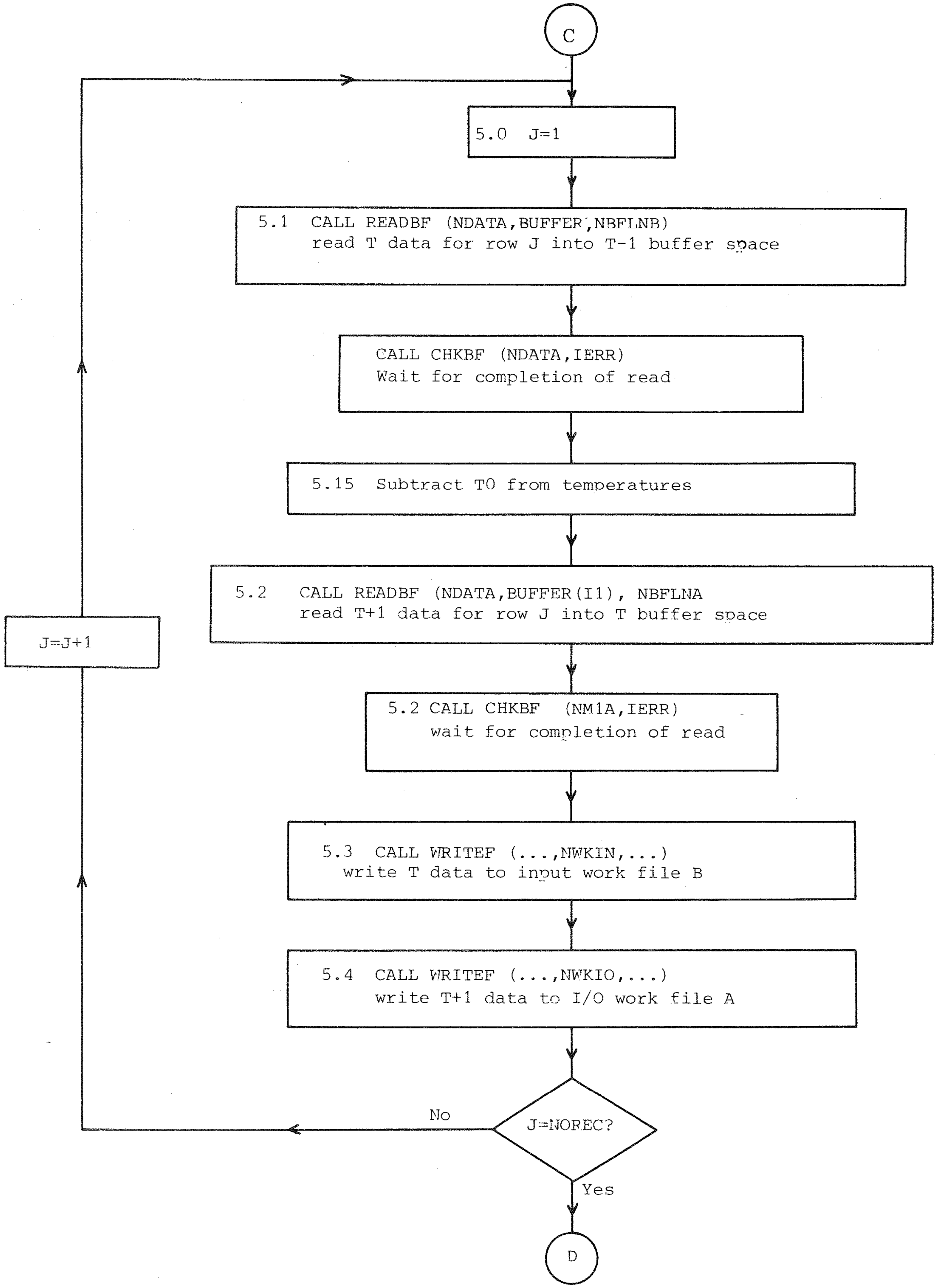
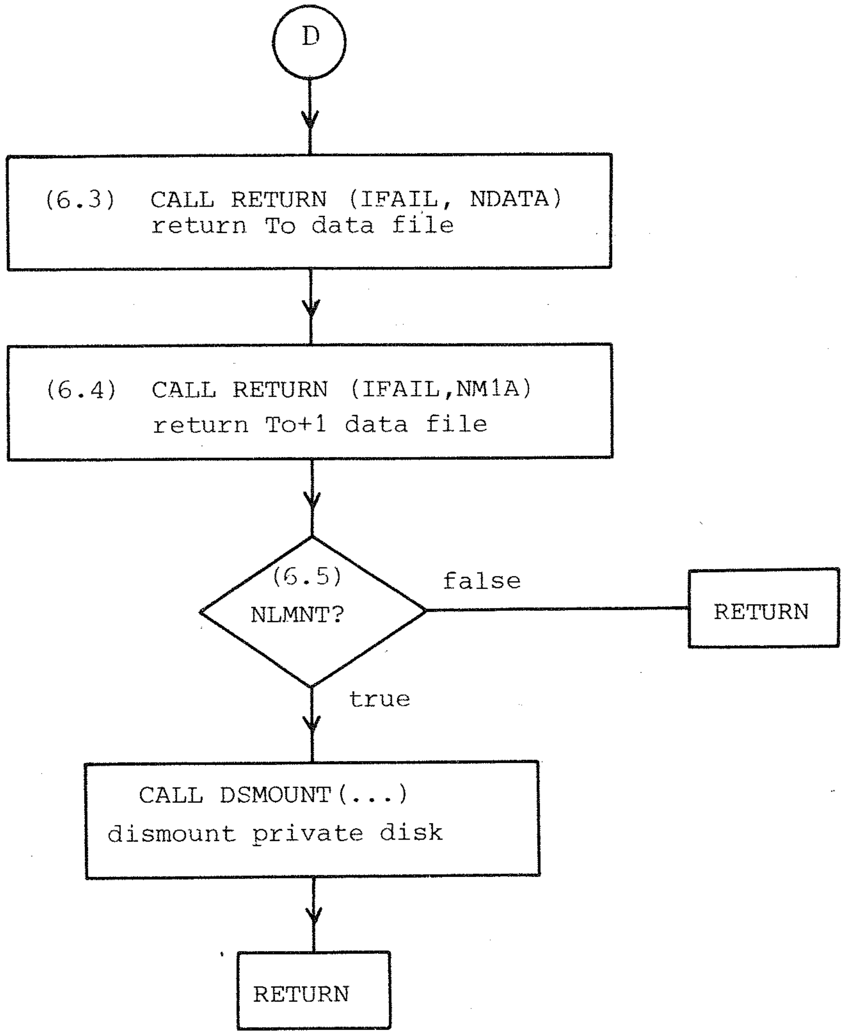


Diagram 4 (contd. 2)

Diagram 4 (contd. 3)



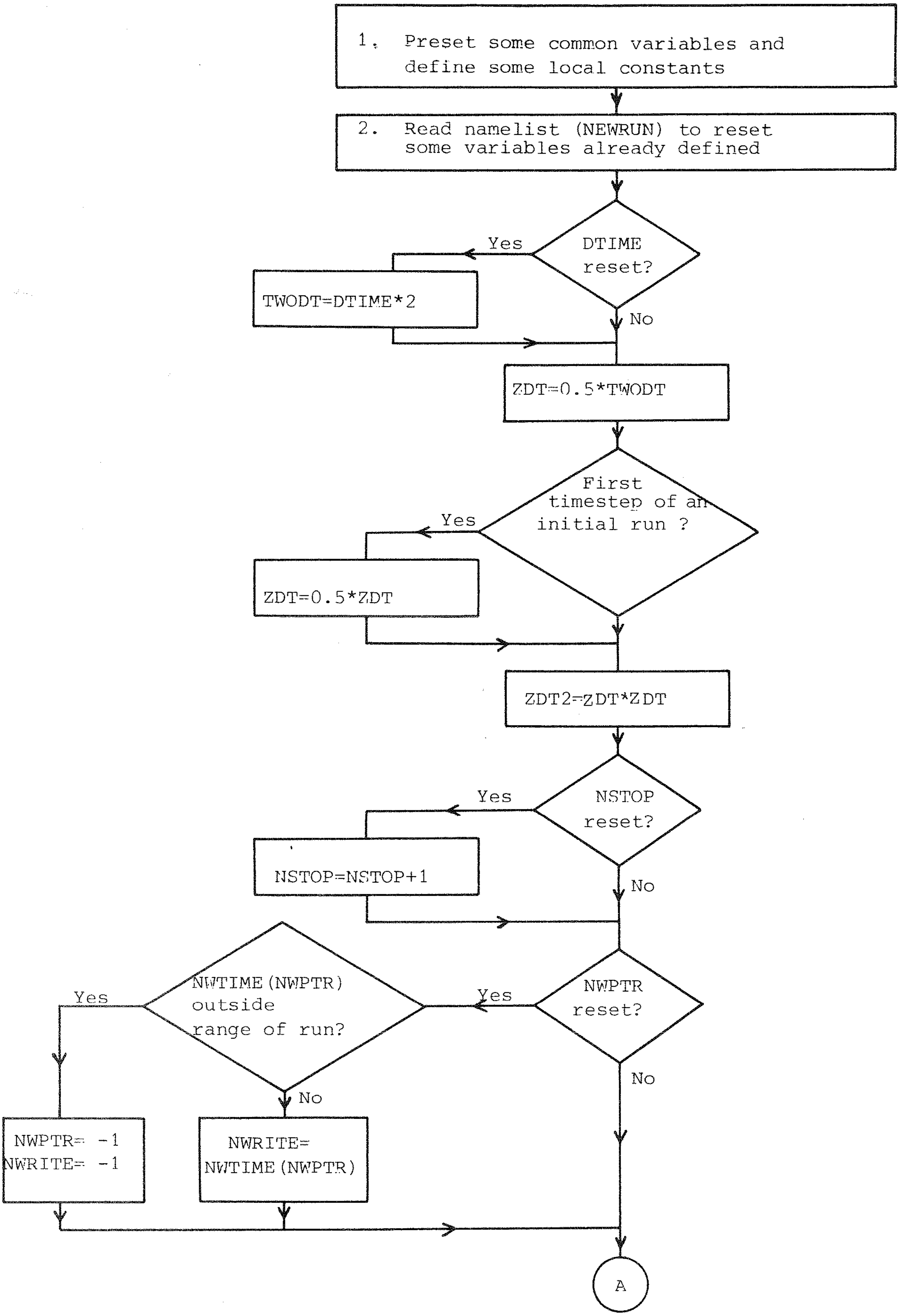
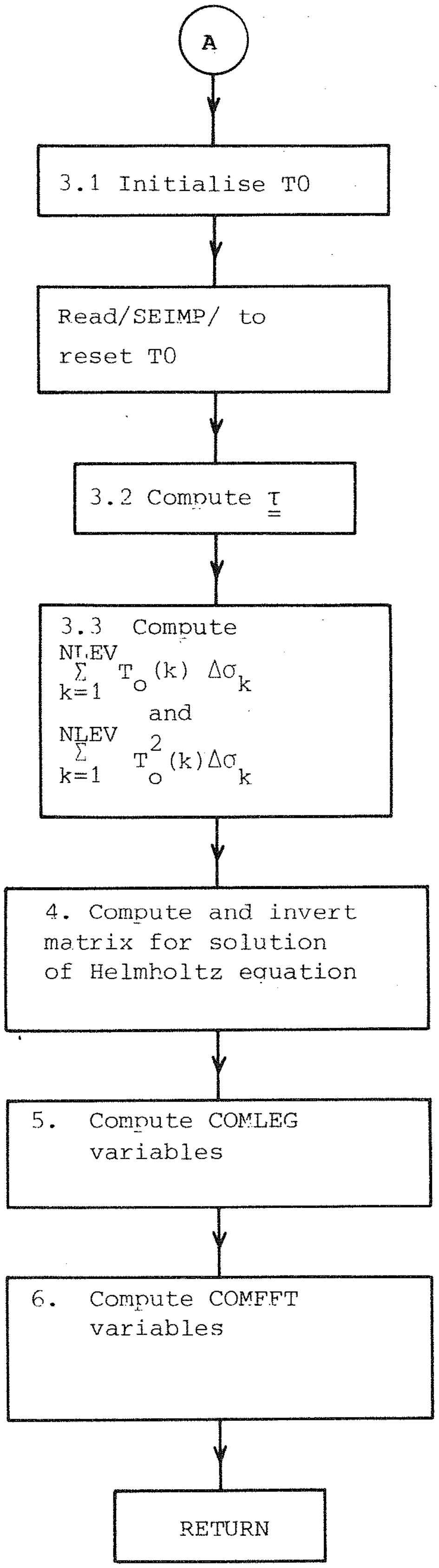


Diagram 5 <1.10> Subroutine DATINI

Diagram 5 (contd.)



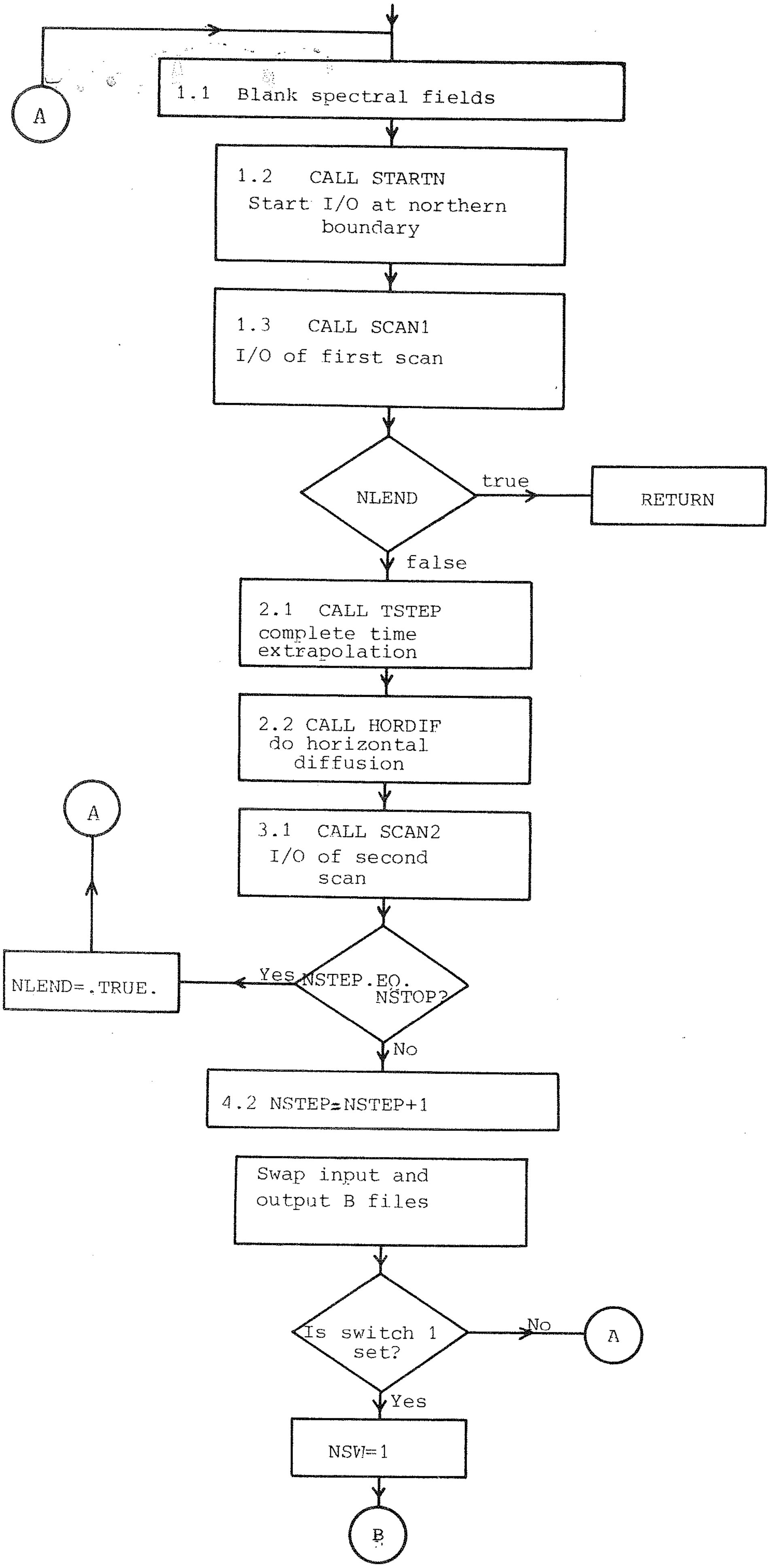


Diagram 6 <2.1> Subroutine STEPON

Diagram 6 (contd.)

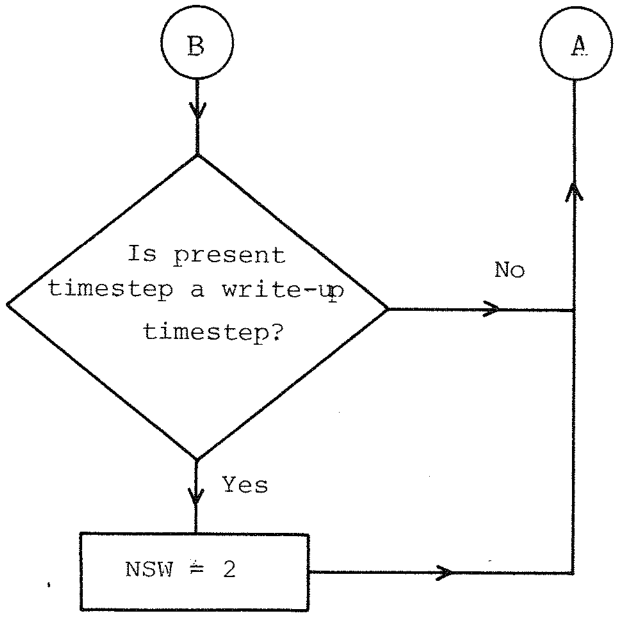


Diagram 7 <2.7> Subroutine STARTN

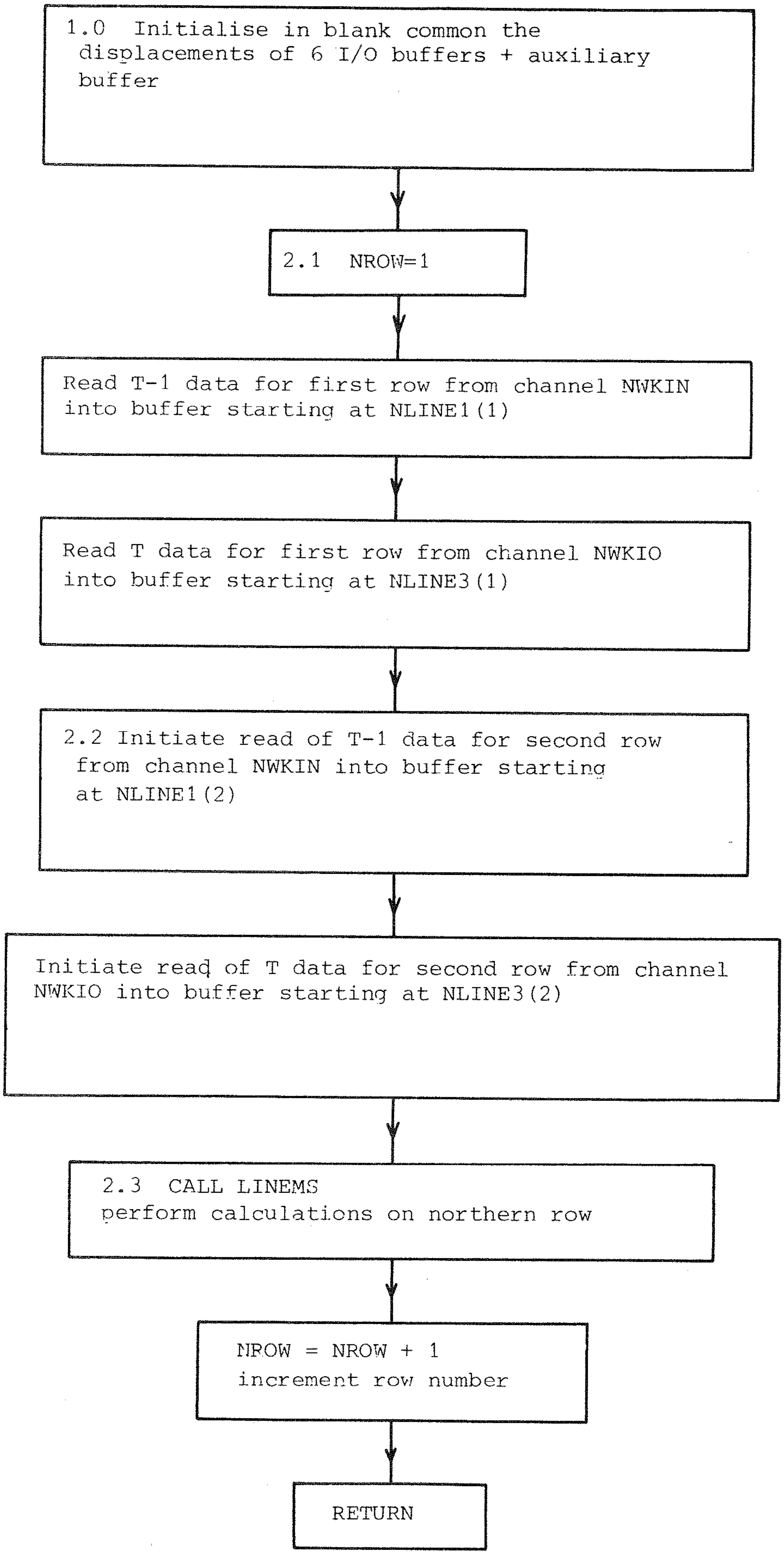
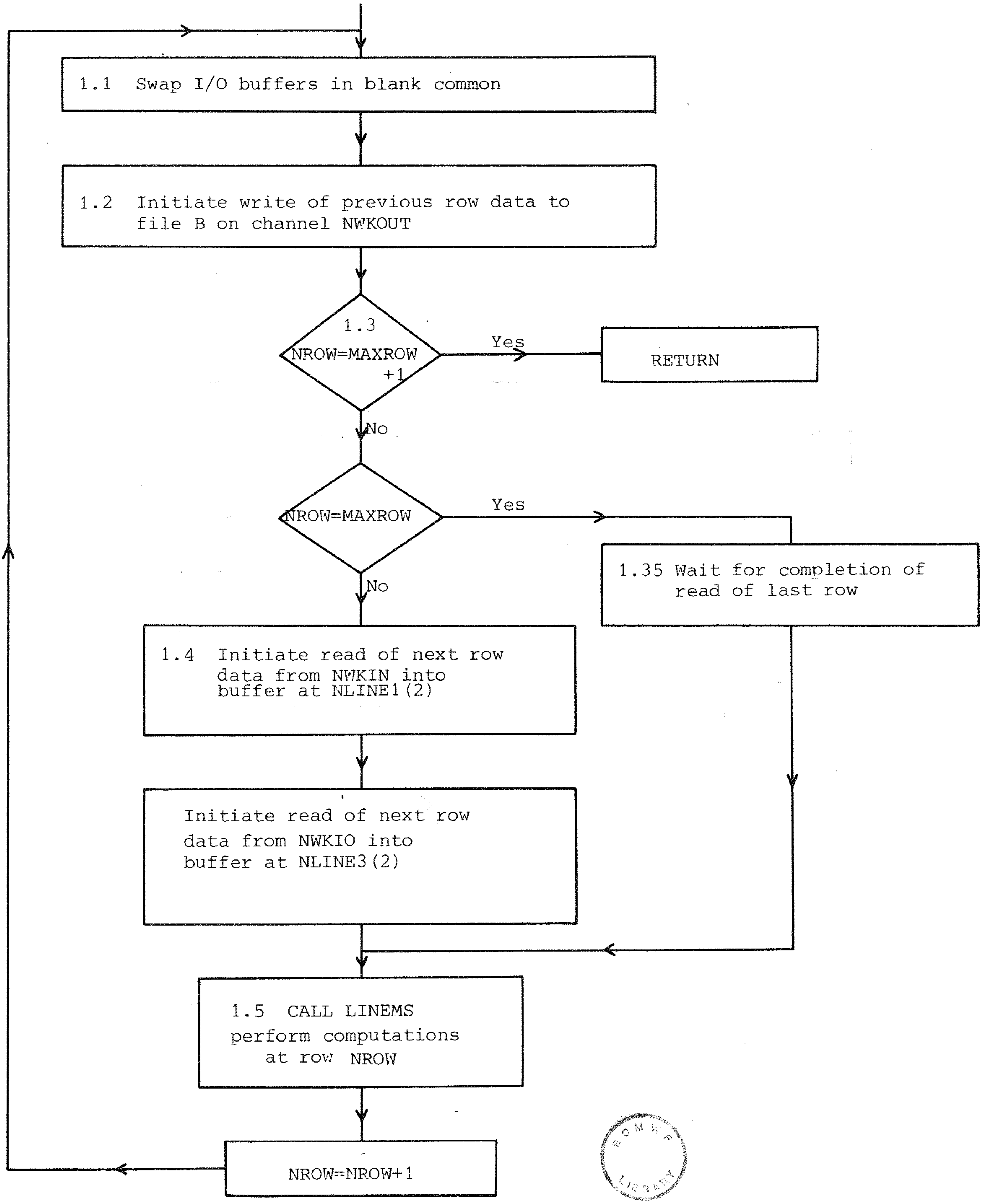
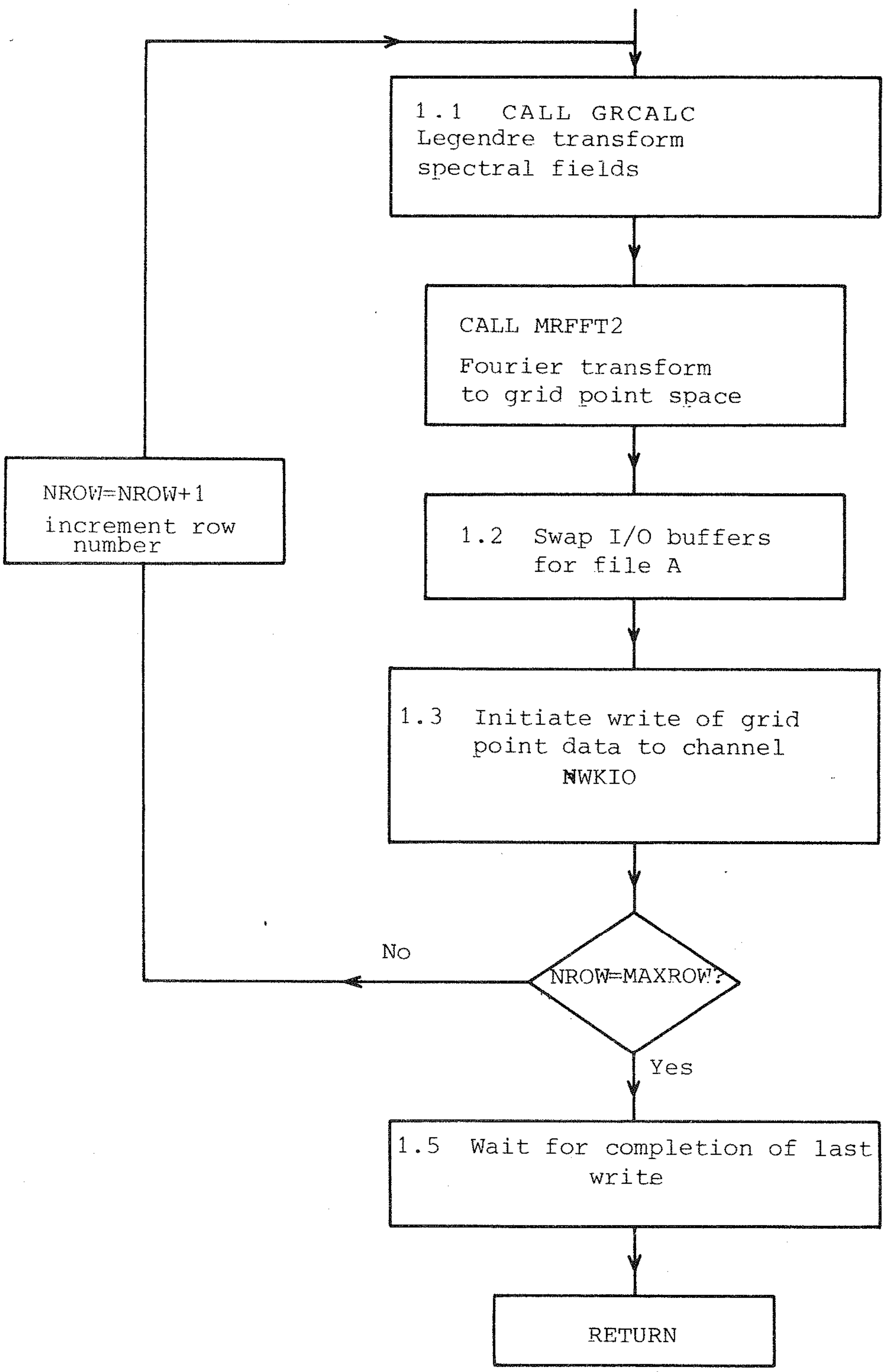


Diagram 8 <2.8> Subroutine SCAN1



ECMWF LIBRARY

Diagram 9 <2.12> Subroutine SCAN2



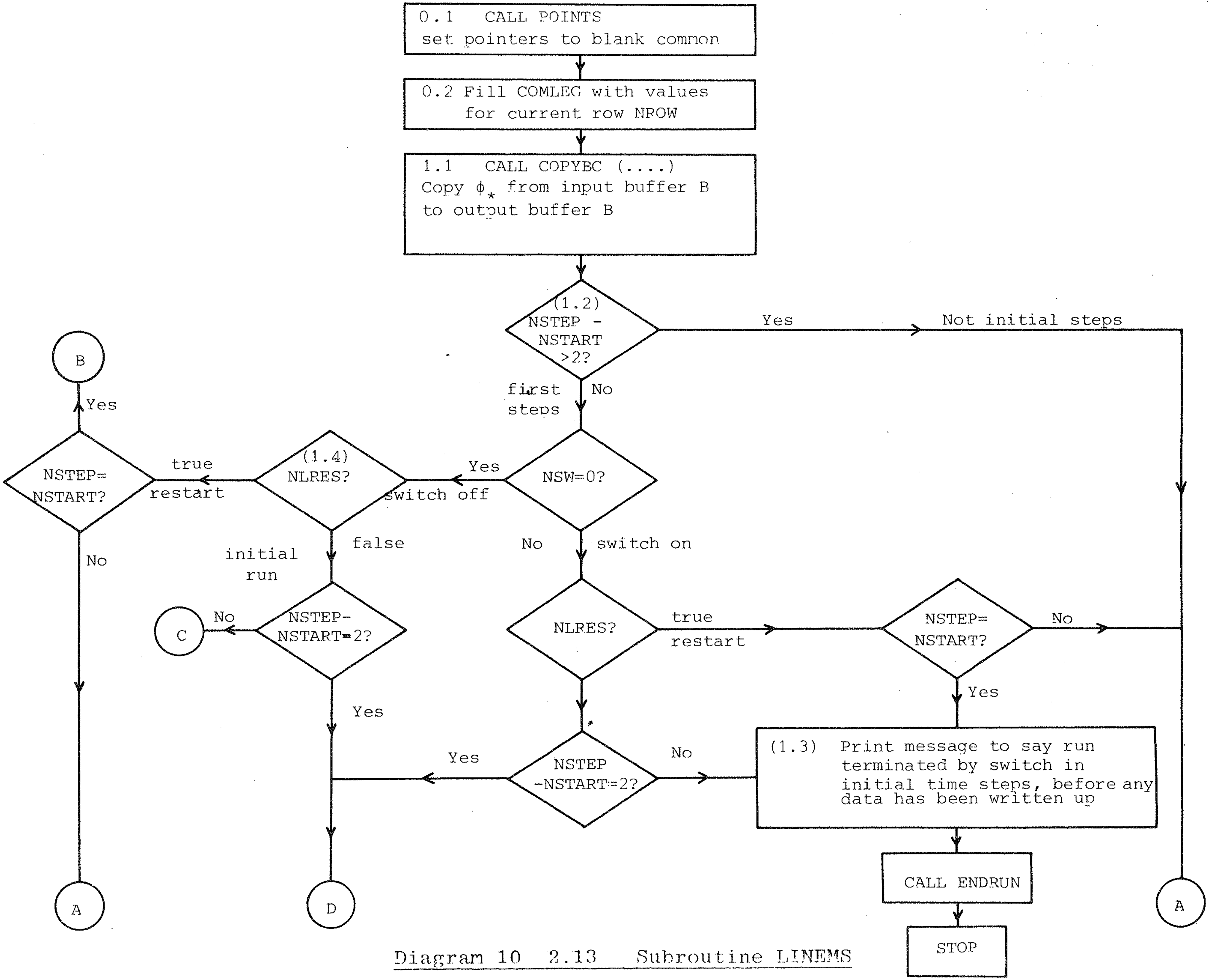


Diagram 10 2.13 Subroutine LINEMS

Diagram 10 (contd. 1)

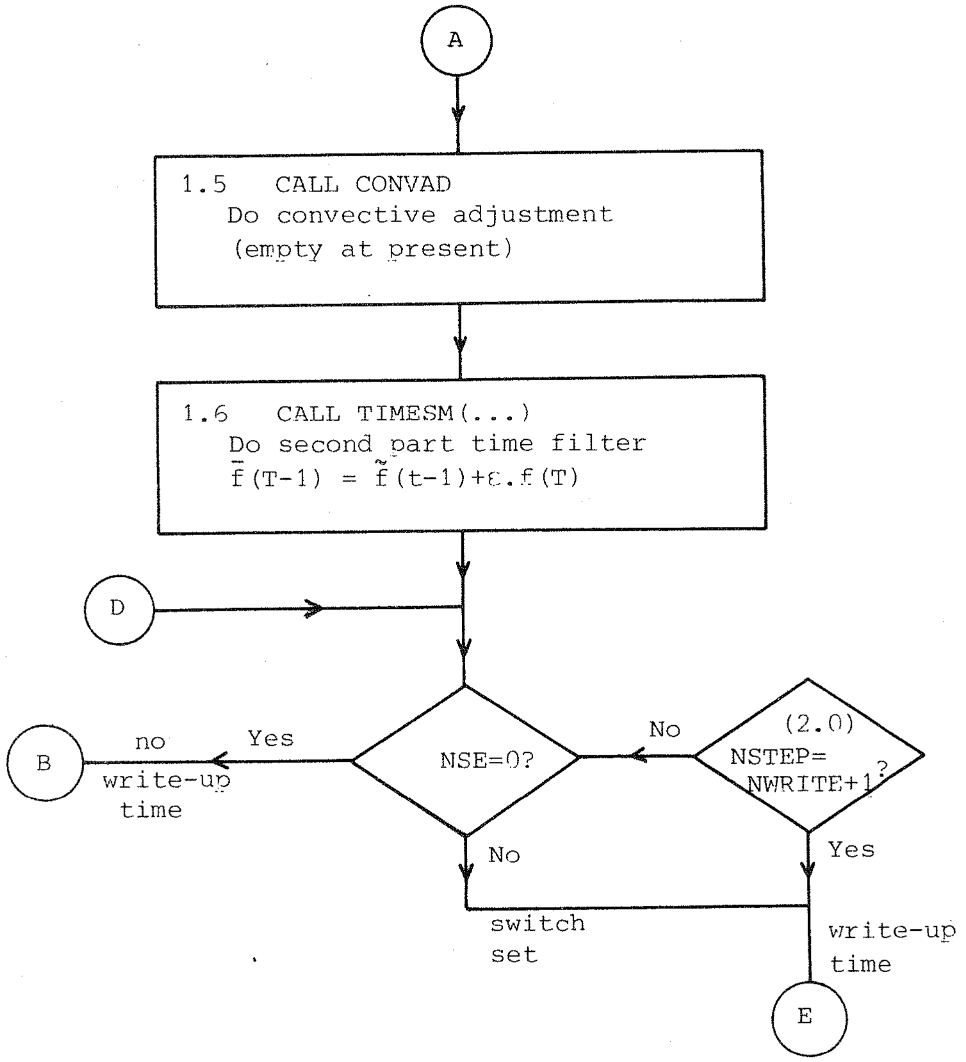
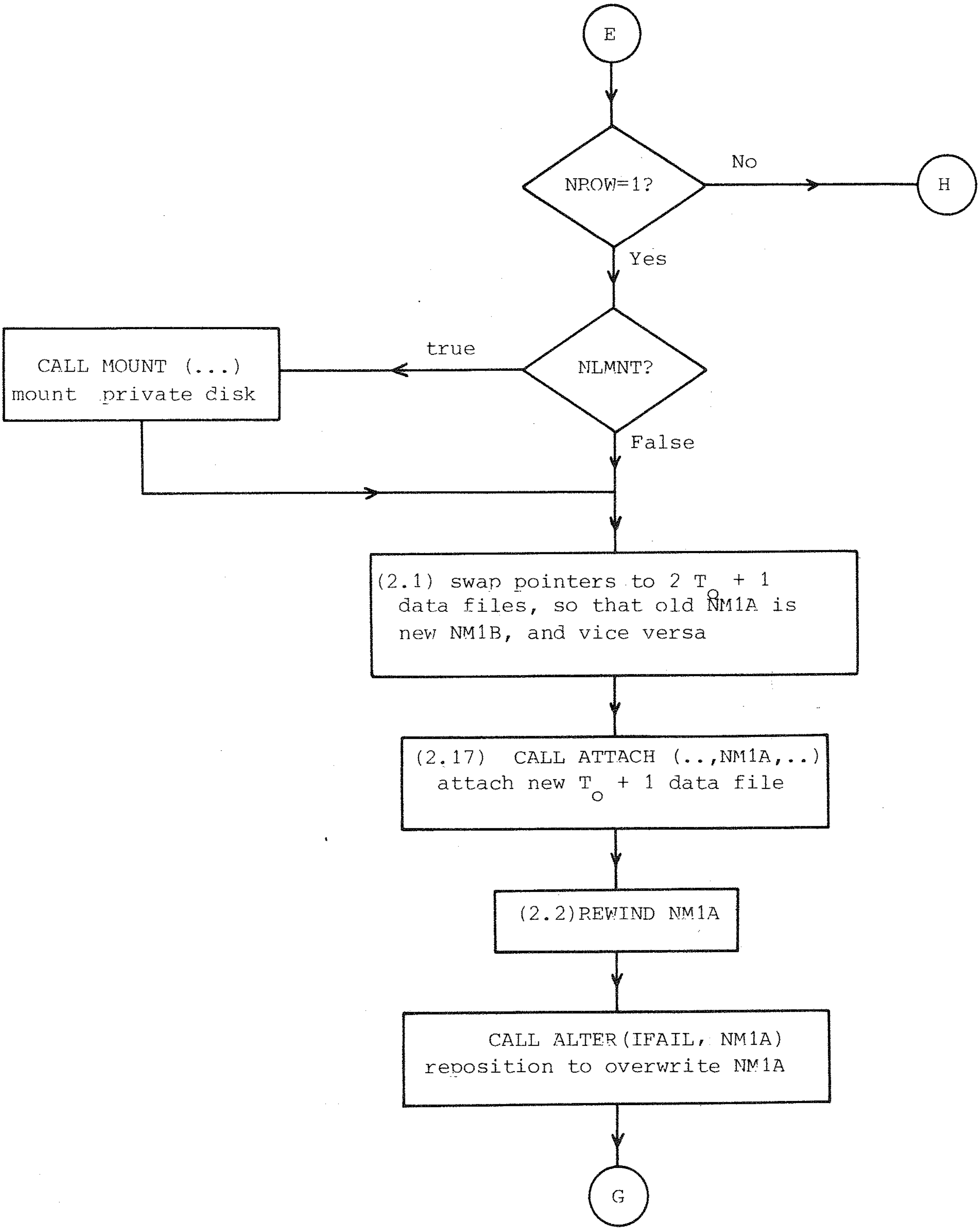


Diagram 10 (contd. 2)



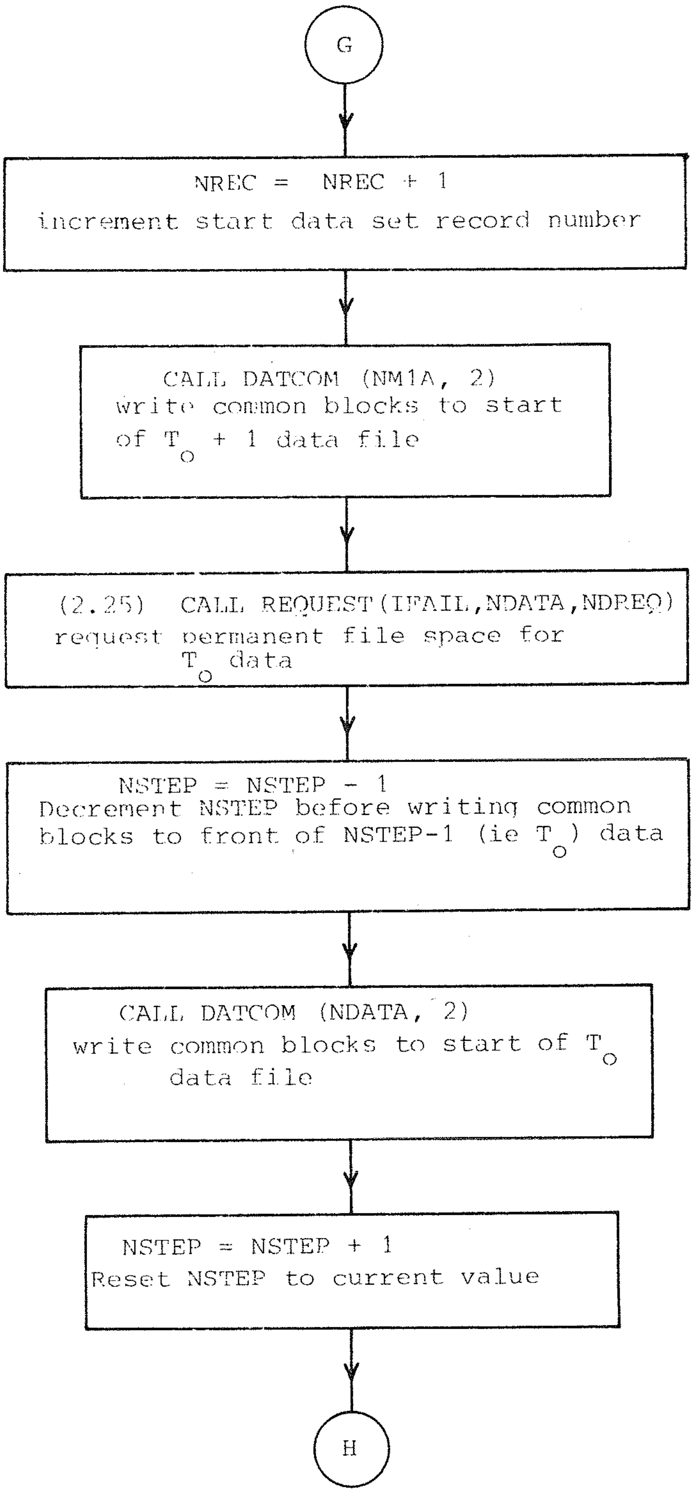


Diagram 10 (contd. 3)

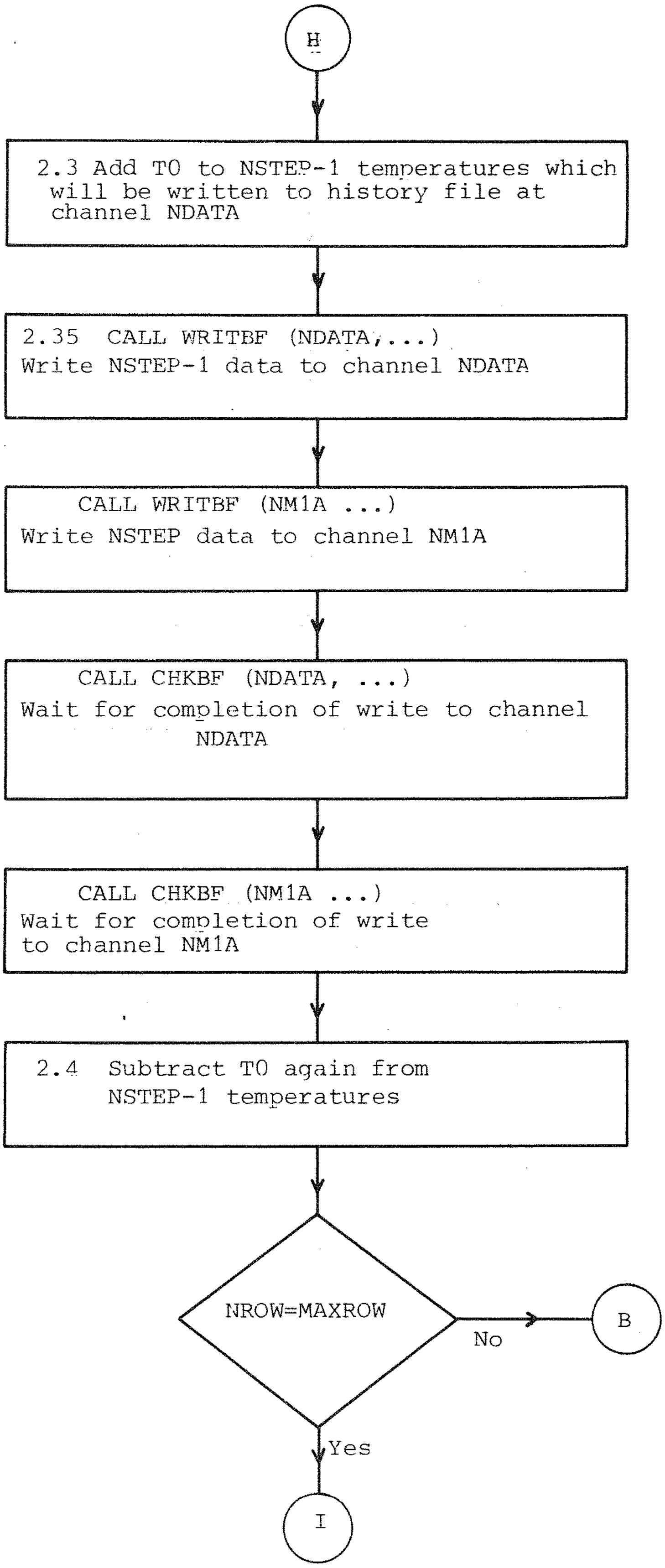


Diagram 10 (contd. 4)

Diagram 10 (contd. 5)

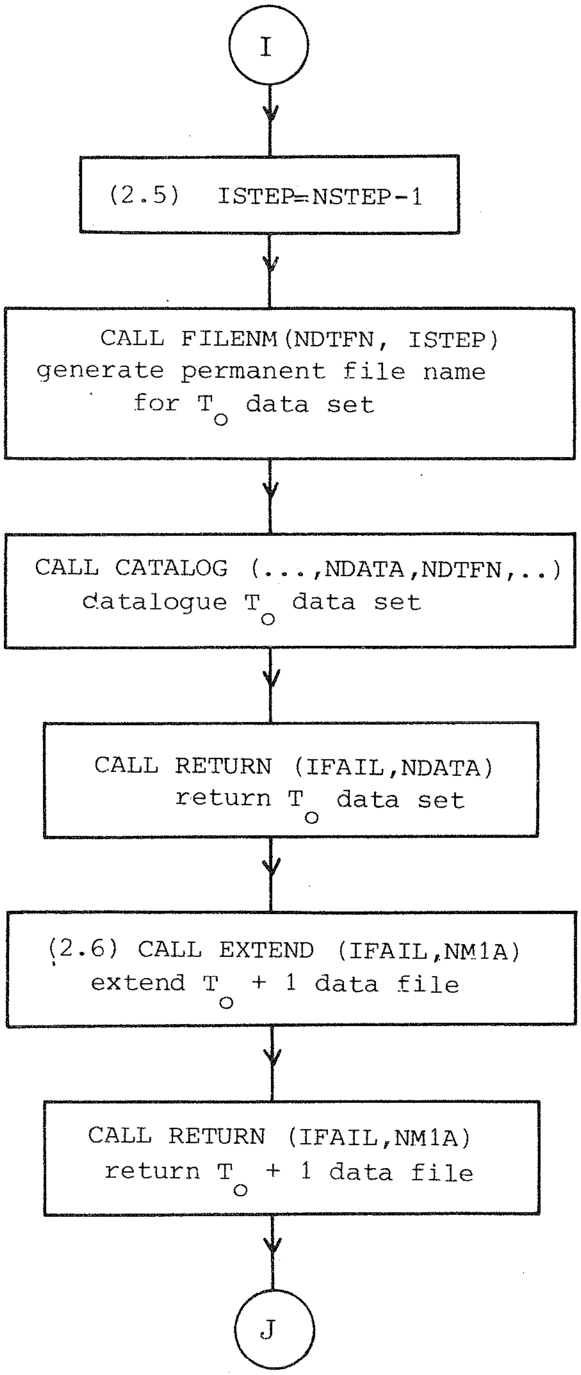


Diagram 10 (contd. 6)

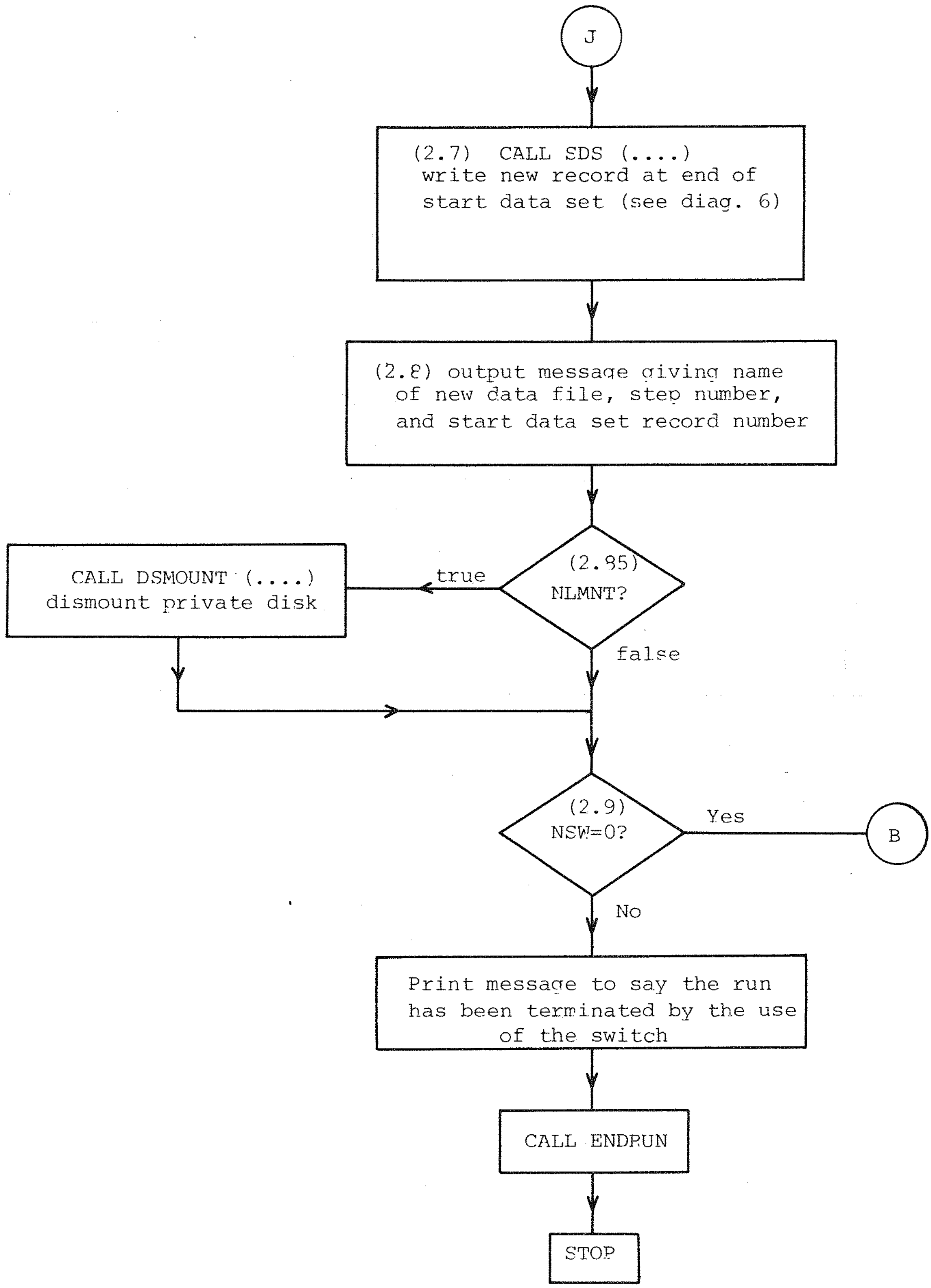


Diagram 10 (contd. 7)

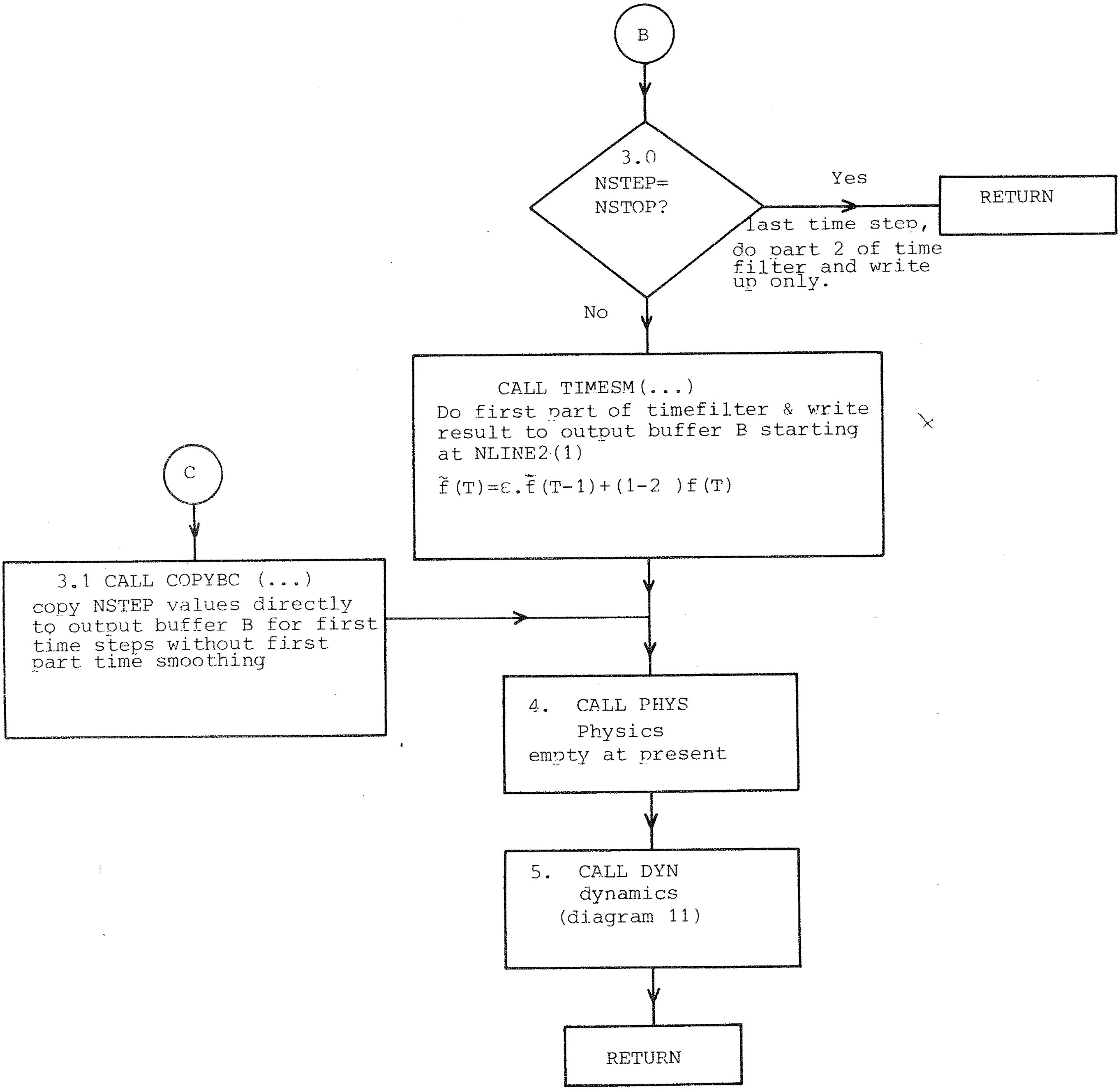
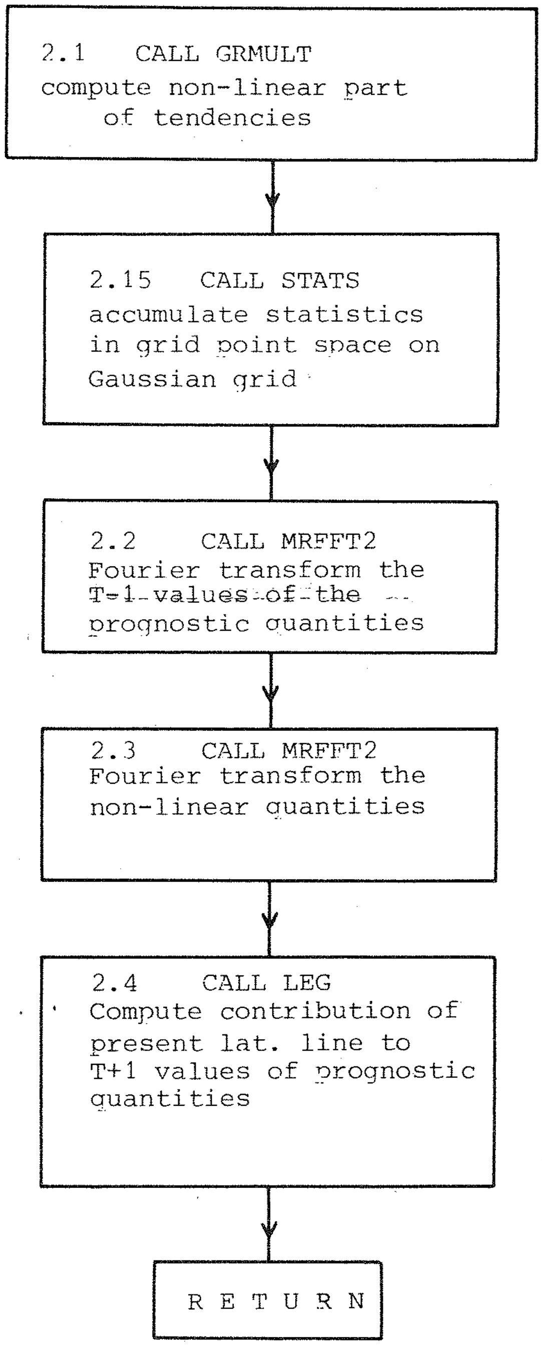


Diagram 11 <2.15> DYN



Appendix 1

App.1.1 Subroutine BLDUMP

The comprehensive system of subroutines for dumping common blocks and parts of blank common in the grid point model has not (yet) been implemented in the spectral model. Instead subroutine BLDUMP offers a convenient way to dump parts of blank common.

The subroutine is called as follows:

```
    CALL BLDUMP (INAME, IADR, ILEN)
with INAME: hollerith string of 8 characters, which can be
            used to specify the character or place of the
            dump.
     IADR : (relative address -1) in blank common of
            first word to be dumped.
     ILEN : number of words to be dumped.
```

Example: Suppose one wishes to dump the T-1 values of vorticity and divergence, which are present in blank common in the buffer of input file B. The following statements will produce the required dump, each number being printed in Format E16.7, 8 numbers per line.

```
ILEN = 2 * NLONP2
CALL BLDUMP (8H Z and D  ,NZM1, ILEN)
```

App.1.2 Subroutine OUTPUT(J)

The subroutine OUTPUT, available at present, produces no output, unless J = 2. In that case the number of completed timesteps is printed.

App.1.3 Subroutines REORD1 and REORD2

These subroutines reorder real spectral fields, stored columnwise in one dimensional arrays, into diagonal storage. Moreover, they allow the output field to be only a part of the input field, such that, for example, a columnwise rhomboidal input field results in a diagonal-wise triangular output field. REORD1 assumes NMAX=MMAX+1, whereas REORD2 assumes NMAX=MMAX.

The calls are

```
CALL REORD1(XOUT,XIN,MOUT,MIN,LFTRI,LTTRI)
CALL REORD2(XOUT,XIN,MOUT,MIN,LFTRI,LTTRI)
with
XOUT :  the reordered output field
XIN  :  the columnwise stored input field
MOUT :  MMAX for the reordered field
MIN  :  MMAX for the input field
LFTRI:  .TRUE. if input field is triangular (from triangular)
        .FALSE.if input field rhomboidal
LTTRI:  .TRUE. if output field is triangular (to triangular)
        .FALSE.if output field rhomboidal
```

Appendix 2

In this appendix we briefly describe some mathematical subroutines used in the spectral model.

App.2.1 Subroutine GAUAW

This subroutine computes the abscissas x_k and weights w_k for the gaussian integration:

$$\int_{-1}^{+1} f(x)dx \simeq \sum_{k=1}^{N} w_k \cdot f(x_k)$$

For the method see:

"Abscissas and Weights for Gaussian Quadratures of High Order", P. Davis and P. Rabinowitz, J. of Research of the National Bureau of Standards, Vol. 56, No. 1, January 1956.

The call is as follows:

```
        CALL GAUAW (A,W,K)
A(K): real array of length K, containing the abscissas
      on return from GAUAW.
W(K): real array containing the weights
K   : Integer number of integration points. K must be > 2.
```

Note that A and W contain the abscissas and weights for the whole globe and not just for one hemisphere.

The program is a modified version of a routine received from the University of Copenhaven, Denmark.
It has been modified at ECMWF by E. Edberg.

App.2.2 Subroutine BSSLZR

The iterative procedure used in the previous routine GAUAW requires the zero's of the Bessel function $J_o(x)$ as a first approximation of the abscissas. These zero's are returned to GAUAW after a call to subroutine BSSLZR. In this routine the first 50 zeros are defined in a DATA statement. Higher zeros are approximated as equidistant with mutual distance of π.

The call is as follows:

        CALL BSSLZR (BES,N)

with BES(N): array containing N zeros of $J_o(x)$ on return from BSSLZR

N: integer number of zeros

App.2.3 Subroutine PHCS

This routine returns associated Legendre polynomials and their derivatives.

The Legendre polynomials are defined as follows:

$$P_{m,n}(\mu) = \sqrt{(2n+1)\,\frac{(n-m)!}{(n+m)!}}\;\frac{1}{2^{n+1}n!}(1-\mu^2)^{m/2}\,\frac{d^{n+m}}{d\mu^{n+m}}(\mu^2-1)^n$$

These functions are normalised as follows:

$$\int_{-1}^{+1}\left[P_{m,n}(\mu)\right]^2 d\mu = \tfrac{1}{2}$$

The derivatives are returned as:

$$H_{m,n}(\mu) = (\mu^2-1)\,\frac{d}{d\mu}\,P_{m,n}(\mu)$$

$P_{m,n}(\mu)$ is computed by the following numerically stable recurrence relation (see Belousov, 1962):

$$P_{m,n}(\mu) = c_{m,n}\,P_{m-2,n-2} - d_{m,n}\,\mu . P_{m-2,n-1} + e_{m,n}\,P_{m,n-1}$$

where

$$c_{m,n} = \left(\frac{2n+1}{2n-3}\cdot\frac{m+n-1}{m+n}\cdot\frac{m+n-3}{m+n-2}\right)^{\frac{1}{2}}$$

$$d_{m,n} = \left(\frac{2n+1}{2n-1}\cdot\frac{m+n-1}{m+n}\cdot\frac{n-m+1}{m+n-2}\right)^{\frac{1}{2}}$$

$$e_{m,n} = \left(\frac{2n+1}{2n-1}\cdot\frac{n-m}{n+m}\right)^{\frac{1}{2}}$$

From these the derivatives are computed by the following relation:

$$H_{m,n} = n . D_{m,n+1}P_{m,n+1} - (n+1)D_{m,n}P_{n,n-1}$$

where $D_{m,n} = \sqrt{\frac{n^2-m^2}{4n^2-1}}$

The call to the subroutine is

CALL PHCS (PMN,HMN,MAX,JMAX,X1)

with PMN(NSIZE): the returned Legendre polynomials in rhomboidal (M,M+J) truncation. They are stored column-wise. The dimension NSIZE must be NSIZE = MAX*JMAX

HMN(NSIZE): the returned derivatives in similar truncation

MAX: highest zonal wavenumber +1, i.e. MMAX

JMAX: length of the columns of the rhomboidal truncation

X1: sine of the latitude: $\mu_j=\sin\phi_j$

<u>Note</u>:

In subroutine DATINI the definition of Legendre polynomials and derivatives is changed in the following way:

1° the Legendre polynomials are multiplied by $\sqrt{2}$. This normalises them to one:

$$\int_{-1}^{+1} \left[P_{m,n}(\mu) \right]^2 d\mu = 1$$

2° the derivatives are multiplied by -1. Their definition therefore is

$$(1-\mu^2) \frac{d}{d\mu} P_{m,n}(\mu)$$

Denoting the Legendre polynomials in the present model by $P_{m,n}(\mu)$ and in the Reading-model (Hoskins and Simmons, 1975) by $P_n^m(\mu)$, the following relation holds:

$$P_{m,n}(\mu) = (-1)^m P_n^m (\mu)$$

App.2.4 Subroutines QREIG, COMHES, COMLR

These subroutines compute the eigenvalues of a real matrix.

The user calls:

```
        CALL QREIG (A,I,J,K,B,C)
```

where

```
A (I*I) :  real matrix
      I :  order of matrix A
    J,K :  not used
   B(I) :  contains eigenvalues on return
      C :  not used
```

Subroutine QREIG calls subroutine COMHES to bring the matrix in the upper Hessenberg form and subroutine COMLR to find the eigenvalues of this matrix. These last two routine are more general and are able to handle complex matrices.

App.2.5 Subroutine MINV

This routine inverts a matrix.

The user calls:

```
        CALL MINV (A,N,D,L,M)
A(N*N)  :  input matrix; on return it contains
           the inverted matrix
     N  :  order of matrix A
     D  :  on return contains determinant
  L(N)  :  work vector of length N
  M(N)  :  work vector of length N.
```

For further documentation see comments in the source listing of the routine.

App.2.6 Subroutines MRFFT2, VPASS2 and

These routines perform a Fourier transform and are designed especially for a vector-machine.

In order to set up some trigonometric table the user calls:

```
CALL RFTSET(TRIGS,N)
```

then the user calls:

```
CALL MRFFT2 (A,WORK,TRIGS,INC,JUMP,N,LOT,ISIGN)
```

where

A(N+2)	:	array containing input/output data. Gridpoint data vectors have length N with two empty words at the end. The spectral data vector has length N+2.
WORK	:	Work array of length (N+1)*LOT
TRIGS(N/4+1)	:	trigonometric tables, prepared in RFTSET
INC	:	the increment within each data vector (i.e. INC=1 for consecutively stored data)
JUMP	:	the increment between the start of each data vector (i.e. N+2 for consecutively stored data vectors)
N	:	length of the gridpoint data vector
LOT	:	number of data vectors
ISIGN	:	+1 transform from spectral to gridpoint -1 transform from gridpoint to spectral

Subroutine MRFFT2 calls subroutine VPASS2.

For further information see comments in source listing of these routines.

The routines have been written by C. Temperton, ECMWF.

Appendix 3 - by M. Jarraud

The above described code was modified for three main reasons:

1) To allow the model to run with any "reasonable" truncation.

2) To allow it to be either global or hemispheric (on the northern hemisphere).

3) To improve some parts of the code.

These three reasons will be referred to as R1, R2, R3 in the following.

A - Modification of the model

A new subroutine (TRUNC) and a new common block (COMTRU) were introduced (R1, R2).

1 - <TRUNC> computes the parameters of a pentagonal truncation either for a global or a hemispheric model. TRUNC is called from DATINI (see further on).

The INPUT parameters are

parameters	location	meaning	defined
NSPHER	COMTRU	= 1 for a global version = 2 for a hemispheric version	DATINI
NOREC	COMHKP	number of latitude circles in the equivalent gaussian grid	MAKEDT
NTRM	COMTRU		DATINI
NTRN	COMTRU	limits of the truncation cf. fig. 1.1	DATINI
NTRK	COMTRU		DATINI

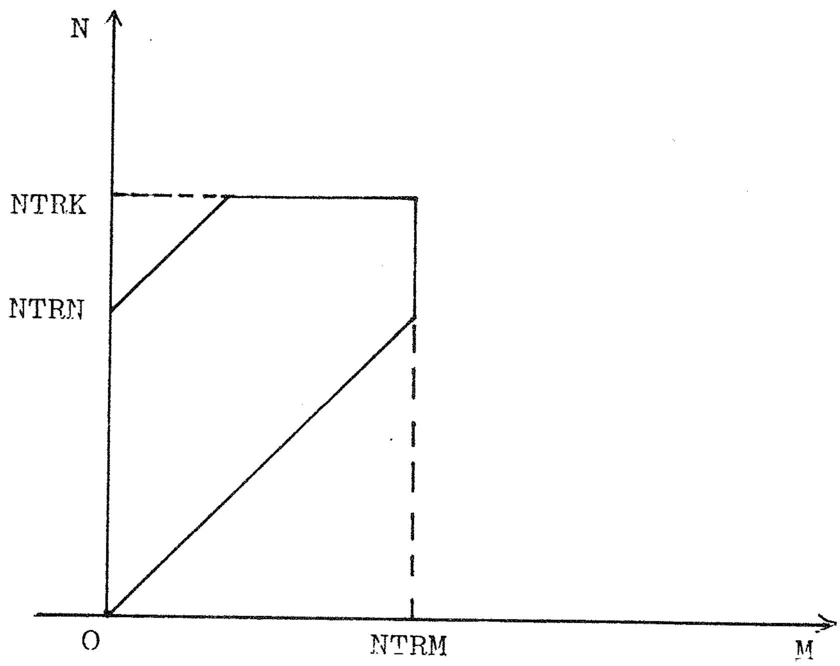


Fig. 1.1 Pentagonal truncation

All common truncations are special cases of the pentagonal one:

trapezoidal: NTRN = NTRK

triangular: NTRN = NTRK = NTRM

rhomboidal: NTRM + NTRN = NTRK

Before going further we have to explain some language conventions:

In a hemispheric model symmetry properties allow us to use for each spectral field F only symmetric [(N+M) even] or antisymmetric [(N+M) odd] spectral components $F_{m,n}$. In the first case the fields will be referred to as "even" fields (even if it is a global version, when all components are used) and in the second case they will be referred to as "odd" fields. Vorticity is an "odd" field.
All the other spectral fields ($D, T, Q, \ell np_*$, ...) are "even" fields.

We are now able to clarify the meaning of the output parameters.

parameters	location	meaning
MAXROW	COMIOC	number of latitude circles used = NOREC/NSPHER
NSPT	COMTRU	number of points included in the truncation
NSPE	"	" " " " " " for even fields
NSPO	"	" " " " " " for odd fields
MMAX	COMHKP	NTRM+1
NMAX	"	NTRK+1
NEMAX	COMTRU	number of diagonals used for even fields
NOMAX	"	" " " odd "
NEVEN (NEMAX+1)	"	address in an even (odd) spectral field of
NODD (NOMAX+1)	"	the 1st element of the Nth diagonal
NALPE (NEMAX)	"	address in a complete spectral field of the first element of the Nth even (odd) diagonal.
NALPO (NOMAX)	"	(Used only as pointers for the Legendre polynomials).

So, in <COMTRU> there are the following parameters

NSPHER, NTRM, NTRN, NTRK,
NEMAX, NOMAX, NSPE, NSPO, NSPT,
NEVEN(NEMAX+1), NODD(NOMAX+1)
NALPE(NEMAX), NALPO(NOMAX)

Now we shall study subroutine by subroutine the implemented modifications and the reasons for them.

2. <INITAL> (R2)

MAXROW and NOREC are now used in accordance with their new meaning.

3. <DATINI> (R1, R2, R3)

The major changes take place in section <5>.
Let us give the new structure of this section:

<5> now compute truncation parameters and constants related to Legendre transforms.

<5.1> read and compute truncation parameters (R1,R2). First, NSPHER, NTRM, NTRN, NTRK are read from a data card (FORMAT (4I5)).

Then TRUNC is called.

<5.2> compute and reorder Legendre polynomials and their derivatives (R3).

We have kept the counter I, but it no longer indicates the hemisphere (N or S). I = NSPHER-1 and is incremented in <5.5> so that <5.2>-<5.4> is done only once for a hemispheric version and twice for a global one. The main change in this part is that there is no longer an extra line for ALP. This line was used to compute U and V in Fourier space, using the formula

$$U_{m,n} = -\varepsilon'_{m,n}\,\zeta_{m,n-1} - \frac{a}{n(n+1)}\,imD_{m,n} + \varepsilon'_{m,n+1}\,\zeta_{m,n+1}$$

and a similar formula for $V_{m,n}$ (see p. 52).

Instead we now use the formulae

$$U_m = -a\left[\, im \sum_n \frac{D_{n,m}}{n(n+1)}\, ALP_{n,m} - \sum_n \frac{\zeta_{n,m}}{n(n+1)}\, DALP_{n,m} \right]$$

$$V_m = -a\left[\, im \sum_n \frac{\zeta_{n,m}}{n(n+1)}\, ALP_{n,m} + \sum_n \frac{D_{n,m}}{n(n+1)}\, DALP_{n,m} \right]$$

The effect of the extra line is already taken into account in the computation of DALP, through PHCS.

This method has the following advantages:

. the truncation is now the same for all the fields and that is a big simplification and improvement.

. we have to call PHCS only once instead of twice.

. we replaced the two subroutines <REORD1> and <REORD2> by a new one <REORD>, shorter, more efficient and more general (written for a pentagonal truncation)

. the two arrays DEL and EPL used in connection with the old manner of computing U and V were removed from COMMAP and the size of ALP in COMLEG was reduced, causing a small but interesting save of core.

<5.3>- <5.5> were almost unchanged.

4. <LINEMS> (R2)

In a hemispheric model, instead of computing $\int_{-1}^{1} A_m P_{n,m} d\mu$ we compute $2\int_{0}^{1} A_m P_{n,m} d\mu$ using symmetry properties. Thus, for this gaussian integration we changed the gaussian weights into WEIGHT * NSPHER.

5. <GRMULT> (R3)

The structure is basically the same but with a more intensive use of pointers this subroutine may now be more efficiently vectorizable.

6. <STATS> (R3)

Same thing as <GRMULT>.

7. <LEG> (R1,R3)

The structure is the same but section <1> has hardly been modified.

<2> and <3> are now very different. We can note first that the number of points over a diagonal does not necessarily decrease as much as with a triangular truncation, since we have a general pentagonal one. So we can choose a truncation which is more efficient with respect to the CRAY vectorization.

We have the following main pointers

ILEVG	=	points to a vertical level of a grid point field
IEV	=	" " " spectral even field
IOV	=	" " " " odd "
IALPE	=	" to the relevant diagonal in ALP and DALP
IALPO	=	used for even (odd) fields

IEVR	=	points to the relevant diagonal in even spectral fields
IOVR	=	" " " odd " "

IM = NEVEN(JN+1)-NEVEN(JN)
or IM = NODD (JN+1)-NODD (JN)
} = length of the JN^{th} diagonal for an even (odd) field

The same do variables are used

JL : loop over levels
JN : - - diagonal
JM : - - points in one diagonal

<2.1> set some pointers

<2.2> compute $\ell N(p_*)$

<2.3> compute other fields (NLEV dependent fields)

<2.3.1> set pointers for first level
<2.3.2> " " for each diagonal

<2.4> contribution to temperature field

<2.5> " " humidity " even fields

<2.6> " " divergence "

<2.7> set pointers for each diagonal for odd fields

<2.8> contribution to vorticity field

<3. > RMS of the Helmholtz equation

<3.1> hardly modified

<3.2> was changed for reasons R1 and R3 so that it is now more easily vectorized.

<3.3> set pointers for next level for <2> and <3>.

8. <TSTEP> (R1, R3)

The structure is not changed but there are new pointers similar to some used in LEG.

9. <HORDIF> (R1, R3)

Similar modifications as for TSTEP.

10. <GRCALC> computes the Legendre transforms for spectral space to Fourier space.

This subroutine was almost completely rewritten (for R1 and R3 purposes).

One of the main differences is that U and V are now computed in the same loops as the other variables (cf. above).

<0> and <1> unchanged.

<2> compute the NLEV dependent variables.

<2.1> set pointers.

IEV has the same meaning as in LEG

IOV " " "

ISPE2 } sets at the first element of the JLth level
ISPO2 } of even (odd) field

<2.2> loop over diagonals for vorticity and for vorticity dependent part of U and V.
In this item, pointers have the same meaning as in LEG except JNO which is used to point at the right place in array RSQ.

<2.3> loop over diagonals for even fields and for the divergence dependent part of U and V.
Again, pointers have the same meaning as in LEG except JNE which is used to point at the right place in array RSQ.

<2.4> a correction is made for U since the model uses the absolute vorticity VZ instead of the vorticity (VZ-EZ) needed for the computation of U and V.

<2.5> multiplication by 1/A to get U and V.

<2.6> update pointers for the new level

<3. > computation for the 1 level variables.
This section was modified in the same manner as <2>.

Let us see now how to run this new version of the model. We shall see later which modifications were introduced into the initial data set creation.

B - How to run the model

1 - All the above modifications were made permanent on a new OLDPL file called <u>CYBERSPECTRALA</u>, ID = EWMJ3

2 - And then the deck running the model was written on a OLDPL file called SPECTRALMODEL, ID = EWMJ3. The listing of this file is given on the following pages.

→ SPMOD 1 = *DECK SPMOD

```
EWMJ3,STPAK.                                                        SPMOD    2
ATTACH,OLDPL,CYBERSPECTRALA,ID=EWMJ3,MR=1.                          SPMOD    3
UPDATE,Q,L=A124.                                                    SPMOD    4
RETURN,OLDPL.                                                       SPMOD    5
FIN,I=COMPILE,L=0.                                                  SPMOD    6
RETURN,COMPILE.                                                     SPMOD    7
FIN,OPT=2,L=0.                                                      SPMOD    8
MOUNT,SN=DSET15,VSN=PA0Q3Y.                                         SPMOD    9
ATTACH,TAPE30,T212121SDS,ID=EWMJ3,SN=DSET115.                       SPMOD   10
ATTACH,LIB3,SPECTROBJ,ID=EWAB3,MR=1.                                SPMOD   11
ATTACH,LIB1,GEMINILIB,ID=EWJC3,MR=1.                                SPMOD   12
ATTACH,LIB2,ECMWF,ID=EWP3,MR=1.                                     SPMOD   13
LIBRARY,LIB3.                                                       SPMOD   14
LDSET,LIB=LIB1/LIB2.                                                SPMOD   15
LGO.                                                                SPMOD   16
*C INITAL                                                           SPMOD   19
*C DATIN                                                            SPMOD   20
*C GRCALC                                                           SPMOD   21
*C LGNDRE                                                           SPMOD   22
*C STPN                                                             SPMOD   23
*C STRIN                                                            SPMOD   24
*C SCAN1                                                            SPMOD   25
*C LINMS                                                            SPMOD   26
*C DYNAM                                                            SPMOD   27
*C LEG                                                              SPMOD   28
*C SCAN2                                                            SPMOD   29
*C TSTEP                                                            SPMOD   30
*C HORDIF                                                           SPMOD   31
*C GRMULT                                                           SPMOD   32
*C STATIS                                                           SPMOD   33
*C TIMESM                                                           SPMOD   34
*C CUSU                                                             SPMOD   35
*C DATCOM                                                           SPMOD   36
*ID MODTRU                                                          SPMOD   37
*D COMMAP.10                                                        SPMOD   38
*D COMMAP.14,19                                                     SPMOD   39
     R SIG(9),   SIGKPH(9),DSIGMA(9),R2DSIG(9),                     SPMOD   40
     R SQ(22),   RSQ(22),                                           SPMOD   41
     R XM(22),                                                      SPMOD   42
     R SIT(16),  W(16),    CS(16),   RCS(16),  ALAT(16),            SPMOD   43
     R G(81),    A(81)                                              SPMOD   44
*D DATCOM.19                                                        SPMOD   45
      COMMON/COMMAP/NC2(353)                                        SPMOD   46
*D DATCOM.51                                                        SPMOD   47
```

→ SPMOD 17 = 7/8/9

```
*D DATCOM.81                                                        SPMOD    49
      ILEN=355                                                      SPMOD    50
*D COMTRU.11                                                        SPMOD    51
     + NEVEN(23),NODD(23), NALPE(22),NALPO(22)                      SPMOD    52
*D COMLEGA.1                                                        SPMOD    53
     R ALP(253), DALP(253)                                          SPMOD    54
*D COMIMP.8                                                         SPMOD    55
     R TO(9),                                                       SPMOD    56
     R BMI(1782),                                                   SPMOD    57
     R TAU(81),  AW(81)                                             SPMOD    58
*D COMSPE.8                                                         SPMOD    59
     R VZ(4554), D(4554),  T(4554),  W(4554),  ALPS(506),KH(4554)   SPMOD    60  -> SPMOD 62 = 7/8/9
7/8/9 PROGRAM SPECTR(INPUT=400,TAPE5=INPUT,OUTPUT=400,TAPE6=OUTPUT  SPMOD    64
     *,TAPE7=OUTPUT                                                 SPMOD    65
     *,TAPE10=0,TAPE11=0,TAPE12=0,TAPE15,TAPE20=0,TAPE21=65,TAPE22=65,  SPMOD    66
     *TAPE30)                                                       SPMOD    67
      COMMON BUFFER(30000)                                          SPMOD    68
      CALL MASTER                                                   SPMOD    69
      STOP                                                          SPMOD    70
7/8/9 END                                                           SPMOD    71  -> SPMOD 73 = 7/8/9
$REST                                                               SPMOD    75
 NLRES=F,                                                           SPMOD    76
$                                                                   SPMOD    77
      CYBER TEST RUN                                                SPMOD    78
      26 JULY 1978                                                  SPMOD    79
      T21.21.21 GLOBAL                                              SPMOD    80
                                                                    SPMOD    81
$NEWRUN                                                             SPMOD    82
      NSTOP=5,                                                      SPMOD    83
      NWPTR=1,                                                      SPMOD    84
      EPS=0.06,                                                     SPMOD    85
      DIF=0.0,                                                      SPMOD    86
      NWTIME=200*0.,                                                SPMOD    87
$                                                                   SPMOD    88
$SEIMP TO=229.304,209.45,218.147,237.6,256.647,268.71,277.454,      SPMOD    89
           283.131,285.666,                                         SPMOD    90
$                                                                   SPMOD    91
   1    21    21    21                                              SPMOD    92
```

So, if we want to run the model with a triangular T21 truncation we have only to route the former deck to the INPUT QUEUE in the following way, e.g.:

```
EWMJ3.
ATTACH, OLDPL, SPECTRALMODEL, ID=EWMJ3.
UPDATE, F,D,8.
ROUTE, COMPILE, DC=IN, TID=AB.
7/8/9
6/7/8/9
```

3 - Let us now examine the modifications necessary to <u>run the model with another truncation</u>.

A general pentagonal truncation will be referred to by Tim in ik

with im = NTRM
in = NTRN
ik = NTRK

To run with a new truncation there are the following cards to change:

→ *D SPMOD.10
ATTACH, TAPE30, TiminikSDS, ID= , SN=

The creation of TiminikSDS is discussed in a following paragraph

→ *D SPMOD.40,44
Change, when needed, the dimensions of arrays
SPMOD.40 = arrays (NLEV)
——— 41 = ——— (NMAX)
——— 42 = ——— (MMAX)
——— 43 = ——— (NG) NG = NOREC/2
——— 44 = ——— (NLEV*NLEV)

```
→ * D SPMOD . 46
      COMMON/COMMAP/NC2(N)
with N = 9+4*NLEV+2*NMAX+MMAX+5*NG + 2*NLEV*NLEV

→ * D SPMOD . 40
      ILEN  = N
→ * D SPMOD . 50
      ILEN  = N
→ * D SPMOD . 52
    + NEVEN(N1+1), NODD(N2+1), NALPE(N1), NALPO(N2)

 with N1 = NEMAX
      N2 = NOMAX
→ * D SPMOD . 54
    R ALP (NSPT) , DALP (NSPT)

→ * D SPMOD. 56,58
    R T0 (NLEV),
    R BM 1 (N3),
    R TAU (NLEV*NLEV) , AQ (NLEV*NLEV)

 with N3 = NMAX * (NLEV*NLEV)

→ * D SPMOD. 60
    R VZ(N4), D(N5), T(N5), Q(N5), ALPS(N6), RH(N5)

 with N4 = 2*NSPO*NLEV
      N5 = 2*NSPE*NLEV
      N6 = 2*NSPE

→ * D SPMOD. 68
      if needed, change the dimension of the blank common

  * D . SPMOD. 79,80
    DATE
    Tim.in.ik GLOBAL (or HEMISPHERIC)} to label the run
```

```
→ * D SPMOD . 84
    NSPHER  NTRM  NTRN  NTRK   :  data card
      I5     I5    I5    I5
```

When NLON has to be changed as well (cf. paragraph on initial data set creation) some dimensions in COMFFT and COMSTA must be modified.

This can be done in the following way:

```
* I SPMOD.38
* TEXT
* D COMSTAA.2,.3
  * VZ2(NLON), D2(NLON), T2(NLON), PE(NLON), PT(NLON), PQ(NLON),
  * TP(NLON), PS(NLON), PFI(NLON)
* D COMFFT. 6
  R TRIG(NLON+1), WORK (NCRAY*(NLON+2)), NTR1A, NRST1A,
  NTR1B, NRST1B, NTR2, NRST2, NCRAY
* ENDTEXT
```

The time step can easily be changed, by adding a line to namelist NEWRUN :

```
→ * I SPMOD.87
     DTIME = new value of  .time step.
```

<u>C - Creation of a new initial data set</u>

The structure of files Timinik SDS and Timinik DATASET is similar to that described above except that the arrays DEL and EPL are removed from COMMAP.

<u>1 - This creation</u> is, at the moment, done in 3 steps.

1-1 Spectral triangular T63 data are truncated down to Timinik (it works even if the Timinik truncation is not included in T63 (the extra components are set to 0). This is done only for the divergence, vorticity, temperature and mixing ratio, surface pressure and orography fields. The results are put on a local file.

1-2 The second program transforms the six above described spectral fields read from the local file into fields over the equivalent gaussian grid for the 6 fields and also for U V components of the wind.

The characteristics of the equivalent gaussian grid for a pentagonal truncation are made clear in an ECMWF Technical Report to be published (Baede and Jarraud, 1978).

Suffice to know here that we must have

$$NLON \geq 3NTRM+1$$

$$NOREC \geq (3NTRK+1)/2 \quad \text{if } NTRM \geq 2(NTRK-NTRN)$$
$$\geq (2NTRN+NTRK+NTRM+1)/2 \quad \text{if } NTRM \leq 2(NTRK-NTRN)$$

Furthermore NOREC must be chosen even.

The results are also written on a local file.

1-3 The third program creates Timinik SDS and Timinik DATASET by adding the 3 common blocks COMMAP, COMHKP and COMBAS just before the grid point fields read from the local file.

This program is very similar to the one described on page 62-66.

The main differences are:

. KMAP no longer appears in the calling of MAKEDT

. The arrays ZDEL (7000) and ZEPL (7000) used to compute DEL and EPL are removed from COMMAP and cut out of MAPFAC

. The namelist INIDAT now contains the following variables

NOREC, NLON, NLEV, NCOM unchanged.

MMAX, NMAX, NSPEC suppressed and replaced by

constant	type	common block	initially defined in	initial value
NTRM	int	COMTRU	MAKEDT	1
NTRN	int	COMTRU	MAKEDT	1
NTRK	int	COMTRU	MAKEDT	1

These parameters are used in MAPFAC.

In the same way as the model, the deck running these 3 programs was written on an OLD PL file called

SPDATACREAT, ID = EWMJ3

The creation of T212121 SES and T212121 DATASET was done by routeing the former deck to the INPUT QUEUE:

```
e.g.    EWMJ3.
        ATTACH, OLDPL, SPECTRALMODEL, ID=EWMJ3.
        UPDATE, F, D, 8.
        ROUTE, COMPILE, DC=IN,TID=AB.
        7/8/9
        6/7/8/9
```

2 . To create new Timinik SDS and Timinik DATASET
we must change the following cards:

```
→ * D SPDATA .39,40
  CATALOG, TAPE7, Timinik DATASET, ID=          .
  ________, TAPE8, ________ SDS, ID=          .

→ * D SPDATA1.9
       COMPLEX  ZN (N), XN(4N)
```

with N=(NTRM+1)(NTRN+1)-(NTRM+NTRN-NTRK+1)(NTRM+NTRN-NTRK)/2

```
→ * D SPDATA1. 32,35
     NTRM =    }
     NTRN =    }  define the new truncation
     NTRK =    }
     NSPT = N  }

→ * D SPDATA2. 20,22
    *      FI (2*N) , SP(2*N),
    *      Z(2*N) , D(2*N), T(2*N), Q(2*N),
    *      ALP(N),  DALP(N)

→ * D SPDATA2.26
    *      IM(NTRM+1), RSQ (NTRK+1)
```

ECMWF LIBRARY

```
* D SPDATA2. 32,33
  *      ZB(NOREC), WEIGHT (NOREC),
  *      WORK(NLON),

* D SPDATA2. 37,38
  *    FIG (N1), PLG (N1), PMG (N1), PJG (N1),
  *    ZG  (N2), DG  (N2), TG  (N2), QG (N2), UG (N2), VG (N2)

with N1 = NLON + 2
     N2 = N1*NLEV

→ * D SPDATA2. 47,52
       NTRM=
       NTRN=
       NTRK=
       NG  =                  (NOREC/2)
       NLON=
       NLEV=

→ * D SPDATA3. 8,12
     Change, when needed, the dimension of arrays:
     SPDATA3. 8          arrays (NLEV)
     _______  9          ______ (NMAX)
     _______ 10          ______ (MMAX)
     _______ 11          ______ (NG)
     _______ 12          ______ (NLEV * NLEV)

→ * D SPDATA3. 16
       R  ZMAP (N),

  with N = 9+4*NLEV+2*NMAX+MMAX+5*NG+2*NLEV*NLEV

→ * D SPDATA. 22
       R   ZC3(N)

  * D SPDATA3. 24
          ILEN = N
```

→ * D SPDATA3. 38,43

Change parameters in namelist INIDAT.

NTRM =

NTRN =

NTRK =

NG =

NLON =

NLEV =

→ * D SPDATA3. 50,53

Change name of Dataset file

T im in ik DATASET

ID = , SN=DSET15.

*SN = DSET15.

DSET15 PA003Y

} i.e. private disk Dset15 Pa003Y will be used

References:

Baede, A.P.M. and Jarraud, M.	1978	Formulation and Organisation of ECMWF's Spectral Model, ECMWF Internal Report 21, to be published.
Burridge, D. and Haseler, J.	7/6/76	Random Access I/O Routines which can proceed in parallel with CPU processing. ECMWF Internal documentation.
Haseler, J. and Burridge, D.	1977	Documentation for the ECMWF grid point model, ECMWF Internal Report 9 (In this report referred to as: HB).
Storer, N.	10/1/76	New permanent file function subroutines, ECMWF Internal documentation.

Acknowledgements

Part of the code was written by Eva Edberg, whom I gratefully acknowledge.

Thanks are due to Mrs. M. Foster-Moore for her patience in typing the manuscript and its subsequent additions and corrections.