European Centre
for Medium Range
Weather Forecasts

Documentation for the E.C.M.W.F. Grid Point Model

Internal Report 9
Research Dept.

May 77

Centre Européen pour les Prévisions Météorologiques
a Moyen Terme

Europäisches Zentrum Für Mittelfristige Wettervorhersagen

DOCUMENTATION FOR THE E.C.M.W.F.
GRID POINT MODEL

by

J. Haseler and D. Burridge

European Centre for Medium Range Weather Forecasts, Bracknell

Internal Report No. 9 May 1977

RESEARCH DEPARTMENT

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

PAGE NUMBER

ABSTRACT

This paper contains a program description of the adiabatic version of the ECMWF grid point model. The mathematical formulation is described in the paper 'A Model for Medium Range Weather Forecasting - Adiabatic Formulation' (BURRIDGE and HASELER) - ECMWF Technical Note No. 4.

Inclusion of physics routines may lead to some modifications to the code.

The program has been written to run on the CDC6600, and there are some machine dependent features, particularly automatic file handling from within the program. These features are indicated later in this documentation.

The program has been written using the Olympus programming system, described in 'ROBERTS - Computer Physics Communications, 7, (1974)'.

CHAPTER 1 - INTRODUCTION

1.1 Program structure

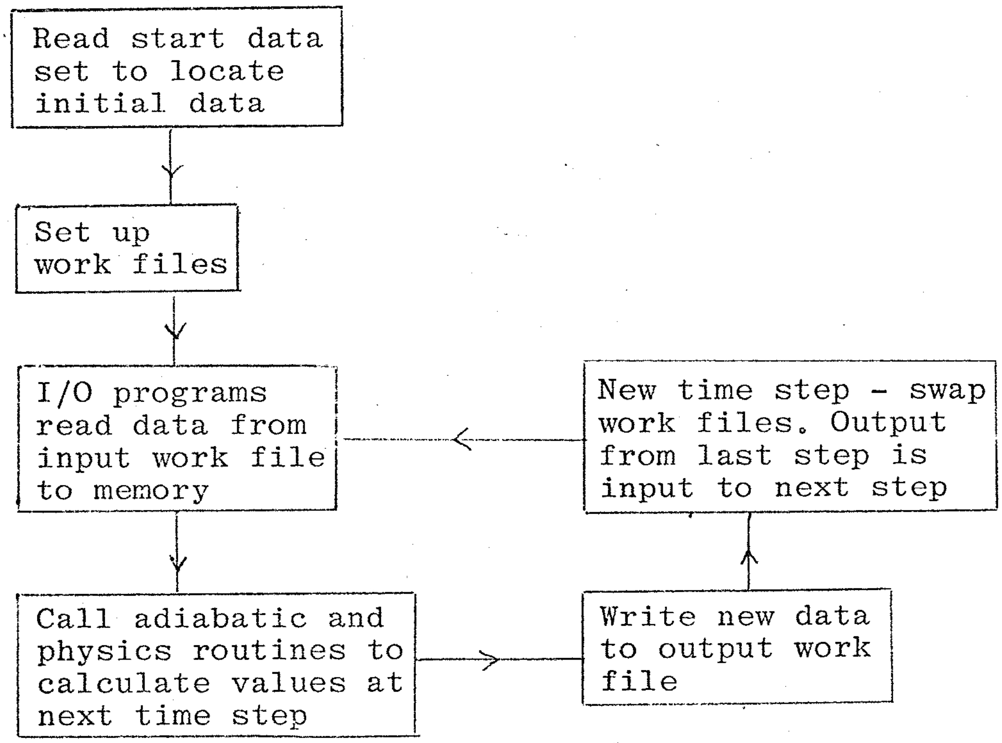


Fig. 1.1.1

Fig. 1.1.1 outlines the basic program flow. First the start data set is read to locate the input data. The start data set contains a record pointing to the initial data, and a record pointing to each forecast data set. Theoretically a run may start from the data file located by any start data set record, but in practice initial runs usually start from the first record, while restart runs carry on from the last record.

Next the initial data is read and work files are generated. There are 2 work files, one for input and one for output.

They each have 1 record for each row of the finite difference grid, containing 2 time levels of data + workspace.

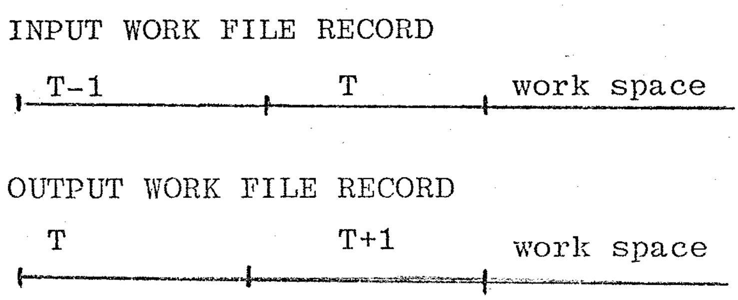


Fig. 1.1.2

These are needed because all the data for the finite difference grid rows cannot be expected to fit in memory at once. Instead, sufficient data to perform the calculations for one row is read from the input work file into memory, and the new values calculated for the next time step are written to the output work file. The I/O subroutines scan from north to south, organising the transfer of data between work files and memory for each row's calculations. At the end of a time step, the work files are swapped, so that the output from the last time step becomes the input for the next step. The old input file is then overwritten by data for the next time step.

The I/O subroutines call the subroutine LINEMS to control the calculations. This calls the subroutine DYN to perform the adiabatic parts of the calculation, and other routines to do the physics. At specified times, forecast data sets are written from LINEMS, and new records pointing to them are added to the start data set. The time stepping scheme is also done in LINEMS.

1.2 Documentation structure

In Chapter 2, the data structure is explained. Chapter 3 describes the different input / output schemes within the model. Chapter 4 describes LINEMS, the subroutine which controls the calculation. The adiabatic calculations, performed in subroutine DYN, are described in Chapter 5. Chapter 6 describes the space filter. Chapter 7 contains an explanation of how the start data set is used, and how it may be created initially. Chapter 8 describes how to create an initial data set and then run the model on the CDC6600. Chapter 9 contains tables for all the common blocks used by the program, explaining the meaning of all the common variables, how they are defined and where they may be redefined. At the start of the chapter there is an index to the common blocks. Chapter 10 contains flow diagrams for all the subroutines. Again there is an index at the beginning of the chapter.

1.3 Notation

The Olympus programming system enables the program code and documentation to be cross-referenced. Olympus programs are split into numbered sub-sections, appearing in the code as:-

```
CL          2.2         READ NEXT ROW

  220     CONTINUE

          ........
```

A reference to this part of the code in the documentation would be given as:-

<2.2>   read next row.

CHAPTER 2 - DATA STRUCTURE

2.1 The grid

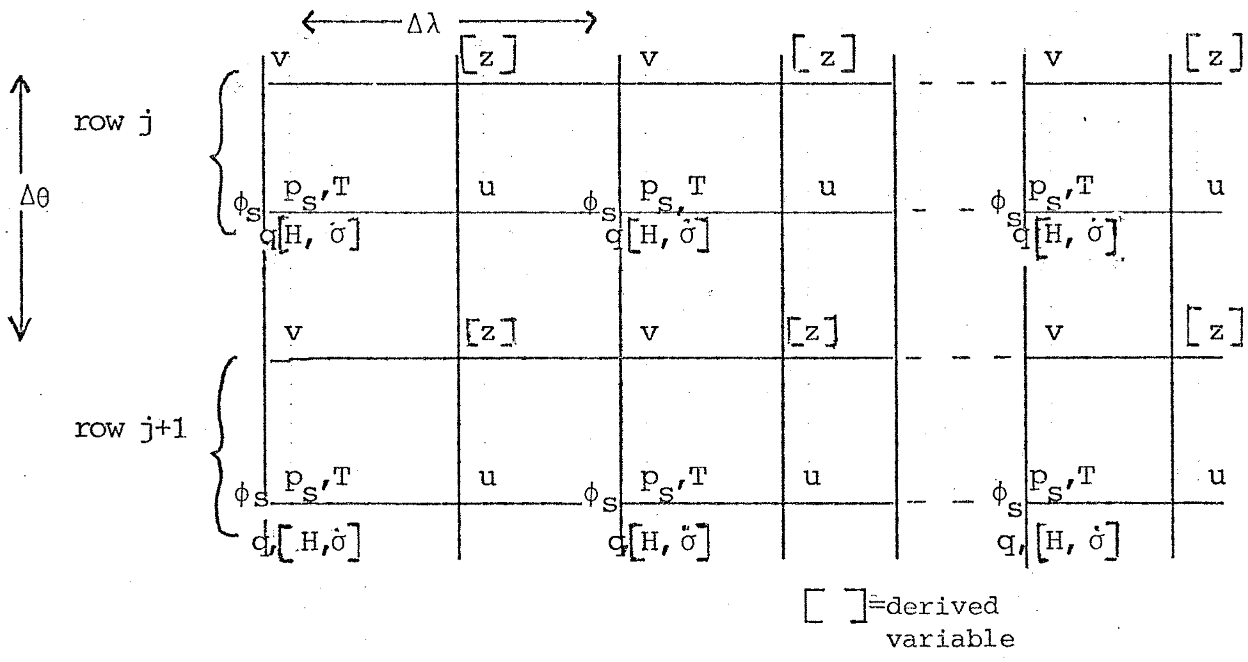
The model has a staggered regular latitude/longitude grid, with a sigma coordinate in the vertical. The variables

- ϕ_s - surface geopotential (geopotential metres)
- p_s - surface pressure (Pa)
- T - temperature (^oK)
- u - E - W velocity component (m/sec)
- v - N - S velocity component (m/sec)
- q - humidity mixing ratio (kg/kg)

are held in the initial data set and temporary work files, while the following variables are derived and stored in the work space in memory :-

- Z - potential absolute vorticity
- H - auxiliary potential
- $\dot{\sigma}$ - vertical velocity

2.2 Horizontal distribution of variables



Throughout this documentation, a 'row' of data will actually mean the data at all longitude points at the latitude of the ϕ_s and u points, plus the data half a grid length to the north, at the latitude of the v points.

The north pole is at the latitude of the ϕ_s and u points of row 1, so that the v points for row 1 are 'to the north of the north pole'. They are preset to 0 and never used in the calculations, but are stored for convenience so that all data rows have the same format. The u values at the pole are not true prognostic variables, but are derived from equations (11) and (12) in Chapter 5, using values of v to the south of the pole. At the pole, NLON (the number of longitude points) identical values of ϕ_s, p_s and q are stored.

The south pole is at the latitude of the ϕ_s and u points of row MAXROW. As at the north pole, NLON identical values of ϕ_s, p_s, T and q are stored, and the u values are calculated. The corresponding v points, half a grid length to the north of the south pole, contain velocities for that latitude.

2.3 Vertical distribution of variables

$\sigma_{\frac{1}{2}}$ ———————— $\dot{\sigma}_1 = 0, \ \sigma = 0$

σ_1 - - - - - - - - - - - - $T_1, u_1, v_1, q_1, [Z_1, H_1]$

$\sigma_{1\frac{1}{2}}$ ———————— $\dot{\sigma}_2$

σ_2 - - - - - - - - - - - - $T_2, u_2, v_2, q_2 \ [Z_2, H_2]$

$\sigma_{NLEV-\frac{1}{2}}$ ———————— $\dot{\sigma}_{NLEV}$

σ_{NLEV} - - - - - - - - - - - - $T_{NLEV}, u_{NLEV}, v_{NLEV}, q_{NLEV}$

$[Z_{NLEV}, H_{NLEV}]$

$\sigma_{NLEV+\frac{1}{2}}$ ———————— $\dot{\sigma} = 0, \quad \sigma = 1$

2.4 Data storage

Data is stored in different ways for

(i) the permanent files on channel NDATA - the initial/restart data set and data files from subsequent write-up times (that is time steps at which a forecast data file is to be saved)

(ii) the temporary work files on channels NWKIN and NWKOUT

(iii) the work buffers in memory.

2.5 Permanent files

The permanent files are organised as sequential data sets. The common blocks COMBAS, COMHKP and COMMAP are stored in the first 3 records, followed by 1 record for each latitude row of data, stored from north to south.

The row records are organised in the following order :-

$\underline{\phi_s}$, $\underline{p_s}$, $\underline{T_1}$, $\underline{T_2}$, ..., $\underline{T_{NLEV}}$, $\underline{u_1}$, ..., $\underline{u_{NLEV}}$, $\underline{v_1}$, ..., $\underline{v_{NLEV}}$, $\underline{q_1}$, ...,

$\underline{q_{NLEV}}$

where $\underline{\phi_s} = \phi_1, \phi_2, \ldots \phi_{NLON}$

i.e. surface geopotential for all longitude points at the given latitude, stored eastwards from the Greenwich meridian

and $\quad \underline{T_k} = T_{k,1}, T_{k,2}, \ldots T_{k,NLON}$

i.e. temperatures at level k for all the longitude points at the given latitude.

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

2.6 Temporary work files

The work files 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 2 time levels, plus extra workspace at the end.

NWKIN - input work file

time T-1	time T	workspace

NWKOUT - output work file

time T	time T+1	workspace

Fig. 2.6.1

The data at a given time-level is organised in the same way as a row record for the permanent files, i.e.

$\underline{\phi_s}, \underline{p_s}, \underline{T_1}$, -- $\underline{T_{NLEV}}$, $\underline{u_1}$, --- $\underline{u_{NLEV}}$, $\underline{v_1}$, --, $\underline{v_{NLEV}}$, $\underline{q_1}$, --, $\underline{q_{NLEV}}$

2.7 Work buffer in memory

```
                    I N P U T

row NROW-1 |--T-1---------|----T-----|work space

      NROW ------------------------------------

      NROW+1-----------------------------------

      NROW+2 -- read row-----------------------



                   O U T P U T

           |--T------|---T+1---|work space  NROW-1
                    write row

           ------------------------------------  NROW
```

Fig. 2.7.1

There are 4 input and 2 output row buffers in core. Each row contains data organised in the same way as a row record of the temporary work files, with 2 time-levels and extra work space. The dynamics require 3 rows of data and while these are being processed, the 4th row is read in parallel from the input work file.

Whilst a new output row is being generated, the previous row is written in parallel to the output work file.

CHAPTER 3 - INPUT / OUTPUT

3.1 I/O between work files and memory

At the start of an I/O cycle, the input work file, on channel NWKIN, contains data for times T-1 and T. Scanning from north to south, this data is read into memory row by row, new output buffers, containing data for times T and T+1 are generated, and these are written to the output work file on channel NWKOUT. At the end of the scan, the work files are swapped, so that the old output file becomes the new input file, and vice versa. On the next scan, the old input file is overwritten with the new T+1, T+2 data.

3.2 Arrangement of data in memory

```
                          INPUT
                                                   row
NLINE1(1) ---------------------------------------- NROW-1

NLINE1(2) ---------------------------------------- NROW

NLINE1(3) ---------------------------------------- NROW+1

               read from NWKIN
NLINE1(4) - - - - - - - - - - - - - - - - - - - - - NROW+2


                          OUTPUT
                                                   row
                 write to NWKOUT
NLINE2(1) - - - - - - - - - - - - - - - - - - - - - NROW-1

NLINE2(2) ---------------------------------------- NROW
```

Fig. 3.2.1

Fig. 3.2.1 shows the data layout in memory when the new values for row NROW (the row counter) are being calculated. The dynamics also require input data from the rows to the north (NROW-1) and south (NROW+1). In parallel with this computation, the next input row (NROW+2) is read from NWKIN, and the previous output row (NROW-1) is written to NWKOUT.

The array NLINE1 contains the displacements in blank common of the beginning of the 4 input buffers, while NLINE2 contains the start addresses of the 2 output buffers. NLINE1(2) always points to the input row which is currently being updated, while NLINE2(2) points to the output row in which the new values are stored.

Cyclic buffering is used for the data. When the computation for row NROW is complete, the program scans 1 row further south, ready to compute the new values for row NROW+1. The input buffer which contained NROW-1 data is overwritten by data for row NROW+3. The input buffer pointers are swapped cyclically, so that the old NLINE1(J) becomes the new NLINE1(J-1) (apart from the old NLINE1(1) which becomes NLINE1(4)).
The output buffers are swapped in a similar fashion.

3.3 Subroutines

The I/O scheme is handled by 4 subroutines.

(i) STEPON (see diagram 13) controls the I/O.

<1.1> STARTN is called to start the I/O at the northern boundary for the first step only.

<1.2> NSSCAN is called to scan from north to south, except at the boundaries.

<1.3> BDYIO is called to perform the I/O control at the southern and northern boundaries. When the last step has been completed, control returns from STEPON. Otherwise the program returns to < 1.2 > to call NSSCAN for the next time step.

(ii) STARTN (see diagram 14) starts the I/O at the northern boundary for the first step only.

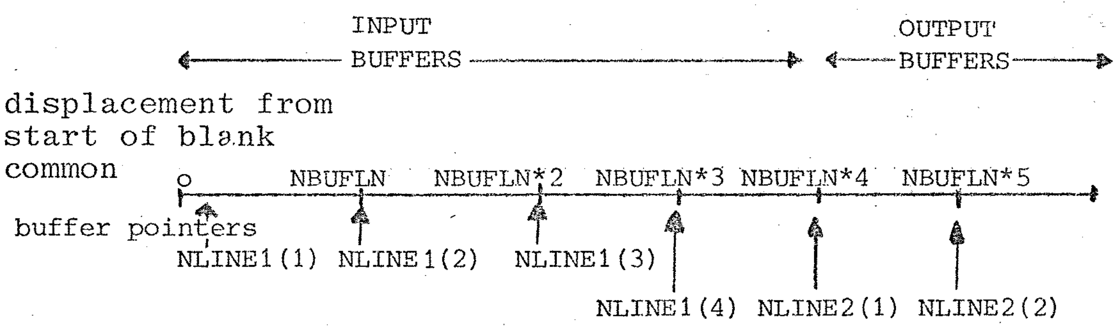


Fig. 3.3.1

<1.1>
<1.2> The 4 input and 2 output buffers are laid out at the start of blank common. Their displacements from the start are stored in the arrays NLINE1 and NLINE2 respectively. Each buffer is of length NBUFLN, where

NBUFLN = 2 time levels of data + extra work space.

NBUFLN, from common COMIOC, is defined in subroutine INITAL (for an initialisation) or RESUME (for a restart).

<2.1> The northern boundary row is read into the buffer to which NLINE1(2) points. Set NROW=1.

<2.2> The next row south, NROW=2, is read into the buffer to which NLINE1(3) points.

<2.3> The read for the next row south, NROW=3,- into the buffer to which NLINE1(4) points, is initialised, and then proceeds in parallel with the computation.

<2.4> LINEMS is called to control the arithmetic calculations for the northern row and store the new values in the buffer to which NLINE2(2) points. Set NROW=NROW+1.

(iii) NSSCAN (see diagram 15) scans from north to south.

<1.1> If NROW = MAXROW-1, i.e. I/O scan has reached southern boundary RETURN.

<1.2> Swap input and output buffers cyclically

<1.3> Initiate read of row NROW+2 into buffer to which NLINE1(4) points.

<1.4> Initiate write of row NROW-1 from buffer to which NLINE2(1) points.

<1.5> Call LINEMS to control the arithmetic calculation. for row NROW and store the new values in the buffer to which NLINE2(2) points. Set NROW = NROW+1. Go to <1.1>

(iv) BDYIO (see diagram 16) performs I/O at the boundaries

<1.0> Set INROW = 1 - the number of the next row to be read. This will be from the northern boundary for the next time step, i.e. from work file NWKOUT, which will become the next input file.
Set IOUTRW = NROW-1 - the number of the next row to be written. This is still for the current time step, and is near the southern boundary.

<1.05> Check that the read for the southern boundary row has finished.

<1.1> Swap input and output buffers cyclically.

<1.2> Initiate read of row INROW from northern boundary.

<1.3> Initiate write of row IOUTRW from southern boundary.

<1.4> Call LINEMS to control the arithmetic calculation for row NROW.
Set INROW = INROW+1, set IOUTRW = IOUTRW+1, set NROW = NROW+1.
If LINEMS was called for 1 row from southern boundary go to <1.1>
If LINEMS was called for northern boundary, RETURN.

<1.5> LINEMS has just been called for the southern boundary row. Test if NSTEP = NSTOP, that is if the last step has just been completed. If so, set NLEND = TRUE, to signify a normal termination, and RETURN.

<1.6> Increment NSTEP, the step number. Set NROW = 1. The next row to be updated will be at the northern boundary. Swap NWKIN and NWKOUT, the work files, so that the old output file becomes the new input file, and vice versa.

<1.65> Test Switch 1. This is a CDC feature, whereby a logical switch may be set externally and tested by the program. Switch 1 is used as a signal to terminate the program prematurely. The switch is tested here at the end of each time step. If it has been set, NSW = 1, unless the next step would normally have been a write-up time, when NSW = 2. LINEMS will then write to a permanent file on the next time step, and terminate the run.

<1.7> Return to < 1.1> to update the northern boundary row.

3.4 I/O for permanent files

The model has been designed to handle its permanent files automatically. The user must attach the start data set to his job through the job control language, but all other files are attached, made permanent or modified by Fortran callable subroutines from within the program. These routines, specially written for the CDC 6600, are not standard Fortran. Other installations may have equivalent Macro instructions. Otherwise file and disk handling would have to be done through the job control language.

3.5 Input from a permanent file

When the model starts an initial or restart run, it searches for the appropriate start data set record (see para. 6.1). This contains enough information to locate the initial data, and attach it to the job.

If NLMNT = TRUE in COMSDS, the data is on a private disk. On the CDC 6600, a private disk has to be 'mounted', that is physically loaded and logically connected to the job. In <2.1> of SUBROUTINES INITAL or RESUME, if NLMNT = TRUE, the private disk is mounted by

CALL MOUNT (IFAIL,NDMTSN,NDMTVS)

where IFAIL is preset to 1 and reset to 0 if the mount is successful.

NDMTSN (from COMSDS) is a Hollerith word containing the private disk set name.

NDMTVS (from COMSDS) is a Hollerith word containing the private disk VSN.

On the CDC 6600, a permanent file is identified to the system by the following parameters :-

(i) a name

(ii) an identifier (ID) which gives the owner of the file

(iii) a cycle number

(iv) passwords, which restrict the use of a file.

The file is connected logically to a job by a local file name or logical unit number.

In <2.2> of INITAL or RESUME, the initial data file is attached to the job by

CALL ATTACH (IFAIL,NDATA,NDTFN,NDTCY,NDTPW)

where IFAIL is preset to 1 and is reset to 0 if the attach is successful.

NDATA (from COMSDS) is the logical unit number

NDTFN(4) (from COMSDS) is a Hollerith array containing the file name, right filled with blanks.

NDTCY (from COMSDS) is the cycle number (NDTCY = 0 gives the highest current cycle number).

NDTPW (ii) (from COMSDS) is a Hollerith array containing the ID, any passwords, any set name, etc.

In <2.5> the common blocks COMBAS, COMHKP and COMMAP are read from the start of the initial data file in SUBROUTINE DATCOM.

3.6 File handling for an initial run - SUBROUTINE INITAL

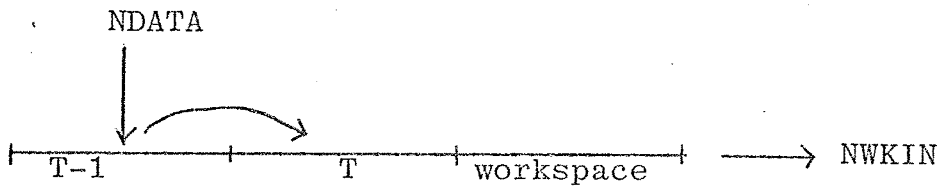


Fig. 3.6.1

The initial data file, on unit NDATA, contains data at one time level only, and the run starts from this with a forward time step. The work files contain space for 2 time levels + extra work space. Row by row from north to south, a data record is read from NDATA into the T-1 space of a buffer in core, then this is copied into the T space and, together with some uninitialised work space, the buffer is written to the input work file on unit NWKIN.

Whereas an initial run starts with a forward time step from a single time level of data, a restart continues from a write-up point with a leapfrog time step,which requires 2 adjacent time levels of data. NDATA contains the data for T-1, and 2 files on units NM1A and NM1B are created to hold the time T data. These are overwritten alternately, so that only the T data for the last 2 write-ups is ever available.

If the user wished to make the program fully restartable from any write-up point, the program could easily be modified so that time T files were treated in the same way as T-1 files. Instead of over-writing an earlier file, a new file would then be generated at each write-up time.

To create a permanent file on the CDC 6600 it is necessary to

(i) set up the permanent file parameters (see para. 3.5)

(ii) 'request permanent file space'. This directs the file to a permanent file device - either a public disk or the user's disk.

(iii) write to the file

(iv) catalogue the file.

The 2 time T files are created in SUBROUTINE INITAL.

<4.0>

(i) The cycle numbers, NM1ACY and NM1BCY (from COMHKP), are set to 0.

(ii) A file name is generated in NM1AFN(4) (from COMHKP) for NM1A by CALL FILENM (NM1AFN,0)

(iii) The passwords NM1APW(11) and NM1BPW(11),(from COMHKP) are set to NDTPW(11), the passwords for the initial data.

(iv) CALL REQUEST (IFAIL,NM1A,NDREQ), requests permanent file space for NM1A,
where IFAIL is preset to 1 and reset to 0 for a successful request
NDREQ(10) (from COMSDS) is a Hollerith array containing the request parameters:-

'*PF." for a public disk, or

'*PF,SN = ...' for a private disk.

<4.2>

(i) ENDFILE NM1A - this creates an empty file
REWIND NM1A

(ii) CALL CATALOG (IFAIL,NM1A,NM1AFN,NM1ACY,NM1APW)-catalogues file NM1A

<4.5>

(i) CALL FILENM (NM1BFN,0)-generates a file name for NM1B in NM1BFN(4) (from COMHKP)

(ii) CALL REQUEST (IFAIL,NM1B,NDREQ)-requests permanent file space for NM1B

<4.7>

(i) ENDFILE NM1B - this creates an empty file
REWIND NM1B

(ii) CALL CATALOG (IFAIL,NM1B,NM1BFN,NM1BCY,NM1BPW)-catalogues file NM1B.

Finally, in subroutine INITAL, the permanent files are returned (i.e. logically disconnected from the job) and, if a private disk is being used, it is dismounted by

<6.6> CALL DSMOUNT(IFAIL,NDMTSN,NDMTVS).

This means that a model which may only write up once every few hours does not need to have a disk mounted all the time.

3.7 File handling for a restart - SUBROUTINE RESUME

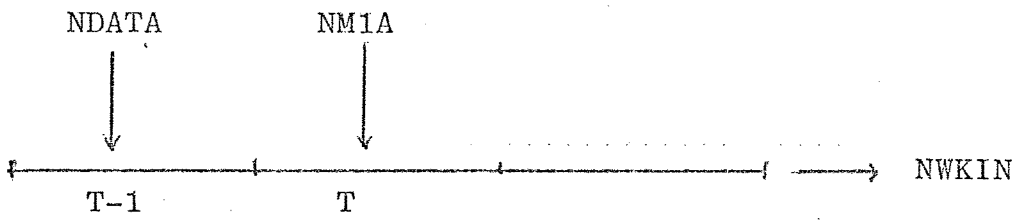


Fig. 3.7.1

As for an initialisation, the initial data on unit NDATA is found from the start data set. SUBROUTINE DATCOM is called to read the common blocks COMBAS,COMHKP and COMMAP from the start of NDATA. There are parameters in COMHKP which enable the data file containing the second time level of data to be attached by

⟨4.0⟩ CALL ATTACH (IFAIL,NM1A,NM1AFN,NM1ACY,NM1APW)

Row by row, from north to south, a record is read from NDATA into the T-1 part of a buffer in core, a record is read from NM1A into the T part of the buffer, and the 2 time levels, together with some uninitialised work space, are written to the input work file, NWKIN.

When all the data has been read, the permanent files are returned, and, if a private disk is being used, it is dismounted.

3.8 Output to the permanent files - SUBROUTINE LINEMS

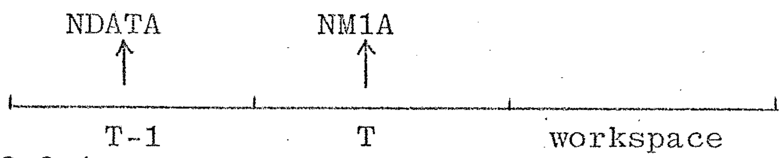


Fig. 3.8.1

At write-up times, data for 2 time levels are saved. The fully time-filtered T-1 field is written to NDATA,while the unfiltered T field is written to NM1A. Different strategies are adopted for handling these 2 files. A new permanent file is created on channel NDATA for each write-up, and it is these files which are preserved to give a record of the run. The files NM1A and NM1B are used solely to hold the second time level for restarts, and are overwritten alternately.

<2.0> Write up if NSTEP = NWRITE + 1 or if switch 1 (see 3.4) has been set. It is data for step NWRITE which is written to NDATA and preserved.

<2.05> If NLMNT = TRUE (i.e. a private disk is being used)
CALL MOUNT (IFAIL,NDMTSN,NDMTVS)

<2.1> Swap NM1A and NM1B, so that the time T file from the last write-up but one will be overwritten.

<2.17> CALL ATTACH (IFAIL,NM1A,NM1AFN,NM1ACY,NM1APW)
Attach the new NM1A file.

<2.2> REWIND NM1A
CALL ALTER (IFAIL,NM1A)

Reposition file NM1A to overwrite it from the beginning. 'Alter' is a CDC feature which permits a sequential file to be overwritten from its current position.

NREC = NREC + 1
CALL DATCOM (NM1A,2)

Update the current start data set record number before writi common blocks COMBAS, COMHKP and COMMAP to the start of NM1A.

<2.25> CALL REQUEST (IFAIL,NDATA,NDREQ)

Request permanent file space for channel NDATA.

NSTEP = NSTEP - 1
CALL DATCOM (NDATA,2)
NSTEP = NSTEP + 1

Write common blocks to the start of NDATA. The step number of the data is one less than the current step number, so NSTEP in COMHKP is adjusted accordingly.

```
                      INPUT
<2.3>  NLINE1(1) ------------------NROW-1
<2.4>  NLINE1(2) ------------------NROW
                  T-1     T
       NLINE1(3) |-----+------+----NROW+1
       NLINE1(4) ------------------
```

Fig. 3.8.2.

Away from the poles the second part of the time filter $\bar{f}(T-1) = f^*(T-1) + \varepsilon f(T)$ is done for row NROW+1.

CHAPTER 4 - SUBROUTINE LINEMS

LINEMS controls the arithmetic calculation. It calls the adiabatic routines and may call physics routines, organises the space and time filters, writes data to the permanent files and carries out the time stepping scheme.

Diagram 17 shows the detailed organisation of the subroutine, but below are some additional explanations.

4.1 Time filter

The time filter used is

$$\bar{f}(T) = f(T) + \varepsilon\{\bar{f}(T - 1) - 2f(T) + f(T + 1)\}$$

where $\bar{f}(T)$ indicates the fully time-filtered value. This is performed in 2 parts.

NLINE1(1)	________________	NROW-1	________________	NLINE2(1)
NLINE1(2)	$\bar{f}(T-1)$ $f(T)$	NROW	$f^*(T)$	NLINE2(2)
NLINE1(3)	________________	NROW+1		
NLINE1(4)	----------------			

Fig. 4.1.1

In <3.0> of LINEMS, the first part is calculated:-

$$f^*(T) = \varepsilon\bar{f}(T - 1) + (1 - 2\varepsilon)\ f(T)$$

This is performed for row NROW, and is how the data for time T is moved into the output buffer. For the first few timesteps (see fig. 4.1.3) the first part of the time filter is not done, and data is copied in <3.1> from the T input buffer to the T output buffer. (The surface geopotential field, which is not modified, is always copied in <1.1> from the input buffer to the T and T+1 fields in the output buffer).

The second part of the time filter is calculated in <1.5>

$$\bar{f}(T - 1) = f^*(T - 1) + \varepsilon f(T)$$

This is actually done at the start of the next time step, and it is done for row NROW+1, except at the northern boundary, where it is also done for row NROW.

________________	NROW-1	________________
________________	NROW	________________
$\bar{f}(T-1)$ $f(T)$	NROW+1	

Fig. 4.1.2

Fig. 4.1.3 shows which parts of the time filter are done in the first steps (starting from step NSTART) for both an initial and restart run.

	Initial run		restart run	
Step	Part 1	Part 2	Part 1	Part 2
NSTART	No	No	Yes	No
NSTART+1	No	No	Yes	Yes
NSTART+2	Yes	No	Yes	Yes
NSTART+3	Yes	Yes	Yes	Yes

Fig. 4.1.3

4.2 Writing to permanent files

This has been described in para. 3.8. It is done in <2> of LINEMS, after the 2nd part of the time filter has been completed, so that fully time-filtered T-1 data is saved.

4.3 Dynamics

The dynamics subroutine, DYN, is called in <4> of LINEMS. This generates tendencies in the T+1 output buffer for row NROW and overwrites various fields in the T input buffer for rows NROW and NROW-1.

	INPUT	OUTPUT
NROW-1		
NROW	$\bar{f}(T-1)$, $f(T)$	$f^*(T)$, DYN$\rightarrow f'(T)$
NROW+1		

4.4 Space filter

The space filter is described in Chapter 6. It is called in <5> of LINEMS if NLSPFL = .TRUE.

4.5 Explicit timestep

The explicit timestep is done in <6> of LINEMS:

$$f(T + 1) = \bar{f}(T - 1) + 2\Delta t\ f'(T)$$

where $\bar{f}(T-1)$ is the fully time-filtered value. The first step of an initial run is done with the forward step:-

$$f(T + 1) = f(T) + \Delta t \, f'(T)$$

u is not predicted at either pole, while v is not predicted to the north of the north pole. In <6.1> and <6.2> the tendencies for these fields are set to 0, so that they can be treated in the same way as the other fields in <6.3>, the loop which generates the T+1 fields.

CHAPTER 5 - SUBROUTINE DYN- the adiabatic calculation

The adiabatic calculations are done in a single, very large subroutine. This description is split into different sections :

5.1 The equations are given, with each term numbered, to simplify the later program description.

5.2 A diagram is given to show how the data is laid out in memory initially, and how fields are overwritten.

5.3 An explanation is given of all the field displacement names used in the program.

5.4 An outline flow diagram is given, to show the order in which different terms in the equations are computed.

5.5 A written description of DYN is given, which cross-references the detailed flow diagram, diag. 23, in Chapter 10.

5.1 Equations

$$\frac{\partial u}{\partial t} = \overset{(1.1)}{\frac{1}{\cos\theta}\,[ZV\cos\theta]} - \overset{(1.2)}{\frac{1}{a\cos\theta}\delta_\lambda\{\bar{\phi}^\sigma + E\}} - \overset{(1.3)}{\frac{R\bar{T}^\lambda}{a\cos\theta}\delta_\lambda(\ln p_s)} - \overset{(1.4)}{\frac{1}{\overline{p_s}^\lambda}\,\overline{\frac{\overline{p_s\dot{\sigma}}^\lambda\,\Delta_\sigma u}{\Delta_\sigma\sigma}}^\sigma} + F_u \qquad \text{(1}$$

$$\frac{\partial v}{\partial t} = \overset{(2.1)}{-[ZU]} - \overset{(2.2)}{\frac{1}{a}\delta_\theta\{\bar{\phi}^\sigma + E\}} - \overset{(2.3)}{\frac{R\bar{T}^\theta}{a}\delta_\theta(\ln p_s)} - \overset{(2.4)}{\frac{1}{\overline{p_s}^\theta}\,\overline{\frac{\overline{p_s\dot{\sigma}}^\theta\,\Delta_\sigma v}{\Delta_\sigma\sigma}}^\sigma} + F_v \qquad \text{(2}$$

where $V = \overset{(1.5)}{\bar{p}_s^\theta\, v}$ v-mass flux

$U = \overset{(2.5)}{\overline{p_s}^\lambda\, u}$ u-mass flux

$E = \frac{1}{2}\big(\overset{(1.6)}{\overline{u^2}^\lambda} + \overset{(1.7)}{\frac{1}{\cos\theta}\,\overline{v^2\cos\theta}^\theta}\big)$ kinetic energy

$Z = \frac{1}{\overline{p_s a\cos\theta}^{\lambda\theta}}\cdot\{\overset{(1.8)}{af\cos\theta} + \overset{(1.9)}{\delta_\lambda v} - \overset{(1.10)}{\delta_\theta u\cos\theta}\}$ vorticity

$\phi_{k+\frac{1}{2}} = \phi_s + \sum_{\ell=k+1}^{NLEV} \overset{(1.11)}{RT_\ell\,(\Delta_\sigma \ln\sigma)_\ell}$ geopotential

$$\frac{\partial p_s}{\partial t} = -\frac{1}{a\cos\theta}\sum_{\ell=1}^{NLEV} \overset{(3.1)}{\{\delta_\lambda U + \delta_\theta(V\cos\theta)\}_\ell(\Delta\sigma)_\ell} \qquad \text{(3}$$

$$p_s\dot{\sigma}_{k+\frac{1}{2}} = -\overset{(4.1)}{\sigma_{k+\frac{1}{2}}\frac{\partial p_s}{\partial t}} - \frac{1}{a\cos\theta}\sum_{\ell=1}^{k} \overset{(4.2)}{\{\delta_\lambda U + \delta_\theta(V\cos\theta)\}_\ell(\Delta_\sigma\sigma)_\ell} \qquad \text{(4}$$

$$\frac{\partial T}{\partial t} = -\frac{1}{p_s}\left\{\frac{1}{a\cos\theta}\left(\overset{(5.1)}{\overline{U\delta_\lambda T}^{\lambda}} + \overset{(5.2)}{\overline{V\cos\theta\,\delta_\theta T^{\theta}}}\right) + \overset{(5.3)}{\frac{\overline{p_s\dot{\sigma}\;\Delta_\sigma T}^{\sigma}}{\Delta_\sigma \sigma}} - \overset{(5.4)}{\left(\frac{\kappa T\omega}{\sigma}\right)}\right\} + Q \tag{5}$$

where

$$\frac{\kappa T\omega}{\sigma} = \frac{\kappa T}{\sigma}\Big(\overset{(5.5)}{\overline{\sigma\frac{\partial p_s}{\partial t} + p_s\dot{\sigma}}^{\sigma}}\Big) + \frac{\kappa}{a\cos\theta}\left\{\overset{(5.6)}{\overline{U\overline{T}^{\lambda}\delta_\lambda(\ell n p_s)}^{\lambda}} + \overset{(5.7)}{\overline{V\cos\theta\,\overline{T}^{\theta}\delta_\theta(\ell n p_s)}^{\theta}}\right\}$$

$$\frac{\partial q}{\partial t} = -\frac{1}{p_s}\left\{\frac{1}{a\cos\theta}\left(\overset{(6.1)}{\overline{U\delta_\lambda q}^{\lambda}} + \overset{(6.2)}{\overline{V\cos\theta\,\delta_\theta q}^{\theta}}\right) + \overset{(6.3)}{\frac{\overline{p_s\dot{\sigma}\Delta_\sigma q}^{\theta}}{\Delta_\sigma \sigma}}\right\} + S \tag{6}$$

<u>Equations at poles</u>

$$\frac{\partial p_{s,p}}{\partial t} = -.\sum_{\ell=1}^{NLEV}\left\{SIGN\,\frac{a\Delta\lambda}{\varepsilon}\overset{(7.1)}{\sum_{i=1}^{NLON}}(V\cos\theta)_{p-\frac{1}{2},i}\right\}_\ell(\Delta_\sigma\sigma)_\ell \tag{7}$$

$$p_s\dot{\sigma}_{k+\frac{1}{2}} = -\overset{(7.2)}{\sigma_{k+\frac{1}{2}}\frac{\partial p_{s,p}}{\partial t}} - \sum_{\ell=1}^{k}\left\{SIGN.\frac{a\Delta\lambda}{\varepsilon}\overset{(7.3)}{\sum_{i=1}^{NLON}}(V\cos\theta)_{p-\frac{1}{2},i}\right\}_\ell(\Delta_\sigma\sigma)_\ell \tag{8}$$

where SIGN = -1 at North pole

= +1 at South pole

ε = area of polar cap

$$= NLON.\tfrac{1}{2}a\Delta\lambda\cos\left(\frac{\pi}{2} - \frac{\Delta\theta}{2}\right).\frac{a\Delta\theta}{2} \qquad (7.4)$$

$$\frac{\partial T_p}{\partial t} = -\frac{1}{p_s}\left\{SIGN.\frac{a\Delta\lambda}{\varepsilon}\overset{(9.1)}{\sum_{i=1}^{NLON}\left(V\cos\theta_{p-\frac{1}{2},i}\;\overline{T}^{\theta}_{p-\frac{1}{2},i}\right)} + \overset{(9.2)}{\frac{\overline{p_s\dot{\sigma}_p\Delta_\sigma T_p}^{\sigma}}{\Delta_\sigma\sigma}} - \frac{\kappa T\omega}{\sigma}\right\} + Q_p \tag{9}$$

$$\text{where } \left(\frac{\kappa T\omega}{\sigma}\right)_p = \overset{(9.4)}{\overline{\frac{\kappa T_p}{\sigma}\left(\frac{\sigma\,\partial p_{s,p}}{\partial t} + p_{s,p}\right)}^{\sigma}} + \frac{\kappa}{a\frac{1}{2}\cos\theta_{p-\frac{1}{2}}} \,x$$

$$x\,\frac{1}{NLON}\overset{(9.5)}{\sum_{i=1}^{NLON} V_{p-\frac{1}{2},i}\cos\theta_{p-\frac{1}{2}}\,.\,\overline{T}^{\theta}_{p-\frac{1}{2},i}\,x\,\delta_\theta(\ell n p_s)_{p-\frac{1}{2},i}}$$

$$\frac{\partial q_p}{\partial t} = -\frac{1}{p_s}\{\text{SIGN}.\frac{a\Delta\lambda}{\varepsilon}\overset{(10.1)}{\sum_{i=1}^{NLON}(V_{p-\frac{1}{2},i}\cos\theta_{p-\frac{1}{2}}\overline{q}^{\theta}_{p-\frac{1}{2},i})} +$$

$$+ \overset{(10.2)}{\overline{p_s\dot{\sigma}_p\Delta_\sigma q_p}^{\sigma}}\ \} + S_p \qquad (10)$$

There is no u equation at the poles, but a zonal polar mass flux, U_p, used in the nearest v equation to the pole is derived from :

$$(U_{p,i+\frac{1}{2}} - U_{p,i-\frac{1}{2}}) \times \frac{a\Delta\theta}{2} + \text{SIGN}.V_{p-\frac{1}{2},i}\cos\theta_{p-\frac{1}{2}}.a\Delta\lambda =$$

$$= \text{SIGN}.\frac{1}{NLON}\sum_{i=1}^{NLON} V_{p-\frac{1}{2},i}\cos\theta_{p-\frac{1}{2}}.a\Delta\lambda \qquad (11)$$

$$\text{and} \quad \sum_{i=1}^{NLON} U_{p,i+\frac{1}{2}} = 0 \qquad (12)$$

The v equation (2) is valid at the nearest v point to the pole, if

$$E_p = \frac{1}{NLON}\cdot\frac{1}{\frac{1}{2}\cos\theta_{p-\frac{1}{2}}}\overset{(13.1)}{\sum_{i=1}^{NLON}\tfrac{1}{2}(v_{p-\frac{1}{2},i})^2\cos\theta_{p-\frac{1}{2}}}$$

$$\text{and} \quad \overset{(13.2)}{Z_{p+\frac{1}{2},i+\frac{1}{2}} = Z_{p-\frac{1}{2},i+\frac{1}{2}}}$$

5.2 Organisation of the time T data + workspace from the input buffers, showing how the data is overwritten in subroutine DYN.

	← NLON →	← NLON →	← NLON → *NLEV	← NLON → *NLEV	— NLON — *NLEV	← NLON* → NLEV	← NLON →	← NLON → *NLEV	← NLON → *NLEV	← NLON →
NLINE1 (1) + LREC	(ϕ_1)	<19.1> 6 IRPT 5 ~~IPT1~~	10 IVT1	15 ITLNP1	19 IVFLX2	23 IVQ1	<8.1> 28 IDLNPL	32 IH1	36 IPSIG1	<8.1> 41 IRPS <5.2> 40 ~~IPSMN~~
NLINE1 (2) + LREC	1 IFI2	< 8.1> 4 IPT2 3 ~~IPS2~~	<18.1> 9 IVT2 8 ~~IT2~~	<15.6> 14 ITLNP2 <8.2> 13 ~~IUFLX2~~ 12 ~~IU2~~	<8.2> 18 IVFLX3 17 ~~IV2~~	<18.1> 22 IVQ2 21 ~~IQ2~~	<8.1> 27 IDLNPT 26 ~~ILNPS2~~	<6.1> 31 IH2 30 ~~IUZ1~~	<13.2> 35 IPSIG2 34 ~~IZ2~~	<19.1> 39 IRPL2 38 ~~IPL2~~
NLINE1 (3) + LREC	(ϕ_3)	2 IPS3	7 IT3	11 IU3	16 IV3	20 IQ3	<8.07> 25 ILNPS3 <7.1> 24 ~~IKE~~	<9.4> 29 IUZ2	<5.1> 33 IZ3	<5.2> 37 IPL3
	← Time T					→	← Workspace			→

An explanation of the contents of each of the numbered fields is given overleaf.

In the diagram

<8.1>
IPT2
~~IPS2~~

indicates that the field IPS2 is overwritten by the field IPT2 in section 8.1 of DYN.

5.3 Displacement names

Displacements in blank common are stored in local variables, whose names usually end with 1, 2 or 3 to indicate whether the data is for the northern row (row 1), central row (row 2) or southern row (row 3). Below there is an explanation of the fields whose displacements were numbered in diag. 5.2. The notation $f_{i,j,k}$ is used where i = 1, 2 or 3 is the row number

j = longitude point number

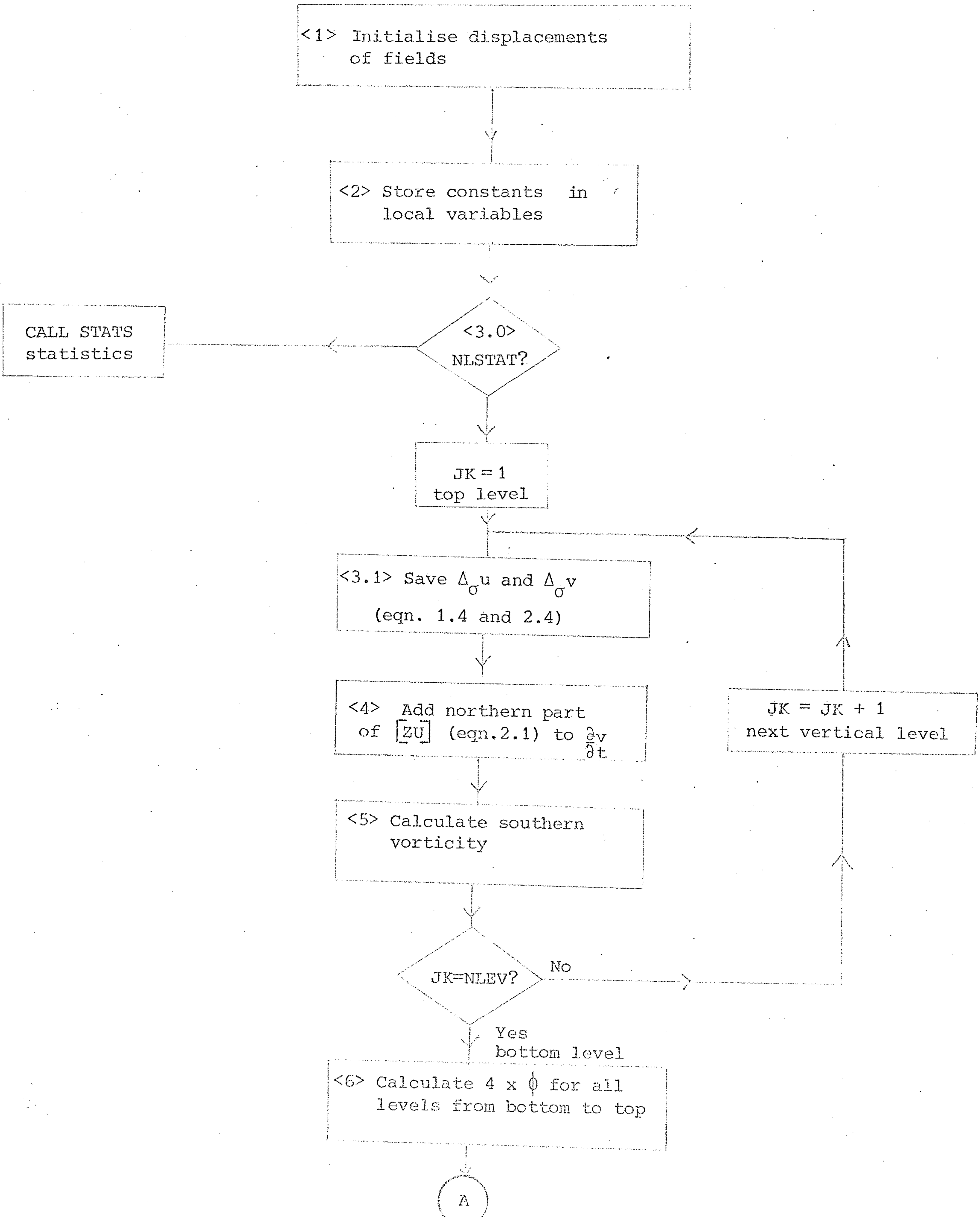
k = level number

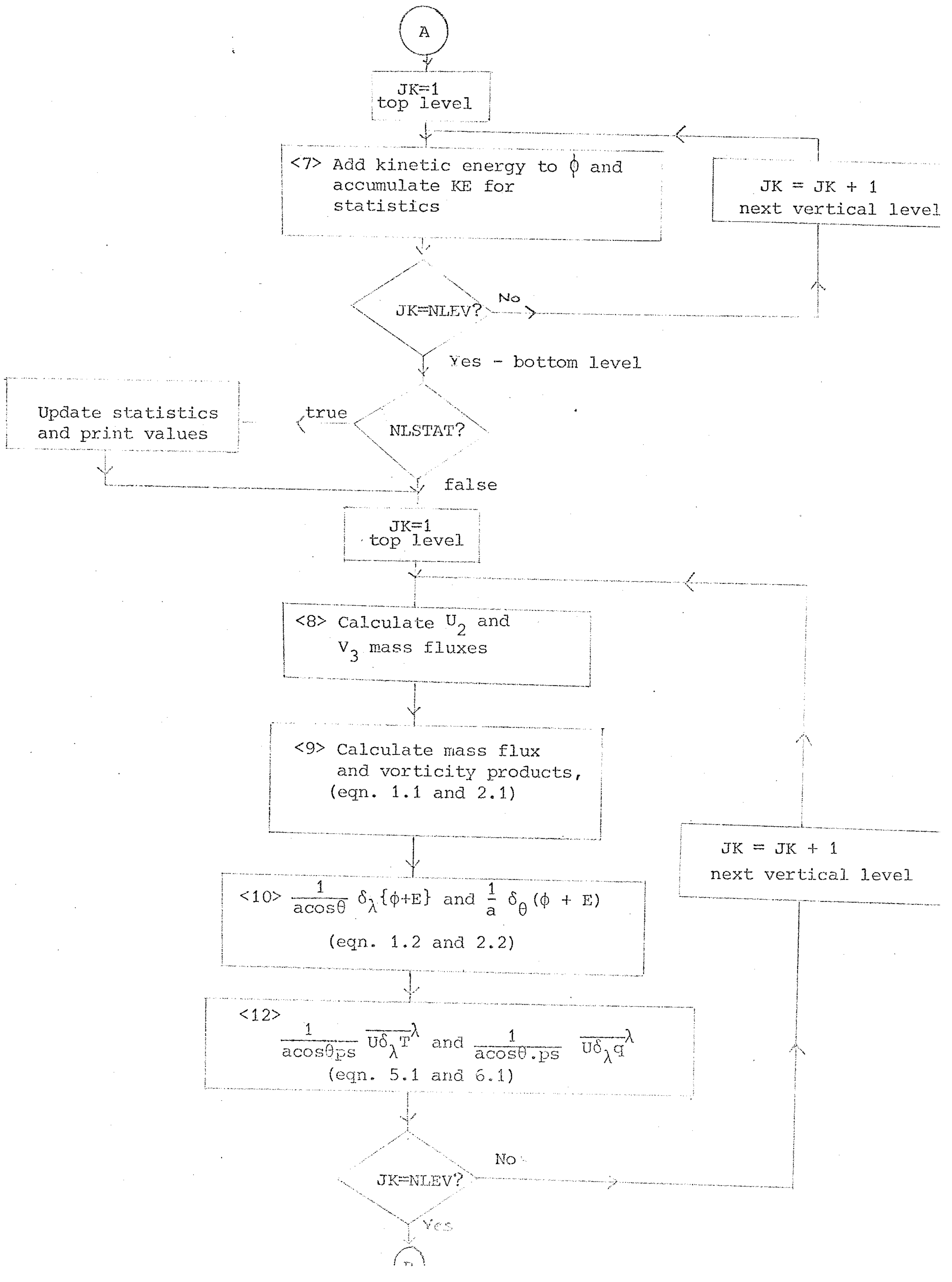
1.	IF12	surface geopotential	$\phi_{s2,j}$
2.	IPS3	surface pressure	$ps_{3,j}$
3.	IPS2	" "	$ps_{2,j}$
4.	IPT2	2xN/S pressure mean	$ps_{2,j} + ps_{3,j}$
5.	IPT1	" "	$ps_{1,j} + ps_{2,j}$
6.	IRPT1	reciprocal N/S pressure mean	$1/(ps_{1,j} + ps_{2,j})$
7.	IT3	temperature	$T_{3,j,k}$,
8.	IT2	"	$T_{2,j,k}$,
9.	IVT2		$V_{3,j,k}(T_{2,j,k} - T_{3,j,k})$
10.	IVT1		$V_{2,j,k}(T_{1,j,k} - T_{2,j,k})$
11.	IU3	u velocity	$u_{3,j,k}$
12.	IU2	"	$u_{2,j,k}$
13.	IUFLX2	u mass flux	$U_{2,j,k} = \frac{1}{2}(ps_{2,j}+ps_{2,j+1})u_{2,j,k}$
14.	ITLNP2	Rx mean Tx$\frac{\partial(\ell n p_s)}{\partial d\theta}$	$\frac{R}{2a\Delta\theta}(T_{2,j,k}+T_{3,j,k})\times(\ell nps_{2,j} - \ell nps_{3,j})$
15.	ITLNP1	" "	$\frac{R}{2a\Delta\theta}(T_{1,j,k}+T_{2,j,k})(\ell nps_{1,j} - \ell nps_{2,j})$
16.	IV3	v velocity	$v_{3,j,k}$

17.	IV2	v velocity	$v_{2,j,k}$
18.	IVFLX3	v mass flux	$V_{3,j,k} = \frac{1}{2\cos\theta_{v3}}(ps_{2,j}+ps_{3,j})v_{3,j,k}$
19.	IVFLX2	" " "	$V_{2,j,k} = \frac{1}{2\cos\theta_{v2}}(ps_{1,j}+ps_{2,j})v_{2,j,k}$
20.	IQ3	moisture content	$q_{3,j,k}$
21.	IQ2	" "	$q_{2,j,k}$
22.	IVQ2		$V_{3,j,k}(q_{2,j,k}-q_{3,j,k})$
23.	IVQ1		$V_{2;j,k}(q_{1,j,k}-q_{2,j,k})$
24.	IKE	4xkinetic energy of level	$\sum_{i,j} ps_{2,j}(u^2_{2,j,k}+u^2_{2,j-1,k}+\frac{1}{\cos\theta_{u2}}\{\cos\theta_{v2}v^2_{2,j,k}+\cos\theta_{v3}v^2_{3,j,k}\})$
25.	ILNPS3	log surface pressure	$\ell n(ps_{3,j})$
26.	ILNPS2	" "	$\ell n(ps_{2,j})$
27.	IDLNPT	$\Delta_\theta(\ell n\ ps)$	$\ell n(ps_{2,j})-\ell n(ps_{3,j})$
28.	IDLNPL	$\Delta_\lambda(\ell n\ ps)$	$\ell n(ps_{2,j+1})-\ n(ps_{2,j})$
29.	IUZ2	vorticity x u flux	
30.	IUZ1	" "	
31.	IH2	auxiliary potential	ϕ + KE
32.	IH1	" "	" "
33.	IZ3	$\frac{1}{6}$ x vorticity	$\frac{1}{6\Delta\lambda}\{v_{3,j+1,k}-v_{3,j,k}\}-\frac{1}{6\Delta\theta}\{u_{2,j,k}\cos\theta_{u2}-u_{3,j,k}\cos\theta_{u3}\}$

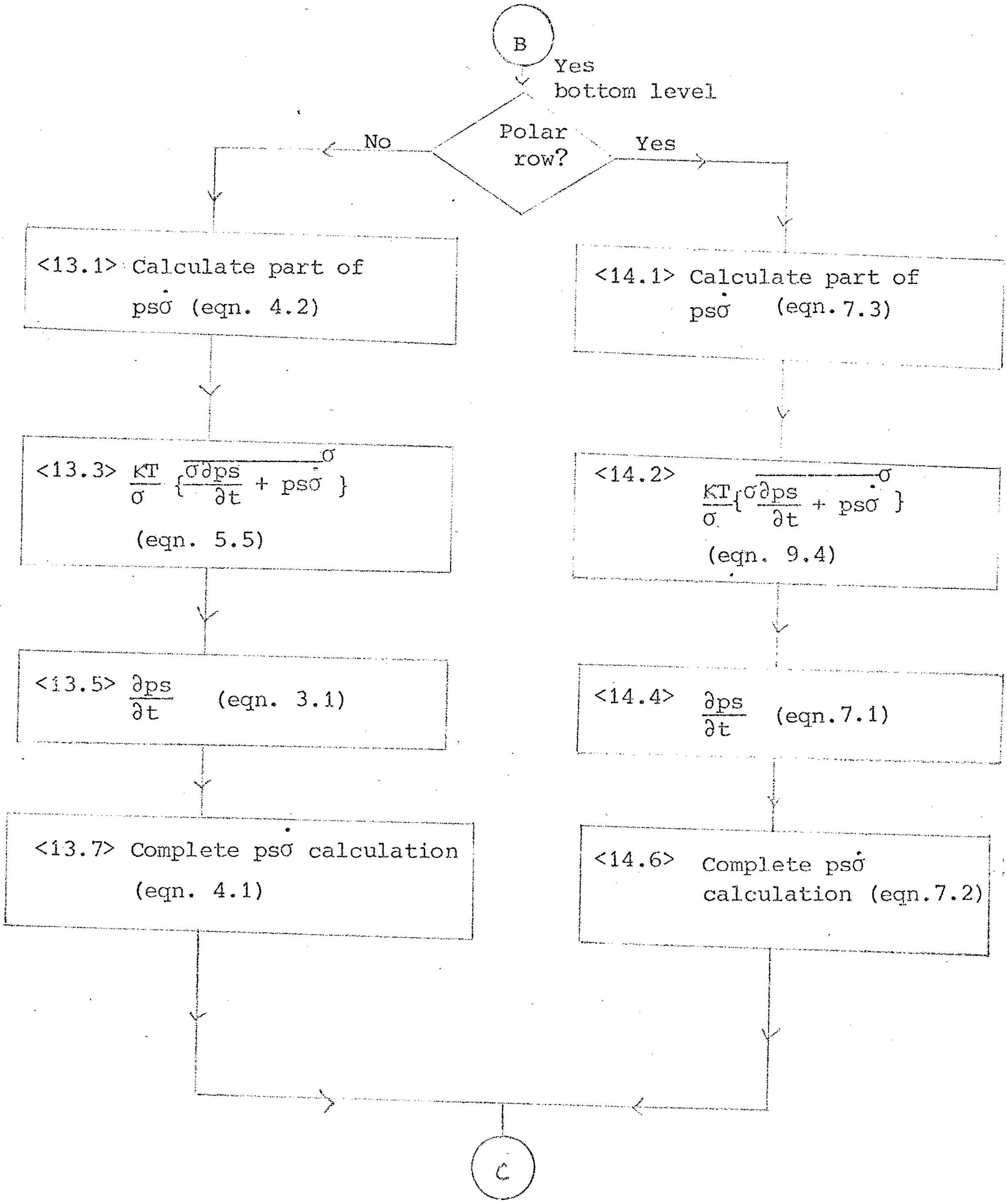
34.	IZ2	$\frac{1}{6}$ x vorticity	$\frac{1}{6\Delta\lambda}\{v_{2,j+1,k}-v_{2,j,k}\} - \frac{1}{6\Delta\theta}\{u_{1,j,k}\cos\theta_{u1} - u_{2,j,k}\cos\theta_{u2}\}$
35.	IPSIG2	ps x $\frac{d\sigma}{dt}$	
36.	IPSIG1	" "	
37.	IPL3	$\overline{ps_3}^{\lambda}$	$\frac{1}{2}(ps_{3,j+1}+ps_{3,j})$
38.	IPL2	"	$\frac{1}{2}(ps_{2,j+1}+ps_{2,j})$
39.	IRPL2	$\frac{1}{2\overline{ps}^{\lambda}}$	$^1/(ps_{2,j+1}+ps_{s,j})$
40.	IPSMN	$1/\overline{ps}^{\lambda\theta}$	$^1/\{\cos\theta_{u2}\cdot\frac{1}{2}(ps_{2,j+1}+ps_{2,j}) + \cos\theta_{u3}\cdot\frac{1}{2}(ps_{3,j+1}+ps_{3,j})\}$
41.	IRPS	$\frac{1}{ps}$	$^1/ps_{2,j}$

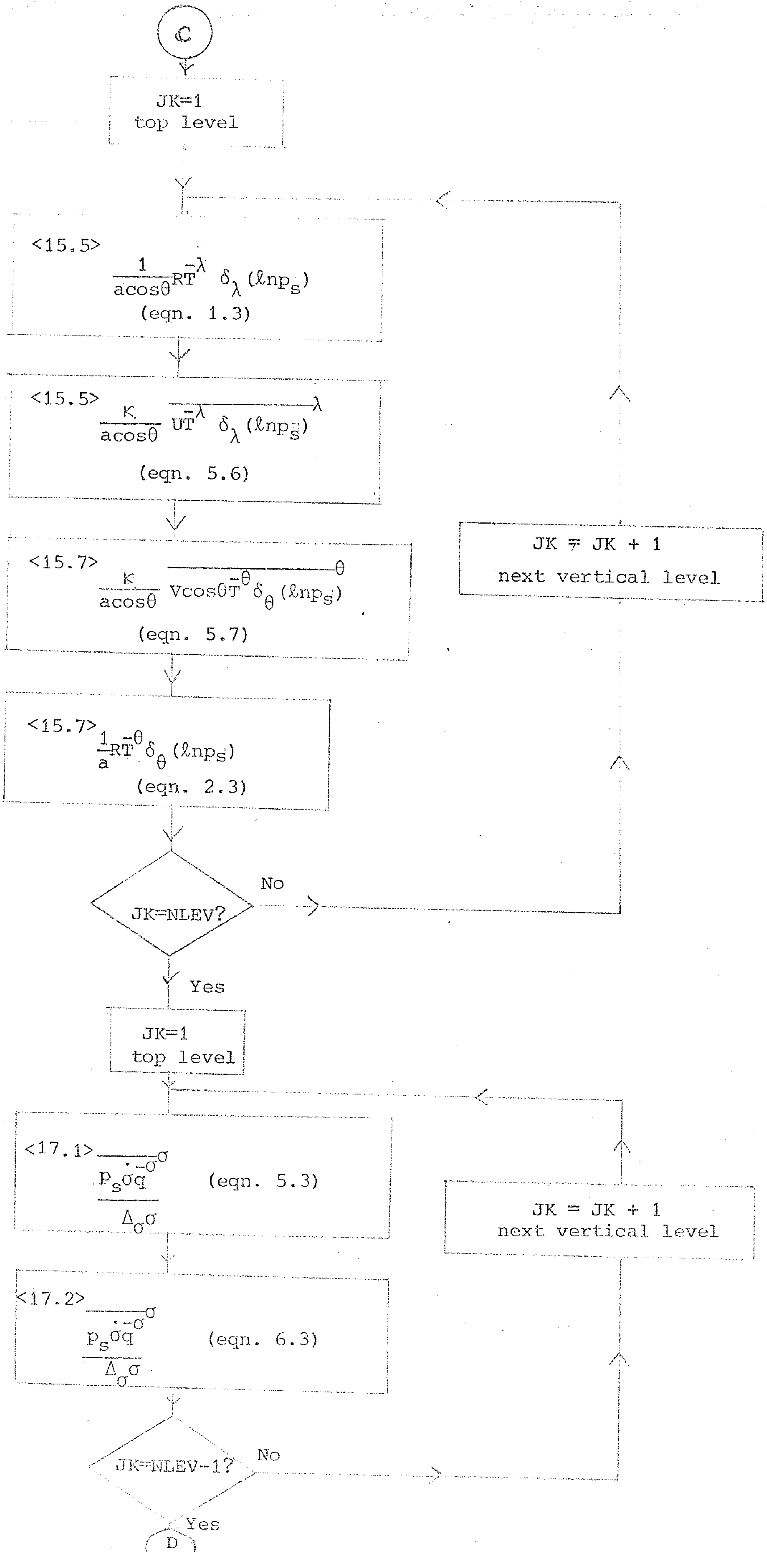
5.4 SUBROUTINE DYN - OUTLINE FLOW DIAGRAM

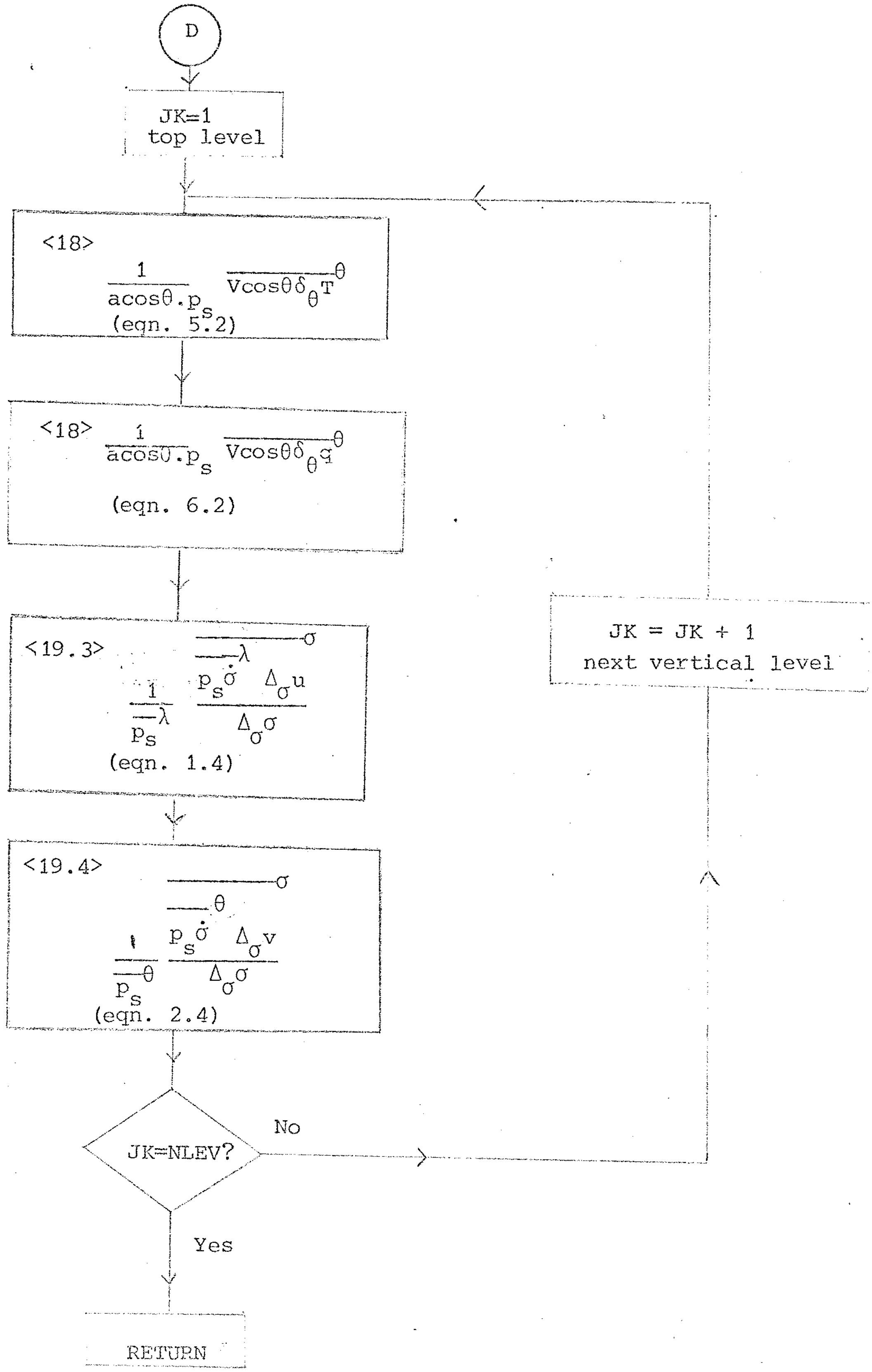




A
JK=1
top level
<7> Add kinetic energy to ϕ and accumulate KE for statistics
JK=NLEV?
No
JK = JK + 1
next vertical level
Yes - bottom level
NLSTAT?
true
Update statistics and print values
false
JK=1
top level
<8> Calculate U_2 and V_3 mass fluxes
<9> Calculate mass flux and vorticity products, (eqn. 1.1 and 2.1)
<10> $\frac{1}{a\cos\theta}\,\delta_\lambda\{\phi+E\}$ and $\frac{1}{a}\,\delta_\theta(\phi+E)$
(eqn. 1.2 and 2.2)
<12>
$\frac{1}{a\cos\theta ps}\,\overline{U\delta_\lambda T}^{\lambda}$ and $\frac{1}{a\cos\theta.ps}\,\overline{U\delta_\lambda q}^{\lambda}$
(eqn. 5.1 and 6.1)
JK=NLEV?
No
JK = JK + 1
next vertical level
Yes


B
Yes
bottom level
Polar row?
No
Yes
<13.1> Calculate part of $ps\dot{\sigma}$ (eqn. 4.2)
<13.3> $\frac{KT}{\sigma}\{\overline{\frac{\sigma\partial ps}{\partial t} + ps\dot{\sigma}}^{\sigma}\}$ (eqn. 5.5)
<13.5> $\frac{\partial ps}{\partial t}$ (eqn. 3.1)
<13.7> Complete $ps\dot{\sigma}$ calculation (eqn. 4.1)
<14.1> Calculate part of $ps\dot{\sigma}$ (eqn. 7.3)
<14.2> $\frac{KT}{\sigma}\{\overline{\frac{\sigma\partial ps}{\partial t} + ps\dot{\sigma}}^{\sigma}\}$ (eqn. 9.4)
<14.4> $\frac{\partial ps}{\partial t}$ (eqn. 7.1)
<14.6> Complete $ps\dot{\sigma}$ calculation (eqn. 7.2)
C


C
JK=1
top level
<15.5>
$\frac{1}{a\cos\theta}\overline{RT}^{\lambda}\,\delta_\lambda(\ell n p_s)$
(eqn. 1.3)
<15.5>
$\frac{\kappa}{a\cos\theta}\overline{\overline{U T}^{\lambda}\,\delta_\lambda(\ell n p_s)}^{\lambda}$
(eqn. 5.6)
<15.7>
$\frac{\kappa}{a\cos\theta}\overline{V\cos\theta\overline{T}^{\theta}\,\delta_\theta(\ell n p_s)}^{\theta}$
(eqn. 5.7)
<15.7>
$\frac{1}{a}\overline{RT}^{\theta}\,\delta_\theta(\ell n p_s)$
(eqn. 2.3)
JK=NLEV?
No
Yes
JK = JK + 1
next vertical level
JK=1
top level
<17.1>
$\frac{\overline{p_s\dot{\sigma}\overline{q}^{\sigma}}}{\Delta_\sigma\sigma}$
(eqn. 5.3)
<17.2>
$\frac{\overline{p_s\dot{\sigma}\overline{q}^{\sigma}}}{\Delta_\sigma\sigma}$
(eqn. 6.3)
JK = JK + 1
next vertical level
JK=NLEV-1?
No
Yes
D


D
JK=1
top level
<18>
1 / acosθ.p_s   Vcosθδ_θT ^θ
(eqn. 5.2)
<18>
1 / acosθ.p_s   Vcosθδ_θq ^θ
(eqn. 6.2)
<19.3>
(eqn. 1.4)
<19.4>
(eqn. 2.4)
JK=NLEV?
No
Yes
RETURN
JK = JK + 1
next vertical level

5.5 Description of SUBROUTINE DYN

This subroutine description cross-references the code, and complements the detailed flow diagram, diag. 23 in Chapter 10.

<1> Initialise displacements of fields. See para. 5.2, for the layout of the fields, and para. 5.3 for an explanation of the contents of each of the fields.

<2> Store constants in local variables, see para. 9.3 for an explanation of the constants in COMMAP.

<3.05> CALL STATS if NLSTAT = .TRUE.. SUBROUTINE STATS uses data from the time T input buffer, so it has to be called at the start of DYN, before any of the fields which it needs have been overwritten.

Start a vertical loop for JK = 1 to NLEV. ILEV gives the displacement from the start of a field of the data for level JK.

<3.1> For non-polar rows, except at the top level, store $\Delta_\sigma u$ in IDUDSG and $\Delta_\sigma v$ in IDVDSG where:

$$\Delta_\sigma u = u_{2,JL,JK} - u_{2,JL,JK-1}$$

$$\Delta_\sigma v = v_{2,JL,JK} - v_{2,JL,JK-1}$$

(eqn. 1.4 and 2.4)

<3.2> At south pole, store $\Delta_\sigma v$ in IDVDSG (eqn. 2.4).

<4.1> Fig.5.4.1 shows the finite difference operators used to evaluate the rotation terms $\frac{1}{\cos\theta}[\overline{ZV\cos\theta}]$ (eqn. 1.1) and $[\overline{ZU}]$ (eqn. 2.1).

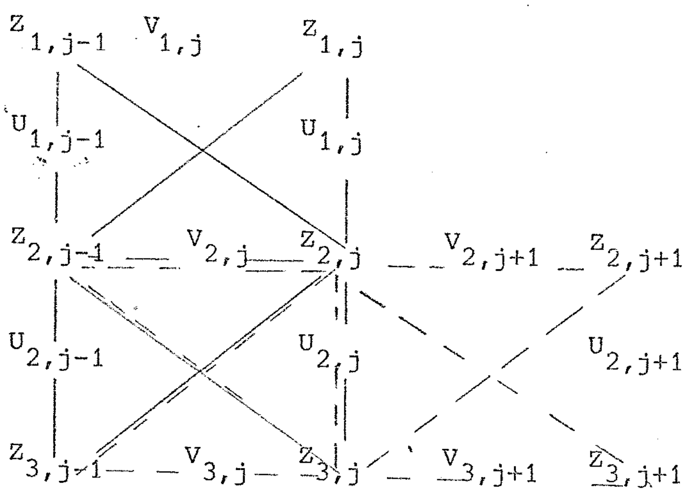


Fig. 5.4.1

[ZU] is calculated from the average over the solid triangles of the product of the U mass flux with the mean vorticity for the triangle.

$$\text{so}\qquad \overset{(2.1)}{[ZU]_{2,j}} = \frac{1}{4}\{\overset{(2.1.1)}{U_{1,j-1}\cdot\frac{1}{3}(Z_{1,j-1} + Z_{2,j-1} + Z_{2,j})} +$$

$$+ \overset{(2.1.2)}{U_{1,j}\cdot\frac{1}{3}(Z_{1,j} + Z_{2,j-1} + Z_{2,j})} +$$

$$+ \overset{(2.1.3)}{U_{2,j-1}\cdot\frac{1}{3}(Z_{2,j-1} + Z_{3,j-1} + Z_{2,j})} +$$

$$+ \overset{(2.1.4)}{U_{2,j}\cdot\frac{1}{3}(Z_{2,j} + Z_{3,j-1} + Z_{3,j})}\}$$

[ZVcosθ] is calculated from the average over the dashed triangles of the product of the V mass flux with the mean vorticity for the triangle.

$$\text{so}\qquad [ZV\cos\theta]_{2,j} = \frac{1}{4}\{\overset{(1.1.1)}{V_{2,j}\cos\theta_{v2}\cdot\frac{1}{3}(Z_{2,j-1} + Z_{2,j} + Z_{3,j})} +$$

$$+ \overset{(1.1.2)}{V_{2,j+1}\cos\theta_{v2}\cdot\frac{1}{3}(Z_{2,j} + Z_{2,j+1} + Z_{3,j})} +$$

$$+ \overset{(1.1.3)}{V_{3,j}\cos\theta_{v3}\cdot\frac{1}{3}(Z_{2,j-1} + Z_{2,j} + Z_{3,j})} +$$

$$+ \overset{(1.1.4)}{V_{3,j+1}\cos\theta_{v3}\cdot\frac{1}{3}(Z_{2,j} + Z_{3,j} + Z_{3,j+1})}\}$$

To calculate all these terms at once would require 3 rows of Z and 2 rows of U. Instead, terms (2.1.1) and (2.1.2) are calculated in the previous call to DYN and stored in the field $IUZ1_3$, so that Z_1 and U_1 are not needed.

In <4.1> the terms in IUZ1 are accumulated into $\frac{\partial v}{\partial t}$.

<5> Calculate $\frac{a}{6}$ x southern vorticity, and store in IZ3.

<5.1> Except at south pole, calculate
$\frac{1}{6}$ x$\{\delta_\lambda v - \delta_\theta(u\cos\theta)\}$ (eqn. 1.9 and 1.10)

$$IZ3_{JL,JK} = \frac{1}{6\Delta\lambda}(v_{3,JL+1,JK} - v_{3,JL,JK}) - \frac{1}{6\Delta\theta}(u_{2,JL,JK}\cos\theta_{u2} - u_{3,JL,JK}\cos\theta_{u3})$$

<5.2> Save surface pressure means at first level, since they only have to be calculated once for all the levels.

At north pole, set $\overline{ps}_2^\lambda = ps_2$

$$IPL2_{JL} = IPS2_1$$

<5.25> Except at south pole, save $\bar{p}_{s_3}^\lambda$ in IPL3 and $\frac{1}{2\overline{p_s\cos\theta}^{\lambda\theta}}$ in IPSMN.

$$IPL3_{JL} = \tfrac{1}{2}(p_{s_{3,JL}} + p_{s_{3,JL+1}})$$

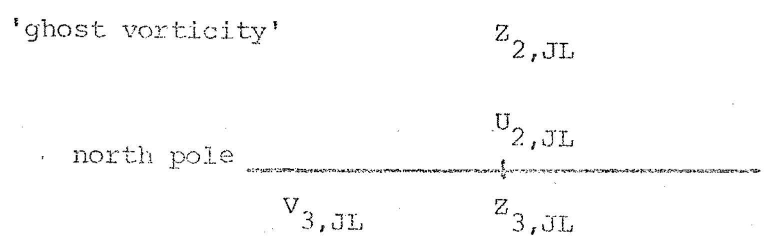
$$IPSMN_{JL} = 1/\{\cos\theta_{u2}\cdot\overline{ps}^\lambda_{2,JL} + \cos\theta_{u3}\overline{ps}^\lambda_{3,JL}\}$$

<5.3> Except at south pole, add $\frac{1}{6}$ x coriolis force to IZ3, and divide by mean surface pressure. (eqn. 1.8)

$$IZ3_{JL,JK} = IPSMN_{JL}*(IZ3_{JL,JK} + \frac{1}{6}\, fa\cos\theta_{v3})$$

<5.4> At the north pole, set the 'ghost vorticity', ½ a grid length to the 'north of the north pole', equal to the genuine vorticity ½ grid length to the south of the pole (eqn. 13.2)

$$IZ2_{JL,JK} = IZ3_{JL,JK}$$

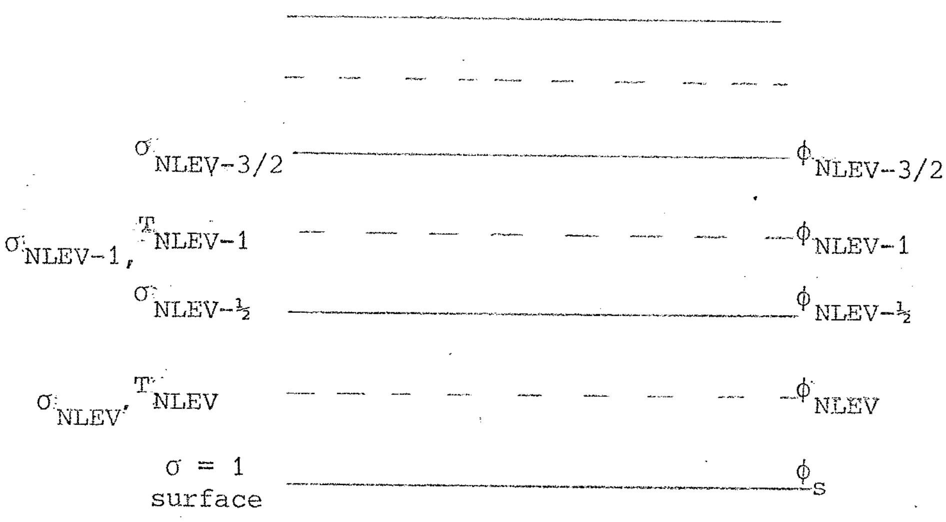


<5.5> Similarly, set up ghost vorticities at the south pole:

$$IZ3_{JL,JK} = IZ2_{JL,JK}$$

<6> Integrate hydrostatic equation, and store $4 \times \phi$ in IH2 (eqn. 1.11)

Integrating the hydrostatic equation from the surface gives ϕ at levels $\sigma_{k-\frac{1}{2}}$. ϕ at levels σ_k is found by taking a weighted average of the adjacent values.



<6.1> Bottom level:

$$2\phi_{NLEV-\frac{1}{2}} = 2\phi_s + 2RT_{NLEV}\,\Delta(\ell n\sigma)_{NLEV}$$

$$4\phi_{NLEV} = \alpha(NLEV).2\phi_s + \beta(NLEV).2\phi_{NLEV-\frac{1}{2}}$$

$\alpha(JK)$ and $\beta(JK)$ are weighting factors which can account for the position of level σ_k relative to $\sigma_{k+\frac{1}{2}}$ and $\sigma_{k-\frac{1}{2}}$. We have chosen $\alpha(JK = \beta(JK) = 1$.

<6.2> Middle levels:

$$2\phi_{JK-\frac{1}{2}} = 2\phi_{JK+\frac{1}{2}} + 2RT_{JK}\,\Delta(\ell n\sigma)_{JK}$$

$$4\phi_{JK} = 2\alpha(JK)\phi_{JK+\frac{1}{2}} + 2\beta(JK)\phi_{JK-\frac{1}{2}}$$

<6.3> Top level:

$$4\phi_1 = 2\alpha(1).\phi_{1+\frac{1}{2}} + 2\beta(1) \times (\phi_{1+\frac{1}{2}} + RT_1\Delta(\ell n\sigma)_1)$$

<7> Add kinetic energy to geopotential to form auxiliary potential in IH2. Except at poles

$$KE = \tfrac{1}{2}\left(\overline{u^2}^{\lambda} + \frac{1}{\cos\theta}\,\overline{v^2\cos\theta}^{\theta}\right) \quad \text{(eqn. 1.6 and 1.7)}$$

<7.1> $$IH2_{JL,JK} = IH2_{JL,JK} + \frac{\cos\theta_{v3}}{\cos\theta_{u2}}\, v^2_{3,JL,JK} + \frac{\cos\theta_{v2}}{\cos\theta_{u2}}\, v^2_{2,JL,JK}$$

<7.2> $$IH2_{JL,JK} = \tfrac{1}{4}\left(IH2_{JL,JK} + u^2_{2,JL,JK} + u^2_{2,JL-1,JK}\right)$$

At poles $$KE = \frac{1}{NLON}\cdot\frac{1}{\frac{1}{2}\cos\theta_{p-\frac{1}{2}}}\sum_{JL=1}^{NLON} \tfrac{1}{2} v^2_{p-\frac{1}{2},JL}\cos\theta_{p-\frac{1}{2}} \quad \text{(eqn.13.1)}$$

If NLSTAT = TRUE accumulate kinetic energy statistics in STKE, where

$$STKE = \frac{1}{NLON\sum_{J=1}^{MAXROW}\text{cosx}(J)}\Big\{\sum_{JK=1}^{NLEV}(\Delta_{\sigma}\sigma)_{JK}\times$$

$$\times\Big[\sum_{J=2}^{MAXROW-1}\text{cosx}(J)\sum_{JL=1}^{NLON} p_s(JL,J)\Big\{\tfrac{1}{2}\overline{u^2}^{\lambda}_{JL,JK,J} +$$

$$+\frac{1}{\cos\theta}\overline{v^2\cos\theta}^{\theta}_{JL,JK,J}\Big\} + \text{cosx}(1)\sum_{JL=1}^{NLON} ps(JL,1)\,v^2_{JL,JK,2} +$$

$$+\text{cosx}(MAXROW)\sum_{JL=1}^{NLON} p_s(JL,MAXROW)\,v^2_{JL,JK,MAXROW}\Big]\Big\}$$

where cosx(J) = cos (latitude of u points in row J), except at poles

= $\frac{1}{4}$ x cos (latitude of v points in row 2) at north pole

= $\frac{1}{4}$ x cos (latitude of v points in row MAXROW) at south pole

<8> Save surface pressure means and mass fluxes.

All terms involving u_2 and v_2 have now been calculated, so that they can now be overwritten by

$$U_{2,JL,JK} = \tfrac{1}{2}(ps_{2,JL} + ps_{2,JL+1})u_{2,JL,JK} \quad \text{i.e. IUFLX2}$$

$$V_{3,JL,JK}\cos\theta_{v3} = \tfrac{1}{2}(ps_{2,JL} + ps_{3,JL})v_{3,JL,JK}\cos\theta_{v3} \quad \text{i.e. IVFLX3}$$

<8.3> At the poles, U_2 is derived from the relations:

$$\frac{\partial U}{\partial \lambda} + \frac{\partial (V\cos\theta)}{\partial \theta} = \frac{1}{2\pi}\int_0^{2\pi} \frac{\partial (V\cos\theta)}{\partial \theta}\, d\lambda \qquad \text{(eqn. 11)}$$

$$\int_0^{2\pi} U d\lambda = 0 \qquad \text{(eqn. 12)}$$

<9> The rotation terms $\overline{ZU}$ and $\overline{ZV\cos}$ are evaluated. See <4> above for a full description of these terms.

In <9.1> the southern part of $\overline{ZU}$ is calculated (the northern part was added in <4>). In <9.2> and <9.3> $\overline{ZV\cos\theta}$ is evaluated. In <9.4> the northern part of $\overline{ZU}$ for the next row is stored in IUZ2.

<10> Except at poles, evaluate $\frac{1}{a\cos\theta}\,\partial_\lambda(\bar{\phi}^\sigma + E)$ (eqn. 1.2)

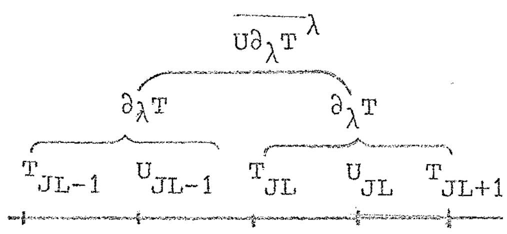
$$= \frac{1}{a\cos\theta_{u2}}(IH2_{JL+1,JK} - IH2_{JL,JK})$$

Except at north pole, evaluate $\frac{1}{a}\partial_\theta(\bar{\phi}^\sigma + E)$ (eqn. 2.2)

$$= \frac{1}{a}(IH2_{JL,JK} - IH3_{JL,JK})$$

<12> Except at poles, evaluate $\frac{1}{a\cos\theta}\overline{U\partial_\lambda T}^\lambda$ (eqn. 5.1)

$$= \frac{1}{a\cos\theta_{u2}}\frac{1}{2\Delta\lambda}(U_{2,JL,JK}(T_{2,JL+1,JK} - T_{2,JL,JK})$$



$$+ U_{2,JL-1,JK}(T_{2,JL,JK} - T_{2,JL-1,JK}))$$

<12.1> Except at poles, evaluate $\frac{1}{a\cos\theta}\overline{U\partial_\lambda q}^\lambda$ (eqn. 6.1)

<13> Integrate continuity equation at internal rows. At top level of model, $\sigma_{\frac{1}{2}} = 0$ and $p_s\dot{\sigma}_{\frac{1}{2}} = 0$.

$$\begin{array}{ll}
\sigma_{\frac{1}{2}} \text{ - - - - - - - - - - - - - -} & \sigma_{\frac{1}{2}} = 0,\ p_s\dot{\sigma}_{\frac{1}{2}} = 0 \\
\sigma_{1} \text{ ―――――――――――――} & T_1 \\
\sigma_{1\frac{1}{2}} \text{ - - - - - - - - - - - - - -} & p_s\dot{\sigma}_{1\frac{1}{2}} \\
\sigma_{2} \text{ ―――――――――――――} & T_2
\end{array}$$

<13.1> Solve (eqn. 4), from top to bottom

$$\left(p_s\dot{\sigma} + \sigma\frac{\partial p_s}{\partial t}\right)_{k+\frac{1}{2}} = \left(p_s\dot{\sigma} + \sigma\frac{\partial p_s}{\partial t}\right)_{k-\frac{1}{2}} -$$

$$- \frac{1}{a\cos\theta}\{\delta_\lambda U + \delta_\theta(V\cos\theta)\}_k(\Delta_\sigma(\sigma))_k$$

<13.2> At the bottom, with $p_s\dot{\sigma} = 0$ and $\sigma = 1$, this gives (eqn. 3) for $\frac{\partial p_s}{\partial t}$

<13.2> At the same time, find (eqn. 9.4)

$$\frac{\kappa T_k}{\sigma_k}\overline{\left(\sigma\frac{\partial p_s}{\partial t} + p_s\dot{\sigma}\ \right)}^{\sigma} = \frac{\kappa T_k}{\sigma_k}\ \frac{1}{2}\{\alpha(k)\left(\sigma\frac{\partial p_s}{\partial t} + p_s\dot{\sigma}\right)_{k+\frac{1}{2}} +$$

$$+ \beta(k)\ \left(\sigma\frac{\partial p_s}{\partial t} + p_s\dot{\sigma}\right)_{k+\frac{1}{2}}\}$$

We have chosen $\alpha(k) = \beta(k) = 1$.

<13.3> $p_s\dot{\sigma}_{k+\frac{1}{2}} = \left(p_s\dot{\sigma} + \sigma\frac{\partial p_s}{\partial t}\right)_{k+\frac{1}{2}} - \sigma_{k+\frac{1}{2}}\frac{\partial p_s}{\partial t}$

<14> Integrate continuity equation at poles. This is similar to <13>, except for

$$\left(p_s\sigma + \sigma\frac{\partial p_s}{\partial t}\right)_{k+\frac{1}{2}} = \left(p_s\sigma + \sigma\frac{\partial p_s}{\partial t}\right)_{k-\frac{1}{2}} -$$

$$- \text{SIGN}.\frac{a\Delta\lambda}{\varepsilon}\sum_{JL=1}^{NLON}\{(V\cos\theta)_{p-\frac{1}{2},JL,k}\}(\Delta_\sigma\sigma)_k \qquad \text{(eqn. 8)}$$

where SIGN = -1 at north pole
+1 at south pole

ε = area of polar cap

$$= \text{NLON}.\frac{a\Delta\lambda}{2}\cos\left(\frac{\pi}{2}-\frac{\Delta\theta}{2}\right)\cdot\frac{a\Delta\theta}{2}$$

<15> Except at poles, add $\frac{-\overline{R\overline{T}}^{\lambda}}{a\cos\theta}\delta_\lambda(\ell np_s)$ to $\frac{\partial u}{\partial t}$(eqn. 1.3)

and $\frac{\kappa}{a\cos\theta}\overline{U\overline{T}^{\lambda}\delta_\lambda(\ell np_s)}^{\lambda}$ to $\frac{\partial T}{\partial t}$ (eqn. 5.6)

All terms involving U_2 have now been completed, so that this field can be overwritten by $\frac{R}{a}\overline{T}^{\theta}\delta_\theta(\ell np_s)$ in ITLNP2.

<15.6> Except at the north pole, add $\frac{-R\overline{T}^{\theta}}{a}\delta_\theta(\ell np_s)$ to $\frac{\partial v}{\partial t}$(eqn.2.3)

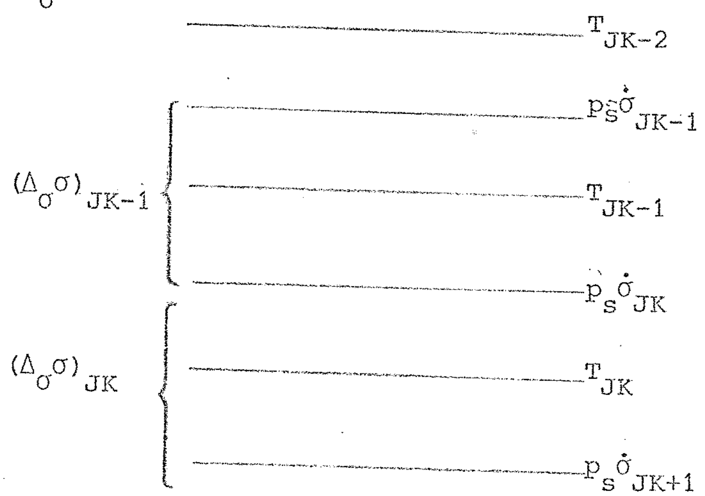
Except at both poles, add $\frac{\kappa}{a\cos\theta}\overline{V\cos\,\overline{T}^{\theta}\delta_\theta(\ell np_s)}^{\theta}$ to $\frac{\partial T}{\partial t}$

(eqn. 5.7)

<16> At the poles, add $\frac{2\kappa}{a\cos\theta_{p-\frac{1}{2}}.\text{NLON}}\sum_{JL=1}^{\text{NLON}}(V\cos\theta\overline{T}^{\theta}\delta_\theta(\ell np_s))_{p-\frac{1}{2},JL}$

to $\frac{\partial T}{\partial t}$ (eqn. 9.5)

<17> Add $\frac{\overline{p_s\dot{\sigma}\Delta\sigma T}^{\sigma}}{\Delta_\sigma\sigma}$ to $\frac{\partial T}{\partial t}$ (eqn. 5.3)



This is calculated in 2 steps:

$$\frac{\partial T}{\partial t}JL,JK-1 = \frac{\partial T}{\partial t}JL,JK-1 - p_s\dot{\sigma}_{2,JL,JK}(T_{2,JL,JK} - T_{2,JL,JK-1})\cdot\frac{1}{2(\Delta_\sigma\sigma)_{JK-1}}$$

and $$\frac{\partial T}{\partial t}JL,JK = \frac{\partial T}{\partial t}JL,JK - p_s\dot{\sigma}_{2,JL,JK}(T_{2,JL,JK} - T_{2,JL,JK-1})\cdot\frac{1}{2(\Delta_\sigma\sigma)_{JK}}$$

<17.2> Add $\dfrac{\overline{p_s\dot{\sigma}\Delta_\sigma q}^{\sigma}}{\Delta_\sigma \sigma}$ to $\frac{\partial q}{\partial t}$ (eqn. 6.3).

<18> All terms involving T_2 and q_2 have now been computed, so they can now be overwritten by the fluxes

$$IVT2 = V\cos\theta_3\ \Delta_\theta(T)$$
$$= V_{3,JL,JK}\cos\theta_{v3}(T_{2,JL,JK} - T_{3,JL,JK})$$

and $IVQ2 = V_{3,JL,JK}\cos\theta_{v3}(q_{2,JL,JK} - q_{3,JL,JK})$

<18.2> Add $\frac{-1}{a\cos\theta}\ \overline{V\cos\theta\,\delta_\theta T}^{\theta} = \frac{1}{a\cos\theta_{u2}}\frac{1}{2\Delta\theta}((V\cos\theta)_{2,JL,JK}\times$

$$\times(T_{1,JL,JK} - T_{2,JL,JK}) +$$
$$+ (V\cos\theta)_{3,JL,JK}(T_{2,JL,JK} -$$
$$- T_{3,JL,JK}))$$

to $\frac{\partial T}{\partial t}$ (eqn. 5.2).

Add $\frac{-1}{a\cos\theta}\ \overline{V\cos\theta\,\delta_\theta q}^{\theta}$ to $\frac{\partial q}{\partial t}$ (eqn. 6.2).

<18.25>Up till now, $p_s.\frac{\partial T}{\partial t}$ and $p_s\frac{\partial q}{\partial t}$ have been calculated.

Divide by p_s to get $\frac{\partial T}{\partial t}$ and $\frac{\partial q}{\partial t}$

<18.4> At poles, add $-\frac{SIGN.\ 2}{NLON.a\Delta\theta.\cos\theta_{p-\frac{1}{2}}}\sum_{JL=1}^{NLON} V_{p-\frac{1}{2},JL}\cos\theta\,\overline{T}^{\sigma}_{p-\frac{1}{2},JL}$

to $\frac{\partial T}{\partial t}$ (eqn. 9.1).

Add a similar term to $\frac{\partial q}{\partial t}$.

<19> Except at poles, add $\frac{-1}{\overline{p_s}^{\lambda}}\frac{\overline{p_s\dot{\sigma}}^{\lambda}\Delta_\sigma u}{\Delta_\sigma\sigma}$ to $\frac{\partial u}{\partial t}$ (eqn. 1.4).

<19.4> Except at south pole, add $\frac{-1}{\overline{p_s}^{\theta}}\frac{\overline{\overline{p_s\dot{\sigma}}^{\sigma}\Delta_\sigma v}^{\sigma}}{\Delta_\sigma\sigma}$ to $\frac{\partial v}{\partial t}$ (eqn. 2.4).

CHAPTER 6 - THE SPACE FILTER

6.1 Theory

The space operator $\underline{\underline{\mathcal{F}}}(\theta) = \underline{\underline{\Phi}}^{-1}\,\underline{\underline{\Lambda}}(\theta)\,\underline{\underline{\Phi}}$ is used to filter the tendencies of all the model's dependent variables. $\underline{\underline{\Phi}}$ is a finite Fourier transform operator and $\underline{\underline{\Phi}}^{-1}$ its inverse. $\underline{\underline{\Lambda}} = \text{diag}\,\{\Lambda_1(\theta),\ldots,\Lambda_N(\theta)\}$ where $\Lambda_k(\theta)$ is the reduction factor for the k^{th} Fourier mode. Let $F_J = F_J(\theta)$ be the grid point values of a particular field at equally spaced grid points around the latitude circle θ ($J = 1$ corresponds to the Greenwich meridian). Then

$$F_J = \frac{A_o}{2} + \sum_{k=1}^{\frac{NLON}{2}}{}^{1}\,\{A_k\cos(k(J-1)\tfrac{2\pi}{NLON}) + B_k\sin(k(J-1)\tfrac{2\pi}{NLON})\}$$

$$J = 1\ \ldots,\ NLON$$

where

$$B_{(\frac{NLON}{2})} = 0$$

and

$\sum^1$ means a factor of ½ in the last term.

The Fourier coefficients A_k and B_k are given by

$$A_k = \frac{2}{NLON}\sum_{J=1}^{NLON} F_J\cos(k(J-1)\tfrac{2\pi}{NLON})\ \text{for}\ k=0,\ldots,\frac{NLON}{2}$$

and

$$B_k = \frac{2}{NLON}\sum_{J=1}^{NLON} F_J\sin(k(J-1)\tfrac{2\pi}{NLON})\ \text{for}\ k=1,\ldots,(\frac{NLON}{2} - 1).$$

The filtered field $(\underline{\underline{\mathcal{F}}}F)_J$ is given by

$$(\mathcal{F}F)_J = \frac{A_o}{2} + \sum_{k=1}^{\frac{NLON}{2}}{}^{1}\ \Lambda_k(\theta)\{A_k\cos(k(J-1)\tfrac{2\pi}{NLON}) + B_k\sin(k(J-1)\tfrac{2\pi}{NLON})\}$$

$$= F_J + \sum_{k=1}^{\frac{NLON}{2}}{}^{1}\ (\Lambda_k(\theta)-1)\{A_k\cos(k(J-1)\tfrac{2\pi}{NLON}) + B_k\sin(k(J-1)\tfrac{2\pi}{NLON})\}.$$

Define for $J = 1,\ldots,\frac{NLON}{4} + 1$ (NLON is assumed to be divisible by 4)

$$F_J^{++} = (F_J + F_{NLON+2-J}) + (F_{\frac{NLON}{2}+2-J} + F_{\frac{NLON}{2}+J})$$

$$F_J^{+-} = (F_J + F_{NLON+2-J}) - (F_{\frac{NLON}{2}+2-J} + F_{\frac{NLON}{2}+J})$$

$$F_J^{-+} = (F_J - F_{NLON+2-J}) + (F_{\frac{NLON}{2}+2-J} - F_{\frac{NLON}{2}+J})$$

$$F_J^{--} = (F_J - F_{NLON+2-J}) - (F_{\frac{NLON}{2}+2-J} - F_{\frac{NLON}{2}+J})$$

where $F_{NLON+1} \equiv F_1$

Then

$$F_J^{++} = 2A_o + 4 \sum_{k=2,2}^{\frac{NLON}{2}1} A_k \cos(k(J-1)\frac{2\pi}{NLON})$$

$$F_J^{+-} = 4 \sum_{k=1,2}^{(\frac{NLON}{2}-1)} A_k \cos(k(J-1)\frac{2\pi}{NLON})$$

$$F_J^{-+} = 4 \sum_{k=1,2}^{(\frac{NLON}{2}-1)} B_k \sin(k(J-1)\frac{2\pi}{NLON})$$

$$F_J^{--} = 4 \sum_{k=2,2}^{\frac{NLON}{2}1} B_k \sin(k(J-1)\frac{2\pi}{NLON})$$

where $\sum_{k=2,2}^{\frac{NLON}{2}1}$ means a sum for $k = 2$ step 2 to $\frac{NLON}{2}$

and $\sum_{k=1,2}^{\frac{NLON}{2}-1}$ means a sum for $k = 1$ step 2 to $\frac{NLON}{2}-1$

Making use of symmetries and antisymmetries we have

$$A_k = \frac{2}{NLON} \sum_{J=1}^{\frac{NLON}{4}+1}{}'' F_J^{++} \cos(k(J-1)\frac{2\pi}{NLON}) \quad k \text{ even}$$

$$A_k = \frac{2}{NLON} \sum_{J=1}^{\frac{NLON}{4}+1}{}'' F_J^{+-} \cos(k(J-1)\frac{2\pi}{NLON}) \quad k \text{ odd}$$

$$B_k = \frac{2}{NLON} \sum_{J=1}^{\frac{NLON}{4}+1}{}'' F_J^{-+} \sin(k(J-1)\frac{2\pi}{NLON}) \quad k \text{ odd}$$

$$B_k = \frac{2}{NLON} \sum_{J=1}^{\frac{NLON}{4}+1}{}'' F_J^{--} \sin(k(J-1)\frac{2\pi}{NLON}) \quad k \text{ even}$$

where $\sum''$ means a factor of ½ in the first and last terms.

Finally

$$4F_J = (F_J^{++} + F_J^{+-}) + (F_J^{-+} + F_J^{--})$$

$$4F_{NLON+2-J} = (F_J^{++} + F_J^{+-}) - (F_J^{-+} + F_J^{--})$$

$$4F_{\frac{NLON}{2}+2-J} = (F_J^{++} - F_J^{+-}) - (F_J^{-+} - F_J^{--})$$

$$4F_{\frac{NLON}{2}+J} = (F_J^{++} - F_J^{+-}) - (F_J^{-+} - F_J^{--})$$

$J = 1,\ldots,\frac{NLON}{4} + 1.$

6.2 Program Formulation

Define

$$\begin{aligned} f_J^{++} &= Po(J)F_J^{++} \\ f_J^{+-} &= Po(J)F_J^{+-} \\ f_J^{-+} &= Po(J)F_J^{-+} \\ f_J^{--} &= Po(J)F_J^{--} \end{aligned} \qquad (6.2.1)$$

where

$$Po(1) = Po(\frac{NLON}{4} + 1) = 1/\sqrt{2}$$

$$Po(J) = 1;\ J = 2,\ \ldots,\frac{NLON}{4}$$

Define functions

$$C(J,k) = Po(J)\sqrt{\frac{2}{NLON}}\cos(k(J-1)\frac{2\pi}{NLON})$$

$$S(J,k) = Po(J)\sqrt{\frac{2}{NLON}}\sin(k(J-1)\frac{2\pi}{NLON}) \qquad (6.2.2)$$

for $k = 1,\ldots,\frac{NLON}{2}-1$

$$C(J,\frac{NLON}{2}) = \frac{1}{\sqrt{2}}\ Po(J)\sqrt{\frac{2}{NLON}}\cos(k(J-1)\frac{2\pi}{NLON})$$

$$C(J,o) = \frac{1}{\sqrt{2}}\ Po(J)\sqrt{\frac{2}{NLON}}$$

$$S(J,\frac{NLON}{2}) = 0.$$

We have

$$f_J^{++} = 4a_o C(J,o) + 4\sum_{k=2,2}^{\frac{NLON}{2}} a_k C(J,k)$$

$$f_J^{+-} = 4\sum_{k=1,2}^{\frac{NLON}{2}} a_k C(J,k)$$

$$f_J^{-+} = 4\sum_{k=1,2}^{\frac{NLON}{2}} b_k S(J,k)$$

$$f_J^{--} = 4\sum_{k=2,2}^{\frac{NLON}{2}} b_k S(J,k)$$

with

$$a_k = \sum_{J=1}^{\frac{NLON}{4}+1} f_J^{++} C(J,k) \qquad k \text{ even} \neq 0$$

$$a_k = \sum_{J=1}^{\frac{NLON}{4}+1} f_J^{+-} C(J,k) \qquad k \text{ odd} \qquad (6.2.3)$$

$$b_k = \sum_{J=1}^{\frac{NLON}{4}+1} f_J^{-+} S(J,k) \qquad k \text{ odd}$$

$$b_k = \sum_{J=1}^{\frac{NLON}{4}+1} f_J^{--} S(J,k) \qquad k \text{ even} \qquad (6.2.3)$$

$$a_o = \sum_{J=1}^{\frac{NLON}{4}+1} f_J^{++} C(J,o)$$

If wave number k is to be filtered by a reduction factor λ_k then

$$\overline{f}_J^{++\,k} = f_J^{++} + (\lambda_k - 1)4a_k C(J,k) \qquad k \text{ even}$$

$$\overline{f}_J^{+-\,k} = f_J^{+-} + (\lambda_k - 1)4a_k C(J,k) \qquad k \text{ odd}$$

$$\overline{f}_J^{-+\,k} = f_J^{-+} + (\lambda_k - 1)4b_k S(J,k) \qquad k \text{ odd} \qquad (6.2.4)$$

$$\overline{f}_J^{--\,k} = f_J^{--} + (\lambda_k - 1)4b_k S(J,k) \qquad k \text{ even}$$

<u>N.B.</u> Filtering is performed for wave numbers k_{min} to $k_{\frac{NLON}{2}}$.

<u>Definition of λ_k</u>

$$\lambda_k(\theta) = \frac{\cos(\theta)}{\cos(\theta_o)} \frac{1}{\sin(\frac{k\Delta\lambda}{2})} \quad \text{if} \quad \frac{\cos(\theta)}{\cos(\theta_o)} \frac{1}{\sin(\frac{k\Delta\lambda}{2})} < 1$$

$$= 1 \qquad \text{if} \quad \frac{\cos(\theta)}{\cos(\theta_o)} \frac{1}{\sin(\frac{k\Delta\lambda}{2})} \geqslant 1$$

$$\Delta\lambda = 2\pi/NLON$$

<u>N.B.</u> No filtering takes place equatorwards of $\pm\theta_o$.

Finally, unfolding the data gives:

$$F_J = \frac{1}{4Po(J)} \{(f_J^{++}+f_J^{+-}) + (f_J^{-+}+f_J^{--})\}$$

$$F_{NLON+2-J} = \frac{1}{4Po(J)} \{(f_J^{++}+f_J^{+-}) - (f_J^{-+}+f_J^{--})\}$$

$$F_{\frac{NLON}{2}+2-J} = \frac{1}{4Po(J)} \{(f_J^{++}-f_J^{+-}) - (f_J^{-+}-f_J^{--})\} \qquad (6.2.6)$$

$$F_{\frac{NLON}{2}+J} = \frac{1}{4Po(J)} \{(f_J^{++}-f_J^{+-}) - (f_J^{-+}-f^{--})\}$$

6.3 SUBROUTINE INIFLT (PCOSØ, KX, KY, KMINP, KMINV)

Where PCOSØ = $\cos(\theta_o)$, with the space filter applied polewards of latitude θ_o.

KX = number of longitude points

KY = number of latitude rows

KMINP(KY) = smallest wavenumbers to be filtered at p_s, T, u and q rows

KMINV(KY) = smallest wavenumbers to be filtered at v rows.

Subroutine INIFLT (see diag. 21) is called once on the first timestep from LINEMS to initialise constants for the space filter. The functions S(J,k) and C(J,k) (see 6.2.2) are set up in the array CS in common COMFL2, and constants from common COMFL1 are initialised.

k_{min}, the smallest wavenumber to be filtered, is determined for each row. The array KMINV(KY) contains the values of k_{min} at the latitudes of the v points, while KMINP(KY) contains values of k_{min} at the latitudes of the p_s, T, u and q points.

6.4 SUBROUTINE FILTER (PFACT, KMIN, KBASE, KFLDS)

Where PFACT = cos (latitude)/$\cos(\theta_o)$

KMIN = smallest wavenumber to be filtered at given latitude

KBASE = displacement in blank common of first field to be filtered

KFLDS = number of scalar fields to be filtered

Subroutine FILTER (see diag. 22) is called from LINEMS to filter the tendency fields. Each call may filter KFLDS adjacent fields in blank common at the same latitude. Since the v points at a given row are at a different latitude to the other fields, there are 3 calls to FILTER for each row:-

1) filter p_s, T and u

2) filter q (at same latitude, but not adjacent to u)

3) filter v.

In FILTER, the data is folded twice (eqn. 6.2.1) and the fourier coefficients are calculated (eqn. 6.2.3). Coefficients for wavenumbers greater than or equal to KMIN are modified, and the data is unfolded to give the new filtered values.

CHAPTER 7 - THE START DATA SET

7.1 General description

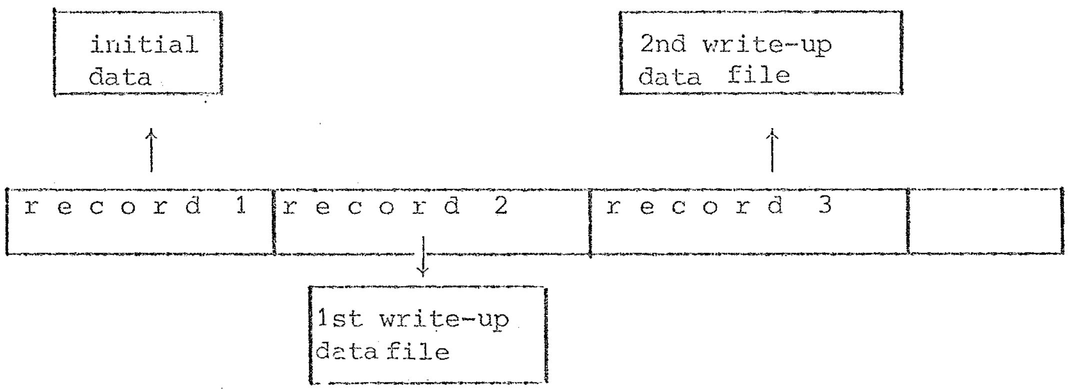


Fig. 7.1.1

Each start data set record consists of the common block COMSDS, which contains sufficient information for the program to locate a data set plus certain run control information. An initial run starts with the start data set containing 1 record only, pointing to the initial data. At each subsequent write-up time, a new data file is created, and a new record is added to the start data set, pointing to it. At a restart, the default procedure in SUBROUTINE SDS is to read the last record of the start data set, and to continue from the latest write-up time.

Record description

The contents of a start data set record can be divided into several different categories :-

7.2 File handling

All the subroutines described in this section were specially writter for the ECMWF CDC 6600.

i) The initial/restart data file is internally attached by

CALL ATTACH (IFAIL,NDATA,NDTFN,NDTCY,NDTPW)

where

IFAIL (local variable) is a return code
NDATA = channel number for data
NDTFN(4) = hollerith array, containing the data file name
NDTCY = cycle number of data file
NDTPW(11) = hollerith array, containing the passowrds of the data file.

ii) Permanent file space for write-up data sets is requested by
CALL REQUEST (IFAIL,NDATA,NDREQ)

where NDREQ (10) = hollerith array, containing the request parameters (eq. "*PF, SN = DSET14.")

iii) A private disk is mounted by

CALL MOUNT (IFAIL,NDMTSN,NDMTVS)

where NDMTSN = hollerith string containing set name of file
NDMTVS = hollerith string containing VSN of disk.

The logical variable NLMNT = .TRUE. if a private disk is to be used and .FALSE. if data is to be kept on public disks.

7.3 Run control

i) TWODT = 2 x time step in seconds. For a leap frog scheme

$$f(T+1) = f(T-1) + TWODT \times \frac{\partial f(T)}{\partial T}$$

ii) NSTOP = final step number. Because the time filter is completed on T-1 data, NSTOP is increased to 1 more than requested by the user. The model will then perform the last part of the time filter, but not the remainder of the calculation on the last extra time step.

iii) NWTIME(200) contains the step numbers at which data is to be written up. NWPTR points to the current element of NWTIME, and NWRITE (from COMIOC) is set to NWTIME (NWPTR). As in (ii) above, the data at step N is not actually written up until step N+1, so that in subroutine LINEMS, write-up times are found by comparing NWRITE with NSTEP-1.

At a write-up time, NWPTR is incremented by 1 in subroutine SDS, and there is a test to see if the next element of NWTIME is a valid step number. If not, NWRITE and NWPTR are set to -1 and not modified again.

iv) NRECRD - the start data set record number. By setting NREC in the namelist REST, the user may start from any record of the start data set in subroutine SDS, over-riding the default values of the first record (for an initial run) or the last record (for a restart).

An initial run, beginning with a forward time step, may start from any record of the start data set. However a restart run, beginning with a leap frog time step needs two adjacent time levels of data, and the time T data files are available only for the last 2 write ups.

v) NSW - the switch flag. In this CDC feature, by setting SWITCH 1, the program can be made to write up at the next timestep and finish. NSW = 0 if the switch has not been set; NSW = 1 if the switch has been set, but it is not a usual write-up time, NSW = 2 if the switch is set at a write-up time, when NWPTR and NWRITE will have to be updated normally.

7.4 Filtering

i) EPS - this is used in the time filter :-

$$\bar{f}(T) = f(T) + EPS \times \{f(T-1) - 2f(T) + f(T+1)\}$$

ii) NLSPFL = .TRUE. if there is to be spatial filtering, and .FALSE. otherwise.

iii) CTHO = COS (θ_o) where θ_o is the latitude at which spatial filtering begins, i.e. filtering is done to the north of latitude θ_o in the northern hemisphere, and to the south of latitude $-\theta_o$ in the southern hemisphere.

7.5 Making a start data set

a) To make a start data set
CALL MAKESD (KIN,KOUT,KSDS)

where KIN = input channel
KOUT = output channel for printed messages
KSDS = channel to which start data set is written.

Subroutine MAKESD presets the variables of COMSDS, according to fig. 7.5.1. Real and integer variables may be overridden by input to namelist STARTD on channel KIN, while hollerith variables such as file names and passwords are read from channel KIN. The necessary control cards for the CDC 6600 are described in para. 7.6.

Values of COMSDS variables

Variable	Meaning	Preset value	Reset value
NRECRD	start data set record number	1	-
NDATA	channel number of initial/re-start data	20	input to namelist STARTI
NDTCY	cycle number of initial/restart number	0	input to namelist STARTI
NWTIME(200)	write-up time step numbers	0's	input to namelist STARTI

Variable	Meaning	Preset value	Reset Value
NWPTR	pointer to current element of NWTIME	-1	If NWTIME(1)> 0, NWPTR = 1
NSTOP	step number of last step	0	If a value of NSTOP > 0 is input to namelist STARTD, then NSTOP = NSTOP+1
TWODT	2 x time step (secs)		TWODT = 2 x DTIME, where DTIME = 0 initially, but may be reset by STARTD
EPS	constant for time filter	0	input to namelist STARTD
CTH0	cos(start latitude for space filter)	1	CTH0 = cos(THETA0xα) where α is a degree to radian conversion factor. THETA0 = 0, initially, but may be reset by STARTD
NLSPFL	NLSPFL = .TRUE. - space filter = .FALSE. - no space filter	FALSE	input to namelist STARTD
NSW	switch flag	0	-
NDTFN	name of initial/restart data file	blanks	input to channel KIN in format 4A10
NDTPW	passwords of initial/restart data file	blanks	input to channel KIN in format 11A10 This must contain at least 'ID = EWxxx.', and may also contain SN, VSN,MR etc. information.
NLMNT	NLMNT = .TRUE. - private disk = .FALSE. - public disks only	.FALSE.	input to namelist STARTD
NDREQ	request call parameters	'*PF'	if NLMNT = TRUE, NDREQ must be input to channel KIN in format 11A10.

Variables	Meaning	Preset Value	Reset Value
NDMTSN	set name for private disk	blank	if NLMNT = TRUE, NDMTSN and NDMTVS must be input to channel KIN in format 2A10.
NDMTVS	VSN of private disk	blank	

Fig. 7.5.1

7.6 A typical card deck (CDC scope)

```
EWJC3.
FTN(A)
MAP(PART)
REQUEST,TAPE1,*PF.
ATTACH,GLIB, GEMINILIB,ID = EWJC3, MR = 1.
LIBRARY,GLIB.
LGO.
CATALOG,TAPE1, MYSDS,ID = EWJC3.
7/8/9
      PROGRAM A (OUTPUT,TAPE6 = OUTPUT,INPUT,TAPE5 = INPUT,TAPE1)
      IN = 5
      IOUT = 6
      ISDS = 1
      CALL MAKESD (IN,IOUT,ISDS)
      STOP
      END

7/8/9
      $STARTD
          NWTIME(1) = 5,10,15,20, NSTOP = 20, DTIME = 120,
      $
NEWGEMINI
ID = EWJC3.
6/7/8/9
```

Fig. 7.5.2 <u28> SUBROUTINE MAKESD (KIN,KOUT,KSDS)

Create start data set.

Arguments : KIN - input channel

KOUT - channel for printed output messages

KSDS - channel to which start data set is written.

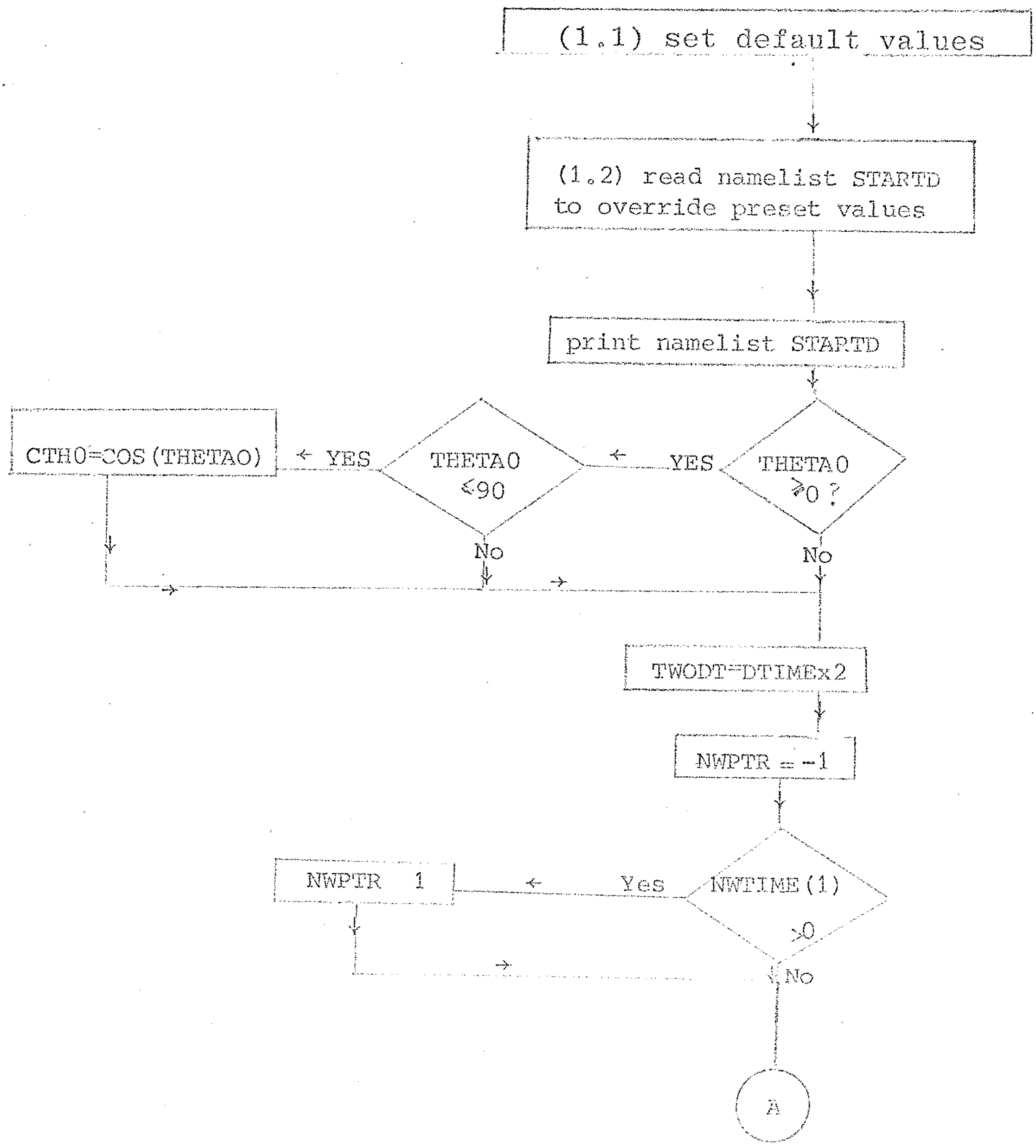
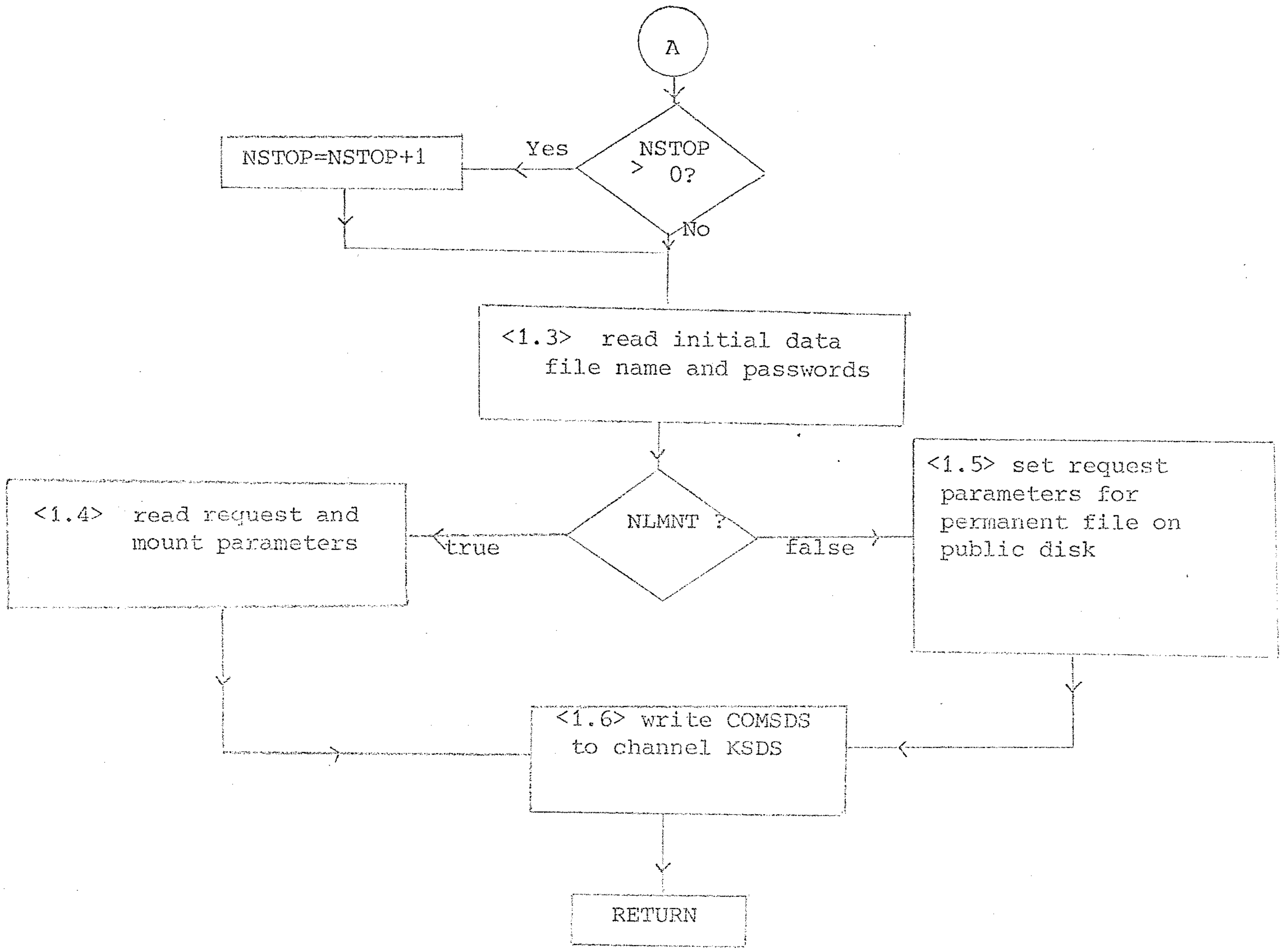


Fig. 7.5.2. (contd.)



CHAPTER 8 - RUNNING THE MODEL

8.1 Creating an initial data set

The initial data set should have common blocks COMBAS, COMHKP and COMMAP as the first 3 records, followed by 1 record for each latitude row of data, ordered from north to south. Two subroutines have been provided, one to make the constants, map factors and other information, COMMAP, and the other to create the initial data set.

8.2 Map factors

A map factor data set, the common block COMMAP, can be created by

```
CALL MAPFAC (KMAP,KPRINT,KLEV,KN)
```

where

```
KMAP   = channel to which COMMAP is written
KPRINT = printed output channel
KLEV   = number of vertical levels
KN     = grid resolution ( eg.  24 = N24 etc.)
```

This subroutine is not generalised at the moment. The dimensions in COMMAP, the dimensions of the local variables and the data statements giving the $\sigma(k)$ and $\sigma(k+\frac{1}{2})$ levels are set for 9 levels, and N24. However, the executable part of the code is general.

8.3 Initial data set

The initial data set can be created by

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

where

```
KMAP   = channel number of map factor data set
KDTIN  = channel number of input grid point data
KRD    = card input channel
KWRITE = printer output channel
KDTOUT = channel to which GEMINI initial data set is written.
```

The user must supply a data set on channel KDTIN which contains the values of ϕ,p,T,u,v and q on the staggered grid, ordered as described in the data structure section. This should have been written with unformatted Fortran WRITE statements with 1 record per latitude row, sequentially from north to south.

The subroutine initialises COMBAS to the same values as are set in subroutine BASIC. COMHKP is initialised as described in para. 9.7. The namelist INIDAT on channel KRD allows NOREC (the number of latitude rows), NLON (the number of longitude points), NLEV (the number of vertical levels) and NCOM (the number of common blocks at the start of the data set) to be reset. COMMAP is read from channel KMAP. The initial data set is then created in the form described in para. 8.1 on channel KDTOUT.

<u 29 > SUBROUTINE MAKEDT (KMAP,KDTIN,KRD,KWRITE,KDTOUT)

Create GEMINI initial data set

Arguments:

KMAP	-	channel number of map factor data set
KDTIN	-	channel number of input grid point data
KRD	-	card input channel
KWRITE	-	line printer output channel
KDTOUT	-	gemini initial data set written to channel KDTOUT

(1.1) initialise COMBAS to same default values as does SUBROUTINE BASIC

(1.2) initialise COMHKP

(1.3) read namelist INIDAT

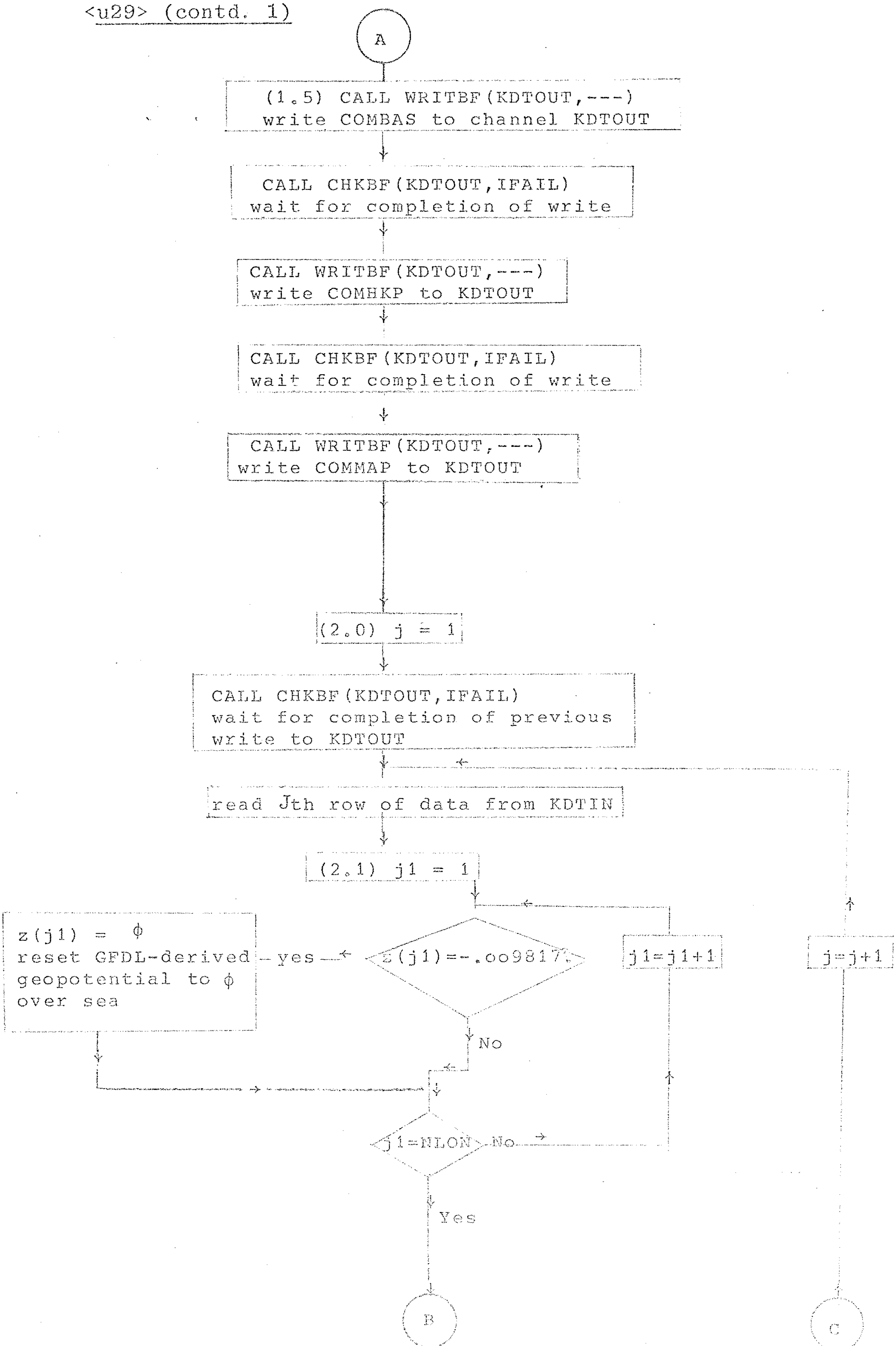
write namelist INIDAT

LREC = NLON*(NLEV x 4+2)
Set data record length, LREC, to 2 scalar + 4 vector fields

(1.4) read COMMAP from KMAP

A

<u29> (contd. 1)


A
(1.5) CALL WRITBF(KDTOUT,---)
write COMBAS to channel KDTOUT
CALL CHKBF(KDTOUT,IFAIL)
wait for completion of write
CALL WRITBF(KDTOUT,---)
write COMHKP to KDTOUT
CALL CHKBF(KDTOUT,IFAIL)
wait for completion of write
CALL WRITBF(KDTOUT,---)
write COMMAP to KDTOUT
(2.0) j = 1
CALL CHKBF(KDTOUT,IFAIL)
wait for completion of previous
write to KDTOUT
read Jth row of data from KDTIN
(2.1) j1 = 1
z(j1) = φ
reset GFDL-derived
geopotential to φ
over sea
yes
z(j1)=-.009817
No
j1=j1+1
j=j+1
j1=NLON
No
Yes
B
C

<μ29> (contd. 2)

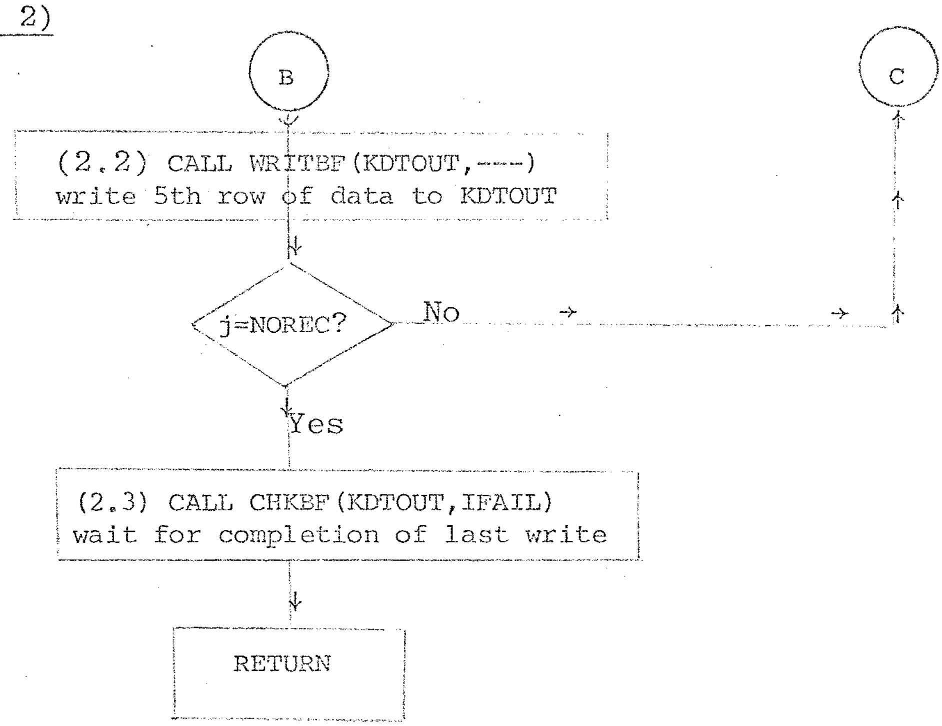


Fig. 8.3.1

8.4 Running the model on the CDC 6600

Below is an example of a typical card deck, with some explanations.

EWJC3, T2000. FTN(A) MAP(PART)	N24,9 levels with space filtering from 45° takes about 20 secs/time step
REQUEST,TAPE10,*VSN=ECMWF8. REQUEST,TAPE11,*VSN=ECMWF9. ATTACH,TAPE30,MYSDS,ID=EWJC3.	Putting the work files on different disks reduces I/O contention. Start data set - not MR=1, since it is written to as well as read from.
ATTACH,GLIB,GEMINILIB, ID=EWJC3, MR=1.	
LIBRARY (GLIB)	
LGO. AUDIT,ID=EWJC3. 7/8/9	The run will create 2 T_o+1 data files + 1 file for each write-up time. Try and keep track of them all!
PROGRAM GEMINI(INPUT, TAPE5=INPUT,OUTPUT, TAPE6=OUTPUT,TAPE7,TAPE10, TAPE11,TAPE20,TAPE21,TAPE22, TAPE30) COMMON B(55296) CALL MASTER STOP END	Main program should declare tapes, and define the size of blank common
7/8/9 $REST $	May reset NLRES(restart)and NREC(start data set record number) in $REST
1ST TEST RUN 3-2-77 JMH ECMWF TIMESTEP=5 MINS,RUN 1 PROGRAM GEMINI	4 data cards with up to 50 characters on each, labelling the run
$NEWRUN NSTOP=45, $	Various parameters may be redefined in $ NEWRUN
6/7/8/9	

CHAPTER 9 - COMMON BLOCKS

Below is an index for all the common blocks

9.1 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.	BDYIO (P)	.TRUE.(after last time step)
NLRES	.TRUE. if run is a restart from an earlier run	BASIC	.FALSE.	MODIFY (P)	input to namelist $REST
NONLIN	channel number for online input/output	BASIC	1	DATINI (P)	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	-	-

9.1 C1.1(contd.)

variable	meaning	where defined	initial value	where redefined*	new value
NSTEP	current step number	BASIC	0	(1) DATCOM (P) (2) BDYIO (A)	1) step number for restart 2) NSTEP = NSTEP+1
STIME	start time (secs)	MASTER	CALL SECOND (STIME)	-	-
LABEL 1(5) LABEL 2(5) LABEL 3(5) LABEL 4(5)	labels used to describe the run	BASIC	BLANK	LABRUN (A)	card input set by user
LABEL 5(5) LABEL 6(5)	labels available to programmer	BASIC	BLANK	-	-
LABEL 7(5) LABEL 8(5)	labels reserved for system use	BASIC	BLANK	-	-
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

*(A) - always
(P) = possibly

9.2 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.	"	" "

9.2 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.	"	" "

9.3 C3.1 COMMAP Map factors and constants

The map factors and constants are generated by a call to SUBROUTINE MAPFAC. Common blocks COMBAS, COMHKP and COMMAP are stored at the front of the initial/restart data set, and are read in by SUBROUTINE DATCOM.

VARIABLE	MEANING
RDLAM	1/DLAMBDA, where DLAMBDA is the longitudinal grid interval in radians
RDTH	1/DTHETA, where DTHETA is the latitudinal grid interval in radians
R6DLA	1/6xDLAMBDA
RNLON	1/NLON, where NLON is the number of longitude points
RQNLON	4/NLON
RA	1/A where A is the radius of the earth in metres
DL2RDT	2xDLAMBDA/DTHETA
RR2ADT	R/2xAxDTHETA where R is the gas constant in joules/kg/^{o}K
CP2	2xCP where CP is the specific heat of air at constant pressure in joules/kg/^{o}k
RNLCSP	1/(NLON x $\sum$cosp), summation taken over all latitudes θ at which pressure is kept, cosp = cos(θ) except at the poles where cosp = 0.25 cos $(\pm \frac{\pi}{2} \pm \frac{\Delta\theta}{2})$.
RHCPXC	2/CPxNLONxcos(v_{POLE}) where v_{POLE} is the latitude of the v grid point ½ grid length from the north pole
HRXACT	2/AxDTHETAxNLONxcos(v_{POLE})
R2DSIG(NLEV)	1/2xDSIGMA(k) where DSIGMA(k) = SIGMA(k+½) - SIGMA(k-½) and NLEV = number of levels

9.3 C3.1 COMMAP (contd.)

VARIABLE	MEANING
R2DLNS(NLEV)	Rx2xD(LN(SIGMA(k))
AKR2SZ(NLEV)	KxD(LN(SIGMA(k))/2xDSIGMA(k) where K = R/cp
SIGKPH(NLEV)	SIGMA(k+½)
S4RXCT(NLEV)	4xDSIGMA(k)/NLONxAxDTHETAxcos(v_{POLE})
DSIGMA(NLEV)	SIGMA(k+½) - SIGMA(k-½)
BETA(NLEV)	2xLN(SIGMA(k+½)/SIGMA(k))/LN(SIGMA(k+½)/SIGMA(k-½)) i.e. the upper logarithmic weighting of level k
ALPHA(NLEV)	2 - BETA(k) i.e. the lower logarithmic weighting of level k.
COSU(NLAT)	cos(θ) at u grid points, where NLAT = number of latitude points
RCOSU(NLAT)	1/COSU(θ)
RACOSU(NLAT)	1/AxCOSU(θ)
R2ACUL(NLAT)	1/2xAxCOSU(θ)xDLAMBDA
R2ACUT(NLAT)	1/2xAxCOSU(θ)xDTHETA
CUR6T(NLAT)	COSU(θ)/6xDTHETA
QKRACL(NLAT)	K/4xAxCOSU(θ)xDLAMBDA
COSV(NLAT)	COS(θ) at v grid points
FACVR6(NLAT)	FxAxCOSV/6 where F is the coriolis parameter

9.3 C3.1 COMMAP (contd.)

VARIABLE	MEANING
R2CPC(NLAT)	1/2xCPxCOSU(θ)
DSRACL(NLAT,NLEV)	DSIGMA(k)/AxCOSU(θ)xDLAMBDA
DSRACT(NLAT,NLEV)	DSIGMA(k)/AxCOSU(θ)xDTHETA
DUMMY	Spare variable space, set to 0.

9.4 C3.3 COMSTA Statistics parameters

Variable	meaning	where defined	initial value	where redefined	new value
STPS	mean surface pressure	STATS	At start of each timestep STPS = 0 STQ = 0 STPE = 0 STKE = 0 in STATS	STATS (A)	during timestep, mean surface pressure over the entire area is accumulated
STQ	mean moisture content	STATS	"	STATS (A)	mean moisture content is accumulated
STPE	mean potential energy	STATS	"	STATS (A)	mean potential energy is accumulated
STKE	mean kinetic energy	DYN	"	DYN (A)	mean kinetic energy is accumulated
STTE	mean total energy	DYN	"	DYN (A)	STTE = STKE + STPE at end of time step

9.5 C4.1 COMIOC Housekeeping parameters

variable	meaning	where defined	initial value	where redefined	new value
NLINE1(4)	displacements from the start of blank common of the 4 input buffers	STARTN	NLINE1(1) = 0 NLINE1(2) = NBUFLN NLINE1(3) = 2 x NBUFLN NLINE1(4) = 3 x NBUFLN	NSSCAN (A) BDYIO (A)	cyclic buffering, i.e. for each new row:- NLINE1(1):= NLINE1(2) NLINE1(2):= NLINE1(3) NLINE1(3):= NLINE1(4) NLINE1(4):= NLINE1(1)
NLINE2(2)	displacements from the start of blank common of the 2 output buffers	STARTN	NLINE2(1)= 4 x NBUFLN NLINE2(2)= 5 x NBUFLN	NSSCAN (A) BDYIO (A)	cyclic buffering, i.e. for each new row:- NLINE2(1):= NLINE2(2) NLINE2(2):= NLINE2(1)
NWKIN	channel number of input work file	DATINI	10	1) DATINI (P) 2) BDYIO (A)	1) input to namelist $ NEWRUN 2) swap NWKIN and NWKOUT at end of timestep
NWKOUT	channel number of output work file	DATINI	11	1) DATINI (P) 2) BDYIO (A)	1) input to namelist $ NEWRUN 2) swap NWKIN and NWKOUT at end of timestep
NROW	current row number	STARTN	1	1) STARTN (A) NSSCAN (A) BDYIO (A) 2) BDYIO (P)	1) NROW = NROW + 1 2) NROW = 1 at start of new timestep
NBUFLN	length of I/O buffer (containing data at 2 time levels + extra work space)	INITAL (initial run) or RESUME (restart)	LREC x 2 + NXTRWK where LREC = length of data at 1 time level NXTRWK = length of extra work space	-	-

9.5 C4.1 COMIOC (contd.)

variable	meaning	where defined	initial value	where redefined*	new value
MAXROW	number of latitude rows	INITAL (initial run) or RESUME (restart)	MAXROW = NOREC	-	-
NORS	flag indicating direction of I/O scan NORS = 2 - north to south NORS = 1 - south to north	STARTN	2	-	-
NSTART	step number at start of run	INITAL (initial run) or RESUME (restart)	NSTEP	-	-
NWRITE	(step number - 1) of next write up time (N.B. it is the T-1 data 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) 3) DATINI (P)	1) NWRITE=NWTIME(NWPTR + 1) at write-up times 2) NWRITE=-1 after last write-up time 3) NWRITE=NWTIME(NWPTR) if NWPTR of NWTIME have been modified by namelist $ NEWRUN 4) NWRITE=-1 if $ NEWRUN input gives write-up time after last step number
NXTRWK	Length of extra work space (2 scalar + 2 vector fields)	INITAL (initial run) or RESUME (restart)	NLON x 2 x (NLEV + 1)	-	-
				*(A) = always (P) = possibly	

9.6 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

9.7 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) NMIA=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,Ø) 3) swap NM1AFN(J) and NM1BFN(J) at write-up time

9.7 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 FDLENM(NM1BFN,Ø) 3) swap NM1AFN(J) and NM1BFN(J) at write-up time

9.7 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
				*(A) = always (P) = possibly	

9.8 C4.4 COMSDS start data set record

variable	meaning	where defined	initial value	where redefined *	new value
NRECRD	start data set record number	SDS	common block COMSDS is read from the start data set	SDS(P)	NRECRD=NREC before a new record is written to the start data set
NDATA	channel number of initial/ restart data set	SDS	"	-	-
NDTCY	cycle number of file NDATA	SDS	"	-	-
NWTIME(200)	array of write-up time step numbers	SDS	"	DATINI(P)	input to namelist $ NEWRUN
NWPTR	pointer to next write-up time in array NWTIME	SDS	"	1) DATINI(P) 3) SDS(P)	1) input to namelist $ NEWRUN 2) NWPTR=-1 if $ NEWRUN input gives write-up time after last time step 3) NWPTR=NWPTR+1 at write-up time 4) NWPTR=-1 after last write-up time
NSTOP	step number of final timestep	SDS	"	DATINI(P)	input to namelist $ NEWRUN
TWODT	2 x timestep in seconds	SDS	"	DATINI(P)	if DTIME (timestep in seconds) is input to namelist $ NEWRUN, then set TWODT=DTIME x 2

9.8 C4.4 COMSDS (contd.)

variable	meaning	where defined	initial value	where redefined *	new value
EPS	constant for time filter: $\bar{f}(t)=f(t)+EPS\{f(t-1)-2f(t)+f(t+1)\}$	SDS	common block COMSDS is read from the start data set	DATINI(P)	input to namelist $ NEWRUN
CTHO	$\cos(\theta_o)$, where space filtering is done north of latitude θ_o and south of $-\theta_o$	SDS	"	DATINI(P)	if THETAO (degrees) is input to namelist $ NEWRUN, then set CTHO=cos(THETAO)
NLSPFL	.TRUE. if there is to be space filtering	SDS	"	DATINI(P)	input to namelist $ NEWRUN
NSW	switch flag NSW=0 - switch off NSW-1 - ONSW1 set by operators to terminate run, not at a write-up time NSW=2 - ONSW1 set at a write-up time	SDS	common block COMSDS is read from the start data set	1) BDYIO(P) 2) SDS (A)	1) check switch 1 at end of each timestep in BDYIO. If switch is on, set NSW=1 (not write-up time) or =2 (write-up time) 2) set NSW=0 when reading start data set
NDTFN(4)	file name of initial/restart data set	SDS	"	LINEMS(P)	CALL FILENM(NDTFN) generates a new file name at each write-up time
NDTPW(11)	password of initial/restart data set	SDS	"	-	-

*(A) = always
(P) = possibly

9.9 C5.1 COMDBC blank common dump parameters

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

*(P) = possibly

9.10 C3.4 COMFLI space filter parameters

variable	meaning	where defined	initial value	where redefined*	new value
NX	number of longitude points	calling parameter to INIFLT	KX	-	-
NXH	NX/2	INIFLT	KX/2	-	-
NXQ	NX/4	"	KX/4	-	-
NXQP1	NXQ+1	"	KX/4+1	-	-
NXP4	NX+4	"	KX+4	-	-
RS2	$1/\sqrt{2}$	"	$1/\sqrt{2}$	-	-
S2	$\sqrt{2}$	"	$\sqrt{2}$	-	-
SK(50)	$1/\mathrm{SIN}(\mathrm{JK}\times\frac{\Delta\lambda}{2})$	"	$1/\mathrm{SIN}(\mathrm{JK}\times\frac{\pi}{\mathrm{KX}})$ for JK=1,... NXH	-	-
F(100)	functions f_J^{++}, f_J^{+-}, f_J^{-+}, f_J^{--} for J=1,...,NXQP1	FILTER	see eqn. 6.2.1	FILTER (A)	see eqn. 6.2.1

9.11 C3.5 COMFL2 space filter parameters

variable	meaning	defined	initial value	redefined*	new value
CS(2400)	functions C(J,K) and S(J,K)	INIFLT	see 6.2.2	–	–

CHAPTER 10 - FLOW DIAGRAMS

Below is an index to all the flow diagrams

Diagram 1

OUTLINE FLOW DIAGRAM

(G) = GEMINI; (O) = OLYMPUS

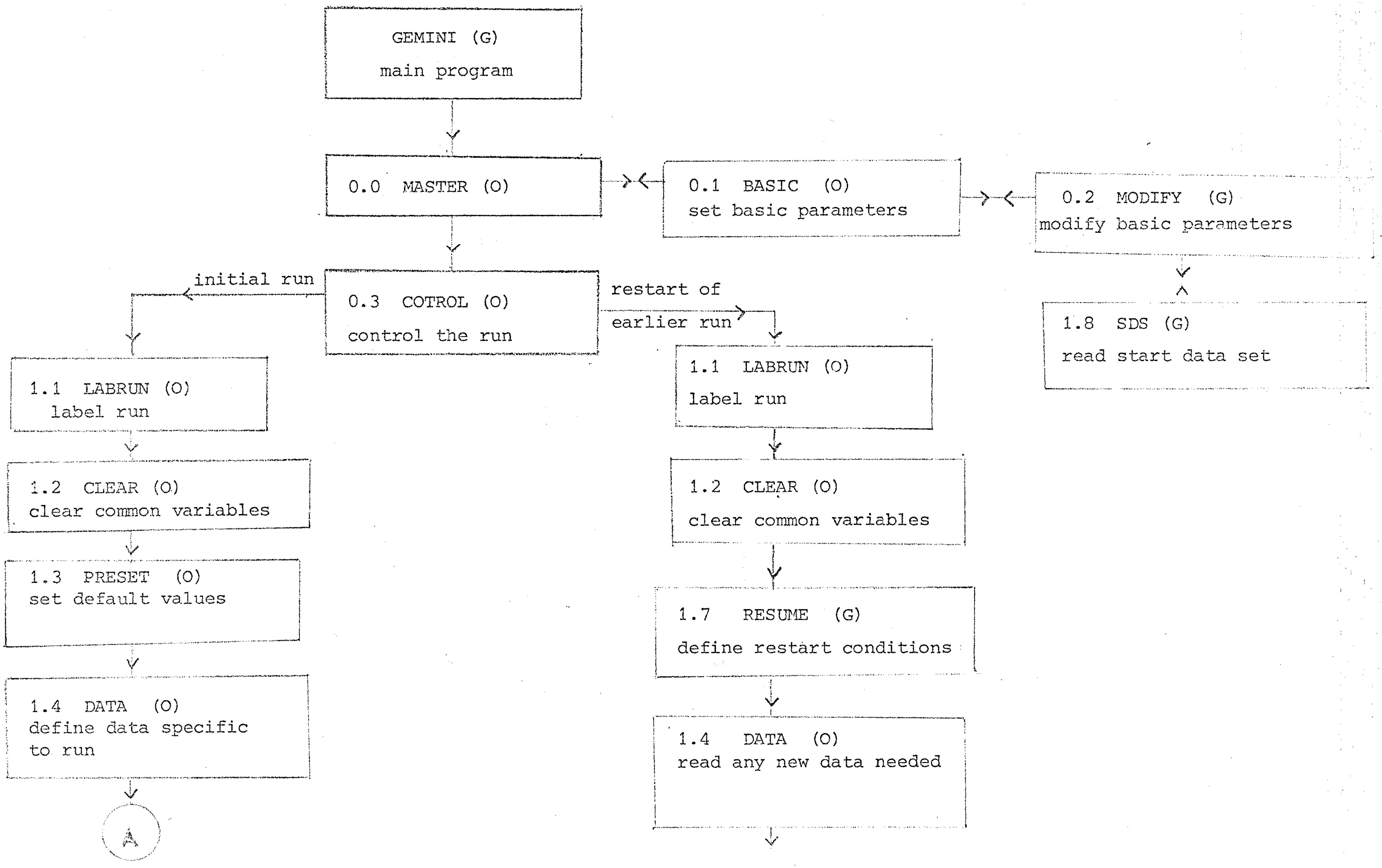


Diagram 1 (contd.)

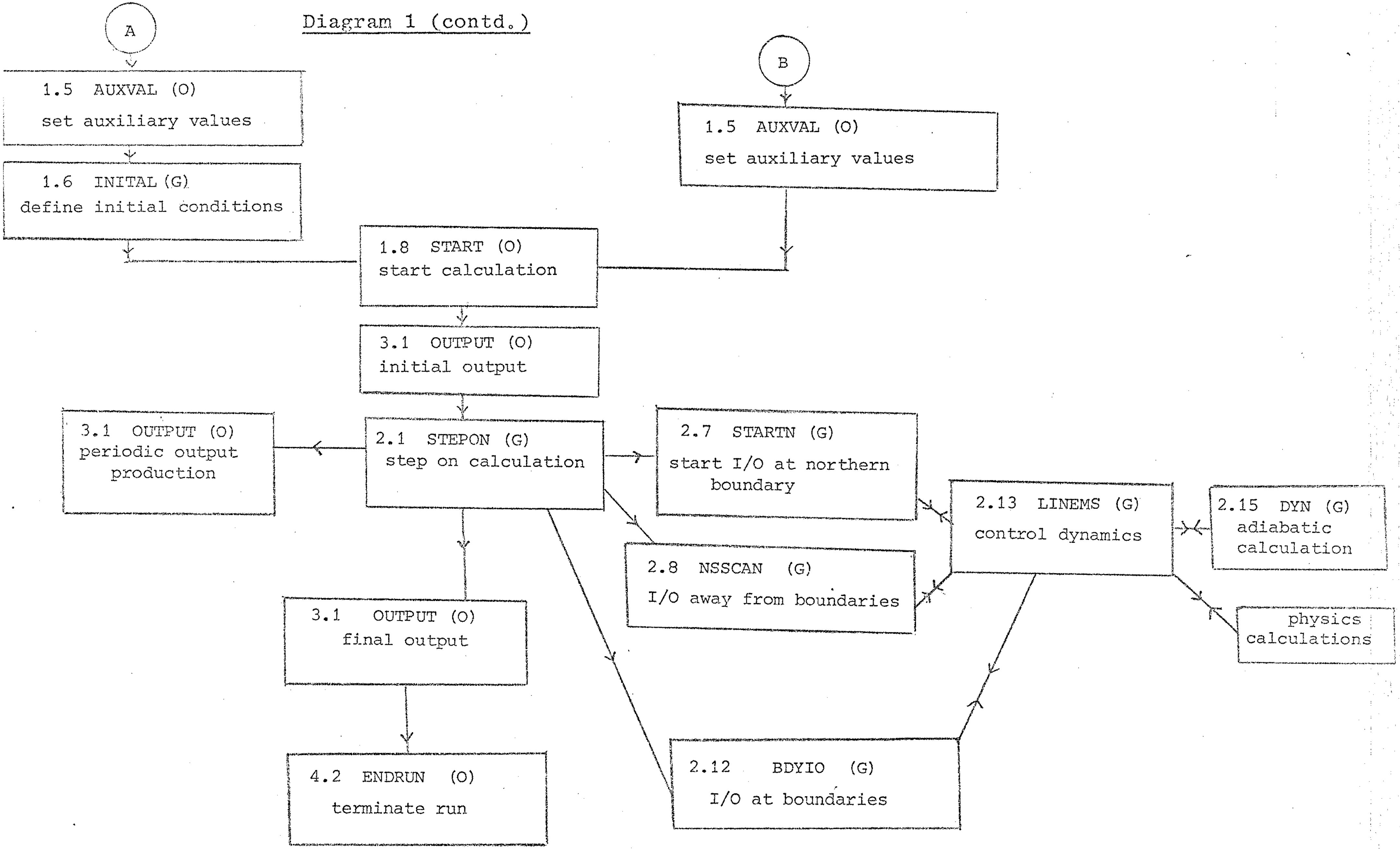
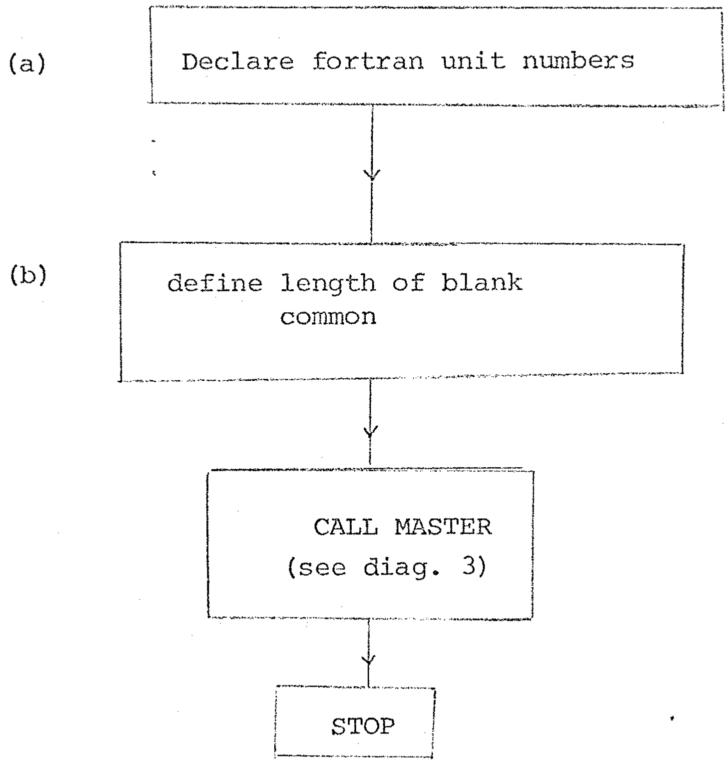


Diagram 2 PROGRAM GEMINI Main program



(a) Fortran unit numbers:

TAPE5	=	INPUT
TAPE6	=	OUTPUT
TAPE7	=	punched card output
TAPE10, TAPE11	=	temporary work files
TAPE20	=	initial/restart data, then write-up data
TAPE21, TAPE22	=	T+1 data files
TAPE30	=	start data set

(b) Length of blank common:

See data structure section. Blank common contains 4 input and 2 output buffers, with 2 time-levels of data + workspace in each buffer.

Diagram 3 <0.0> SUBROUTINE MASTER start run

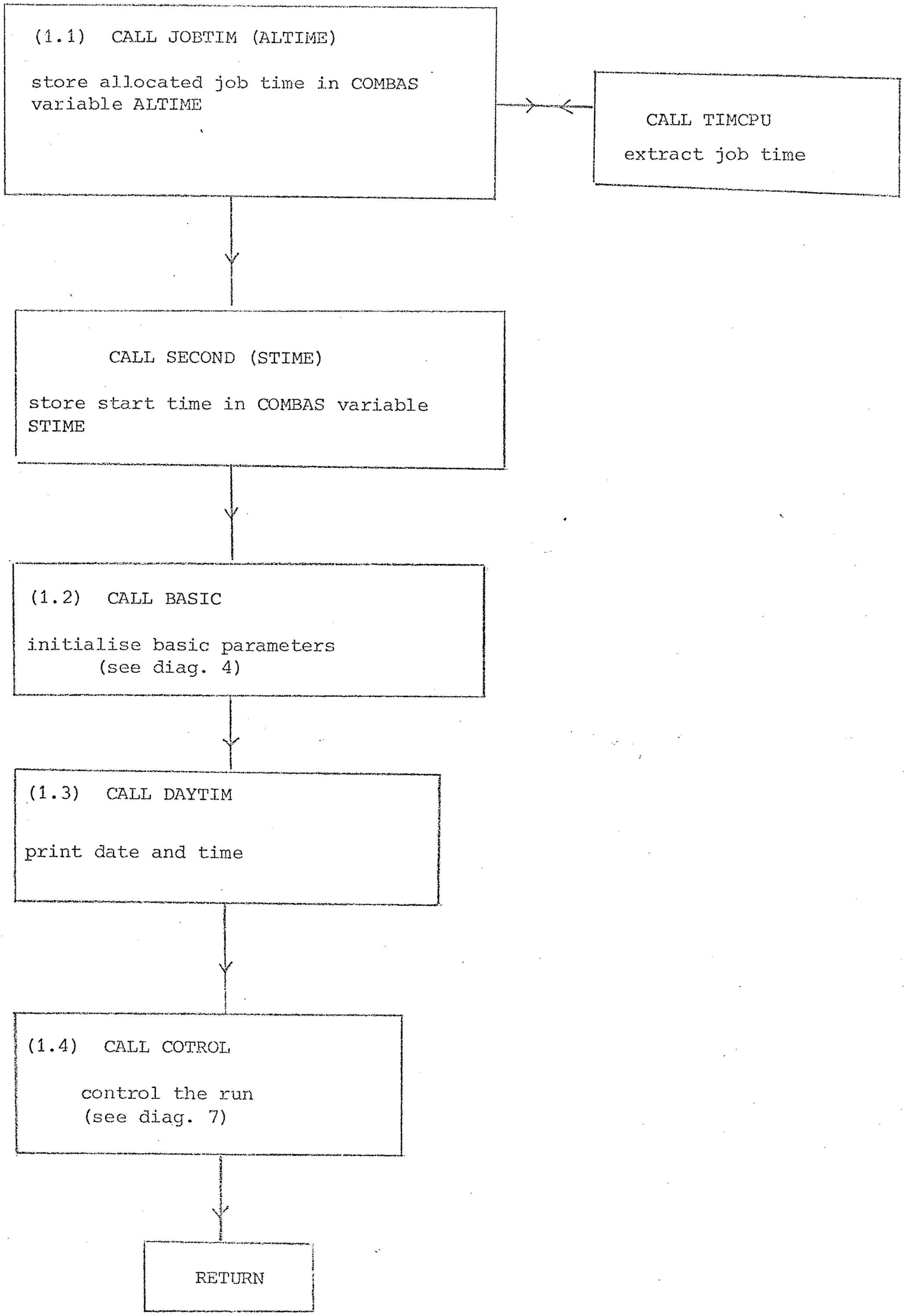


Diagram 4

<0.1> SUBROUTINE BASIC set basic parameters

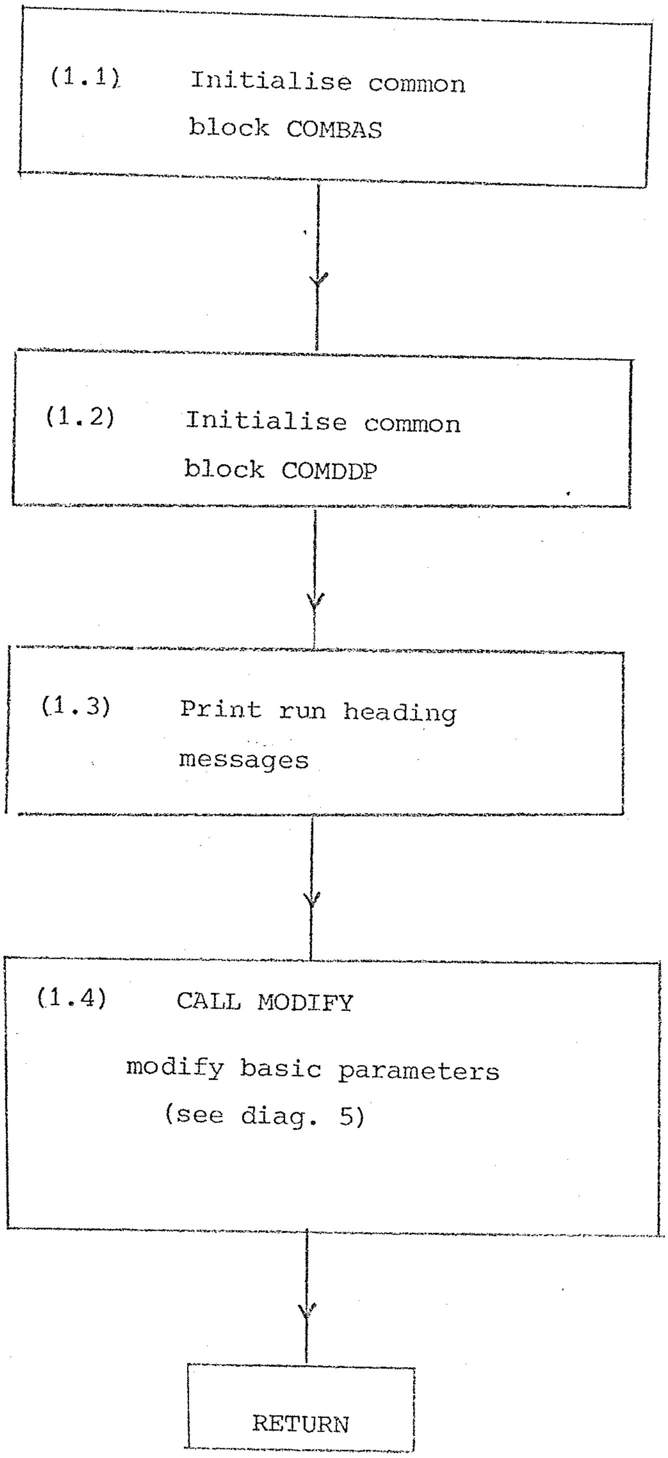
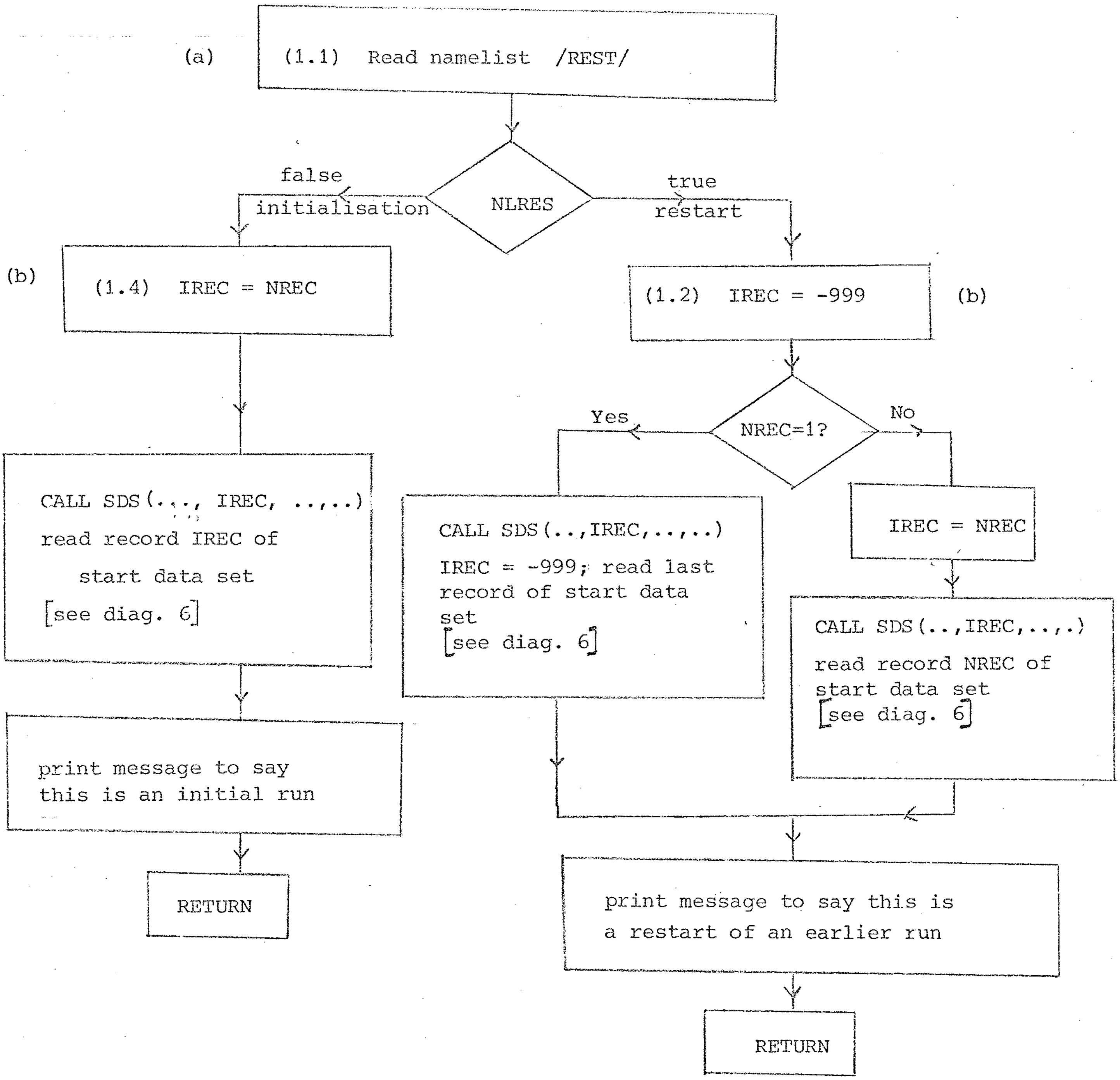


Diagram 5 <0.2> SUBROUTINE MODIFY modify basic parameters



(a) Namelist /REST/ may reset variables NLRES and NREC from common COMBAS.

If NLRES = FALSE (the default value, preset in BASIC), the run is an initialisation.

If NLRES = TRUE, the run is a restart from an earlier run.

(b) NREC is the record number of the start data set which is to be read. It is preset to 1 in BASIC.

(i) Initialisation: read first record of start data set, unless NREC has been reset in /REST/, when record NREC is read.

(ii) Restart: read last record of start data set unless NREC has been reset in /REST/, when record NREC is read.

Diagram 6 <1.8> SUBROUTINE SDS (KLEDGE,KREC,KERR,KCALL)

Read or write start data set record

Arguments:-

KLEDGE: Fortran unit number of start data set.

KREC: Start data set record number to be read

KREC = -999: read last record. (Dummy parameter for writes)

KERR: Error return code.

KERR = 1: successful call for subroutine

KERR = 2: record KREC not on file

KCALL: KCALL = 1: write next record at end of start data set

KCALL = 2: read record KREC

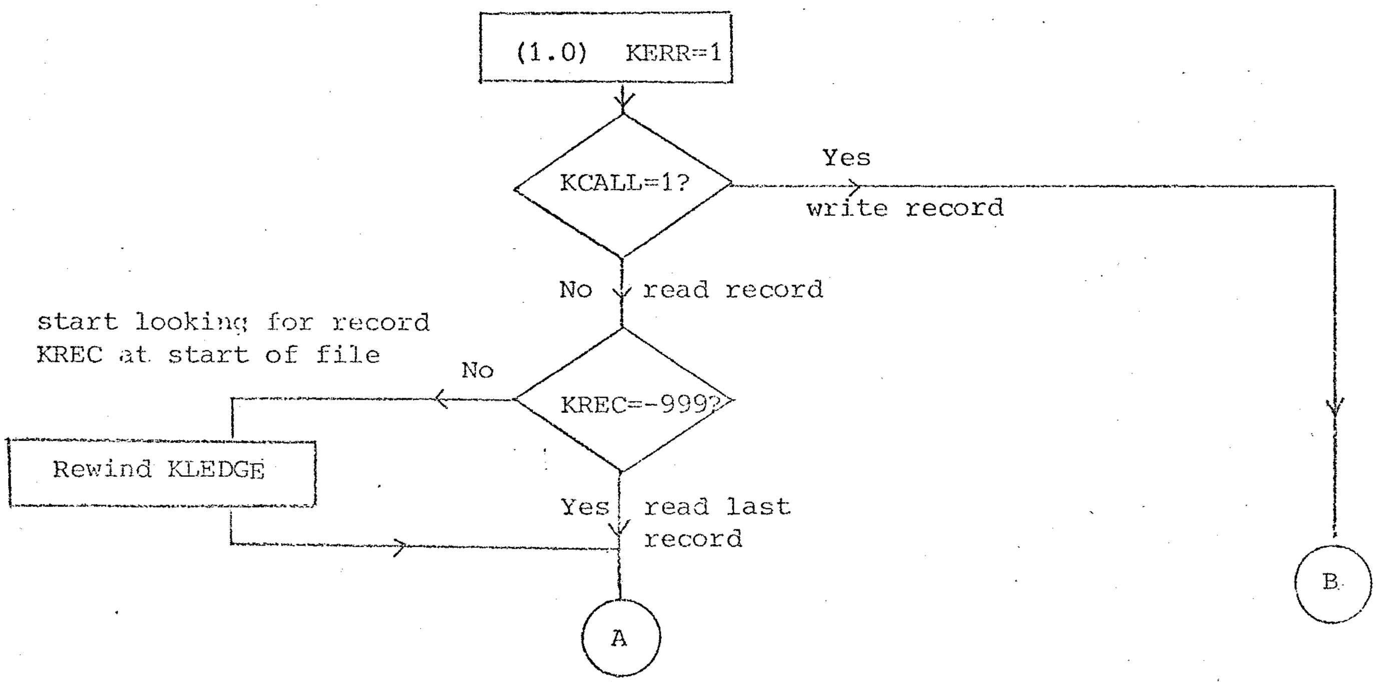


Diagram 6 (contd.1)

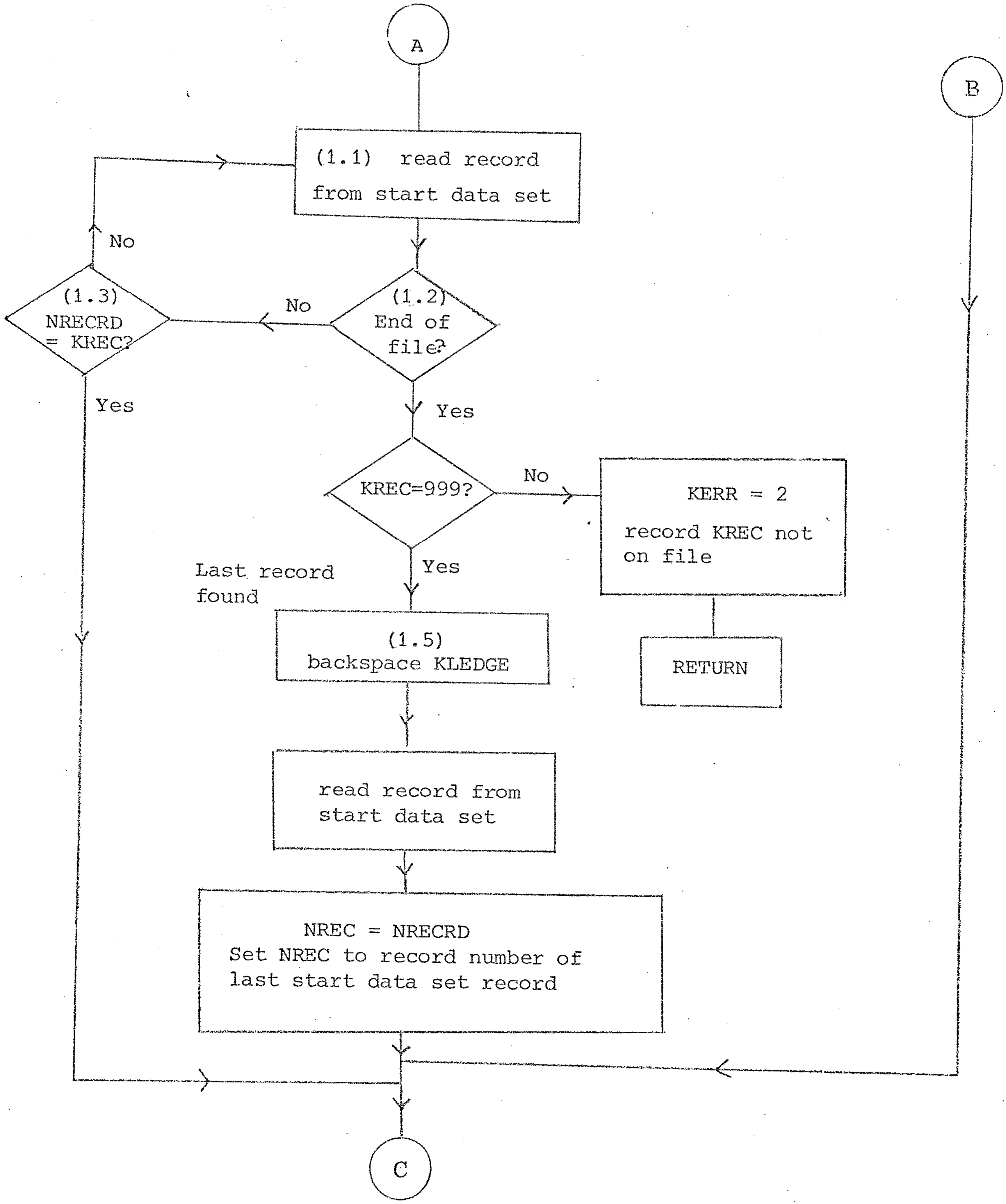


Diagram 6 (contd.2)

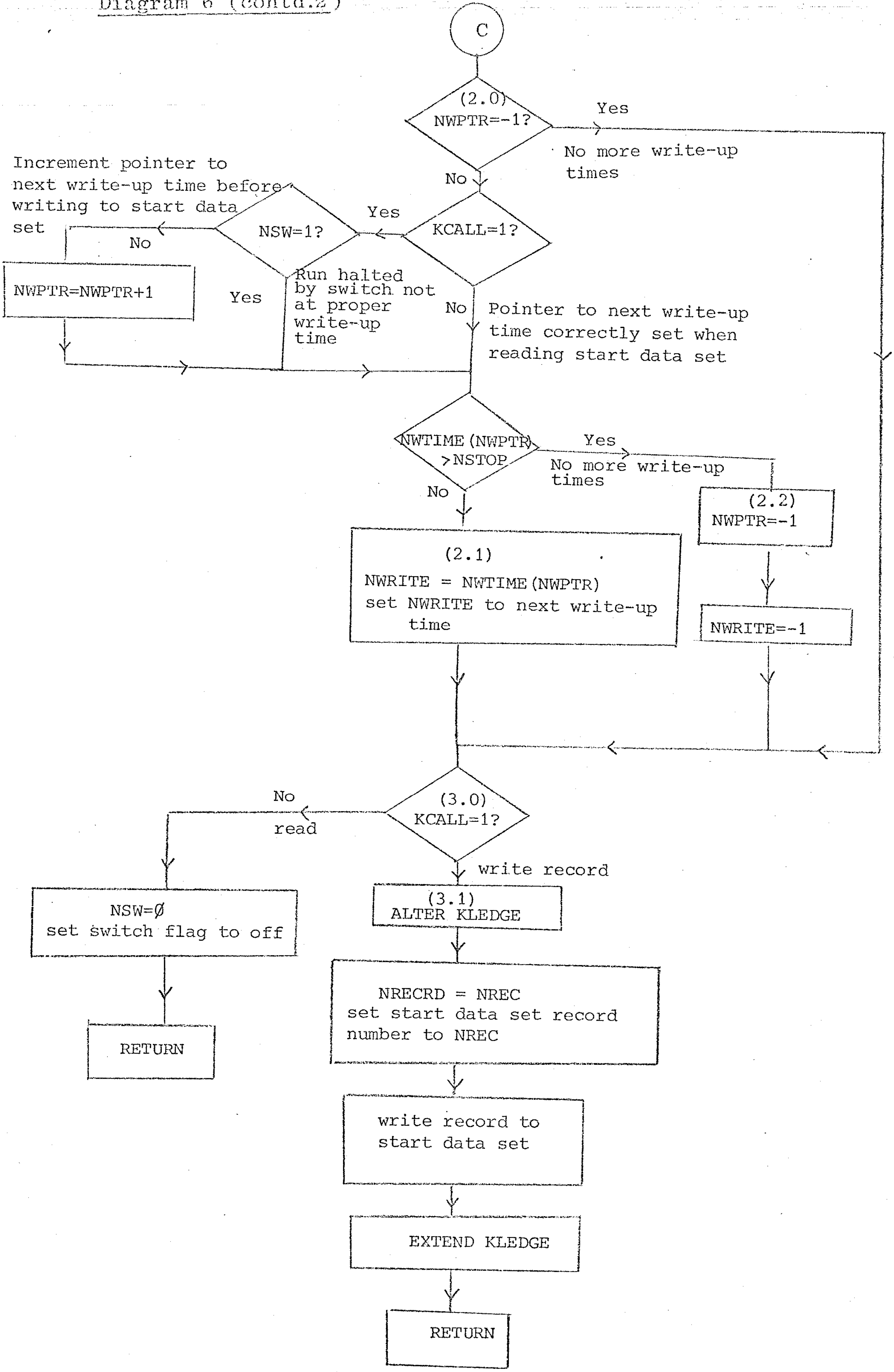


Diagram 7 <0.3> SUBROUTINE COTROL control the run

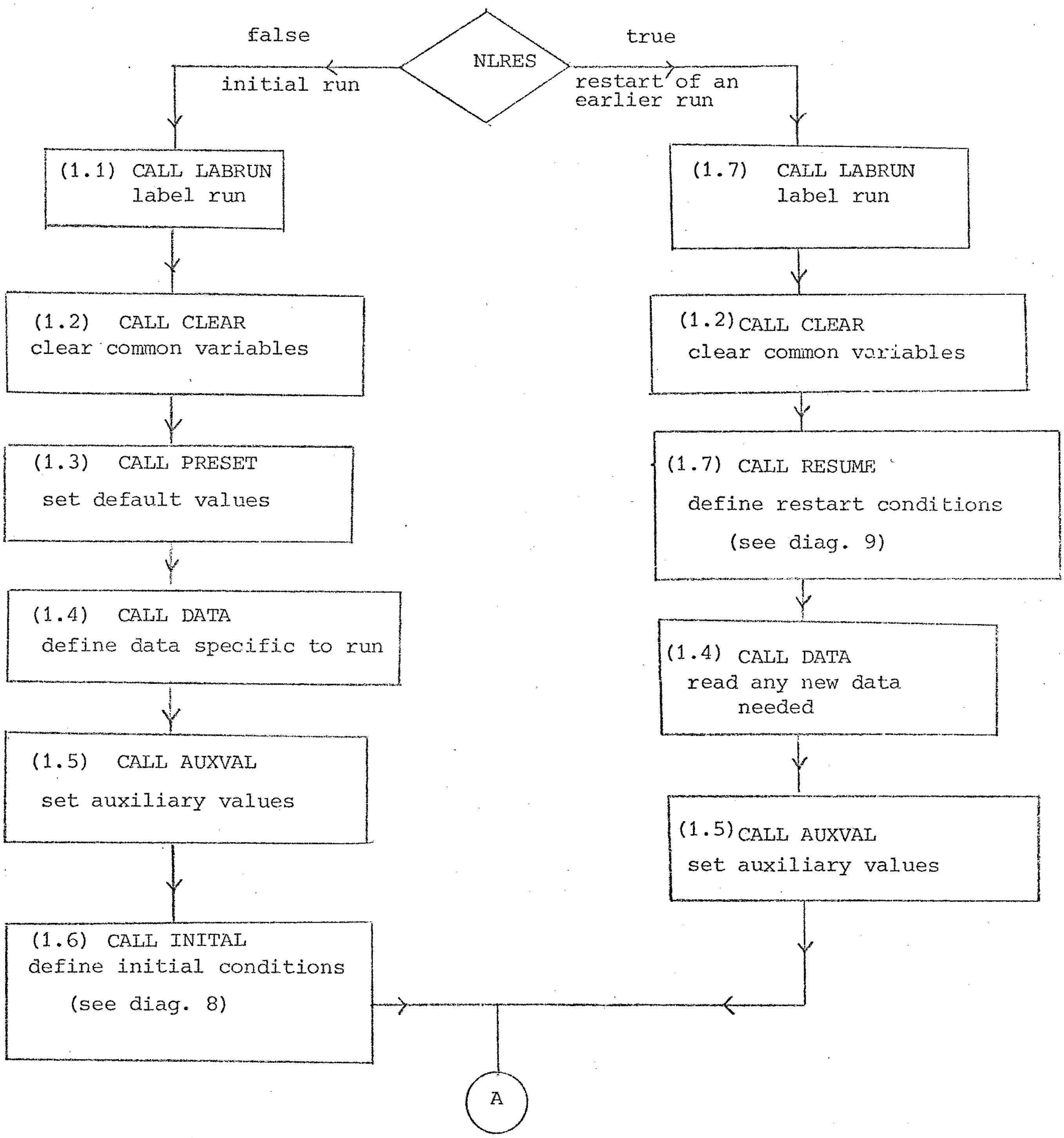


Diagram 7 (contd.)

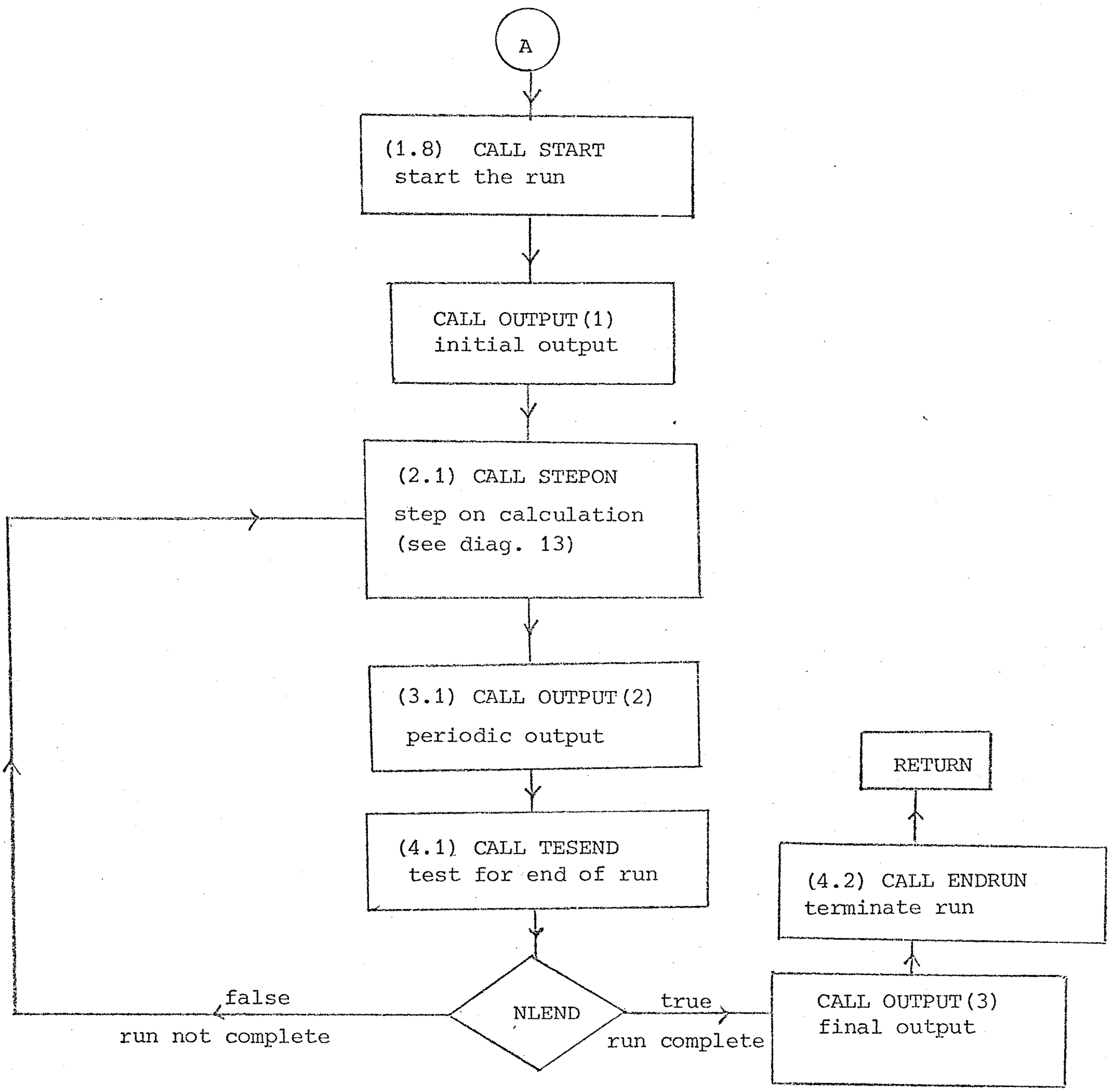


Diagram 8 <1.6> SUBROUTINE INITAL define initial conditions

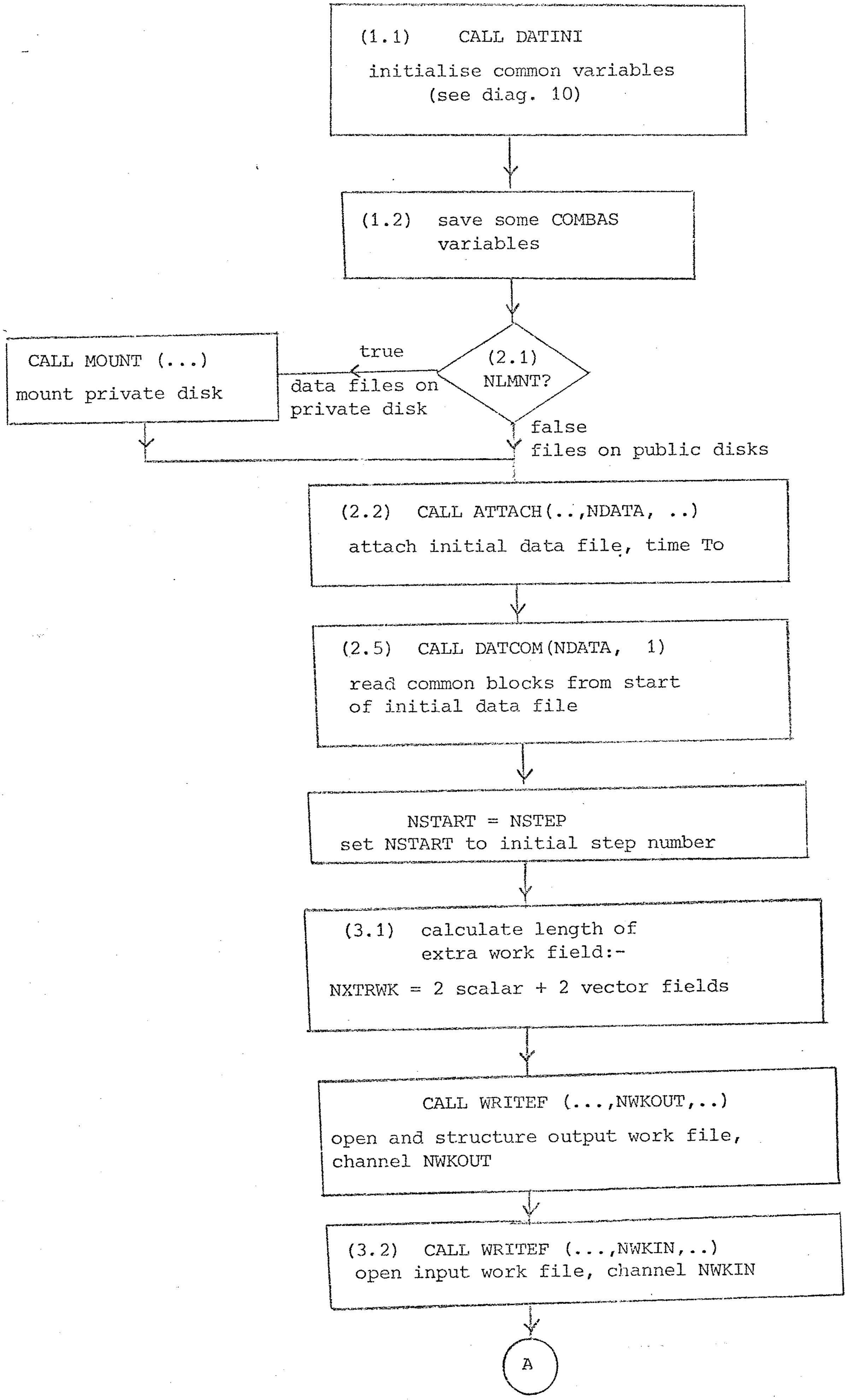


Diagram 8 (contd.1)

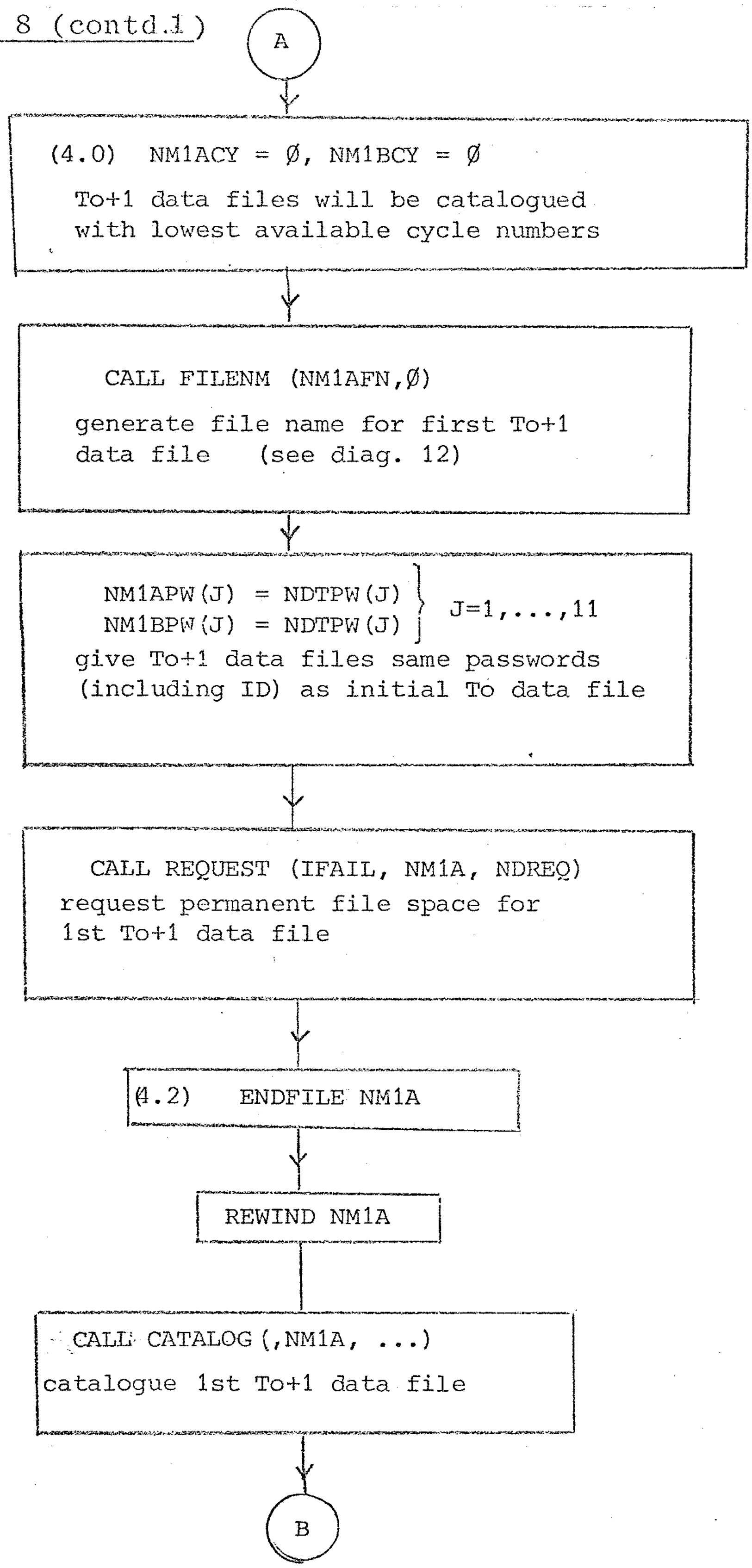


Diagram 8 (contd. 2)

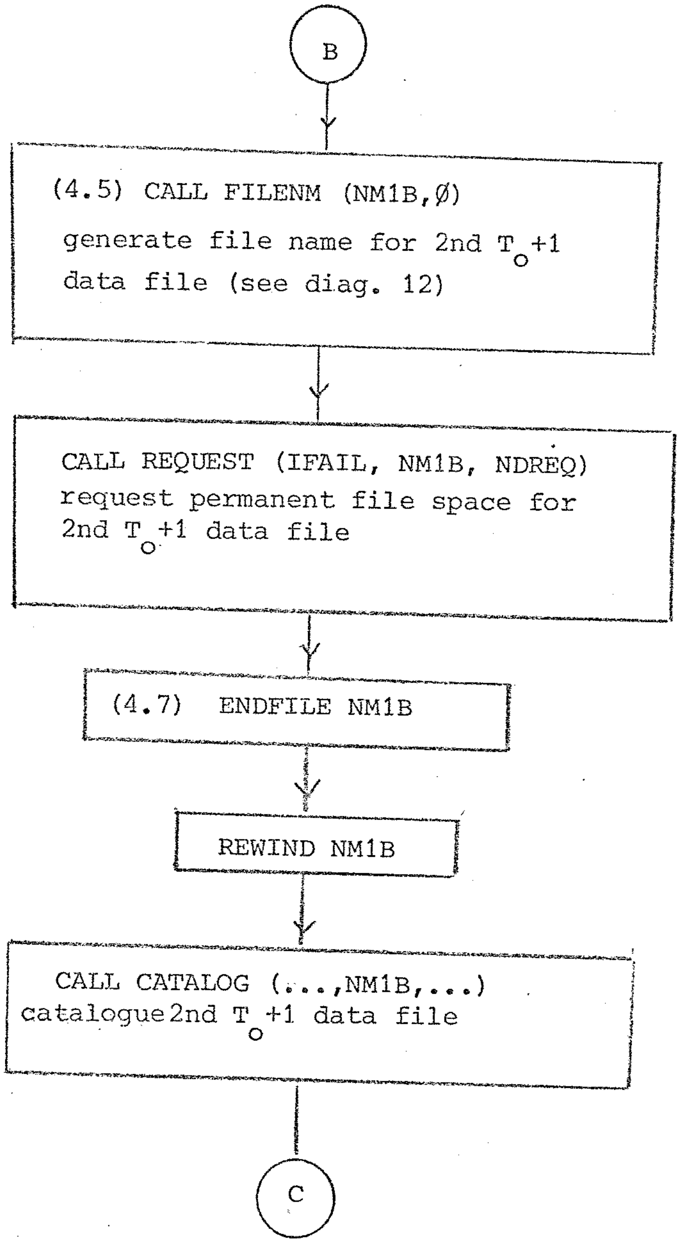


Diagram 8 (contd. 3)

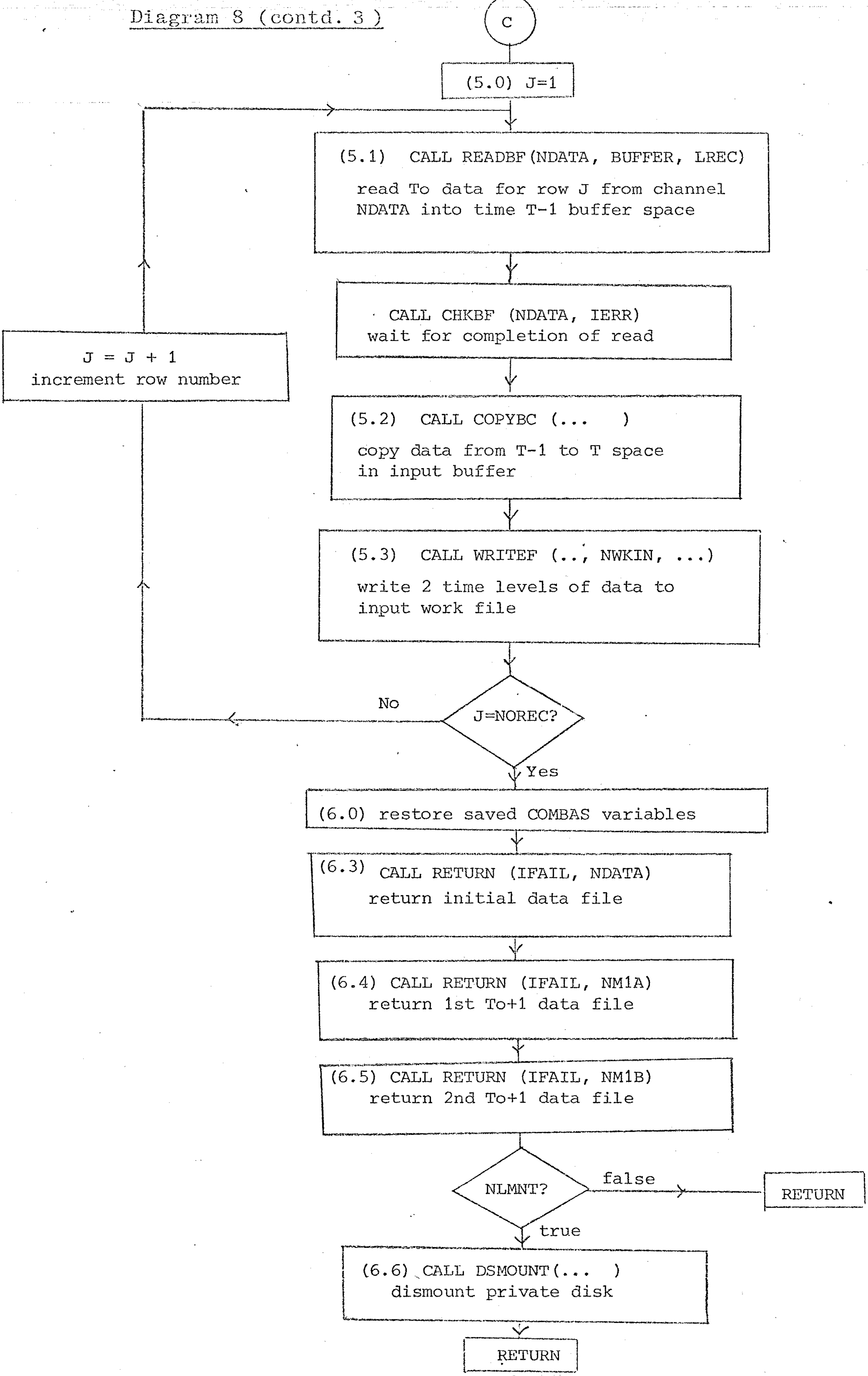


Diagram 9 <1.7> SUBROUTINE RESUME define restart conditions

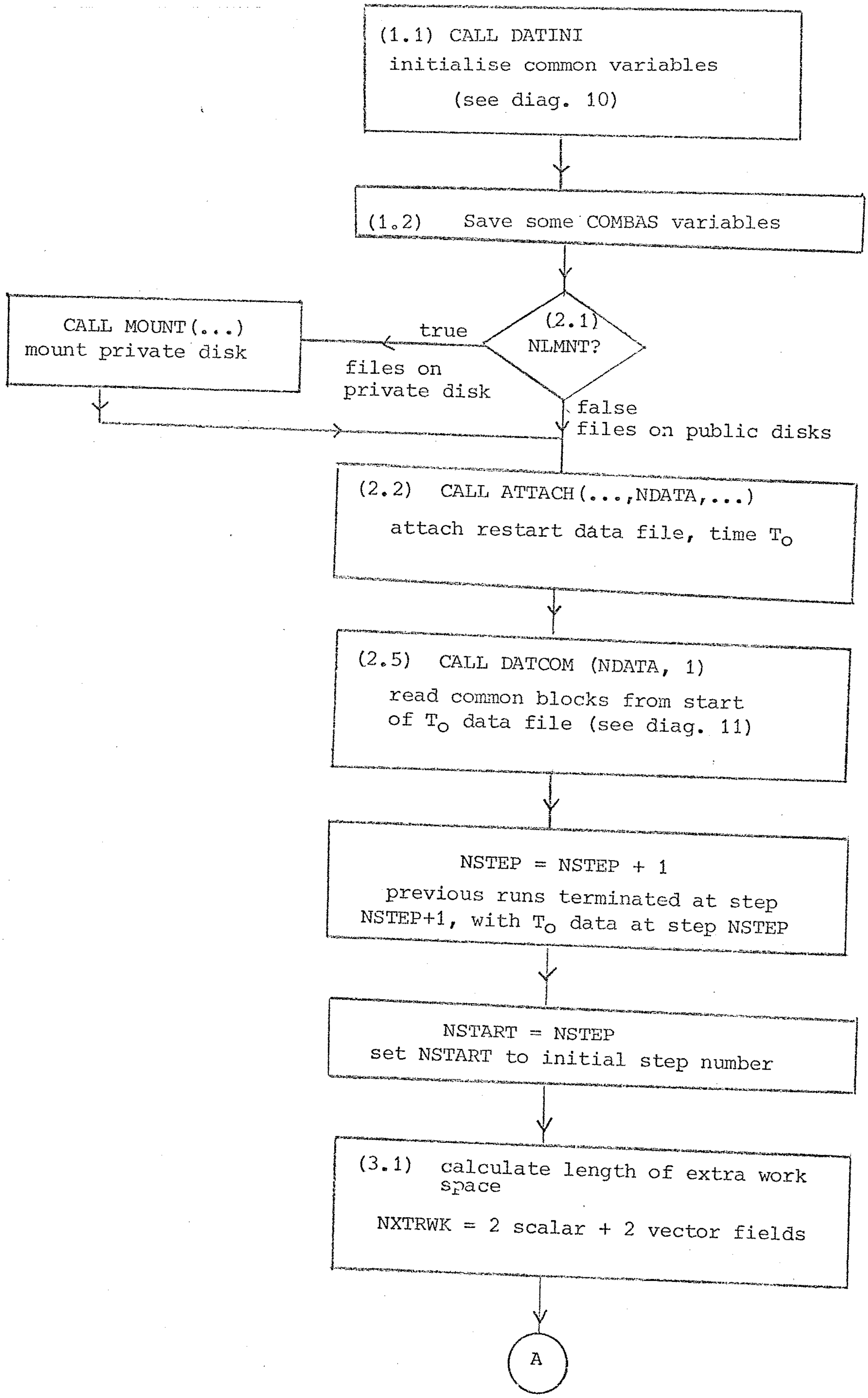


Diagram 9 (contd. 1)

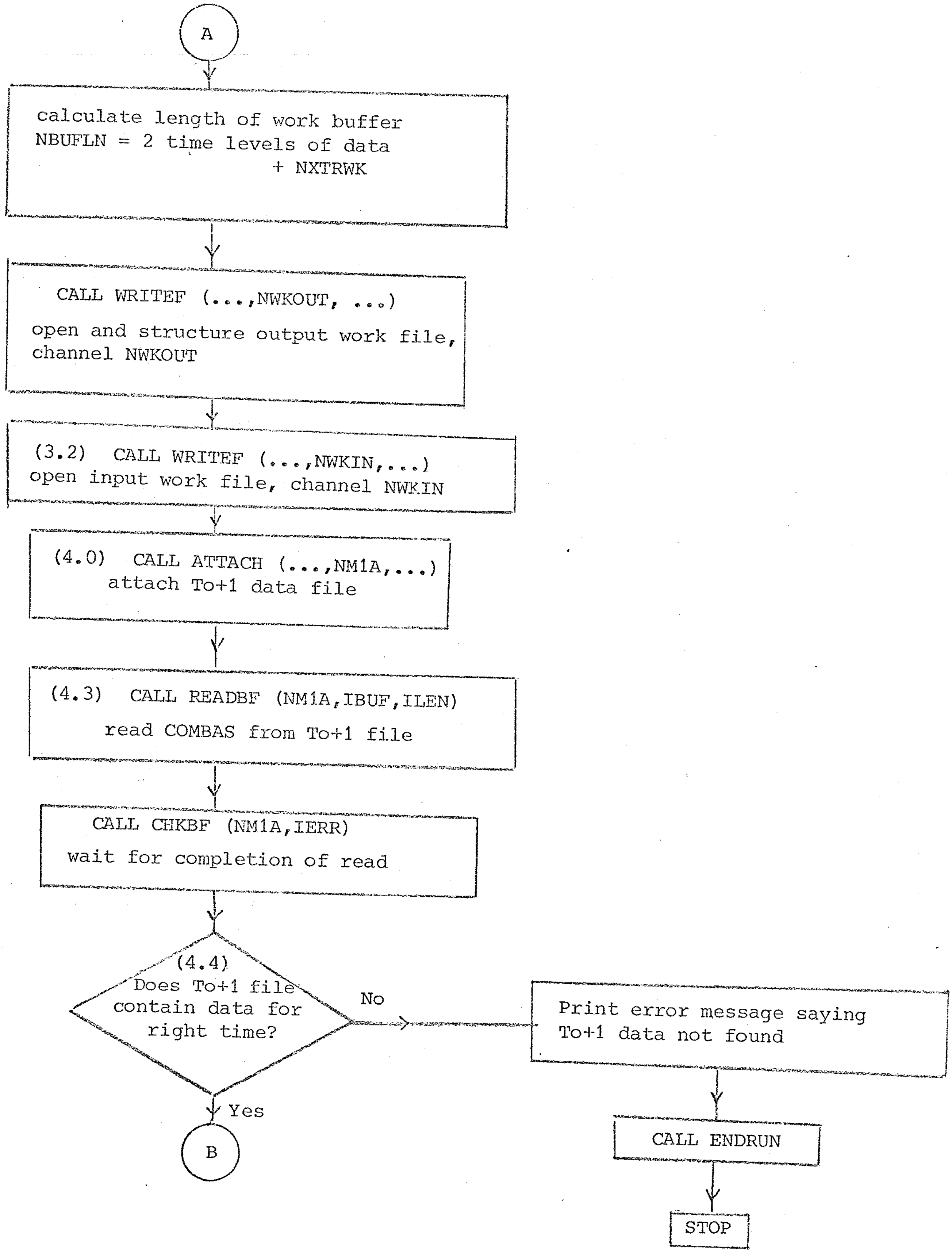


Diagram 9 (contd. 2)

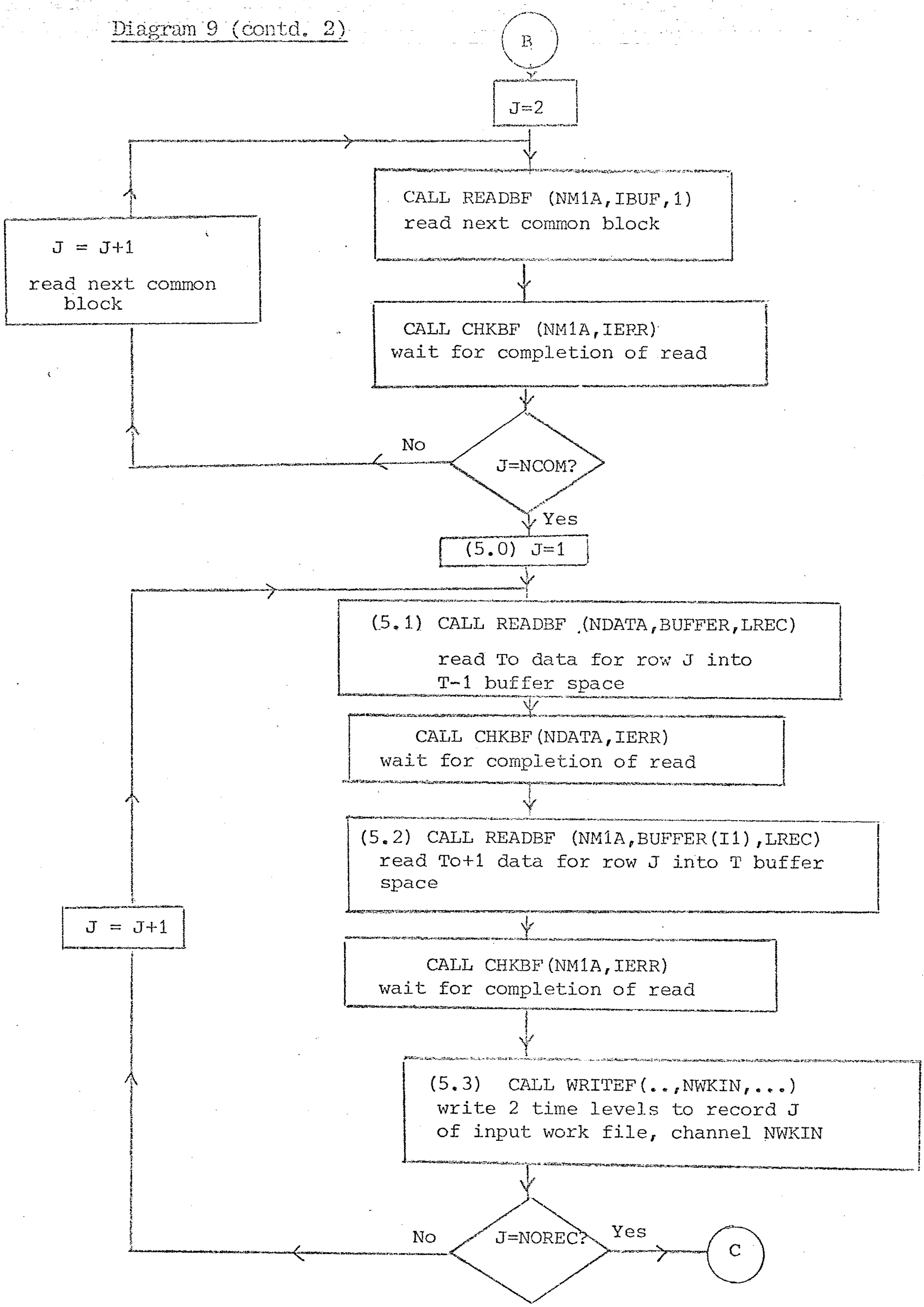


Diagram 9 (contd.3)

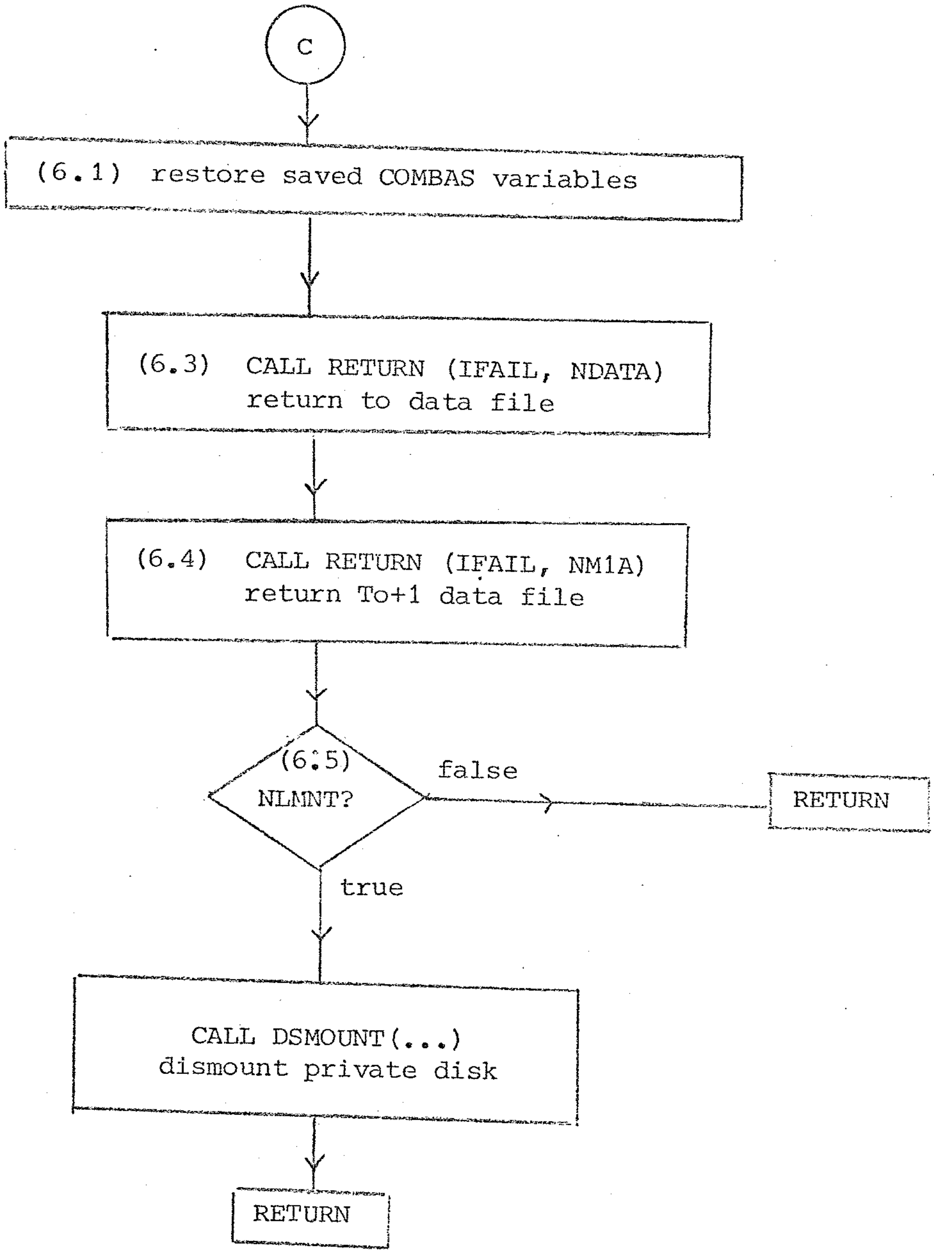


Diagram 10 1.10 SUBROUTINE DATINI initialise common variables

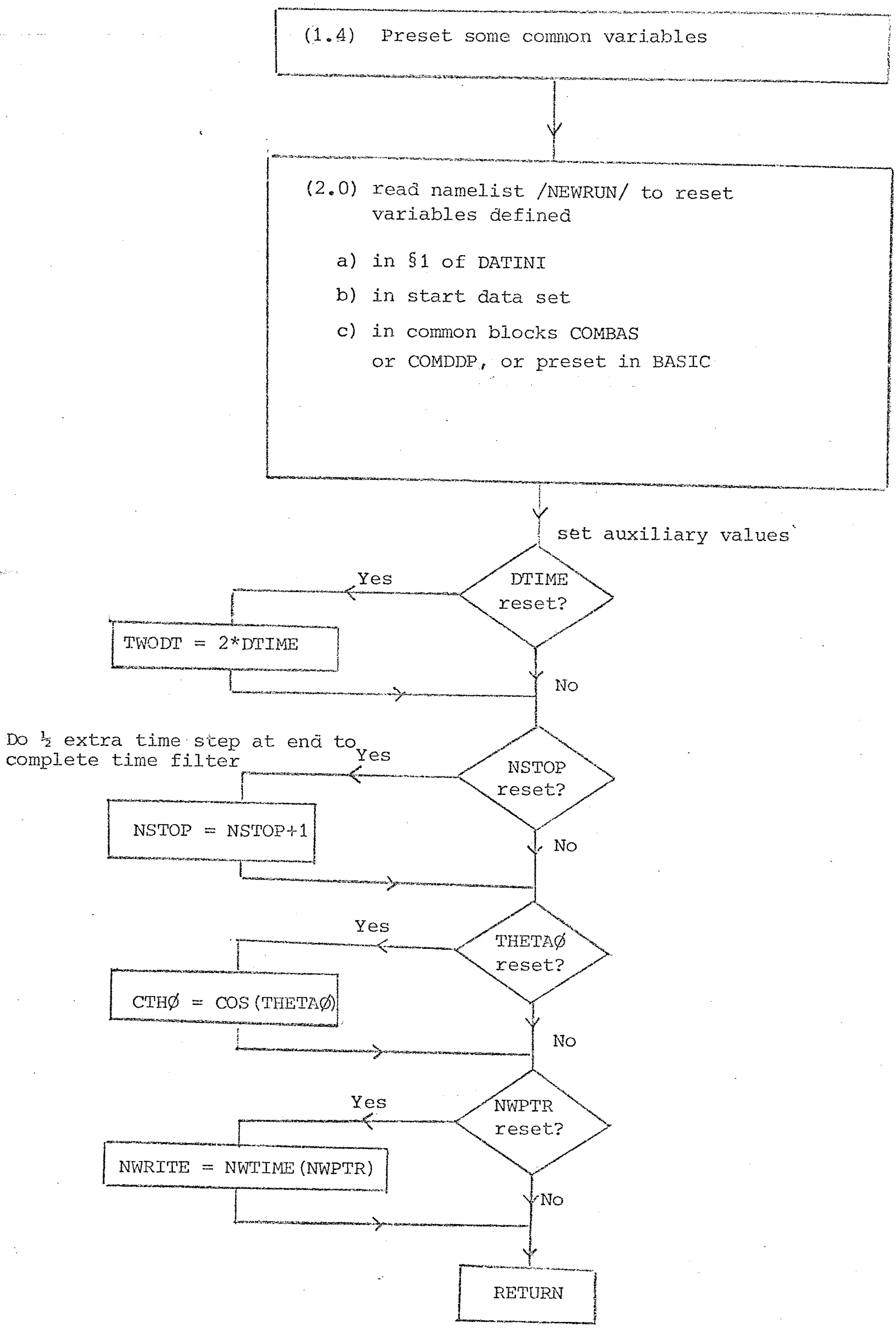


Diagram 11 <1.9> SUBROUTINE DATCOM (KDATA, KCALL)

read or write common blocks

Arguments: KDATA - channel number of data

KCALL=1 - read common blocks

KCALL=2 - write common blocks

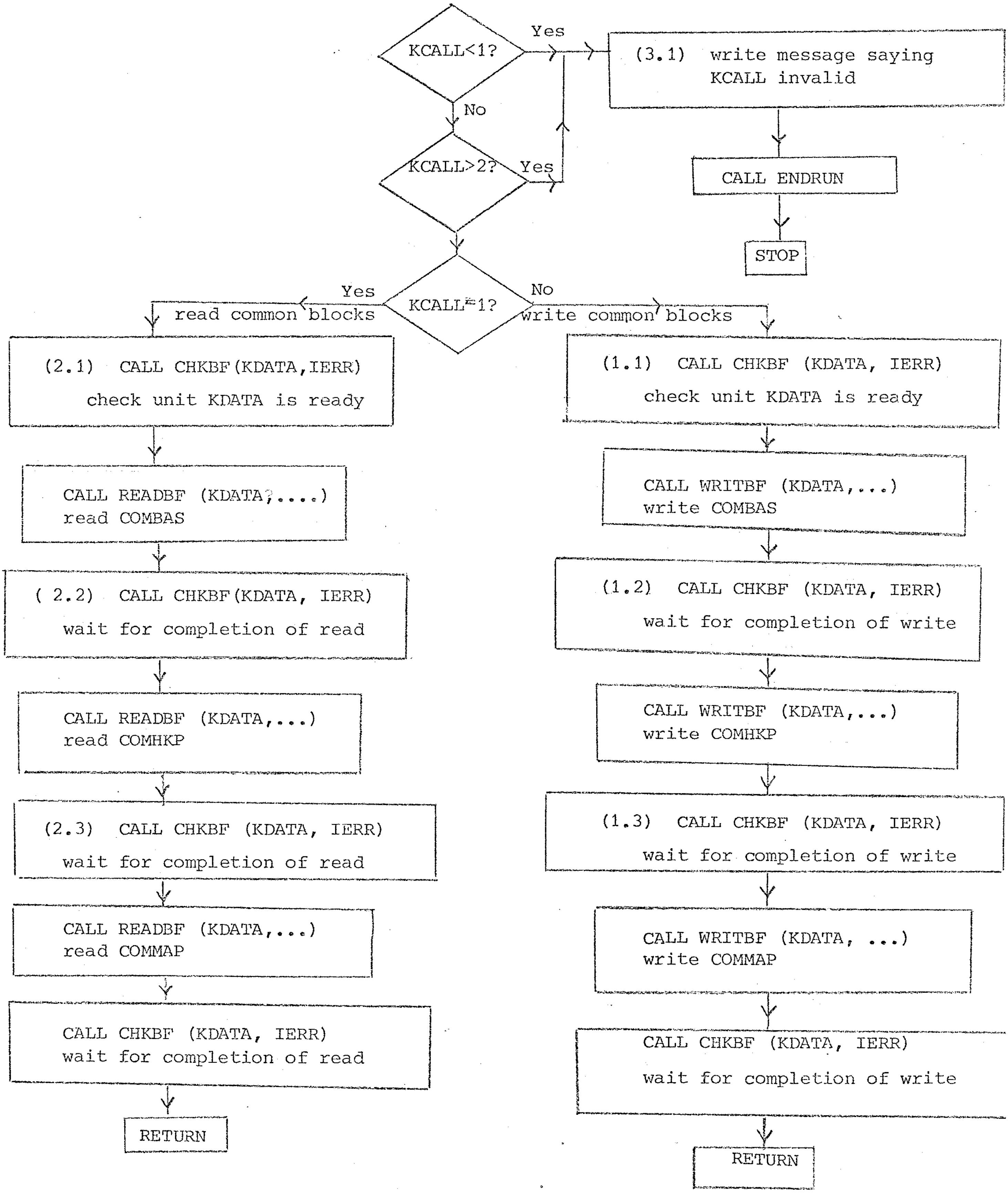
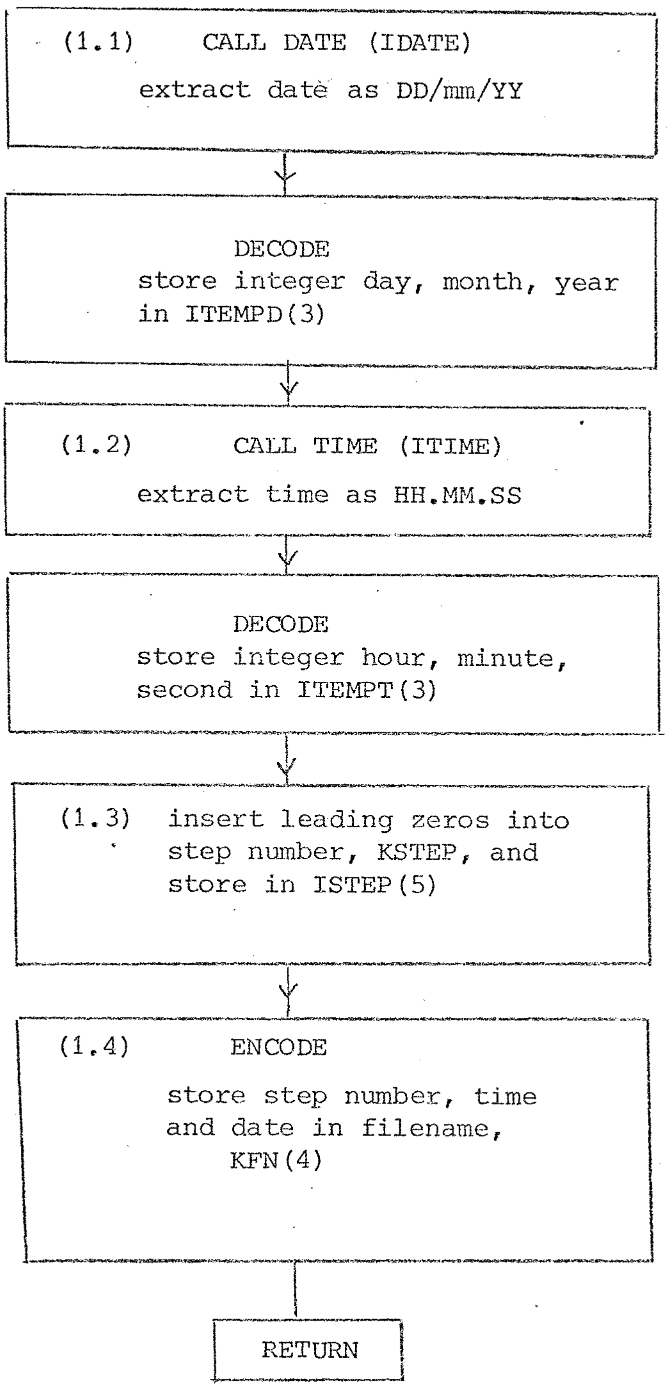


Diagram 12 <2.14> SUBROUTINE FILENM (KFN, KSTEP)

generate unique permanent file name

Arguments: KFN(4) - Hollerith characters for name, filled to right with blanks

KSTEP - step number to be included in name



e.g. file created at 10.21.43am on 12/2/77 at step KSTEP=38 would have the name GEMINISTEPØØ38X10X21X43X12X02X77

Diagram 13 <2.1> SUBROUTINE STEPON Advance calculation

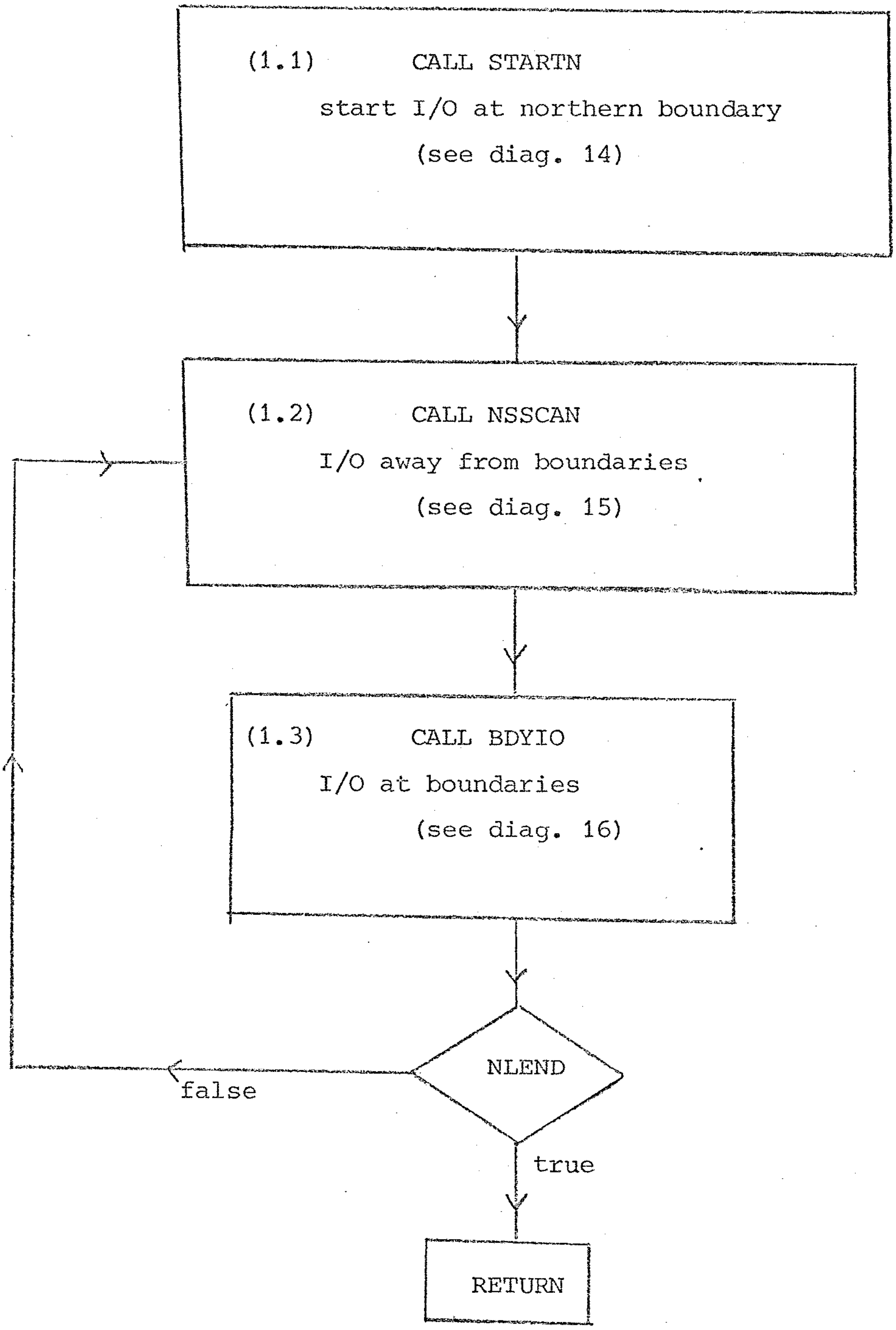


Diagram 14 <2.7> SUBROUTINE STARTN Start I/O at northern boundary

(1.1) Initialise displacements in blank common of 4 input buffers.
Each buffer is of length NBUFLN,
with start addresses held in NLINE1(4)

(1.2) Initialise displacements in blank common of 2 output buffers, held in NLINE2(2)

NROW = 1
row number

(2.1) read data for northern row from input workfile, channel NWKIN, into buffer starting at NLINE1(2)

(2.2) read data for 1 row to south from NWKIN into buffer starting at NLINE1(3)

(2.3) start to read data for 2 rows to south from NWKIN into buffer starting at NLINE1(4)

(2.4) CALL LINEMS
perform calculations on northern row
(see diag. 17)

NROW = NROW + 1
increment row number

RETURN

Diagram 15 <2.8> SUBROUTINE NSSCAN I/O away from boundaries

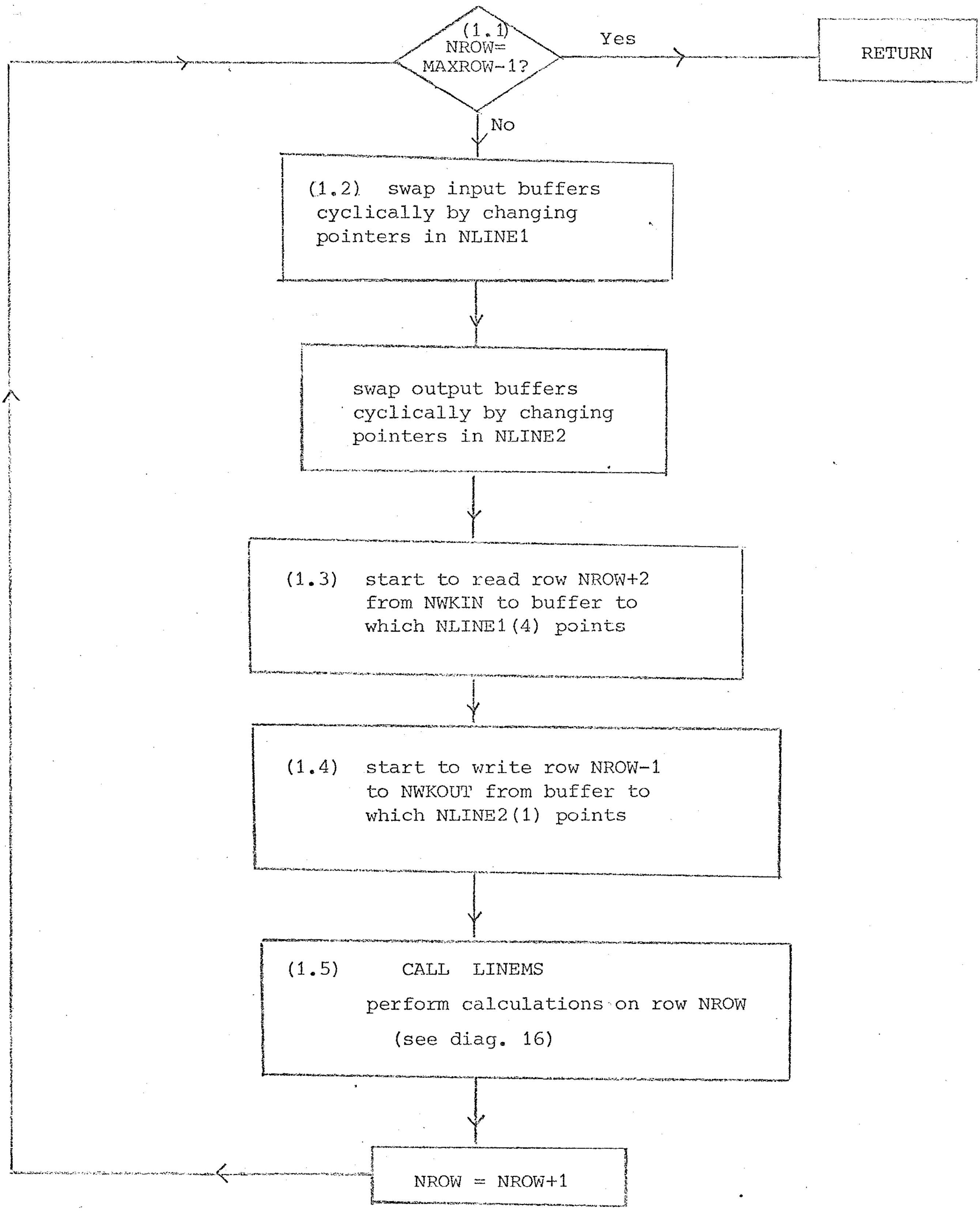


Diagram 16 <2.12> SUBROUTINE BDYIO I/O at boundaries

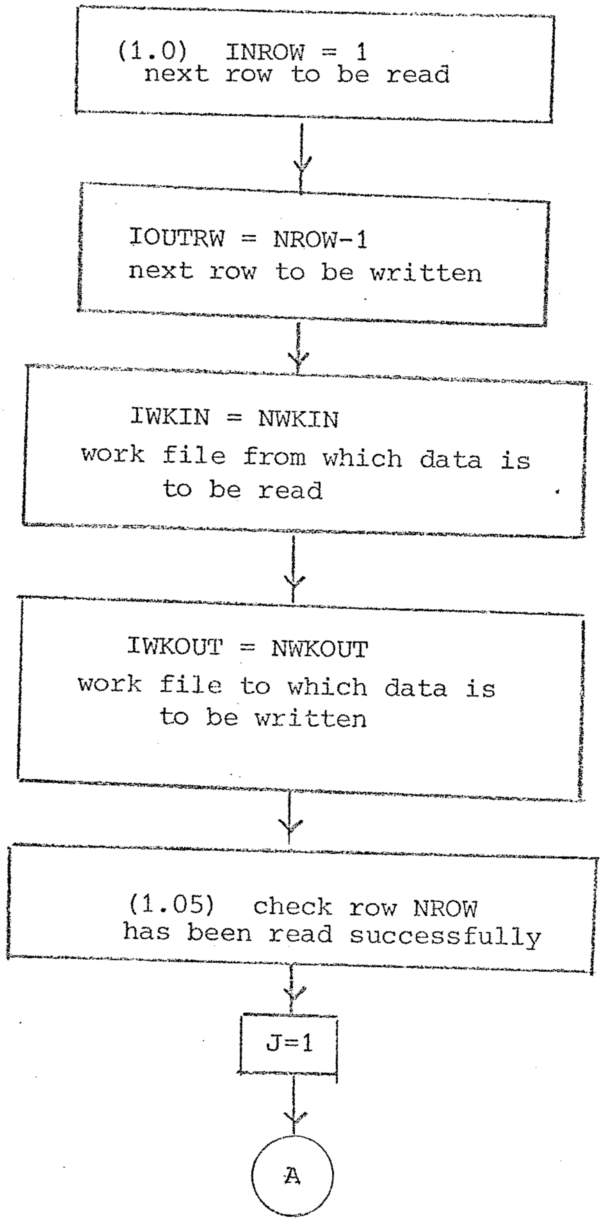


Diagram 16 (contd.1)

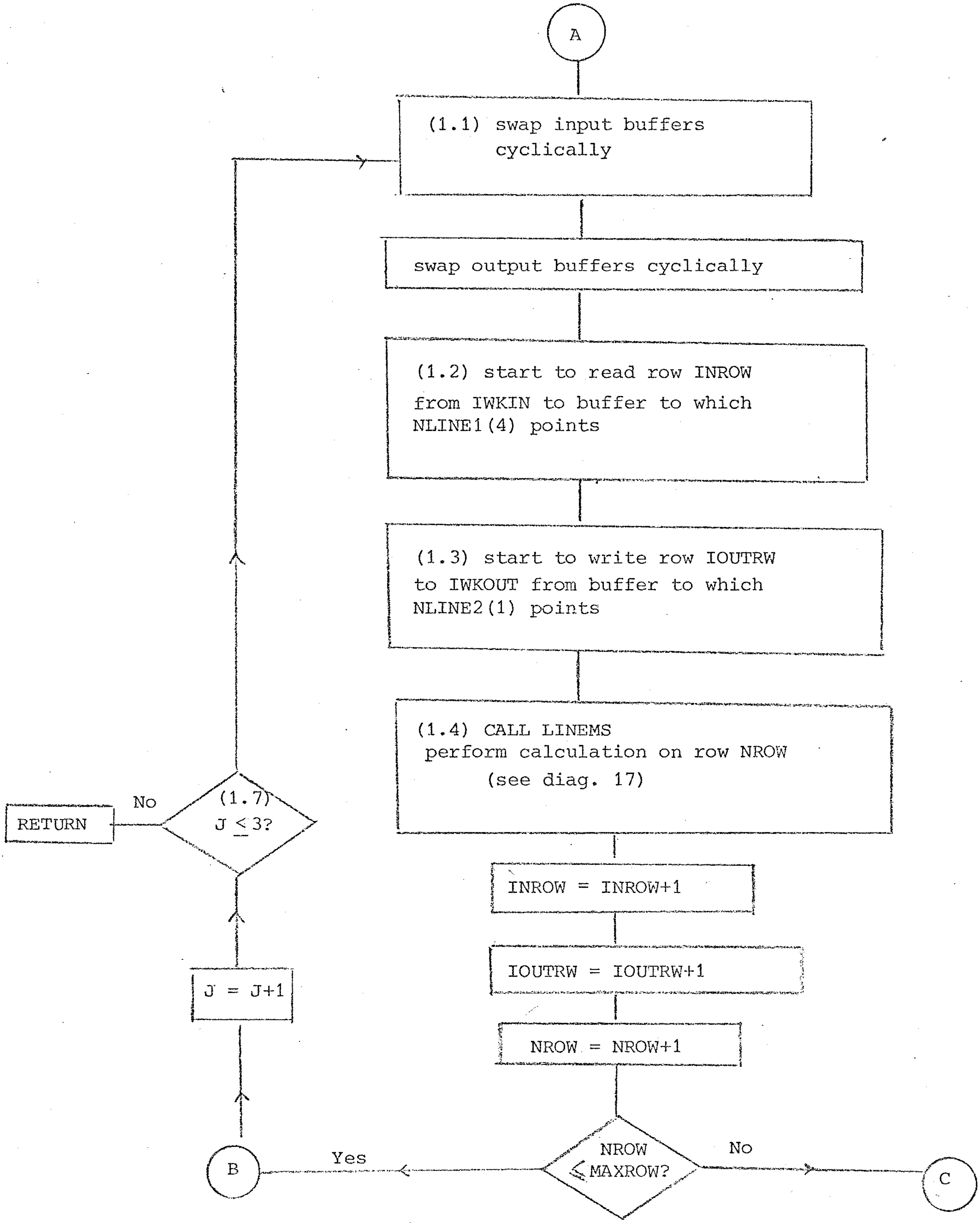


Diagram 16 (contd.2)

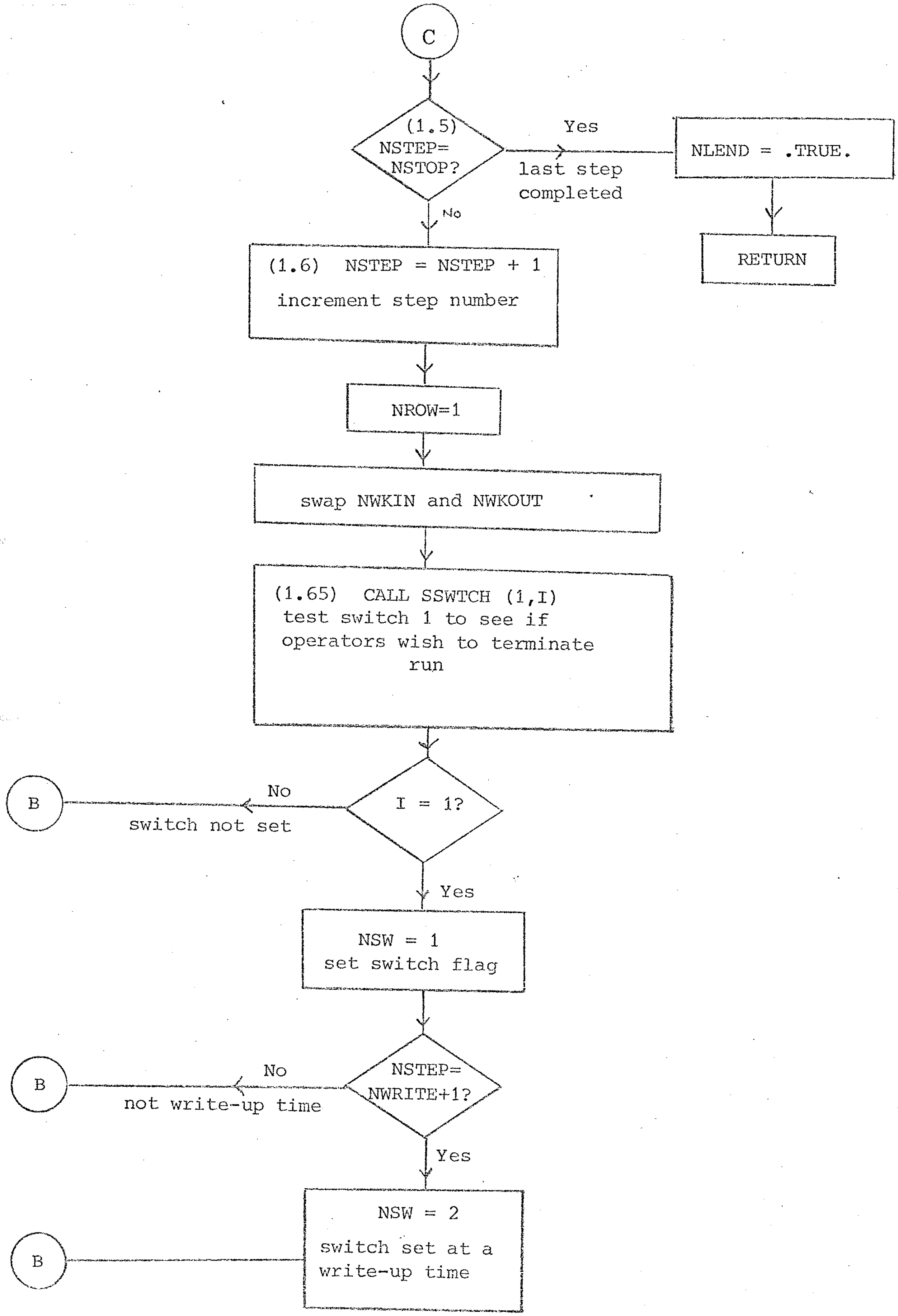


Diagram 17 <2.13> SUBROUTINE LINEMS control calculation

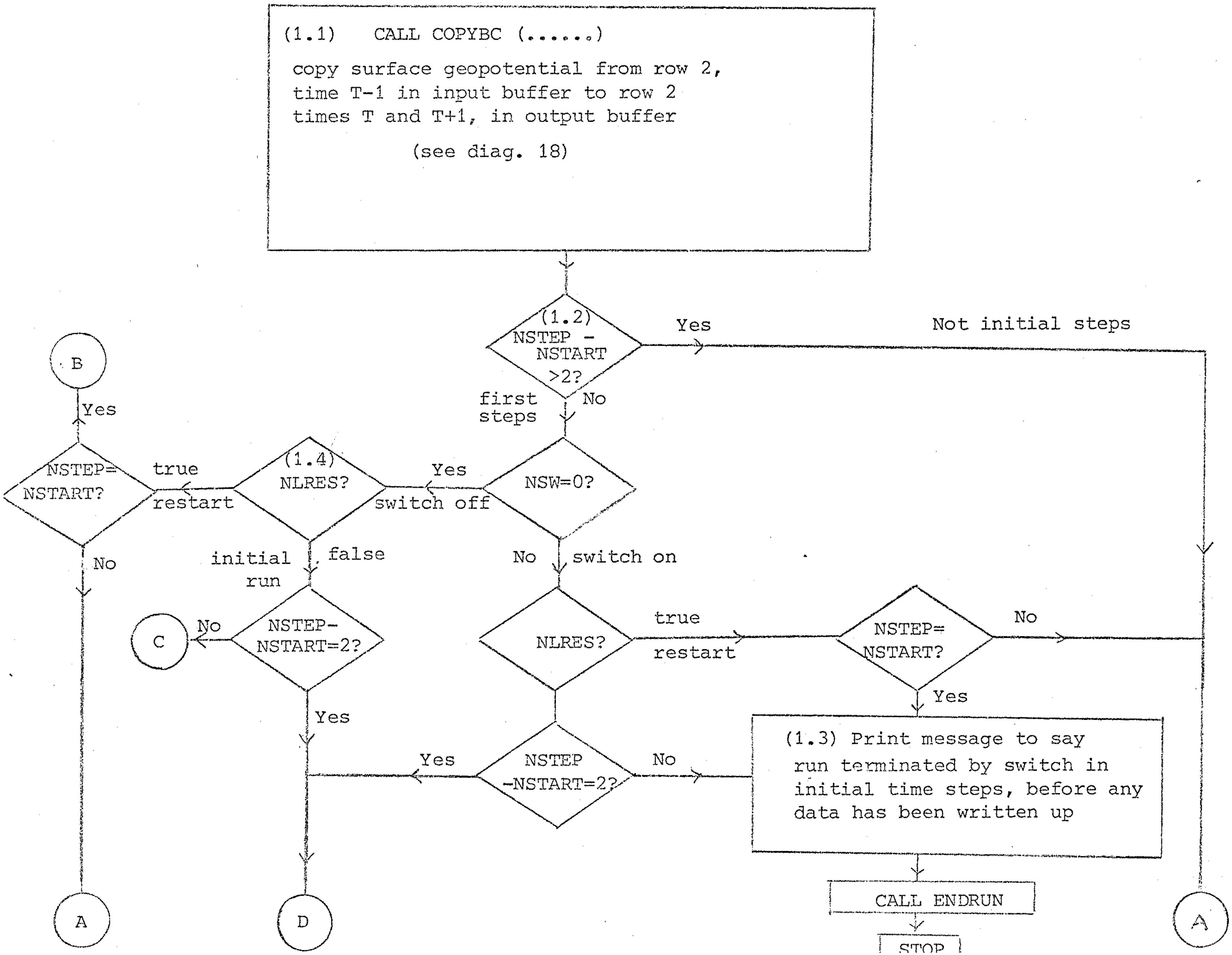


Diagram 17 (contd. 1)

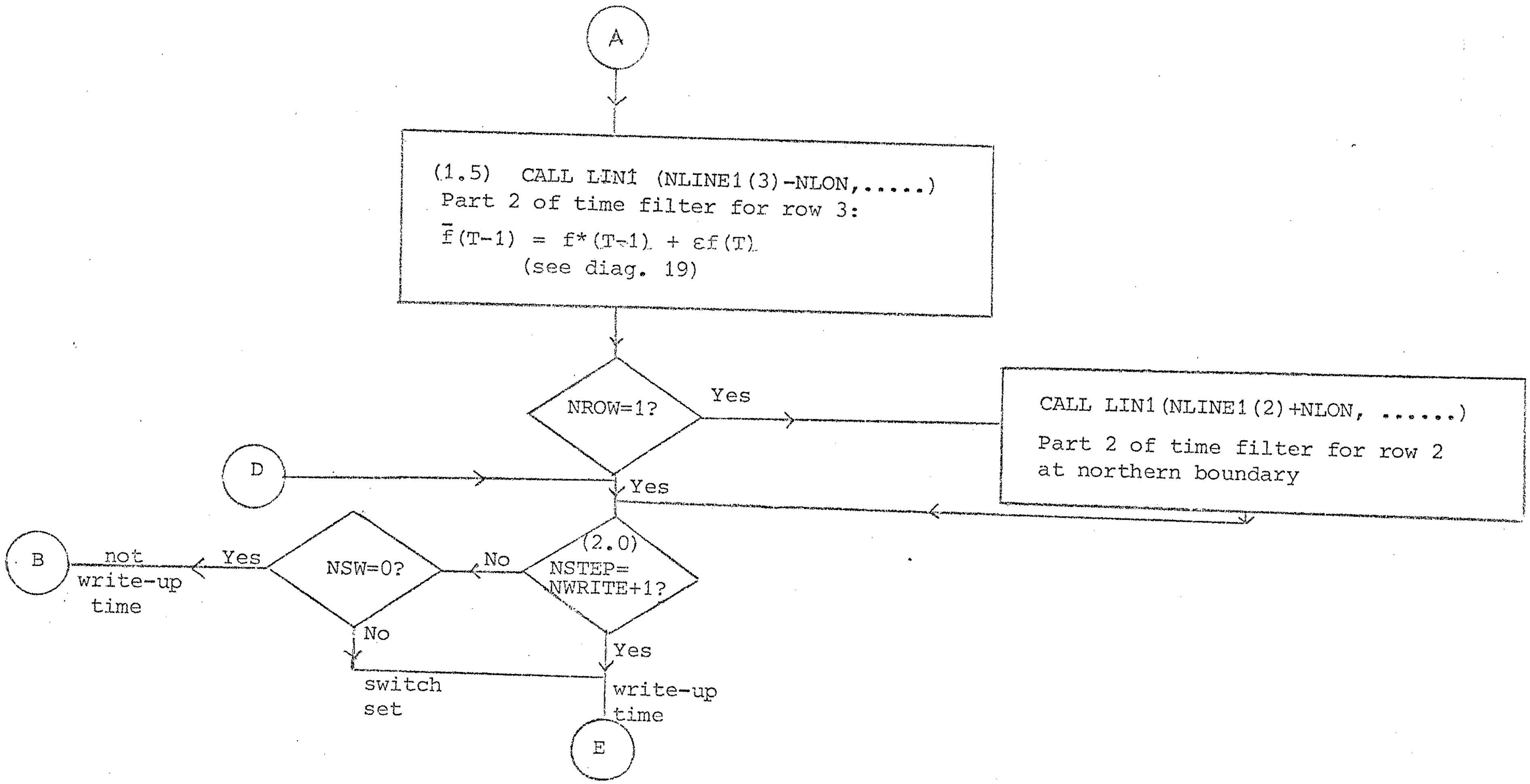


Diagram 17 (contd. 2)

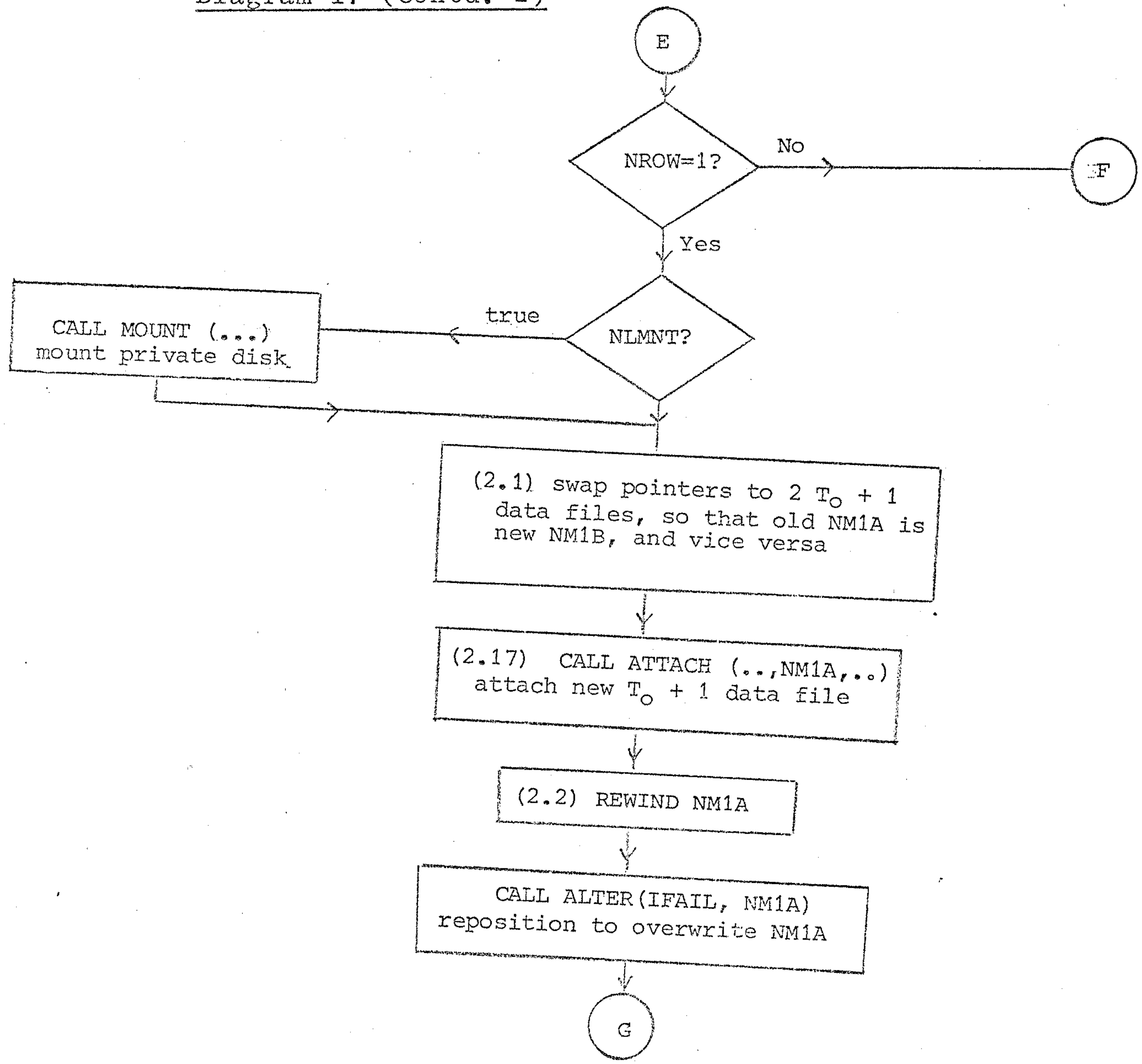


Diagram 17 (contd. 3)

G

NREC = NREC + 1
increment start data set record number

CALL DATCOM (NM1A, 2)
write common blocks to start
of T_o + 1 data file

(2.25) CALL REQUEST (IFAIL,NDATA,NDREQ)
request permanent file space for
T_o data

NSTEP = NSTEP - 1
Decrement NSTEP before writing common
blocks to front of NSTEP-1 (ie T_o) data

CALL DATCOM (NDATA, 2)
write common blocks to start of T_o
data file

NSTEP = NSTEP + 1
Reset NSTEP to current value

H

Diagram 17 (contd.4)

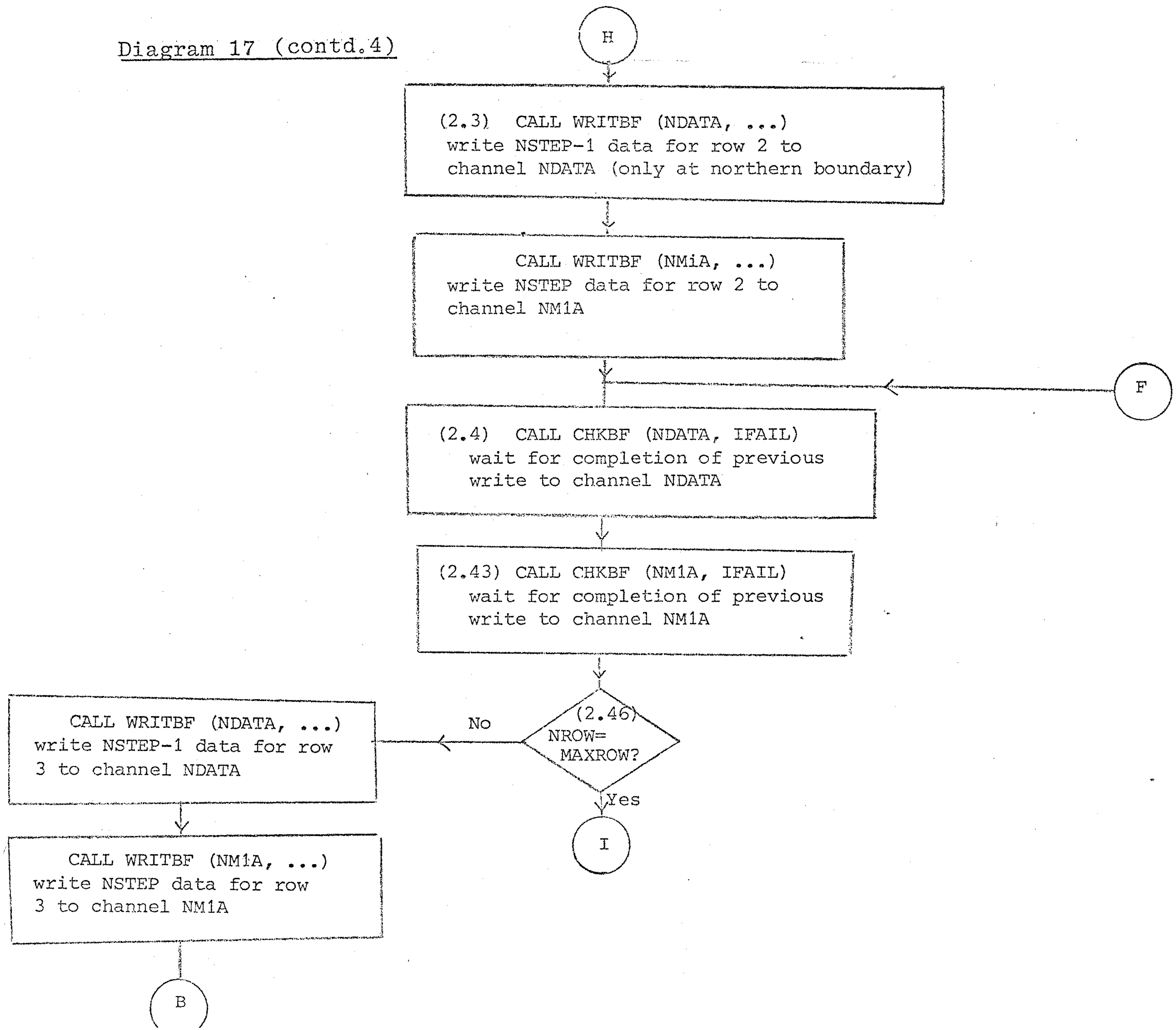


Diagram 17 (contd. 5)

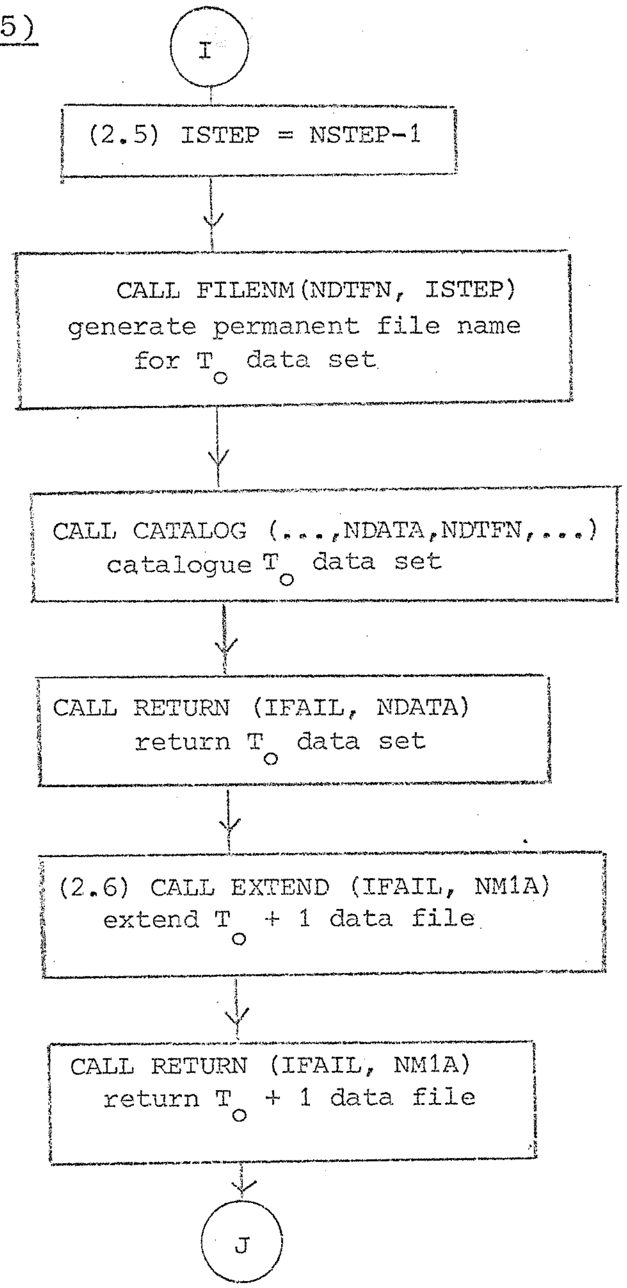


Diagram 17 (contd.6)

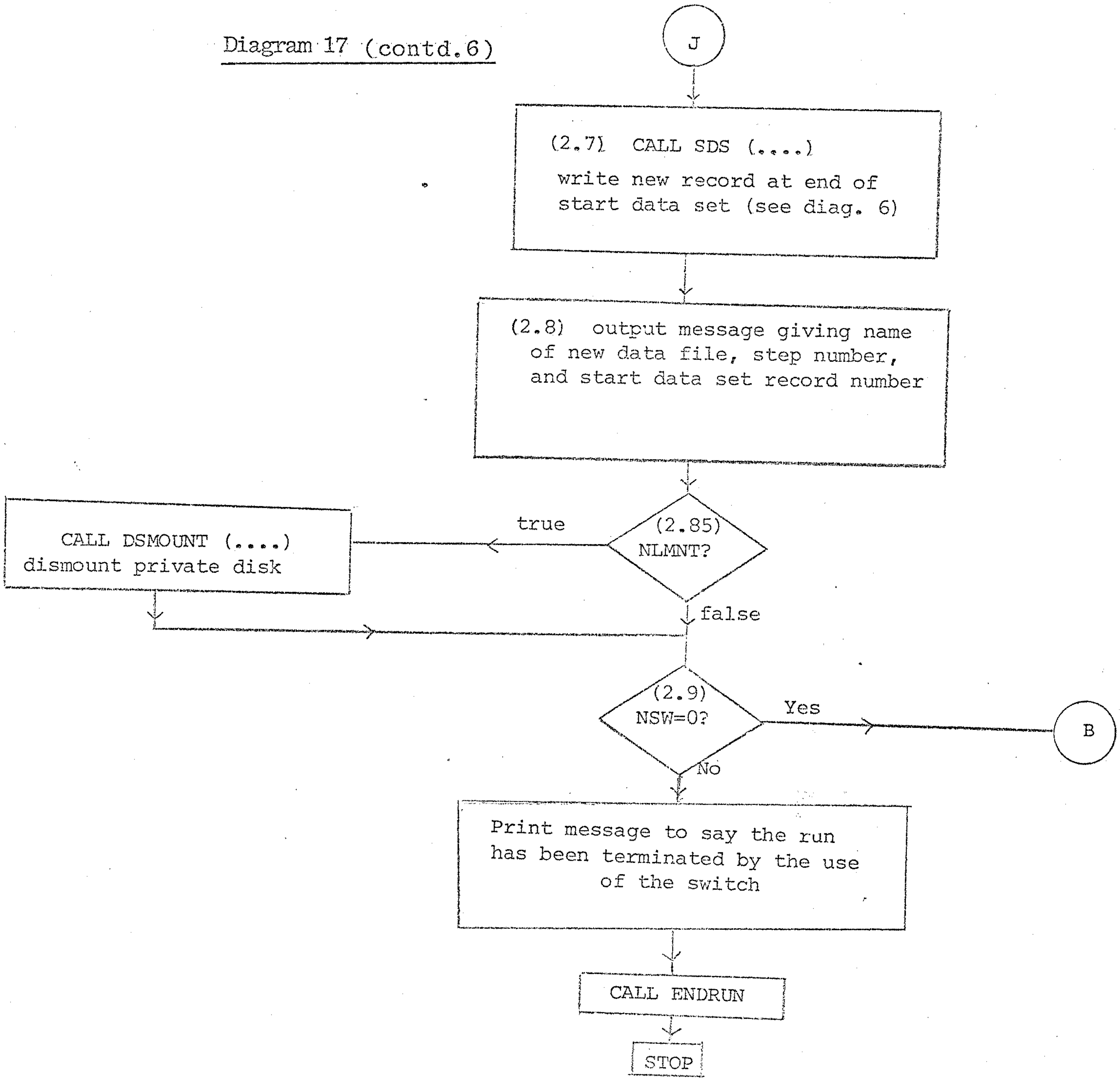


Diagram 17 (contd. 7)

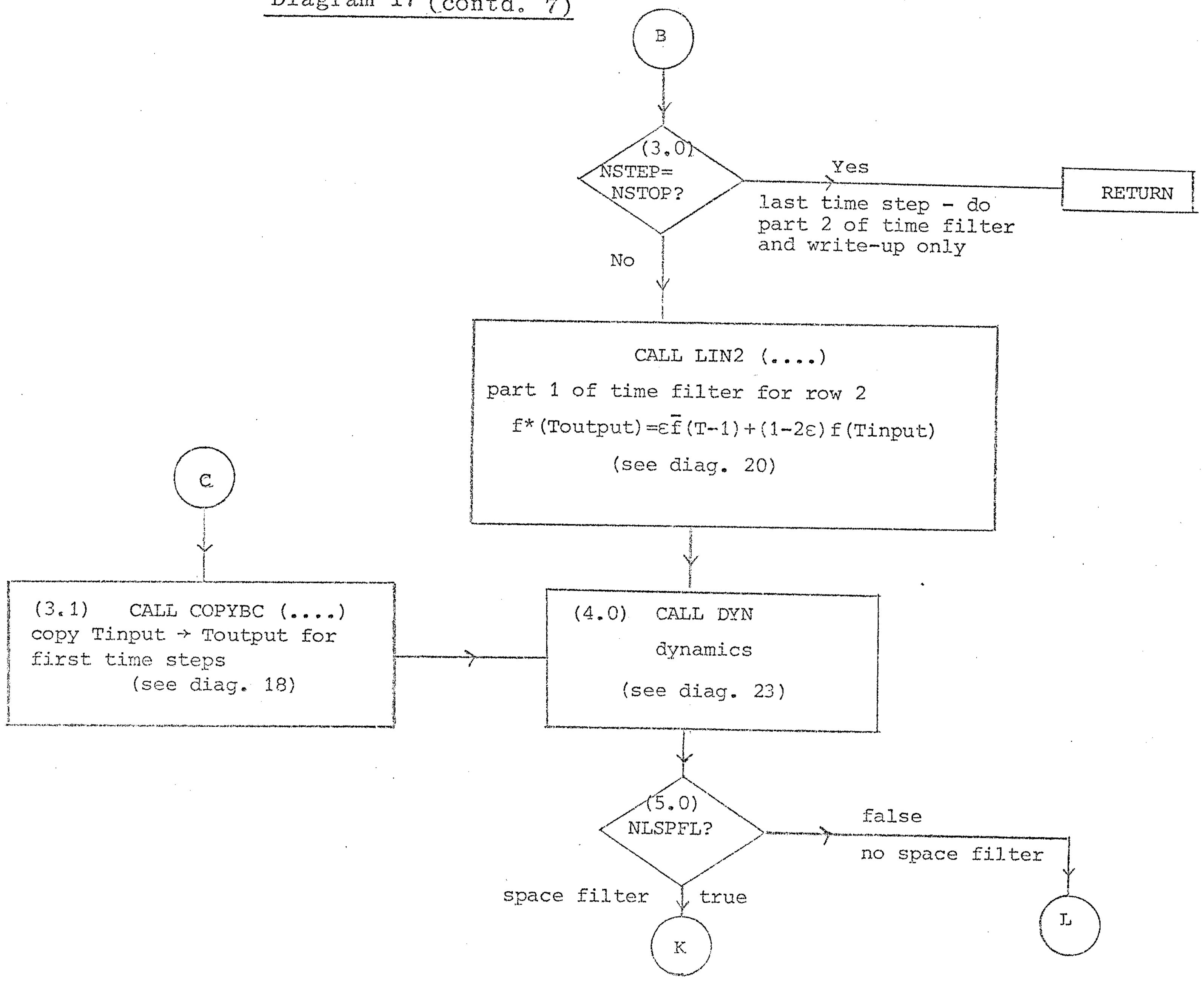


Diagram 17 (contd. 8)

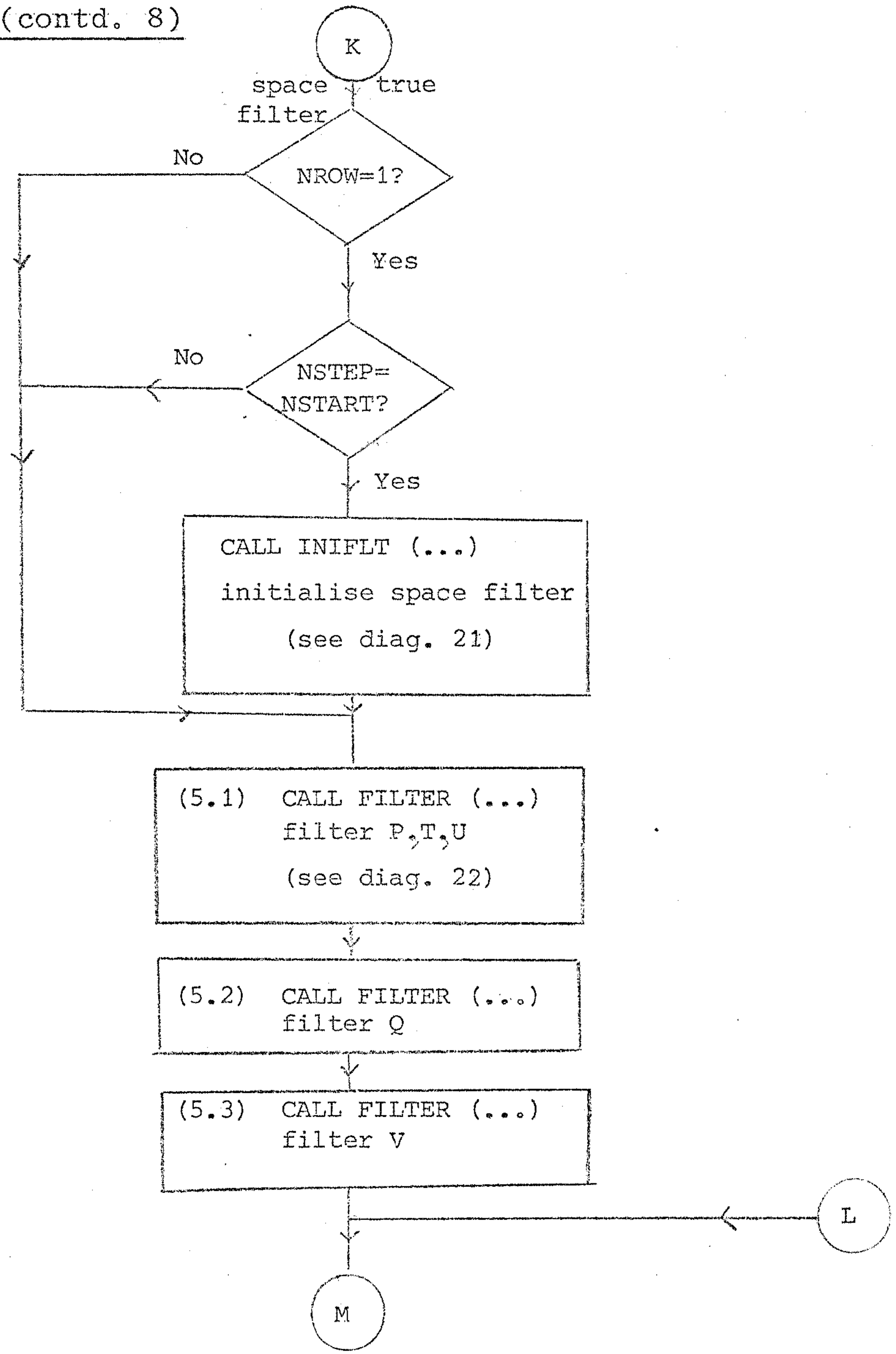


Diagram 17 (contd. 9)

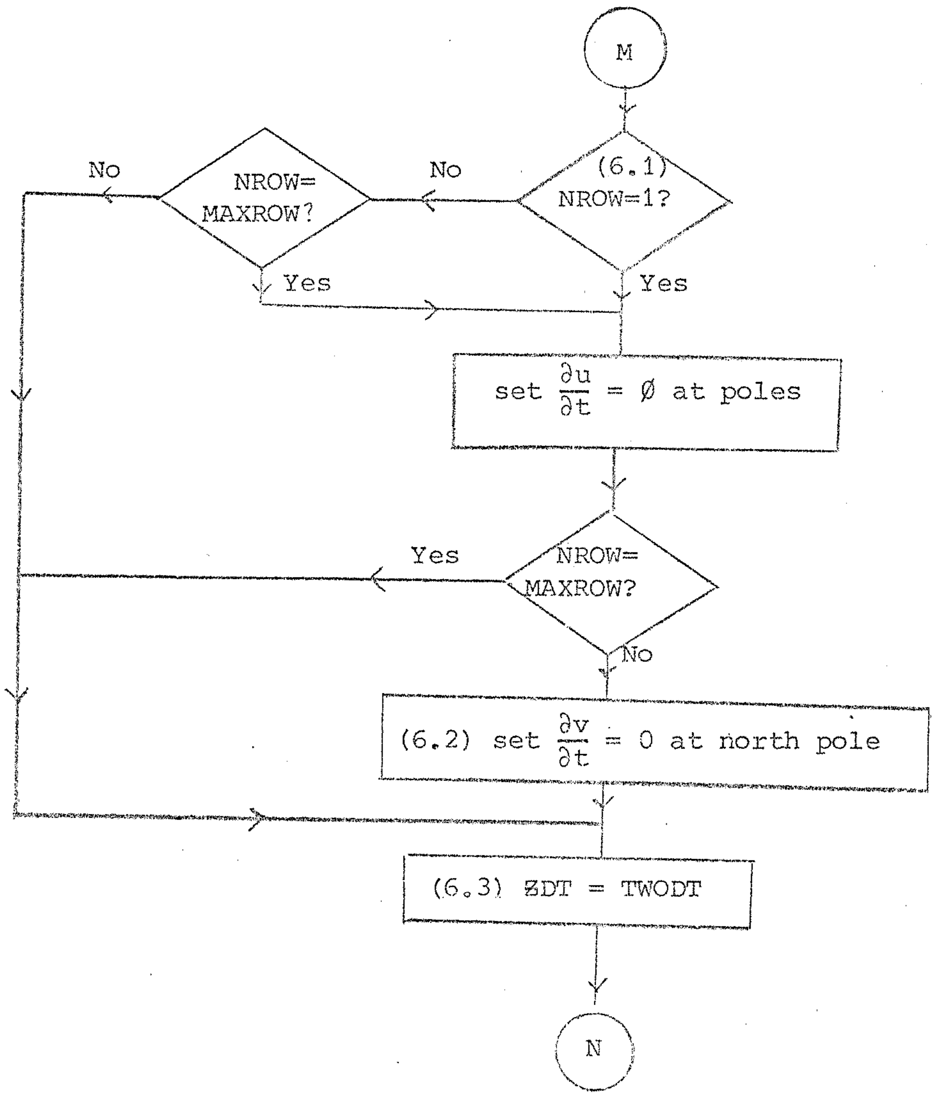


Diagram 17 (contd. 10)

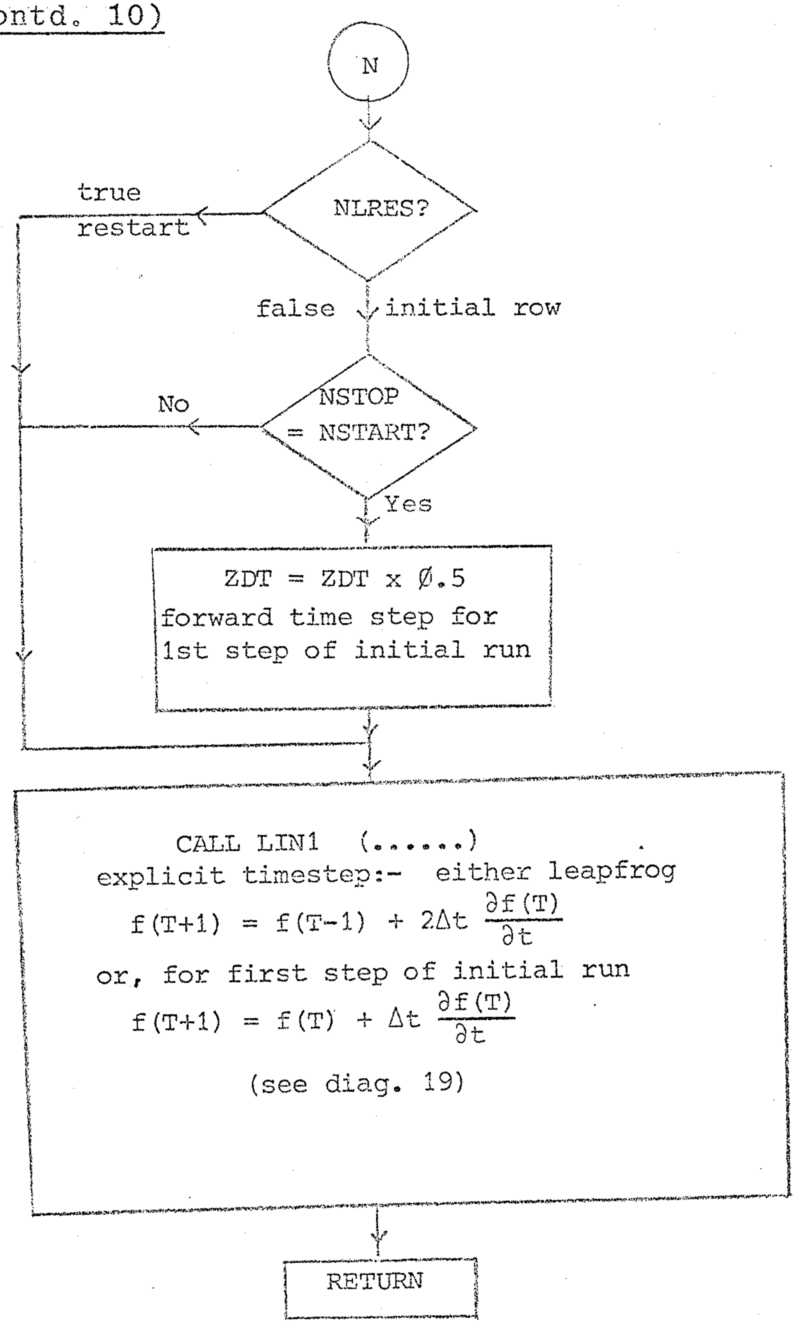


Diagram 18 <5.6> SUBROUTINE COPYBC (KDISA, KDISB, KNO)

copy data in blank common

Arguments: KDISA: displacement in blank common of first element to be copied

KDISB: displacement in blank common of first element of area into which data is to be copied

KNO : number of words to be copied

Assign subroutine arguments to local variables

IDISA = KDISA, etc.

↓

CALL MOVLEV (BUFFER(IDISA), BUFFER(IDISB), INO)

where Buffer(1) is the first element of blank common

↓

RETURN

Diagram 19 <2.16> SUBROUTINE LIN1(KDISA, KDISB, KDISC, PB, KNO)

linear combination of 2 arrays in blank common: C = A + PB x B

Arguments:

KDISA: displacement in blank common of first element of A

KDISB: displacement in blank common of first element of B

KDISC: displacement in blank common of first element of C

PB : constant such that C(J) = A(J)+PBxB(J)

KNO : length of arrays

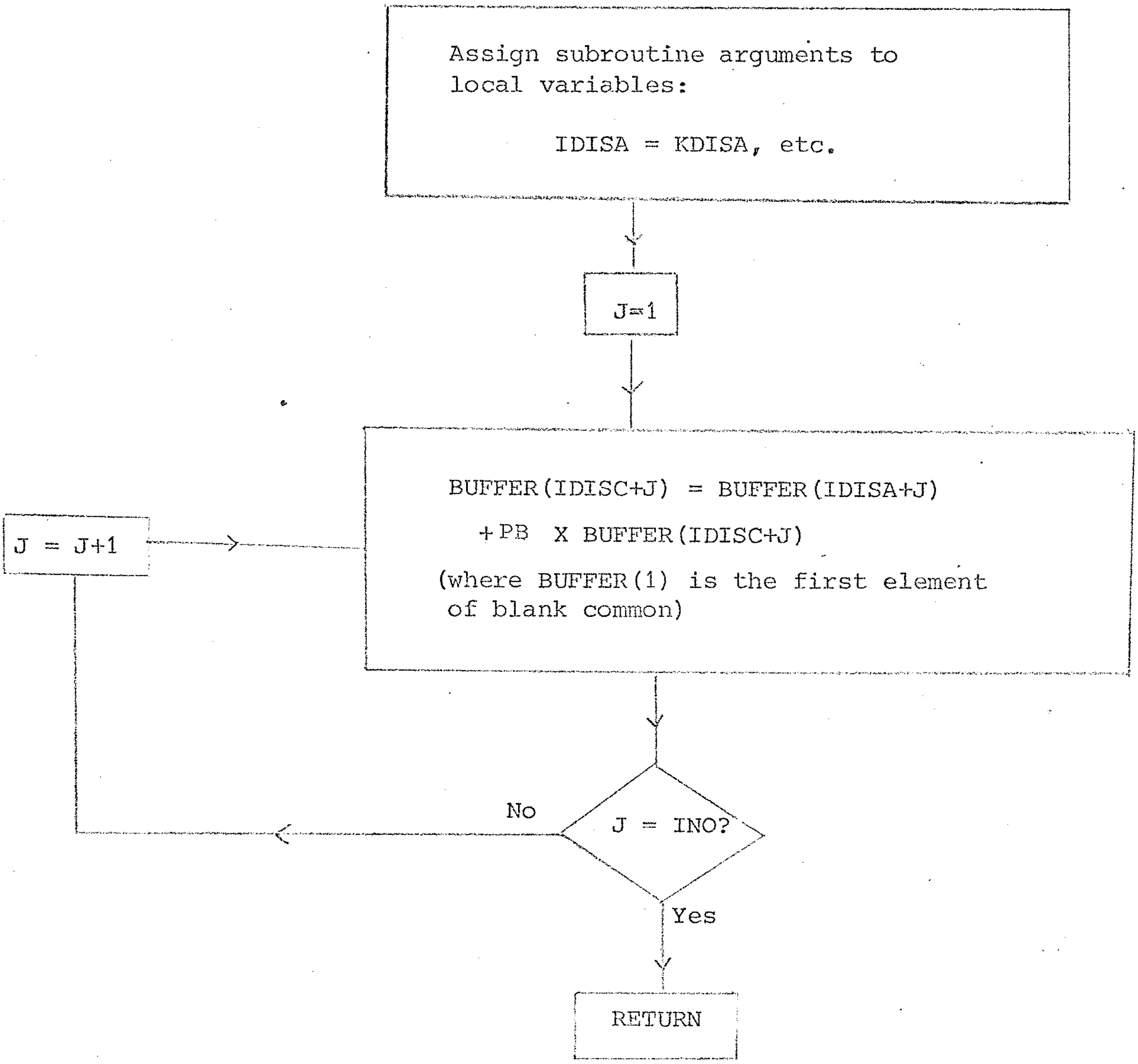


Diagram 20 <2.17> SUBROUTINE LIN2(KDISA, KDISB, KDISC, PA,PB,KNO)

linear combination of 2 arrays in blank common: C = PA×A+PB×B

Arguments:

KDISA: displacement in blank common of first element of A

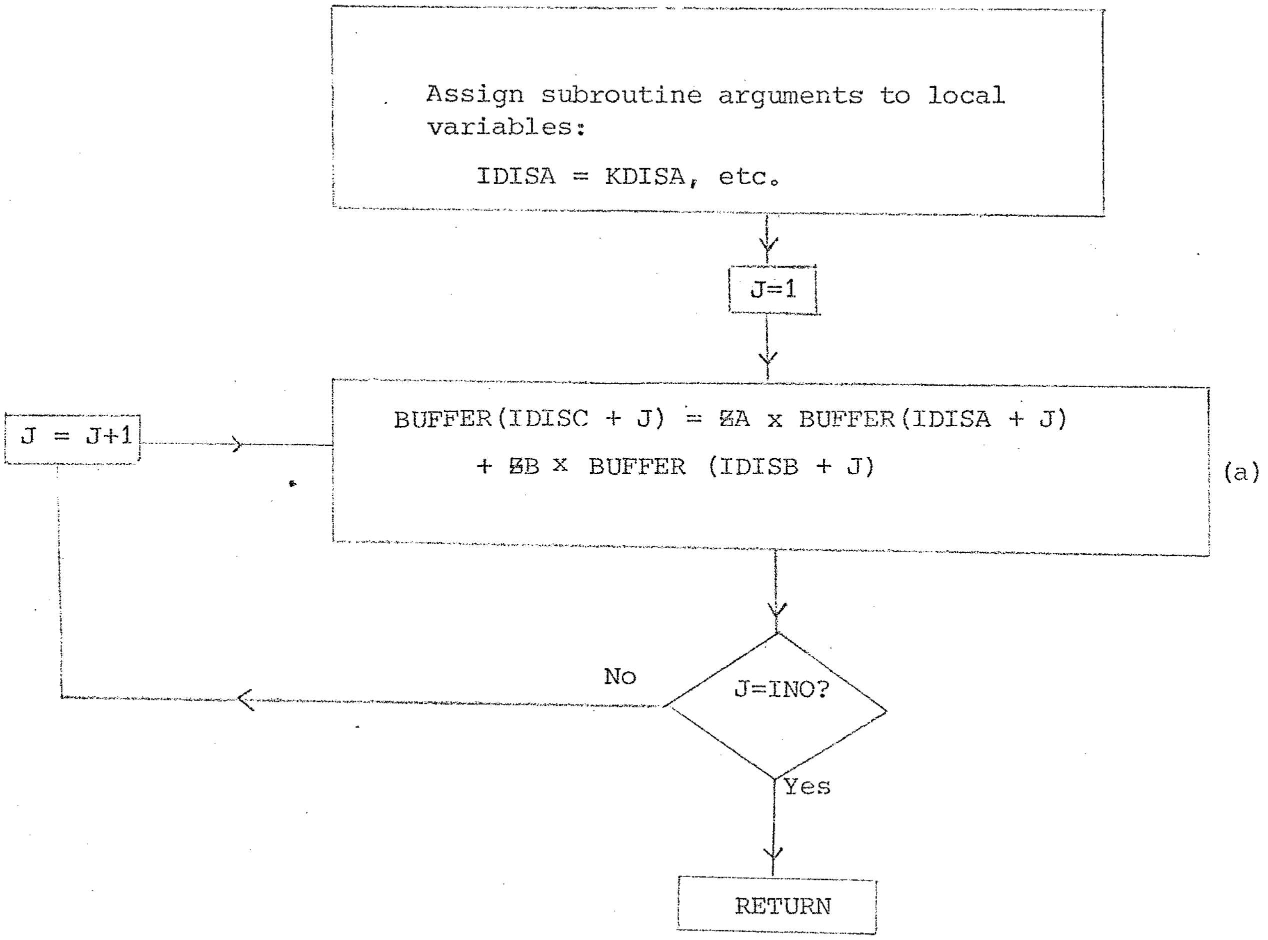
KDISB: displacement in blank common of first element of B

KDISC: displacement in blank common of first element of C

PA : constant multiplying array A

PB : " " " B

KNO : length of arrays



(a) BUFFER(1) is the first element of blank common

Diagram 21 <2.19> SUBROUTINE INIFLT (PCOSØ, KX, KY, KMINP, KMINV)

initialisation for space filter

where PCOSØ = $\cos(\theta_o)$, where the space filter is to be applied polewards of latitude θ_o

KX = number of longitude points

KY = number of latitude rows

KMINP(KY) = k_{min}, lowest filtered wavenumber, for temperature rows

KMINV(KY) = k_{min}, for v-velocity rows

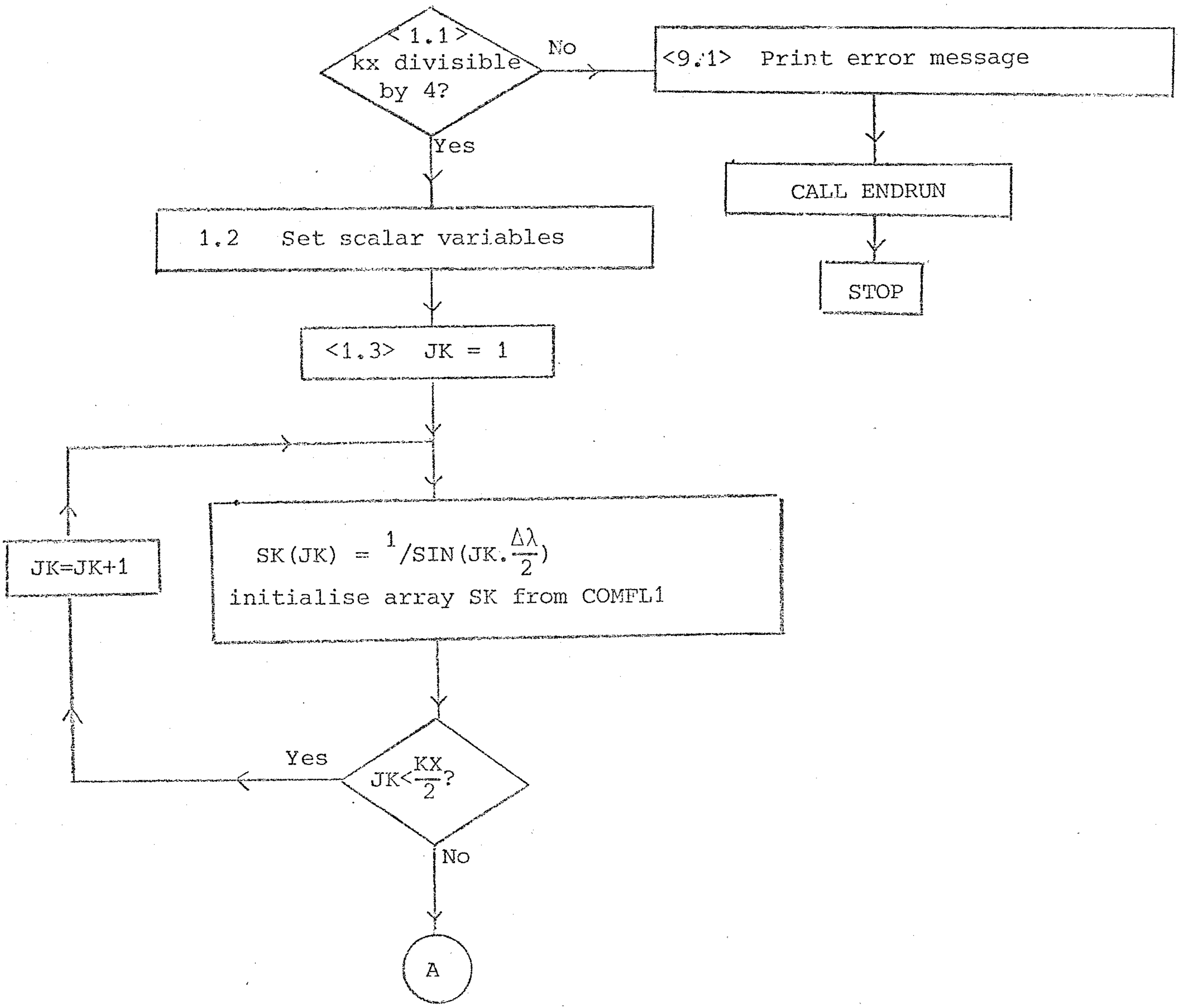


Diagram 21 (contd. 1)

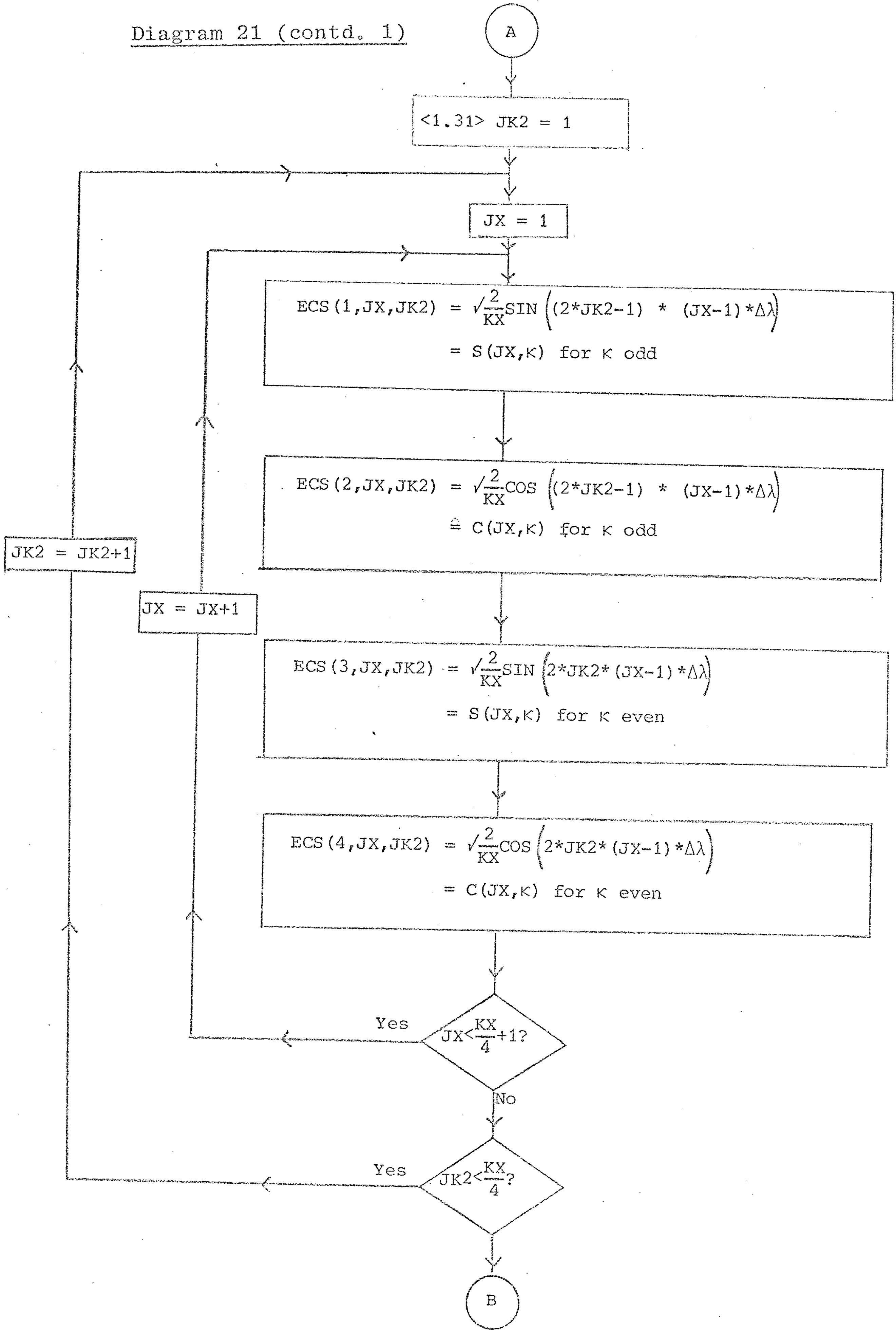


Diagram 21 (contd. 2)

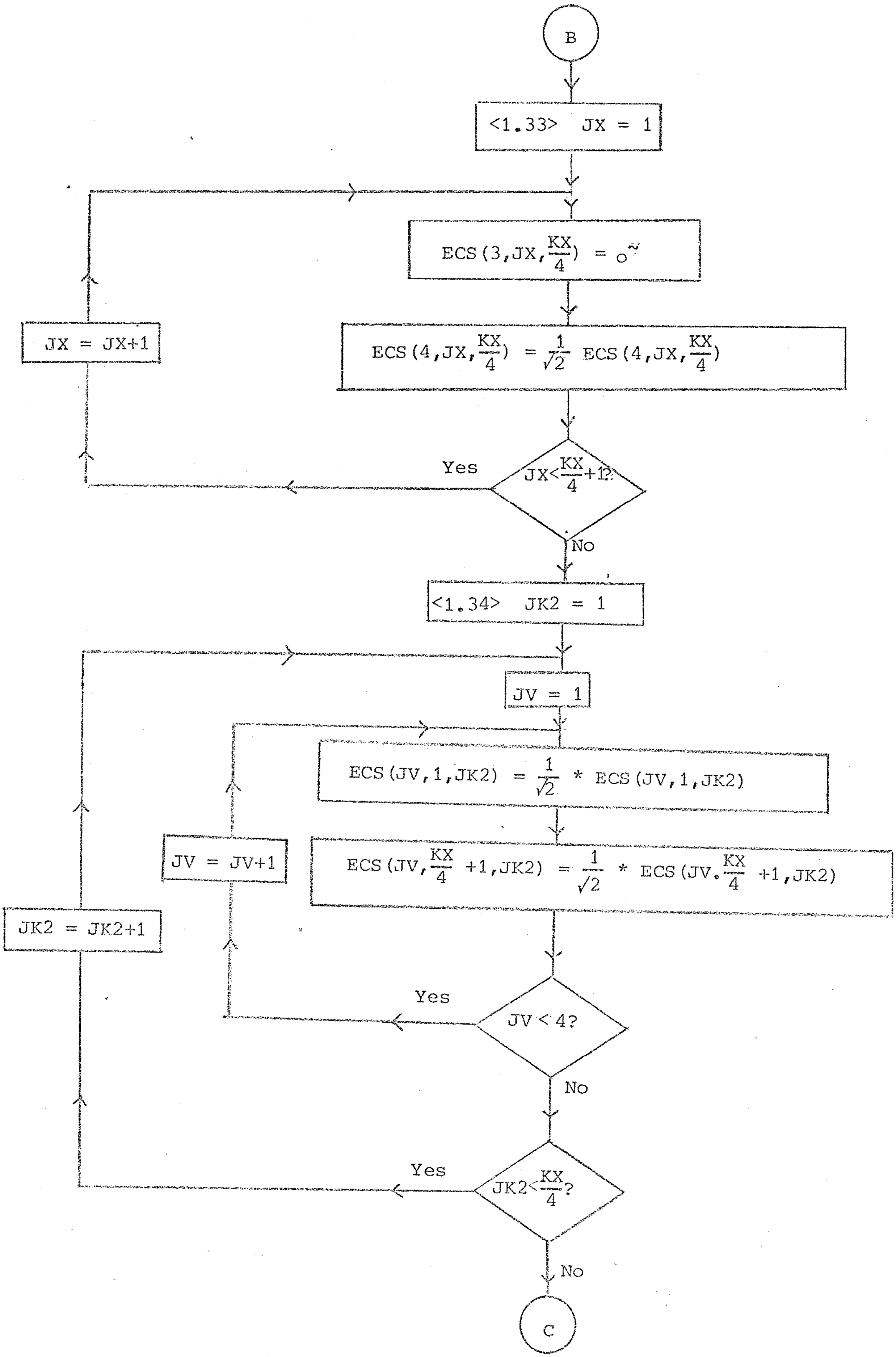


Diagram 21 (contd. 3)

C

Find smallest wavenumber to be filtered at each latitude. Do not filter wavenumbers larger than $\frac{KX}{2}$

<1.4> KMINP(1) = KMINV(1) = $\frac{KX}{2}$ +1

do not filter at north pole

JY = 2

KMINP(JY) = KMINV(JY) = $\frac{KX}{2}$ + 1

default: no filter at row JY

E

$ZRP = SIN((JY-1) \times \Delta\theta) \cdot \frac{1}{PCOS0}$

$= \frac{COS\ (\text{latitude of temperature points})}{COS(\theta_o)}$

$ZRV = SIN((JY-\frac{3}{2}) \cdot \Delta\theta) \cdot \frac{1}{PCOS0}$

$= \frac{COS\ (\text{latitude of V points})}{COS(\theta_o)}$

D

Diagram 21 (contd. 4)

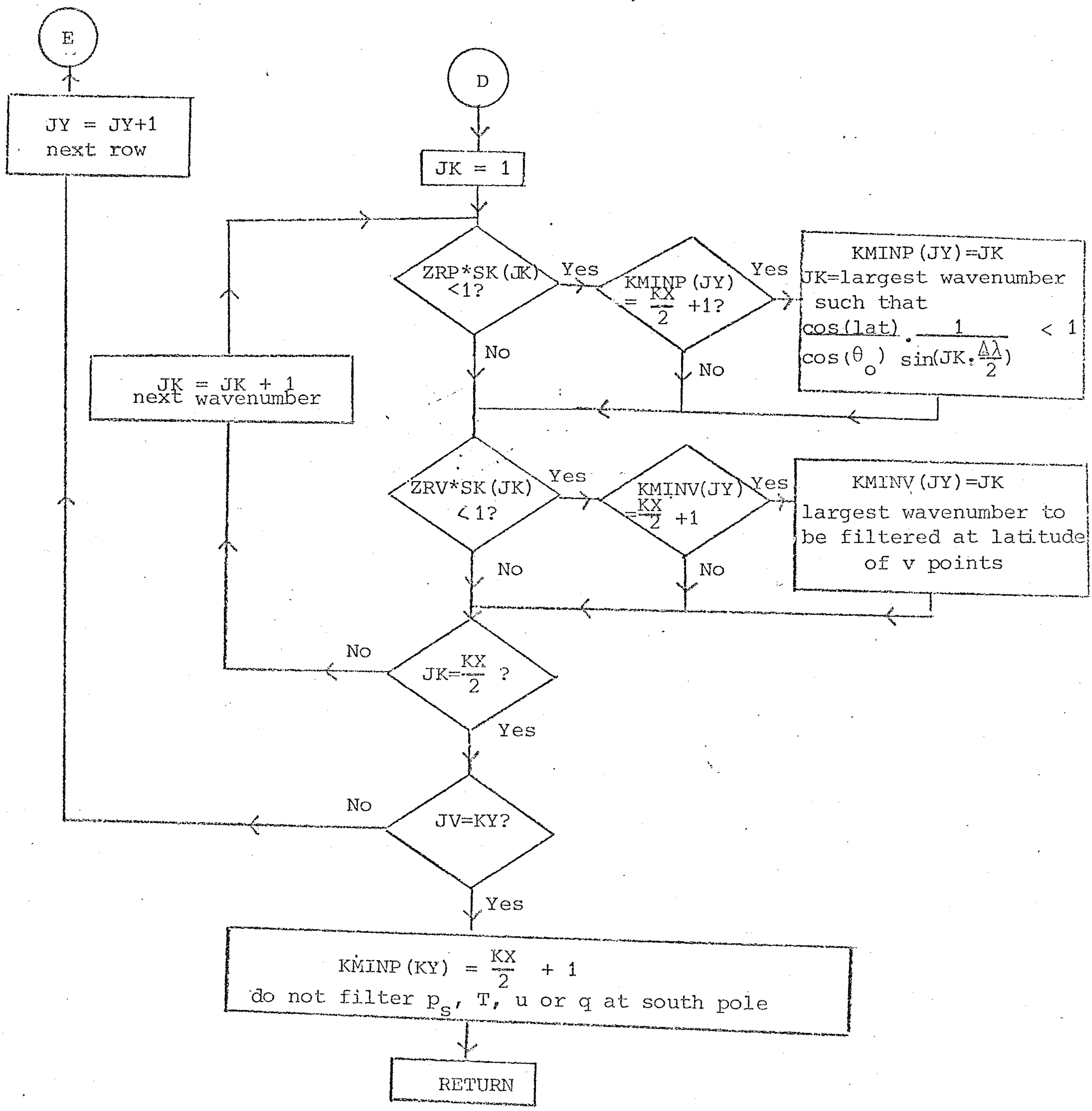


Diagram 22 <2.18> SUBROUTINE FILTER (PFACT,KMIN,KBASE,KFLDS)

spatial filter

where PFACT = cos (latitude)/cos(θ_o) where the space filter is to be applied polewards of latitude θ_o.

KMIN = largest wavenumber to be filtered at this latitude.

KBASE = displacement in blank common of data to be filtered.

KFLDS = number of fields to be filtered.

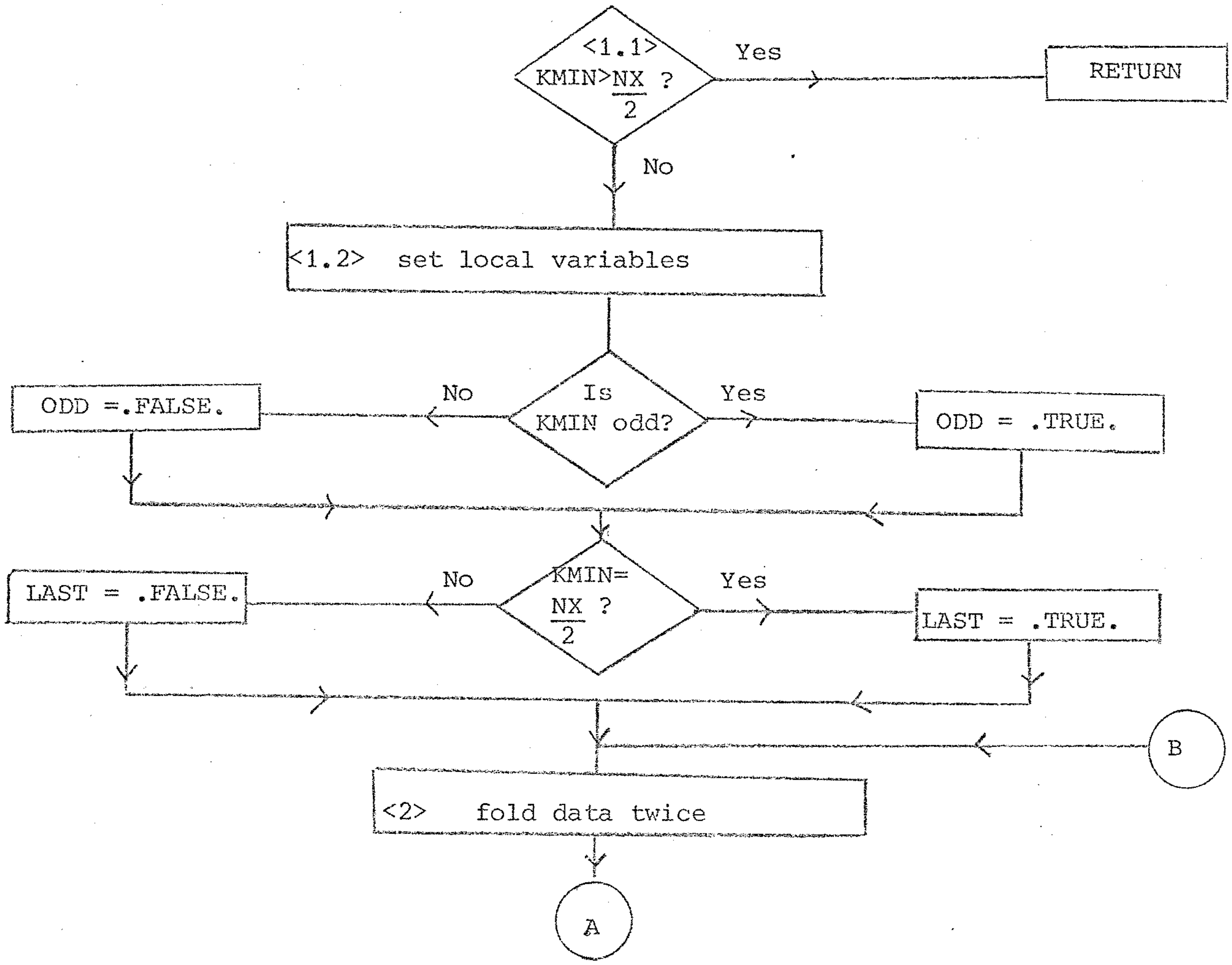


Diagram 22 (contd. 1)

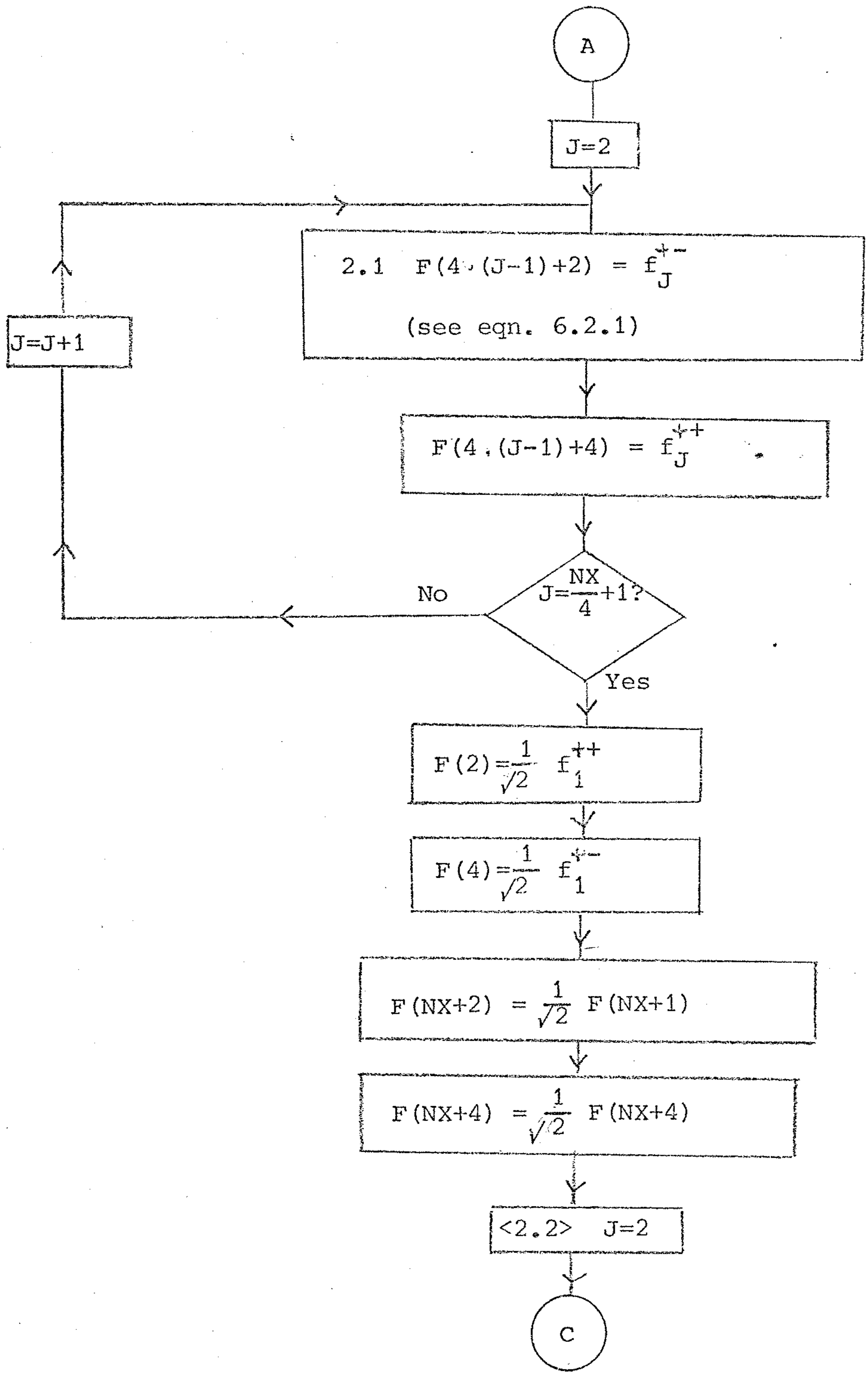


Diagram 22 (contd. 2)

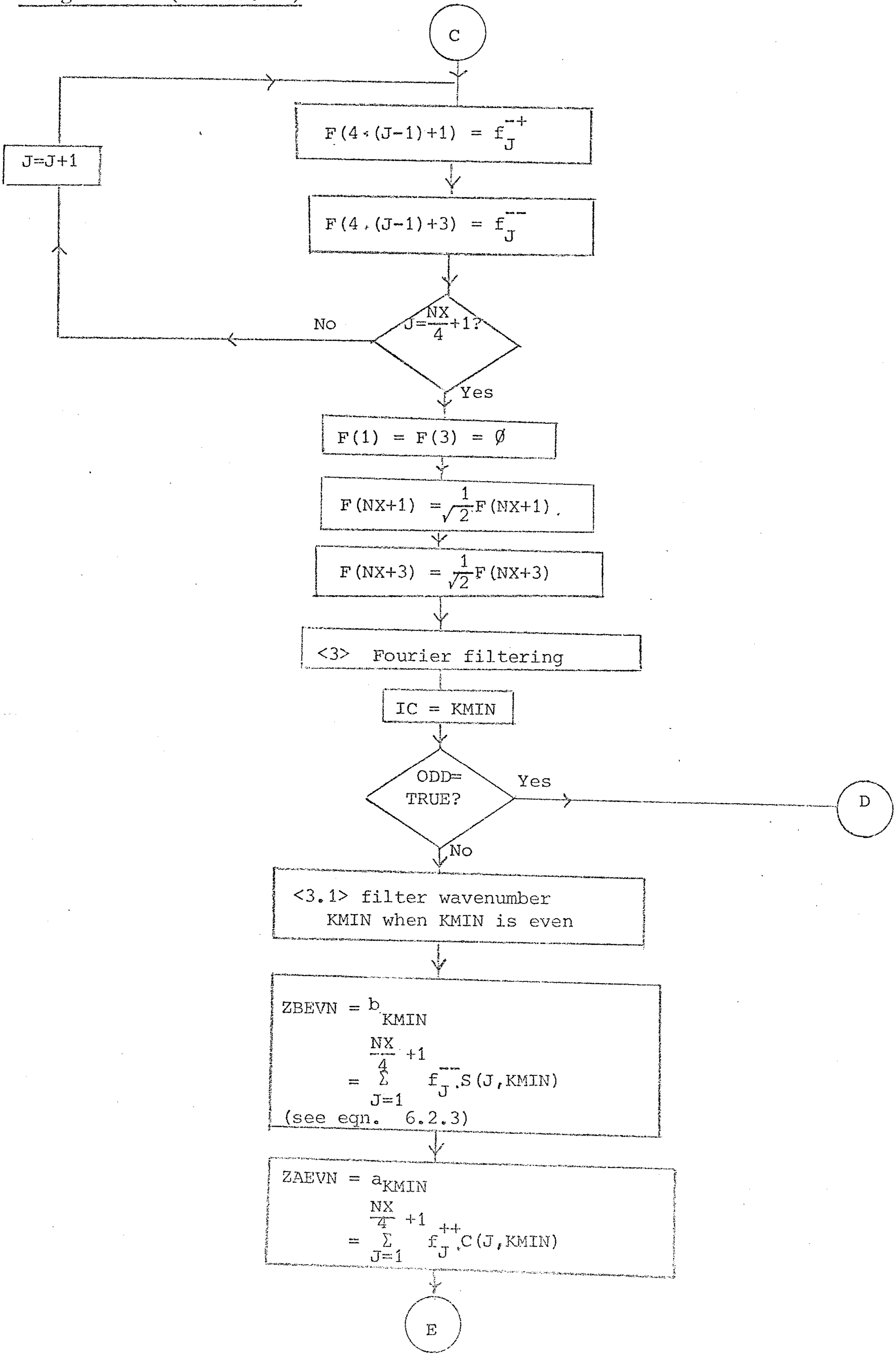


Diagram 22 (contd. 3)

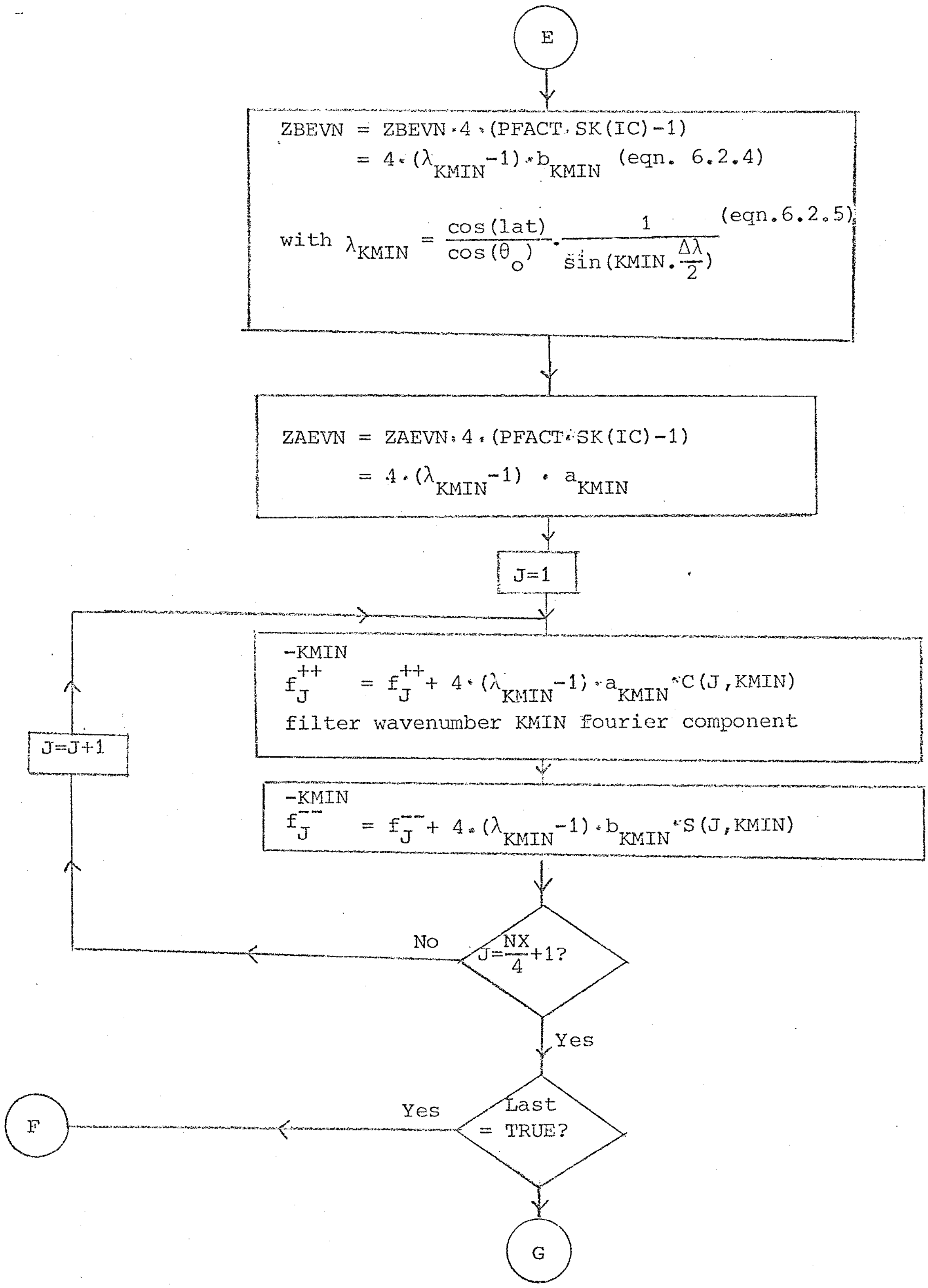


Diagram 22 (contd. 4)

G

IC=IC+1

D

<3.2> filter wavenumbers IC(odd) and IC + 1 (even)

ZBODD = b_{IC}

$$= \sum_{J=1}^{\frac{NX}{4}+1} f_J^{-+} \cdot S(J,IC)$$

(see eqn. 6.2.3)

ZAODD = a_{IC}

$$= \sum_{J=1}^{\frac{NX}{4}+1} f_J^{+-} \cdot C(J,IC)$$

ZBEVN = b_{IC+1}

$$\sum_{J=1}^{\frac{NX}{4}+1} f_J^{--} \cdot S(J,IC+1)$$

ZAEVN = a_{IC+1}

$$\sum_{J=1}^{\frac{NX}{4}} f_J^{++} \cdot C(J,IC+1)$$

ZBODD = ZBODD · 4 · (PFACT · SK(IC) − 1)

$= 4 \cdot (\lambda_{IC} - 1) \cdot b_{IC}$

(see eqns. 6.2.4 and 6.2.5)

ZAODD = ZAODD · 4 · (PFACTXSK(IC) − 1)

$= 4 \cdot (\lambda_{IC} - 1) \cdot a_{IC}$

ZBEVN = ZBEVN · 4 · (PFACT · SK(IC+1) − 1)

$= 4 \cdot (\lambda_{IC+1} - 1) \cdot a_{IC+1}$

ZAEVN = ZAEVN · 4 · (PFACT · SK(IC+1) − 1)

$= 4 \cdot (\lambda_{IC+1} - 1) \cdot a_{IC+1}$

H

Diagram 22 (contd. 5)

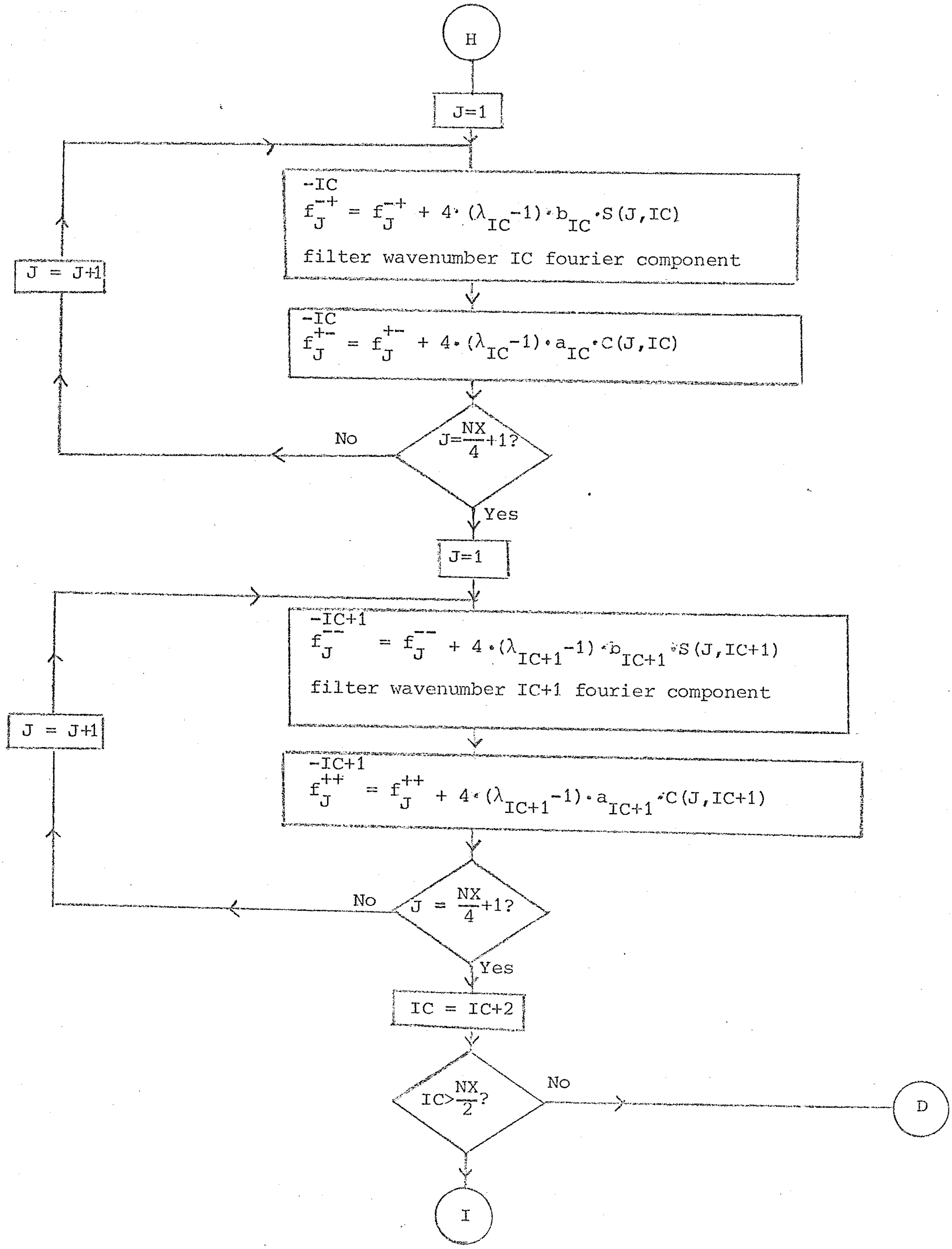


Diagram 22 (contd. 6)

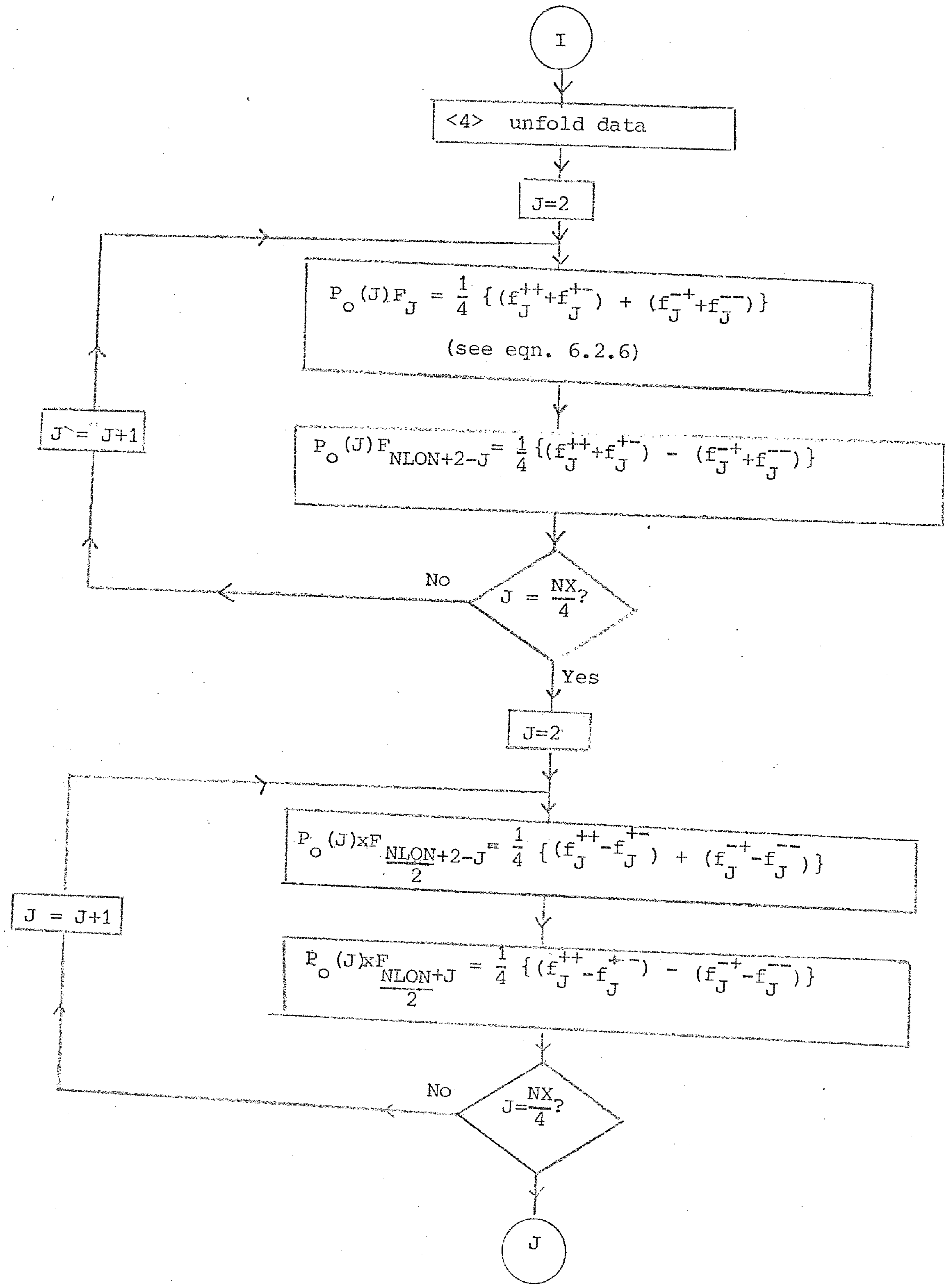


Diagram 22 (contd. 7)

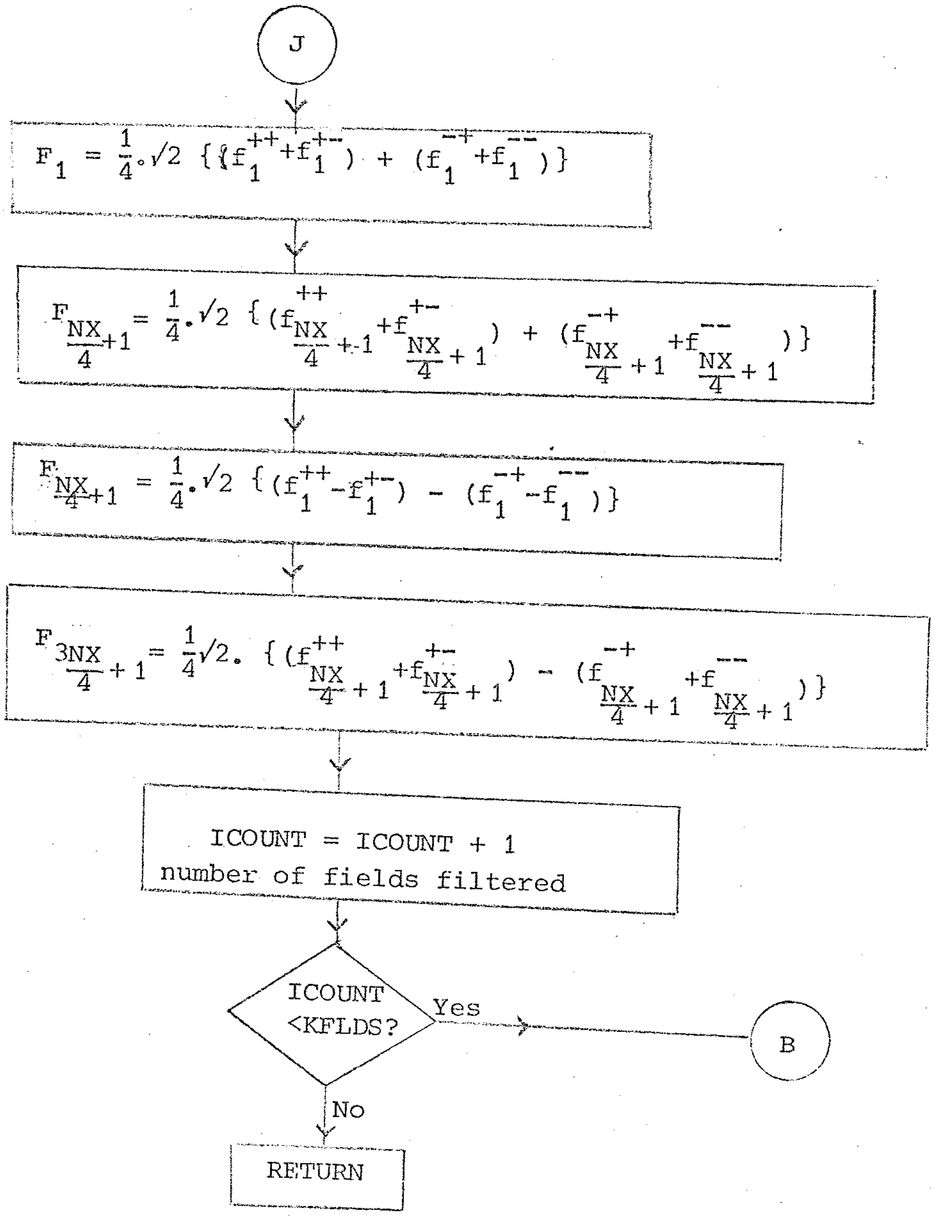


Diagram 23 SUBROUTINE DYN - detailed flow diagram

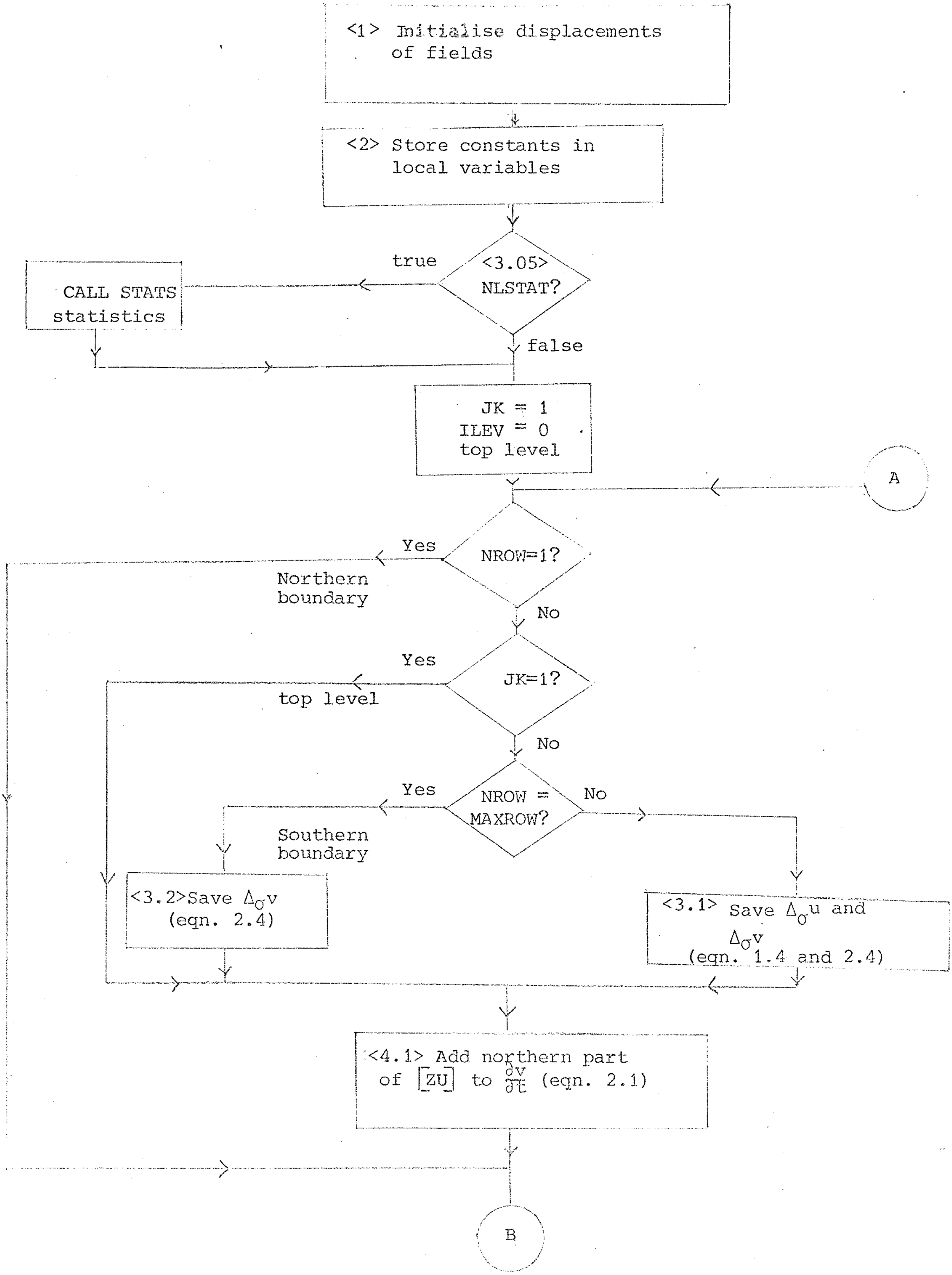


Diagram 23 (contd. 1)

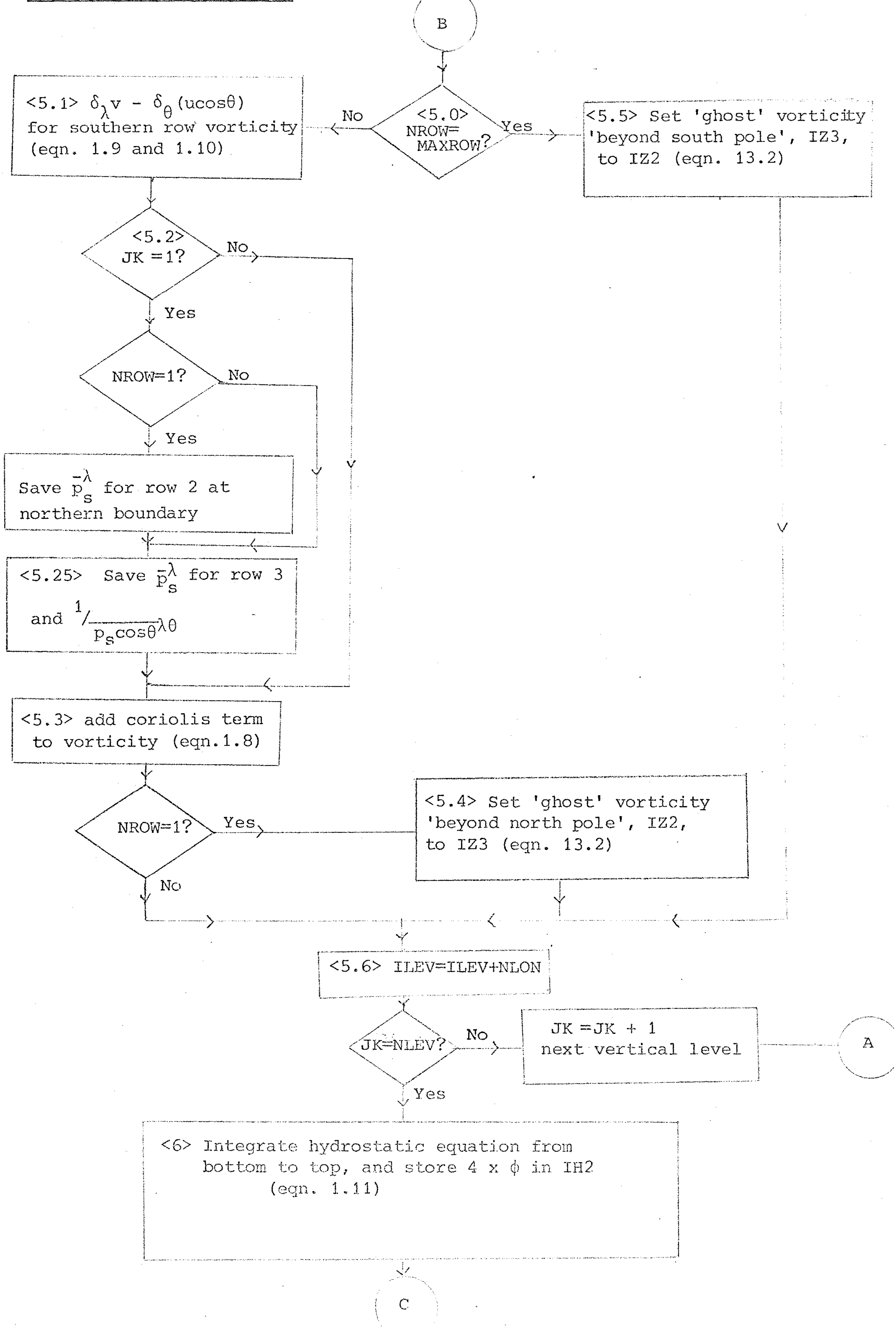


Diagram 23 (contd. 2)

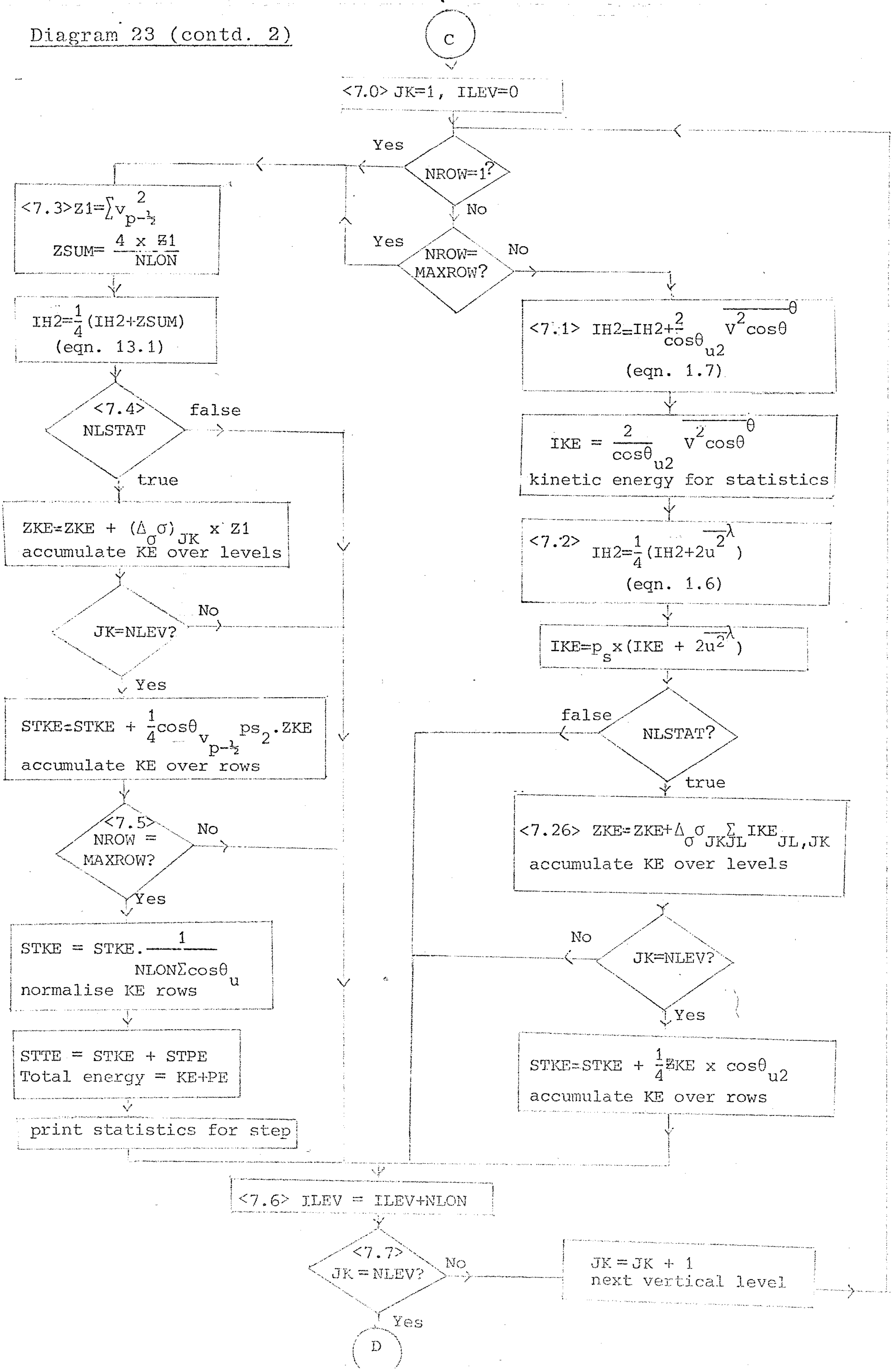


Diagram 23 (contd. 3)

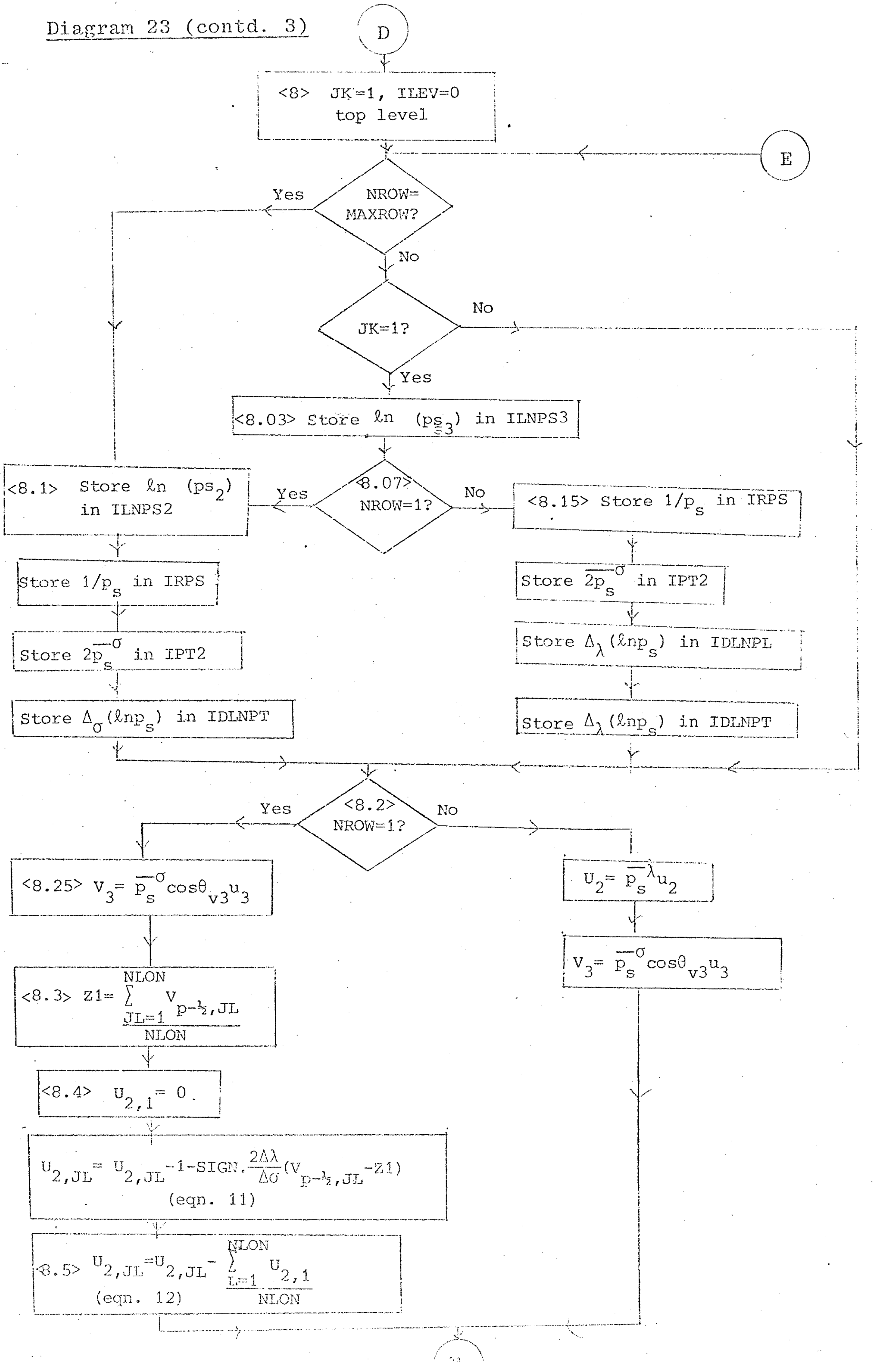


Diagram 23 (contd. 4)

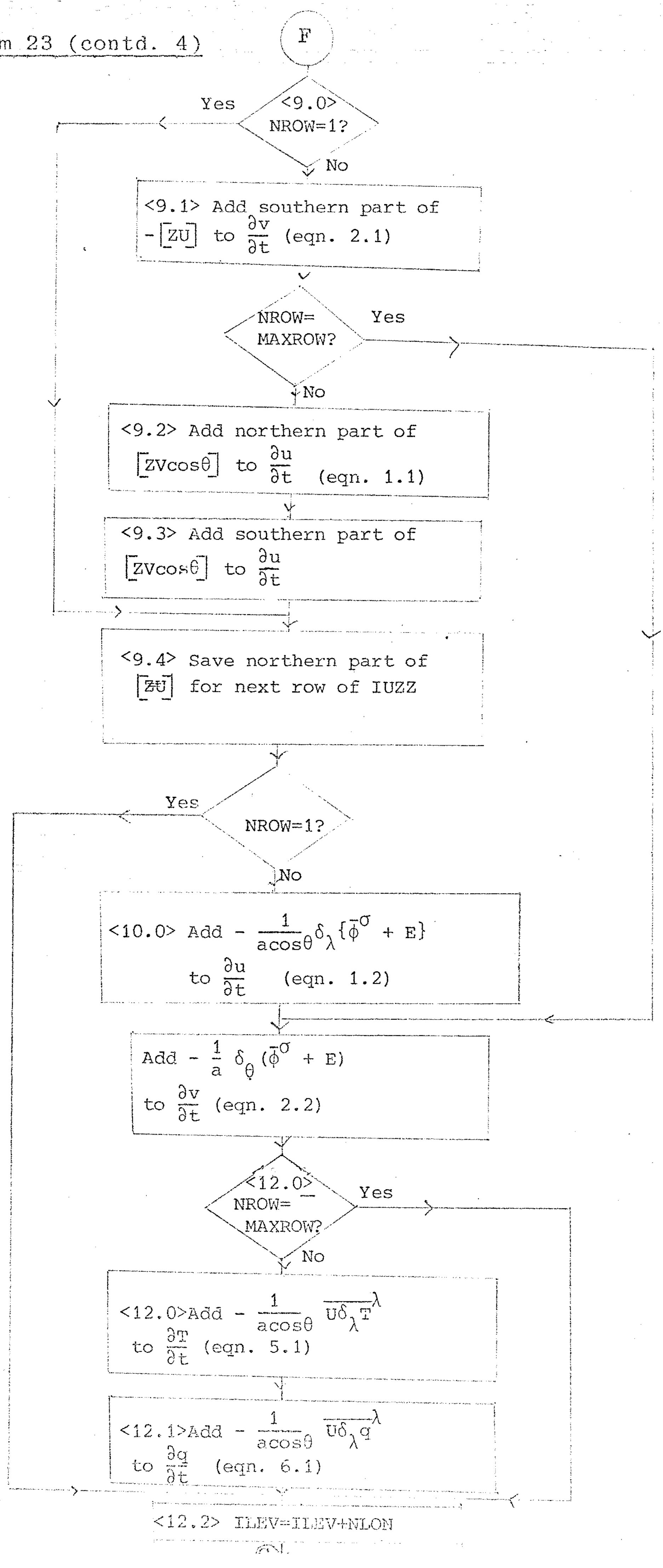


Diagram 23 (contd. 5)

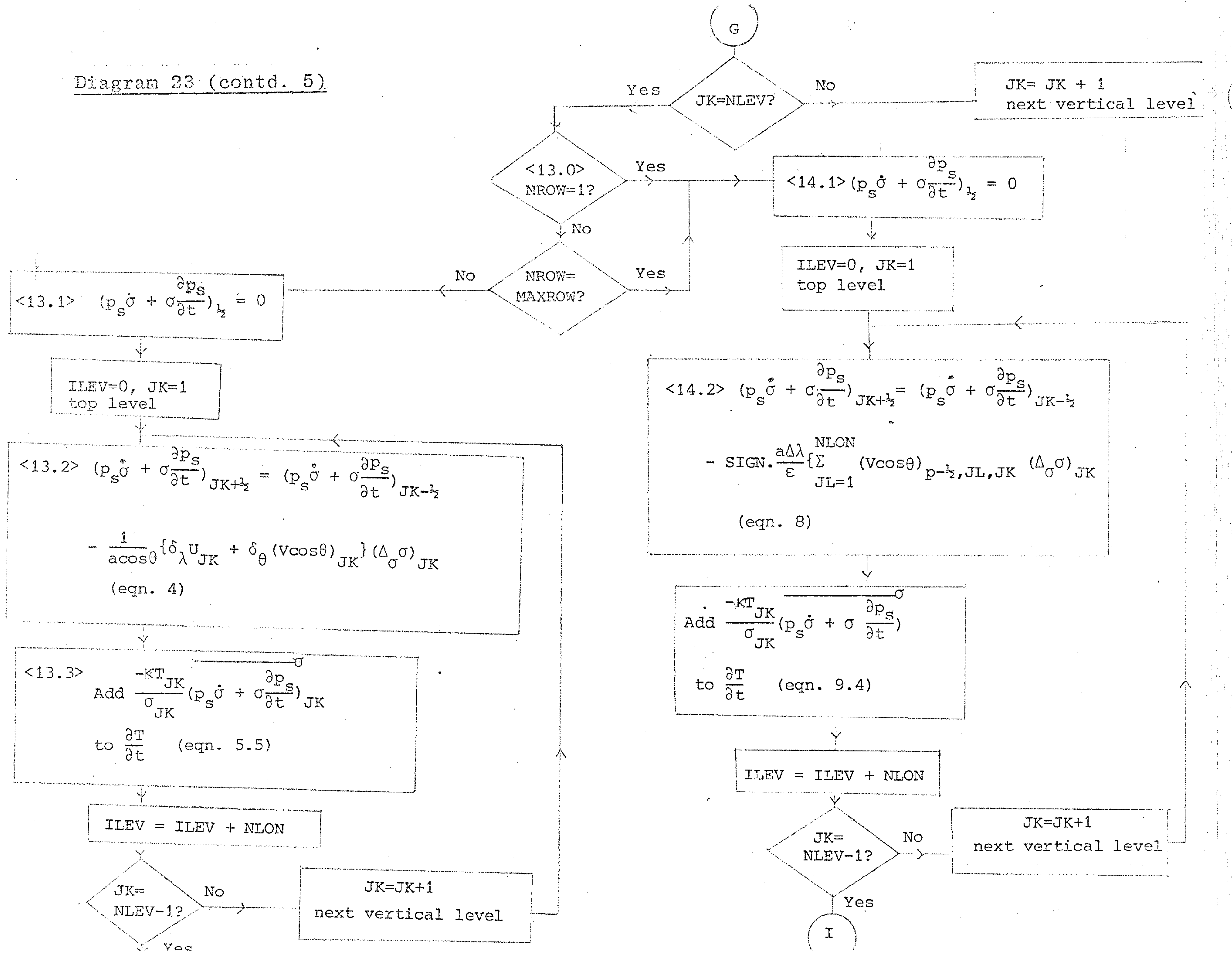


Diagram 23 (contd. 6)

H

<13.5> $$\frac{\partial p_s}{\partial t} = (p_s\dot{\sigma} + \sigma\frac{\partial p_s}{\partial t})_{NLEV-\frac{1}{2}} - \frac{1}{a\cos\theta}\{\delta_\lambda U + \delta_\theta V\cos\theta\}_{NLEV}(\Delta_\sigma\sigma)_{NLEV}$$ (eqn. 3)

<13.6> Add $\frac{-\kappa T_{NLEV}}{\sigma_{NLEV}}\{\alpha\frac{\partial p_s}{\partial t} + \beta(p_s\dot{\sigma} + \sigma\frac{\partial p_s}{\partial t})_{NLEV-\frac{1}{2}}\}$ to $\frac{\partial T}{\partial t}$ (eqn. 5.5)

<13.7> ILEV=NLON, JK=2
second to top level

$$p_s\dot{\sigma}_{JK+\frac{1}{2}} = (p_s\dot{\sigma} + \sigma\frac{\partial p_s}{\partial t})_{JK+\frac{1}{2}} - \sigma_{JK+\frac{1}{2}}\frac{\partial p_s}{\partial t}$$

ILEV=ILEV+NLON

JK=NLEV?

Yes → J

No → JK = JK + 1
next vertical level

I

<14.4> $$\frac{\partial p_s}{\partial t} = (p_s\dot{\sigma} + \sigma\frac{\partial p_s}{\partial t})_{NLEV-\frac{1}{2}} - SIGN.\frac{a\Delta\lambda}{\varepsilon}\{\sum_{JL=1}^{NLON}(V\cos\theta)_{p-\frac{1}{2},JL,NLEV}\}(\Delta_\sigma\sigma)_{NLEV}$$ (eqn. 7)

Add $\frac{-\kappa T_{NLEV}}{\sigma_{NLEV}}\{\alpha\frac{\partial p_s}{\partial t} + \beta(p_s\dot{\sigma} + \sigma\frac{\partial p_s}{\partial t})_{NLEV}\}$ to $\frac{\partial T}{\partial t}$ (eqn. 9.4)

<14.6> ILEV=NLON, JK=2
second to top level

$$p_s\dot{\sigma}_{JK+\frac{1}{2}} = (P_s\dot{\sigma} + \sigma\frac{\partial p_s}{\partial t})_{JK+\frac{1}{2}} - \sigma_{JK+\frac{1}{2}}\frac{\partial p_s}{\partial t}$$

ILEV=ILEV+NLON

JK=NLEV?

Yes → J

No → JK = JK + 1
next vertical level

Diagram 23 (contd. 7)

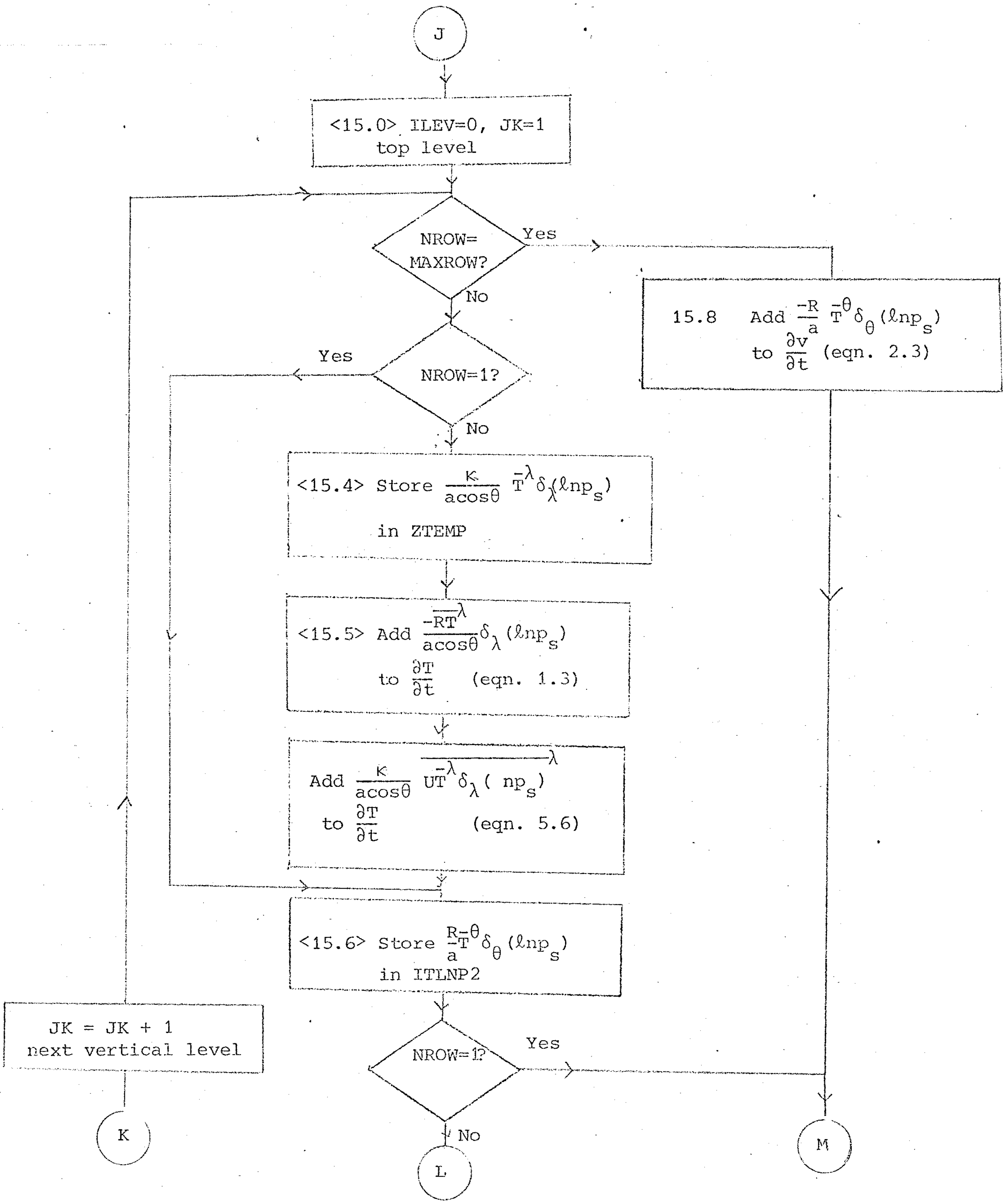


Diagram 23 (contd.8)

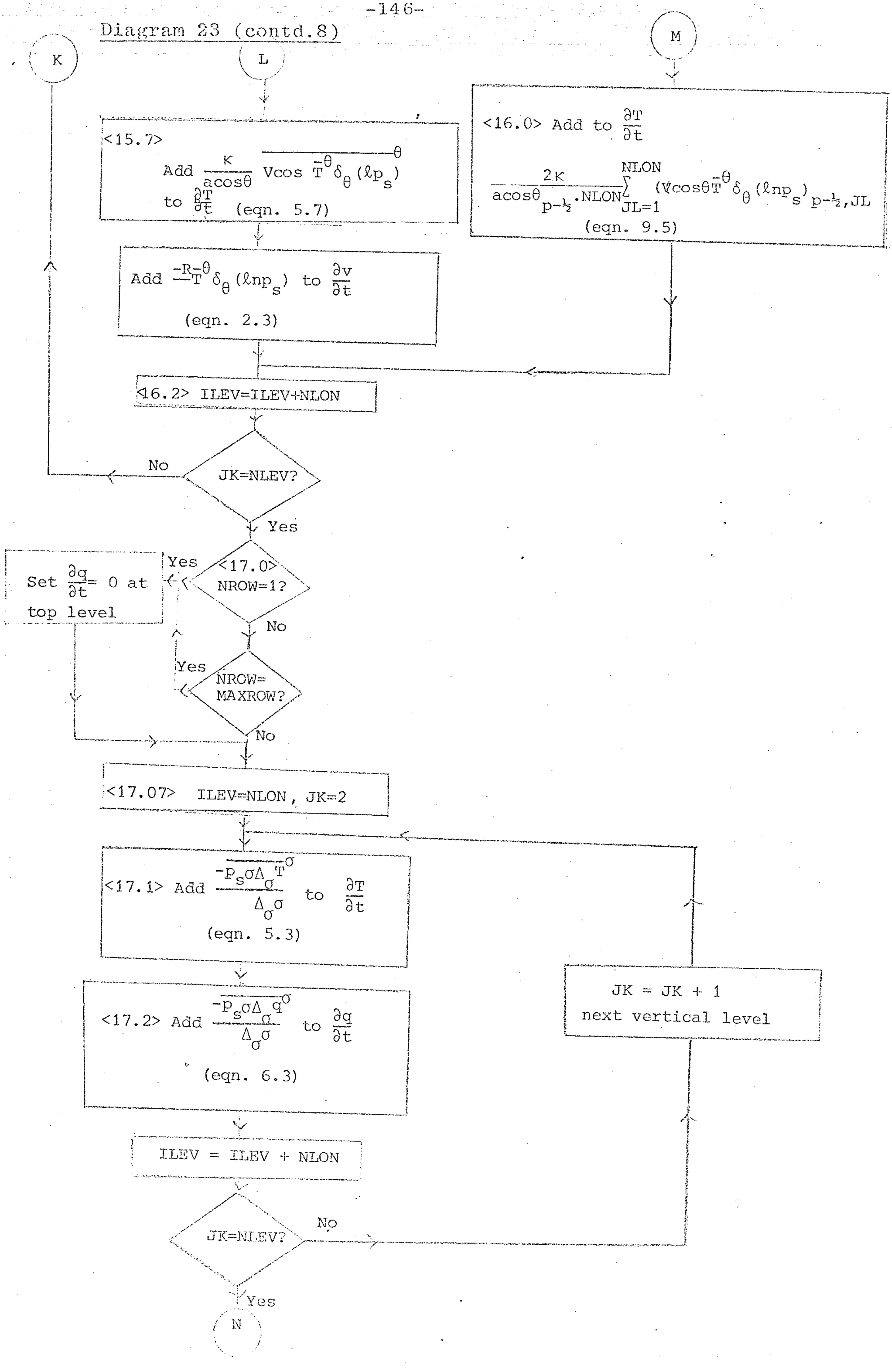


Diagram 23 (contd.9)

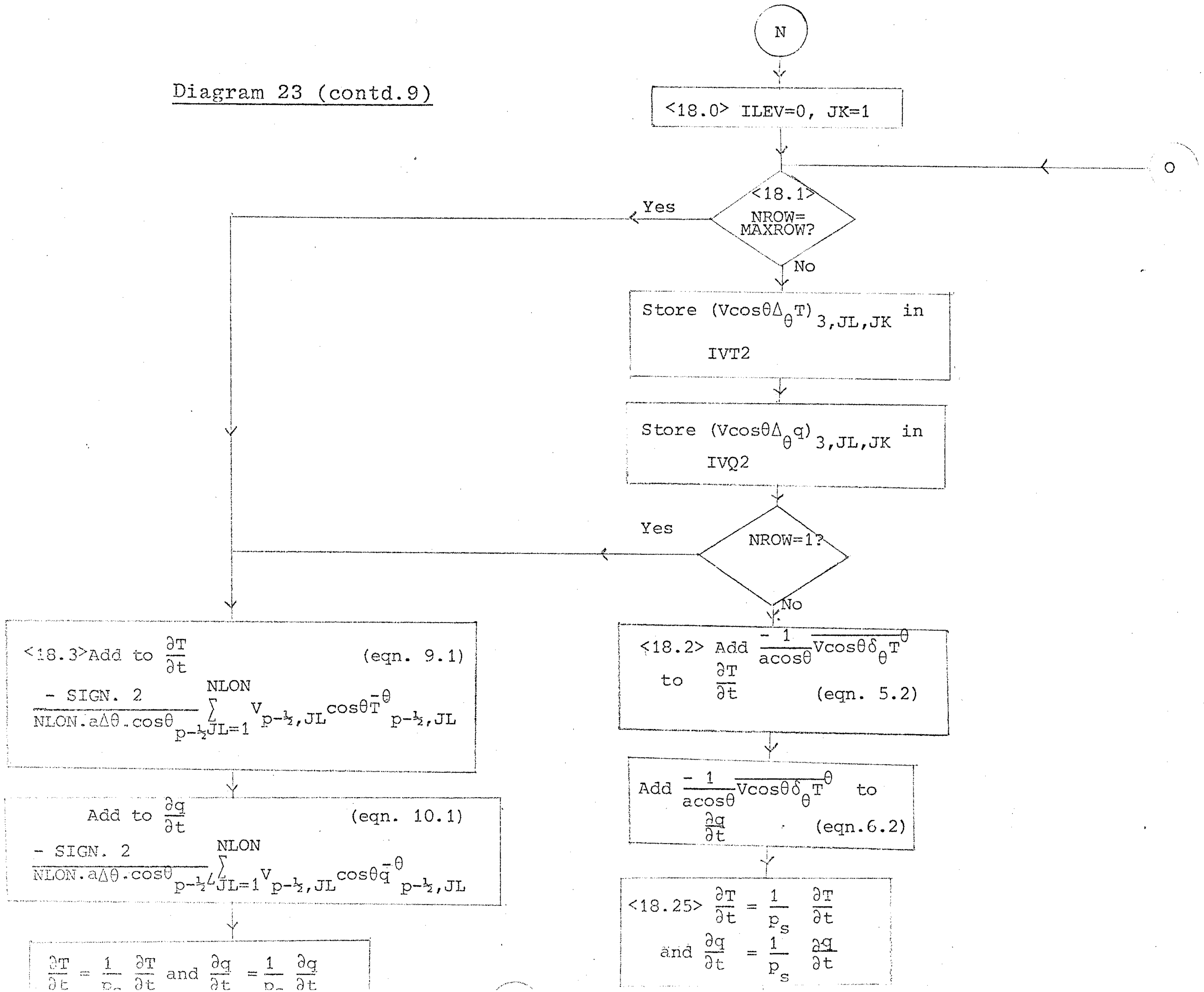
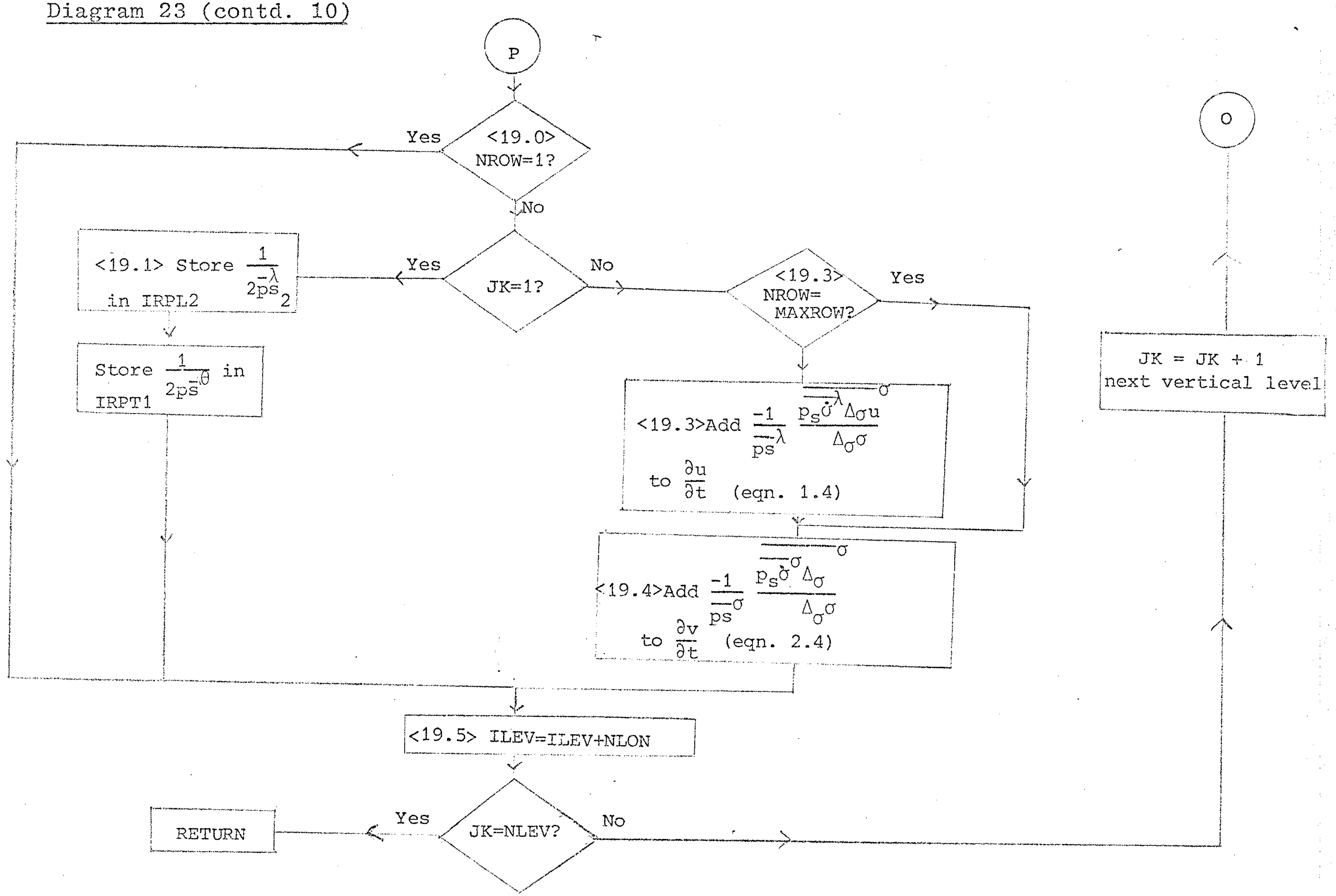


Diagram 23 (contd. 10)



EUROPEAN CENTRE FOR
MEDIUM RANGE WEATHER
FORECASTS

Research Department (RD)

Internal Report No. 9

No. 1 Users' Guide for the G.F.D.L. Model
(November 1976)

No. 2 The Effect of Replacing Southern Hemispheric Analyses by Climatology on Medium Range Weather Forecasts
(January 1977)

No. 3 Test of a Lateral Boundary Relaxation Scheme in a Barotropic Model
(February 1977)

No. 4 Parameterisation of the surface fluxes
(February 1977)

No. 5 An Improved Algorithm for the Direct Solution of Poisson's Equation over Irregular Regions
(February 1977)

No. 6 Comparative Extended Range Numerical Integrations with the E.C.M.W.F. Global Forecasting Model 1: The N24, Non-Adiabatic Experiment
(March 1977)

No. 7 The E.C.M.W.F. Limited Area Model
(March 1977)

No. 8 A Comprehensive Radiation Scheme Designed for Fast Computation
(May 1977)

No. 9 Documentation for the E.C.M.W.F. Grid Point Model
(May 1977)