European Centre
for Medium Range
Weather Forecasts

The E.C.M.W.F. Limited Area Model

Internal Report 7
Research Dept.

March 77

Centre Européen pour les Prévisions Météorologiques
a Moyen Terme

Europäisches Zentrum Für Mittelfristige Wettervorhersagen

THE E.C.M.W.F. LIMITED AREA MODEL

by

J.K. Gibson and P. Kållberg

European Centre for Medium Range Weather Forecasts, Bracknell

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.

1. Introduction

The ECMWF Limited Area Model uses the same basic formulation as the first ECMWF Global Forecasting Model. Indeed, most of the routines are identical, and tests have been made to ensure that, under carefully specified conditions, identical results can be obtained from both models. This document contains details of the boundary scheme used, and illustrates the changes made to those routines of the Global model which required alteration as a result of constraints imposed by a limited area.

Although every effort was made to provide a variable limited area, two possibilities were deliberately omitted. They are :

a) The possibility of including the North or South Poles within the limited area.

b) The possibility of including continuity in the East-West direction by using cyclic boundary condition techniques.

Both of these eventualities could be catered for by means of a much less drastic alteration to the Global Model, while to include them within the current Limited Area scheme would have reduced its efficiency.

For full details of the forecast formulation and implementation the reader is invited to consult the papers describing the Global Model (BURRIDGE and HASELER, 1977). A description of the lateral boundary scheme and its implementation, and details of the subroutines used follows.

2. The Boundary Scheme

The interior variables are smoothly adjusted towards their prescribed boundary values using a relaxation technique suggested by Davies (1976) and modified by Kållberg (1977). Within the boundary zone

$$s^{\tau+1} = (1 - \alpha)\,(s^{\tau-1} + 2\,\Delta t.\,Ds^{\tau}) + \alpha.\tilde{s}^{\tau+1} \qquad (1)$$

for any predicted parameter s, where

Ds^{τ} is the interior tendency

$\tilde{s}^{\tau+1}$ is the boundary value, and

$$\alpha = 1 - \tanh\,(a.j) \qquad (2)$$

where a is a constant and j is proportional to the distance to the nearest boundary.

The resulting scheme is dependent on two factors - the value of the constant, a, and the width of the boundary zone over which it is applied. The currently coded boundary routines use a value of a such that

$$a = \frac{2}{N - 4} \tag{3}$$

where N is the width of the boundary zone in grid points, and is supplied to the forecast programme as data. In most of the experiments so far completed, N has been given a value of 8, giving a = ½ with a resulting boundary width of 8 grid points. Table 1 gives α values corresponding to j for these values.

j	$\alpha = 1 - \tanh (j/2)$
0	1 . 0
1	0 . 538
2	0 . 238
3	0 . 095
4	0 . 036
5	0 . 013
6	0 . 005
7	0 . 002

Table 1 - relaxation factors

3. Implementation of the Boundary Scheme

3.1 General Considerations

The main disadvantage with the boundary scheme outlined above is the large amount of boundary data required. For ease of coding, and to preserve the ability to vary the size of the boundary zone without having to re-create boundary data, it was decided that data for the whole limited area would be used. Boundary data sets are thus prepared and interpolated in exactly the same way as the initial data. By reading the boundary data line by line using asynchronous input, the overheads resulting from this are reduced to a minimum.

Equation (1) can best be applied immediately after the leap-frog calculation at 6.3 in SUBROUTINE LINEMS (BURRIDGE, D. and HASELER, J.M., 1977). At this point

$S^{\tau-1} + 2\ \Delta t.\ Ds^{\tau}$ has been stored, and is available to the boundary updating routines.

3.2 The Boundary Routines

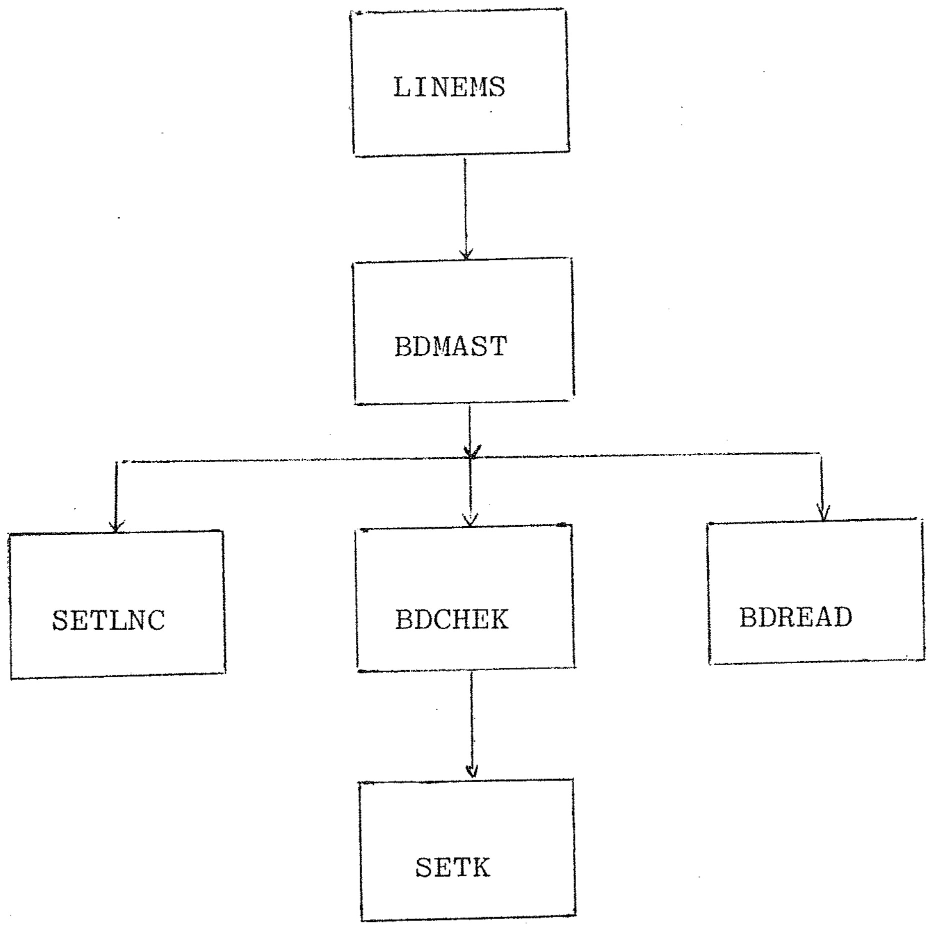

Figure 1. Block diagram of boundary scheme subroutines.

The boundary master routine, BDMAST, is called at section 6.4 in LINEMS, and calls BDCHEK. On a normal call, BDCHEK simply checks reads initiated by BDREAD and returns control. On the first call to BDCHEK the appropriate boundary data sets are located. The first records of those data sets are read and checked. SETK is called to set an array containing relaxation coefficients, and control is returned to BDMAST.

BDMAST next calls SETLNC to extract appropriate relaxation coefficients for the current row from the array set by SETK. These coefficients are applied, taking into account the location of the boundaries with respect to variables staggered in different ways.

Finally, BDMAST calls BDREAD to initiate the input of boundary values for the next row, then returns control to LINEMS.

The boundary routines are written up fully in sections 6 onwards.

3.3 Interpolation

The methods used to interpolate both the initial data and the boundary data are likely to be crucial to this type of boundary scheme. Kållberg (1977) has shown that the continuity in the derivatives preserved in cubic spline interpolation results in much smoother divergence and vorticity fields. It is thought that bi-cubic spline interpolation with respect to space will be suitable for the preparation of initial and boundary data, while linear interpolation with respect to time is a sufficient approximation to obtain boundary values at a specific time from data at an earlier and a later time.

4. Running the Limited Area Model

4.1 Summary

Before the Limited Area Model can be run, it is necessary to set up the following data sets:-

a) Initial data set, consisting of 3 records containing initial values for the common areas COMBAS, COMHKP and COMMAP, followed by the initial data in line form.

b) Boundary data sets, each containing data for common area COMBDY as the first record, followed by the boundary data in line form.

c) Start/restart data set, in the same form as for the Global Model.

Programmes are available for creating a data set containing COMMAP, for bi-cubic spline interpolation of data, and for producing each of the types of data set outlined above.

4.2 The Programme

The programme is contained on deck TRY of the library, and calls the subroutine MASTER which controls the model. A typical set of cards to produce a run to 500 time steps is given as an example.

```
EW--- (T60000)
MOUNT (SN = DSETnn, VSN = ECMWnn)
ATTACH (OLDPL, LAMODELSRCE, ID = EWRG3, MR = 1)
UPDATE (Q)
RETURN (OLDPL)
FTN (I, OPT = 2)
RETURN (COMPILE)
MAP (PART)
REQUEST (TAPE 10, *VSN= -----)code so as to have tape 10 and
REQUEST (TAPE 11, *VSN= -----) tape 11 on different channels
                                if possible.
```

```
ATTACH (TAPE 30, start data set)
ATTACH (LLIB, LAMODELOBJ, ID = EWRG3, MR = 1)
ATTACH (BLIB, BUFFIOLIB, ID = CDFS, MR = 1)
LIBRARY (LLIB, BLIB)
LGO
EXIT (U)
REWIND (ZZZZMP)
COPYBF (ZZZZMP, OUTPUT)
AUDIT (ID = -----, SN = DSETnn)
7
 8
  9
*C TRY
7
 8
  9
£ REST  namelist input  £
followed by 3 cards of title information, then
£ NEWRUN namelist input £ NSTOP = 5ØØ,  £
6
 7
  8
   9.
```

5. Common Blocks Used by the Boundary Scheme

Three extra COMMON blocks are introduced to assist the boundary scheme - COMBDY containing constants used within the scheme, COMBDB containing the boundary buffers, and BDYFLN containing file name information for the input of boundary data.

5.1 COMBDY

Common COMBDY contains the following variables:-

NBDVRB	Number of variables in the boundary data i.e. number of parameters per line, where the same basic variable stored at N levels counts as N parameters.
NBDPVR	Number of preliminary variables stored at beginning of the line data (e.g. invariant items such as topography).
NBDBFL	Buffer length required to accommodate one data record from one boundary data set.
NBDPTS	Number of data points within the boundary zone.
NBDNUM	Number of this boundary data set.
NBDMAX	Maximum number of boundary sets required.
NBDLIN	Line counter for boundary data.

LSTAR2	Location of the start of the second boundary buffer (usually NBDBFL +1)
LEND2	Location of the end of the second boundary buffer.
NBDDS1	Pointer to the earlier of a pair of boundary data sets.
NBDDS2	Pointer to the later of a pair of boundary data sets.
NBDRDS	Number of times the current pair of boundary data sets have been read.
NBDRDR	Number of times the current pair of boundary data sets is required to be read (i.e. number of Limited Area Model time steps between the pair of boundary data sets).
NESKP1	First parameter with an Eastern boundary offset one grid point West.
NESKP2	Last parameter with an Eastern boundary offset one grid point West.
NNSKP1	First parameter with a Northern boundary offset one grid point South.
NNSKP2	Last parameter with a Northern boundary offset one grid point South.
CTWODT	Time step of boundary steps (i.e. time between boundary data sets)
SPARE	Not used.
AKD	Array size 100 used to contain the relaxation factors for the North West corner of the grid. (This array needs to be extended if NBDPTS > 10 to NBDPTS x NBDPTS).
RKDPS1	Array size 100 - not used.
TIMRAT	Time ratio - CTWODT/TWODT where TWODT = time step for the Limited Area Model.

5.2 COMBDB

Common COMBDB contains a single array BDYBUF which is used as the buffers for boundary input. It must be at least NLON x NBDVRB x 2 in length, i.e. NBDBFL x 2 words long.

5.3 BDYFLN

Common BDYFLN contains information required to attach boundary data sets.

NYDFLN	is 4 words long, and is used to contain a file name.
NCY	contains a cycle number, usually 0.
NPLIST	contains other details about the attributes and locations of boundary data sets, and is 4 words long.

Details of files are initiated within BDCHEK, and this routine should be altered to present the correct details for a particular user.

6. SUBROUTINE BDMAST - 2.21 CONTROL BOUNDARY SCHEME

6.1 Common Areas

This subroutine uses

COMBAS, COMIOC, COMHKP and BLANK COMMON (see Global Model documentation).

COMBDY and COMBDB (see section 5 above).

6.2 Local Variables Used

ZNXRAT	Proportion of difference between next boundary time and previous boundary time required to be added to previous boundary time to obtain the time of this time step.
ILNSBD	.TRUE. within a North or South boundary zone, otherwise .FALSE.
ILNBD	.TRUE. within the Northern boundary zone corresponding to variables whose Northern boundary is displaced one grid length Southwards.
ISTR1, ISTR2, ISTR3 ISTP1, ISTP2, ISTP3	DO LOOP control variables.
IISTP2, IISTP3, IISTR3	Modified DO LOOP variables.
ICON, ICON2, ICON3	Displacements.

6.3 Parameters

None.

6.4 Method

Section 1.1 calculates DO LOOP control limits, displacements within the main buffers, and displacements within the boundary buffers. BDCHEK is called to complete the reading of boundary data for the current row. Logical markers are set to assist in identification of the location of the boundary zone. SETLNC is called to set relaxation coefficients into the part of the boundary buffers containing topography, since topography is not required.

Section 1.2 adjusts the boundary values. Tests are made to find the type of boundary updating required. These fall into the following categories:-

a) Within the Northern boundary, normal variable - relaxation factors for the current line are applied to all points.

b) Within the Northern boundary, variable with boundary displaced Southwards - relaxation factors for the previous line are applied to all points, except for the Northern most line, where boundary values are substituted.

c) Within the central zone, normal variable - relaxation factors for the current line are applied within the East and West boundary zones only.

d) Within the central zone, variable with Eastern boundary displaced Westwards - relaxation factors for the current line are applied within the East and West boundaries, but displaced one point West on the Eastern boundary, while the Eastern most point is replaced by the boundary value.

e) Within the Southern boundary zone, all variables have relaxation factors for the current line applied to all points.

f) In cases a) and e) above where variables with Eastern boundaries displaced Westwards are encountered appropriate adjustments are made to the Eastern relaxation factors.

Section 1.3 initiates the reads for the next row of boundary data by calling BDREAD.

6.5 Results

Equation (1) is applied, with appropriate values of α, to all points within the boundary zone.

7. SUBROUTIN BDCHEK - 2.22 CHECK BOUNDARY INPUT

7.1 Common Areas

This subroutin uses COMBAS, COMIOC, COMHKP and BLANK COMMON (see Global Model documentation) COMBDB, BDYFLN, and a modified form of COMBDY (see section 5 above).

The area COMBDY is declared as a single array NCMBDY, and appropriate variables are equivalenced to this array.

7.2 Local Variables Used

IFLN	Contains uncompleted file name for boundary data sets.
IPLIST	Contains file details for boundary data sets.
ILFST	Set .TRUE. on first entry, becomes and remains .FALSE. after first entry.
IFAIL	Used to detect return codes.
IBDRDR	Used to calculate the number of times a boundary set is to be read.
IBDDS1, IBDDS2	Temporary device numbers.
IBDNUM	Boundary data number.

7.3 Parameters

None.

7.4 Method

First, a test is made to establish whether or not this is the first time the routine has been called. If it is the first time, section 2 is processed first; if not, section 1 only is processed.

Section 1 checks for completion of the reading of the current row from each of two boundary data sets. If reading is not complete, the programme waits until it is. When reading is complete, the quality of the read is assessed. If read errors have been noted, control passes to section 9; if not, control is returned to the calling routine.

Section 2 initiates boundary reading.

Section 2.05 inserts file name information into common area BDYFLN, and attaches the first boundary data set.

Section 2.1 reads information into COMBDY. First, COMBDY is read from the first boundary data set. This information is compared with other parameters to determine which boundary data is actually required, in case a restart has taken place, and the appropriate boundary data is attached. COMBDY is then re-initialised from this data set, and the next boundary data in time is attached. The common area COMBDY for this second boundary data set is skipped and initial settings of common constants are completed. If read errors occur, control is passed to section 9.

Section 2.2 initiates the reads for the first row of both boundary data sets. ILFST is switched to .FALSE., and subroutine SETK is called to set up the array of relaxation coefficients. Control is then passed back to section 1 to check the reads of the first rows.

Section 9 prints error messages and calls ENDRUN.

7.5 Results

Provided all reads have concluded satisfactorily, return is made to the calling programme with the completed rows of boundary data ready to use in the boundary buffers. If read errors are detected messages are printed and ENDRUN called to terminate the run.

8. SUBROUTINE BDREAD - 2.23 READ BOUNDARY DATA

8.1 Common Areas

This subroutine uses COMBAS, COMIOC, COMHKP and BLANK COMMON (see Global Model documentation) COMBDY, COMBDB, and BDYFLN (see section 5).

8.2 Local Variables Used

IFAIL	Used to detect return codes.
ITEMP	Temporary variable used in exchanging values.

8.3 Parameters

None.

8.4 Method

Section 1 begins with a test for whether or not the row currently in the buffer is the last row. If it is, control passes to section 2.

Section 1.1 initiates the read of the next row, updates the boundary line counter, and returns control to the calling routine.

Section 2 is entered with the boundary data sets at end of data. The count of number of times they have been read is updated, both data sets are rewound, and a test is made to determine whether they should be read again. If so, control passes to section 2.2.

Section 2.1 allocates new boundary data sets when the forecast time reaches the most recent boundary time. The pointer to the first boundary data is altered to point to the second data set, the first data set is returned. A new boundary data set is attached, and the pointer to the second data set points to the new data set.

Section 2.2 is entered with both boundary data sets in rewind state. The first record of each data set is skipped, and, provided no errors are encountered, control passes to section 1. Should errors be encountered, control is passed to section 9.

Section 9 prints error messages and terminates the run by calling ENDRUN.

8.5 Results

Provided no errors are encountered a read is initiated for the next row of boundary data required. If errors are detected the run is terminated after printing a suitable message.

9. SUBROUTINE SETK - 2.24 CALCULATE BOUNDARY RELAXATION FACTORS

9.1 Common Areas

None used.

9.2 Local Variables Used

Z	$\frac{1}{2}$ (N - 4) where N = number of points in boundary zone.
RZ	1/Z
DIST	Distance of a point from the boundary point nearest to it.

9.3 Parameters

PK	Not used.
PKD	Array (N by N) to receive relaxation coefficients.
PBDCON	Not used.
N	Number of points in boundary zone.

9.4 Method

The array PKD represents the N by N locations of the North West corner of the grid. For each point in PKD its distance from the nearest boundary point is obtained and the value of 1 - α stored in PKD, where α is given by equation (2). The value of the constant, a, in equation (2) is calculated according to equation (3).

9.5 Results

PKD is returned set to the values of 1 - α corresponding to the North West corner of the grid.

10. SUBROUTINE SETLNC - 2.24 EXTRACT RELAXATION FACTORS

10.1 Common Areas

None used.

10.2 Local Variables Used

ISTART, IEND	DO LOOP control variables.
ILIN	Row of array PKD required.

10.3 Parameters

PLINEA	Array length KDIM to receive a row of relaxation factors corresponding to the previous row.
KDIM	Number of grid points in a row.
PK	Not used.
PKD	Array N by N containing relaxation factors for the North West corner of the grid, as set by SETK.
N	Number of points in the boundary zone.
KPLINE	Row number
KLNSBD	.TRUE. if current line within the North or South boundary zone.
KLNBD	.TRUE. if current line between 2 and N + 1.

10.4 Method

Section 1	A test is made to determine whether or not the current line lies within the North or South boundary zones. If not, control passes to section 2.

Section 1.1	Values corresponding to East and West boundaries only are inserted into array PLINA, by locating the required values within PKD. A test is then made to determine the value of KLNBD; if this is .TRUE. control passes to section 2.4, where PLINB is set to values for the previous line. Control then passes back to the calling routine.
Section 2	deals with lines within the North and South boundary zones. First ILIN is set.
Section 2.1	sets the range of DO LOOP control for the central points.
Section 2.2	inserts appropriate values into points within the Eastern and Western boundary zones.
Section 2.3	inserts values into central points.
Section 2.4	performs a similar function for the previous row, placing the results in PLINEB.

10.5 Results

On exit, PLINA contains required values of $(1 - \alpha)$ where α is defined by equation (2). If values are likely to be required for the previous row, such values are returned in PLINEB.

REFERENCES :

Burridge, D. and Haseler, J. (1977) "A Model for Medium Range Weather Forecasting - Adiabatic Formulation" Technical Report No. 4 ECMWF, Bracknell

Haseler, J. and Burridge, D. (1977) "Documentation for the ECMWF Grid Point Model" ECMWF, Bracknell

Davies, H.C. (1976) "A Lateral Boundary Formulation for Multi-Level Prediction Models" Quart. J. R. Met. Soc. (1976), 102, pp. 405-418.

Kållberg, P. (1977) "Test of a Lateral Boundary Relaxation Scheme in a Barotropic Model" Internal Report No. 3, ECMWF, Bracknell

Appendix A - Ancillary Programme

This Appendix will contain details of ancillary programmes and subroutines designed to facilitate data preparation for the limited area model. At present the following are available :-

a) A programme to produce global data of any resolution from GFDL data, using bi-cubic spline interpolation.

b) A subroutine to stagger global on a latitude/longitude grid, using bi-cubic spline interpolation.

c) A programme to produce fine mesh fields from coarse mesh fields, with stagger correction, using bi-cubic spline interpolation.

d) Routines for converting field data to line data, line data to fields, etc.

Details of the above can be obtained from the authors and will be made available in this appendix at a later date.

-.-.-.-.-.-