5. TERRAIN Source Code

The TERRAIN code consists of the main program, TERRAIN, and thirty-three functions and subroutines. There are seven common blocks shared by the main program and most of the subroutines. This chapter describes in detail the common block structure, memory requirements, the main program design, and the subroutines' argument lists.

Similar to the GRAPH program, the TERRAIN program dynamically allocates the memory. Therefore, no parameterized dimensions are required from the user. The user does not need to recompile the program if he or she has no special modifications. The model domain specifications are defined through the TERRAIN namelist variables.

5.1 Common blocks and memory estimates

5.1.1 Common blocks and namelists

Each common block is discussed in two aspects: content and where it is used (main program or/and subroutines).

1. Common blocks

(1) /FUDG/

Content: information on land-use fudge option

Used in: CRLND, SETUP

(2) /LTDATA/

Content: information on the source terrain height and land-use dataset

Used in: CRLND, INTERP, SETUP, TERDRV

(3) /MAPS/

Content: information on the map background

Used in: TERRAIN, BNDRY, LATLON, LLXY, MXMNLL, PLTTER, RELP, TERDRV, XYOBSLL

(4) /NESTDMN/

Content: information on the model domain specifications

Used in: TERRAIN, ANAL2, CRTER, PLTTER, SETUP, TERDRV

(5) /OPTION/

Content: information on the options

Used in: TERRAIN, BNDRY, CRLND, CRTER, LATLON, MXMNLL, NESTLL, PLTTER, REPLACE, SETUP, TERDRV, XYOBSLL

(6) /SPACE/

Content: pointers of the arrays holding the input data

Used in: RDLDTR, TERDRV

(7) /TRFUDGE/

Content: information on the terrain fudge area

Used in: SETUP, TERDRV

(8) /HEADER/

Content: header information

Used in: TERRAIN, REPLACE, SETUP, TERDRV, TFUDGE, FINPRT

A complete list of the constants and variables in each of the common blocks and their descriptions is shown in Table 5.1.

2. Namelists

There are five namelists defined in subroutine SETUP. The default values and description of the namelist variables can be found in Section 4.2 and Table 5.1a–e. In subroutine SETUP, the namelist variables are read in from the TERRAIN script and override the default values. The following is a brief description of each of the namelists.

(1) /MAPBG/: information on map background

(2) /DOMAINS/: model domain specifications

(3) /OPTN/: options on how to run the TERRAIN program

(4) /FUDGE/: land-use fudging information

(5) /FUDGET/: information on terrain fudging area

Table 5.1a. Constants and Variables in Common Block /FUDG/
(maxnes = maximum number of domains)

Name	Type	Description	Source of the value	Default
IFFUG(maxnes)	logical	Flag for land-use fudging	Namelist input	.F.
NDFUG(maxnes)	integer	The number of fudging points	Namelist input	1
IFUG(100, maxnes)	integer	I indices of fudging points	Namelist input	0
JFUG(100, maxnes)	integer	J indices of fudging points	Namelist input	0
LNDFUG(100, maxnes)	integer	Land-use at point (I,J)	Namelist input	0

Table 5.1b. Constants and Variables in Common Block /LTDATA/

Name	Type	Description	Source of the value
ID1(5)	integer	The number of land-use categories	DATA statement in Sub. SETUP
ID2(5)	integer	The number of longitude data points	DATA statement in Sub. SETUP
NREC(5)	integer	The number of latitude data records	DATA statement in Sub. SETUP
LND(5)	integer	1: land-use data available 0: land-use data unavailable	DATA statement in Sub. SETUP
ITR(5)	integer	1: terrain height data available 0: terrain height data unavailable	DATA statement in Sub. SETUP
XLATI(5)	real	Initial integer latitude for the dataset	DATA statement in Sub. SETUP
XLONI(5)	real	Initial integer longitude for the dataset	DATA statement in Sub. SETUP
CENTER(5)	real	The resolution (degree) for the dataset	In Sub. SETUP
NIOLD	integer	The old dataset ID number	Initial value = 0 update in Sub. TERDRV
IVOL	integer	The unit number (=20) for 1 deg. land-use source data	DATA statement in Sub. SETUP
LNDNAME(3)	character*35	MSS file names for land-use source data	DATA statement in Sub. SETUP
TERNAME(5)	character*32	MSS file names for terrain height source data	DATA statement in Sub. SETUP

Table 5.1c. Constants and Variables in Common Block /MAPS/

Name	Type	Description	Source of the value	Remark
PHIC	real	The central latitude of the coarse domain	Namelist input	must be set in job script
XLONC	real	The central longitude of the coarse domain	Namelist input	must be set in job script
XN	real	The cone constant	Computed in Sub. SETUP	depends upon map projection
PSI1	real	Co-latitude	Computed in Sub. SETUP	See Section 3.1.2
POLE	real	The latitude of poles	Set in Sub. SETUP	90. for PHIC > 0 -90. for PHIC < 0
CONV	real	Constant to convert degree to radian	DATA statement in Sub. SETUP	= 57.29578
A	real	The earth radius(km)	DATA statement in Sub. SETUP	= 6370.
C2	real	A map parameter for Mercator projection	Computed in Sub. SETUP and RFLP	See Section 3.1.1
DISOBS(5)	real	The resolution (km) for the dataset	DATA statement in Sub. SETUP	The approximate values
IXEX	real	The I dimension of expanded coarse grid	Computed in Sub. SETUP	If IEXP = .FALSE., IXEX = NESTIX(1)
JXEX	real	The J dimension of expanded coarse grib	Computed in Sub. SETUP	If IEXP = .FALSE. JXEX = NESTJX(1)
XJC	real	The central J index of the coarse domain	Computed in Sub. SETUP	XJC = (NESTJX(1)+1)/2.0
XCNTR	real	Distance from center to the central longitude	Set in Sub. SETUP	Always equal to zero
YCNTR	real	Distance from center to the pole (km)	Computed in Sub. SETUP	See Section 3.1
CNTRJ0	real	The central J index of the coarse domain	Computed in Sub. SETUP	for expanded grid if IEXP=.TRUE.
CNTRI0	real	The central I index of the coarse domain	Computed in Sub. SETUP	for expanded grid if IEXP=.TRUE.
IOFFST	integer	The number of points extended on N-S side	Computed in Sub. SETUP	IOFFST=0 if IEXP=.TRUE.
JOFFST	integer	The number of points extended on W-E side	Computed in Sub. SETUP	JOFFST = 0 if IEXP=.TRUE.
NPROJ(3)	character *6	3 kinds of map projections	DATA statement Sub. SETUP	'LAMCON','POLSTR' 'MERCAT'
PROJECT	character *2	Specified map projection	Set in Sub. SETUP based on IPROJ	The value is 'LC' or 'ST' or 'ME'
IPROJ	character *6	Specified map projection	Namelist input	Must be set in job script

Table 5.1d. Constants and Variables in Common Block /NESTDMN/

Name	Type	Description	Source of the value
MAXNES	integer	The total number of domains	Namelist input
NESTIX(100)	integer	I (Y-direction) dimension for domains	Namelist input
NESTJX(100)	integer	J (X-direction) dimension for domains	Namelist input
NESTI(100)	integer	lower left corner (I) in the mother domain	Namelist input
NESTJ(100)	integer	lower left corner (J) in the mother domain	Namelist input
DIS(100)	real	grid distance (km) for domains	Namelist input
IRATIO(100)	integer	Ratio of mother to nest grid size	Namelist input
RID(100)	real	Radius of influence in ANAL2	Namelist input
MTHRD(100)	integer	"Mother" domain ID number	Namelist input
NTYPE(100)	integer	User specified source data ID number	Namelist input
NRATIO(100)	integer	Ratio of coarse to nest grid size	Computed in Sub. SETUP
XSOUTH(100)	real	South bdry. location of nest in coarse domain	Computed in Sub. SETUP
XNORTH(100)	real	North bdry. location of nest in coarse domain	Computed in Sub. SETUP
XWEST(100)	real	West bdry. location of nest in coarse domain	Computed in Sub. SETUP
XEAST(100)	real	East bdry. location of nest in coarse domain	Computed in Sub. SETUP
MIX	integer	The maximum I dimension of all domains	Computed in Sub. SETUP
MJX	integer	The maximum J dimension of all domains	Computed in Sub. SETUP

Table 5.1e. Constants and Variables in Common Block /OPTION/

Name	Type	Description	Default
DATASW	logical	.T.– user source dataset(ntype) .F.– program choose dataset	.F.
FIN	real	Contour interval (meter) for terrain height plot	100.
IBNDRY	logical	.T.– zero ter. ht. bdry. gradient .F.– not processing the boundary	.T.
IEXP	logical	.T.– expand the coarse domain .F.– do not expand the coarse domain	.F.
IFANAL	logical	.T.– use option ANAL2 .F.– use option INTERP	.F.
IFILL	logical	.T.– generate color plots for terrain height and land-use .F.– no color plots generated	.F.
IFTER	logical	.T.– create terrain height files .F.– no terrain height files created	.T.
IFFUDG	logical	.T.– fudging land-use .F.– no land-use fudging	.F.
IFTFUG	logical	.T.– use land-use 7 to redefine terrain height=-0.001 .F.– do not redefine terrain height	.F.
IPFLAG	logical	.T.– print fields in TFUDGE and REPLACE .F.– do not print in TFUDGE and REPLACE	.F.
IPRINT	integer	1– print some information 0– no information printed	0
IPRNTD	logical	.T.– print Lat.-Lon. on mesh .F.– no Lat.-Lon. printed	.F.
IPRTHT	logical	.T.– print ter. ht. on mesh .F.– no ter. ht. printed	.F.
IPRTLU	logical	.T.– print land-use on mesh .F.– no land-use printed	.F.
ISMTHTR	integer	1– use 1-2-1 smoother 2– use 2 pass smoother/desmoother	1
ISURF	logical	.T.– create land-use files .F.– no land-use files created	.T.
IWRITE	logical	.T.– write out results to disk .F.– do not write out to disk	.T.
NSTTYP	integer	1– create one-way nest terrain 2– create two-way nest terrain	2
TRUELAT1	real	define the first true latitude	91.
TRUELAT2	real	define the second true latitude	91.

Table 5.1f. Constants and Variables in Common Block /SPACE/

Name	Type	Description	Source of the value	Default
IBLOCK	integer	The length of arrays for source data	Computed in Sb. TERDRV	see Section 5.1.2
PTRD11	real	The location in core memory for array ZLON(IBLOCK)	Computed by routine HPALLOC	see Section 5.1.2
PTRD12	real	The location in core memory for array ZLAT(IBLOCK)	Computed by routine HPALLOC	see Section 5.1.2
PTRD13	real	The location in core memory for array HTOB(IBLOCK)	Computed by routine HPALLOC	see Section 5.1.2
PTRD21	real	The location in core memory for array ZLAND(13,IBLOCK)	Computed by routine HPALLOC	see Section 5.1.2

Table 5.1g. Constants and Variables in Common Block /TRFUDGE/

Name	Type	Description	Source of the value
STARTLAT	real	Latitude of the lower left corner of terrain fudge domain	Namelist input
ENDLAT	real	Latitude of the upper right corner of terrain fudge domain	Namelist input
STARTLON	real	Longitude of the lower left corner of terrain fudge domain	Namelist input
ENDLON	real	Longitude of the upper right corner of terrain fudge domain	Namelist input

Table 5.1h. Constants and Variables in Common Block /HEADER/

Name	Type	Description	Source of the value
MIF(200,10)	integer	header information represented by integer numbers	assigned in Sub. SETUP and TERDRV
MRF(200,10)	real	header information represented by real numbers	assigned in Sub. SETUP and TERDRV
MIFC(200,10)	character *80	description of the array MIF	assigned in Sub. SETUP and TERDRV
MRFC(200,10)	character *80	description of the array MRF	assigned in Sub. SETUP and TERDRV

5.1.2 Memory estimates

The terrain height and land-use files are created one domain at a time. Therefore, the program only needs to dynamically request the memory that is required to create the current working domain. Because the terrain height and land-use fields on the mesoscale grids are two-dimensional arrays, they usually do not require much memory. The majority of the program's memory is used to hold the source terrain height and land-use data. The source terrain and land-use data contain a total of 16 two-dimensional arrays. These include 2 arrays for latitude and longitude, 1 array for terrain height data, and 13 arrays for 13 land-use categories. When using high resolution data, the memory requirement increases substantially. The estimate of the length of the 2-D array that holds the source data is

$$length = ((\lambda_{max} - \lambda_{min}) \times \frac{1}{reso} + 2) \times ((\phi_{max} - \phi_{min}) \times \frac{1}{reso} + 2)$$

where $\lambda_{max}, \lambda_{min}$, ϕ_{max}, and ϕ_{min} are the maximum and minimum longitudes and latitudes of the search area for the given domain (see Section 3.2). $reso$ is the source data resolution ($^\circ$). $length$ has units of word.

The pointers, PTRD11, PTRD12, PTRD13, and PTRD21 in common block /SPACE/, are used to position the four arrays (three with size $length$ and one with the size $13 \times length$) which hold longitude, latitude, height, and land-use in the core memory. The CRAY-dependent routines HPALLOC and HPDEALLC which are called by the subroutine TERDRV will dynamically allocate this large part of core memory during program execution. If the user's requested maximum memory in the TERRAIN job script is smaller than what is actually required, the message "ERROR IN HEAP ALLOCATION" and a suggested memory size in Mb will appear in the user's terrain output. The program then stops. The user needs to change the QSUB -lM command in the TERRAIN job script and resubmit the job script.

5.2 Main program design

The main program TERRAIN is a very short routine and calls other subroutines to (1) initialize the variables, (2) create the terrain and land-use fields on the mesoscale grids, (3) reset the boundary values for the nested domains, (4) provide feedback to the mother domains, and (5) plot and print the final results. The program starts with assigning a gmeta file, 'TER.PLT', for the output plot fields, and then calls the start-up NCAR GKS routines. It then calls SETUP for initialization. After SETUP, there are four DO-loops, DO-loop 10, DO-loop 21, DO-loop 22, and DO-loop 30. If only one domain is desired, the program skips DO-loops 21 and 22. If the nest type is one-way, the program skips Do-loop 22. The main program exits with the final calls to close NCAR GKS.

The flow chart for the main program is shown in Fig. 5.1. The purpose of the subroutines that are called by the main program are described below.

5.2.1 Begin initialization

(SETUP)

The constants and the variables, which can be determined based on the input namelist information, are assigned or calculated in SETUP. These constants and variables are then used by the main program and other subroutines. In addition to a set of DATA statements at the beginning of SETUP which provide the source dataset information residing in MSS, the subroutine SETUP also performss the following.

(1) The namelist variables are assigned their default values, and are then over-written with user-selected values by reading in the namelist file.

(2) The map parameters are calculated based on user defined map projection and center latitude and longitude. These parameters include cone factor and the distance from the domain center to the pole.

(3) The coarse domain extended dimensions and the increments of the extension are calculated if the extended grid option is chosen, and the central indices I_c, J_c of the coarse domain and the maximum dimensions of all the domains are determined.

(4) The compatibility of the nested domains with the coarse domain. The ratio of the grid distances between the nest and its mother domain must be close to an integer, and the four corners of the nest must be located on the mother domain's grid points.

(5) The coarse domain grid locations (real numbers) that define the 4 corner points of the nest domains.

(6) The TERRAIN binary output record header arrays.

5.2.2 Create the terrain and land-use

(TERDRV, PLTTER, LATLON, MXMNLL, NESTLL, RDLDTR, CRTER, INTERP, XYOBSLL, ANAL2, CRLND, BINT, FUDGER, SMTHTR/SMTH121, LAKES, BNDRY)

The subroutine TERDRV is a driver routine which calls other subroutines that complete the task of creating the terrain height and land-use data on the mesoscale grids (Fig. 5.2). The main program calls TERDRV inside DO-loop 10 for each of the desired domains. The purpose of each call inside the TERDRV is listed below.

(1) Call subroutine LATLON to calculate the latitudes and longitudes of the mesoscale grid points.

(2) Call subroutine MXMNLL to determine the search area. In MXMNLL the subroutine NESTLL is called if the pole is inside the coarse domain.

(3) Call subroutine RDLDTR to read in the source terrain data.

(4) Call subroutine CRTER to create the terrain height field. In CRTER call subroutine INTERP for overlapping parabolic interpolation, and call subroutines XYOBSLL and ANAL2 for Cressman-type objective analysis.

(5) Call subroutine RDLDTR to read in the source land-use data. If the source terrain data resolution is higher than 10 min, use the 10-min source land-use data.

(6) Call subroutine CRLND to create the land-use field. In CRLND the function BINT is used for interpolation, and subroutine FUDGER is called to fudge the land-use.

(7) Create the final terrain and land-use fields. Call subroutines SMTHTR or SMTH121 to smooth the terrain field. If the model domain includes the great lakes, call subroutine LAKES to correct the heights over great lakes. Call subroutine BNDRY to zero the terrain gradient on the coarse domain boundaries. Check for consistency between the terrain height and land-use category.

(8) Fill in the rest of the terrain output header record arrays, and write out the terrain height and land-use files on the local working disk.

Also, some minor subroutines and functions, such as NUNPACK, IUNPACK, RFLP, and ONED, etc., are called by some of the subroutines and functions mentioned above. The user can find these subroutines or functions in Section 5.3.

5.2.3 Reset the boundary values for the nested domains

(TFUDGE, EXAINT, SINT, BINT)

The procedure to reset the boundary values for the nested domain is described in Section 3.5.1. Since the terrain binary output is written out in subroutine TERDRV for each domain, subroutine TFUDGE then reads in the terrain height and land-use fields for the nest and its mother domain. After the boundary values for the nested domain have been adjusted, the new terrain height and land-use are written out to the disk again with the same unit number.

The main program calls the subroutine TFUDGE inside the DO-loop 21 for $maxnes-1$ times. If only one domain is desired, the subroutine TFUDGE is not called by the main program.

5.2.4 Feedback to the mother domains

(REPLACE)

If the nest type is two-way, subroutine REPLACE is called by the main program inside DO-loop 22 from *maxnes* to 2 to replace the mother domain terrain height at the overlapping grid points with its nest values. Again, the terrain height and land-use are read in from the disk. After the replacement, the final terrain height and land-use for the mother domain are written out to the disk with the same unit number.

5.2.5 Print and plot the final results

(FINPRT, OUTPT, PLTTER)

In addition to the binary terrain height and land-use outputs, TERRAIN also generates ASCII and gmeta plot files. The main program calls subroutine FINPRT to produce both ASCII and gmeta outputs. If the variables IPRTHT and IPRTLU are set to be TRUE in the TERRAIN job script, subroutine OUTPT is called from FINPRT to print out the gridded terrain height and land-use values. FINPRT also calls subroutine PLTTER to produce gmeta plots. The MM5 modeling system plot utilities, MAPDVR and CONDRV, are called from the PLTTER to create maps and contours. This plot utility is saved as an object code and is read in from MSS file /MESOUSER/plots/object/plots.o. Its Fortran source code can be found in the MSS under the name /MESOUSER/plots/source/plots.f.

TERRAIN Main Program Flow Chart

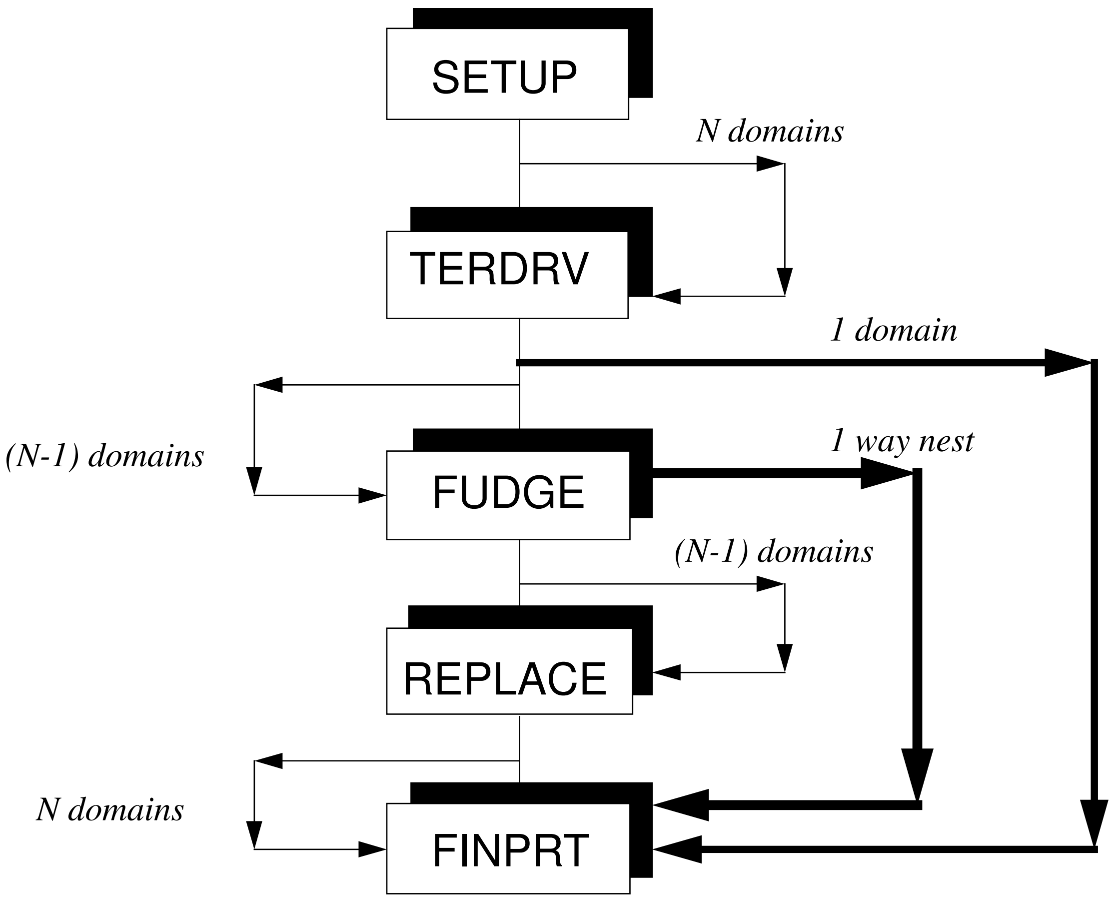

Figure 5.1 TERRAIN Main Program Flow Chart

Subroutine TERDRV Flow Chart

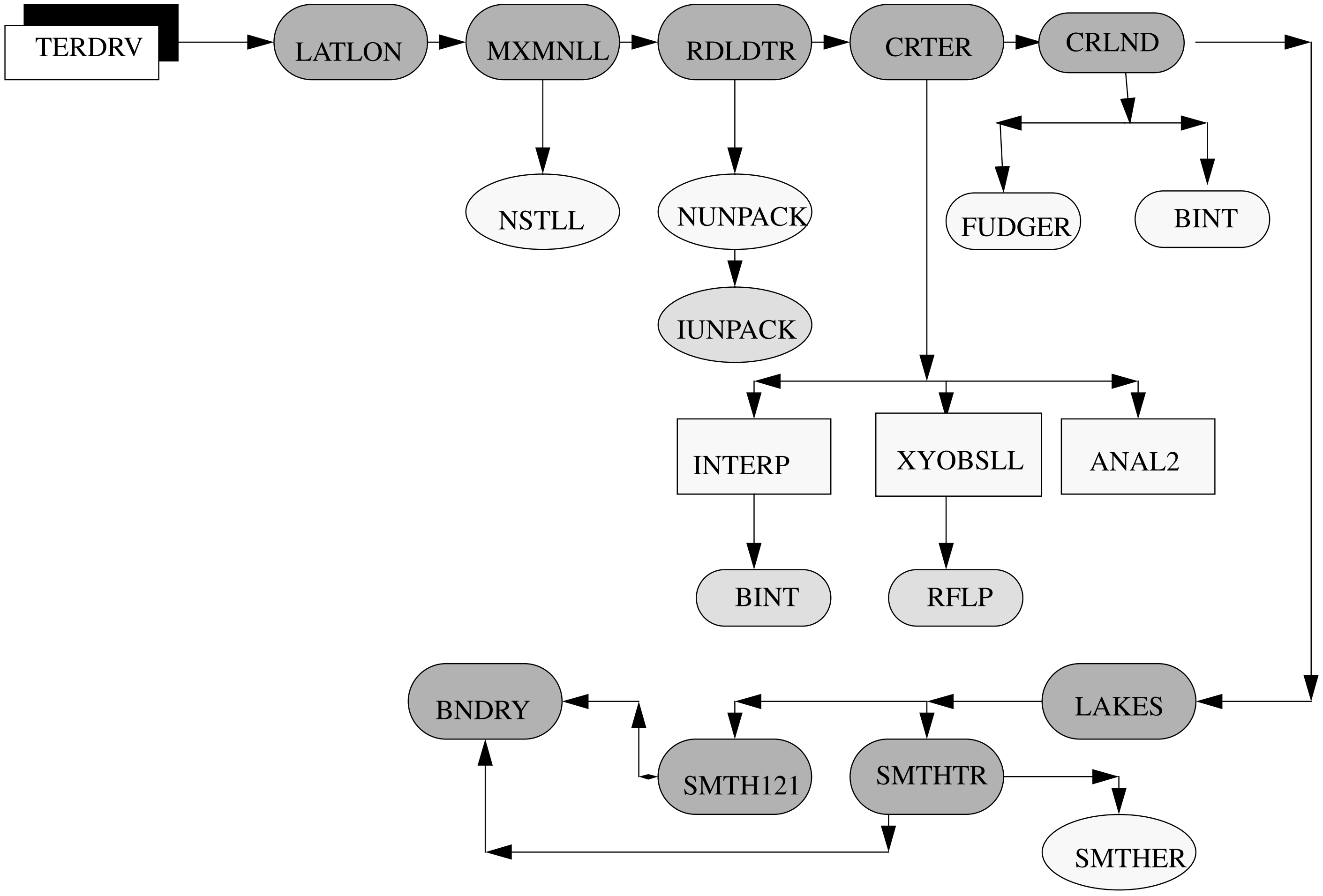

Figure 5.2 Subroutine TERDRV Flow Chart

5.3 TERRAIN Subroutines

This section discusses the TERRAIN subroutines and functions in alphabetical order. For each of the subroutines and functions, important input and output variables and arrays (from both the argument list and the common blocks) are discussed.

The subroutines and functions are discussed according to the following pattern:

Purpose:	A brief statement summarizing the main purpose or purposes of the subprogram.
On Entry:	Descriptions of the state of arguments and certain common variables on entry to the subprogram. All important input to the subprogram is listed in this category.
On Exit:	Descriptions of the state of important arguments and common variables on completion of the subprogram. All important output from the subprogram is listed in this category.
Calls:	A list of any subprograms which are called by the subprogram being discussed.
Called by:	A list of any routines which call the subprogram.
Comments:	A more detailed discussion than was provided in the **Purpose** section. The comments may include a discussion of the methods which the subprogram uses to achieve the purpose as well as more detailed descriptions of the subprogram's code itself.

Subroutine **ANAL2**(A2, ASTA, XOBS, YOBS, IMAX, JMAX, NSTA, N, NI)

Purpose: Create the terrain height fields on mesoscale grids by using the Cressman-type objective analysis technique.

On Entry:

ASTA(NSTA):	Terrain heights from the source data.
XOBS(NSTA):	J indices of the source data.
YOBS(NSTA):	I indices of the source data.
NSTA:	Number of data points.
IMAX,JMAX:	The dimension of the mesoscale domain.
N:	The ID number of the mesoscale domain.
NI:	The ID number of the source data file. NI = 1: 1 degree resolution data NI = 2: 30 minutes resolution data NI = 3: 10 minutes resolution data NI = 4: 5 minutes resolution data NI = 5: 30 seconds resolution data

On Exit: A2(IMAX,JMAX): The terrain height field on the mesoscale domain N.

Calls: Subroutine ANAL2 calls no other subroutines.

Called by: Subroutine CRTER.

Comments: No first-guess fields are used, and a single scan only is performed in the Cressman-type objective analysis. The radius of influence RID is defined through the common block /NESTDMN/. Before calling ANAL2, subroutine XYOBSLL must be called first to convert the latitudes and longitudes of the source data points to the mesoscale indices XOBS and YOBS. Please refer to Section 3.3.1 in this documentation for more details.

Function **BINT**(XX, YY, LIST, III, JJJ, FLAG)

Purpose: Perform the overlapping-quadratic interpolation to the mesoscale grid point (XX, YY) from 16 surrounding grid points.

On Entry:	LIST(III,JJJ):	The input source terrain height or land-use data, *not* on the mesoscale grid.
	XX,YY:	X and Y index values (interpolation location) of the mesoscale grid point in the coordinate system of the input grid.
	FLAG:	FLAG = .FALSE.: 16-point interpolation; FLAG = .TRUE.: 4-point interpolation.
On Exit:	BINT:	The interpolated value of the variable in LIST, at point (XX, YY).

Calls: Function ONED.

Called by: Subroutines CRLND and INTERP.

Comments: Function BINT is called for each mesoscale grid point in each field. For each mesoscale grid point, BINT constructs the array STL(4,4) which contains the values at the sixteen surrounding input grid points. Next, BINT calls function ONED four times to calculate four interpolated values along the line x = XX. Finally, BINT calls function ONED one more time to interpolate from the four new values to the point (XX,YY). In case of FLAG = .TRUE., the outer 12 elements of array STL are always set to zero. Only the central 4 elements have the values from LIST and function ONED will complete the linear interpolation. In the TERRAIN program, the FLAG is always set to be .FALSE..

If the point (XX,YY) is near the boundary of the input field and is not surrounded by 16 input grid points, the missing edge points are not included in the interpolation procedure. In this case, BINT repeats the process. This time, however, BINT interpolates first to the four points along line y = YY and then from those four points to point (XX,YY). BINT then returns the average of the two methods as the value at the interpolated point.

Subroutine BNDRY(IX, JX, N, HT, LAND, LNDOUT)

Purpose: Zero the terrain height and land-use normal gradient at the boundary of the domain N.

On Entry:	IX, JX:	The dimension of the mesoscale domain N.
	N:	The ID number of the mesoscale domain.

	HT(IX,JX):	The terrain height field.
	LAND(IX,JX):	The land-use field. This is an integer array.
	LNDOUT(IX,JX):	The land-use field. This is a real array for the use of print and plot.
On Exit:	HT(IX,JX):	The terrain height field with the zero normal gradient at the boundary.
	LAND(IX,JX):	The land-use field with the zero normal gradient at the boundary.
	LNDOUT(IX,JX):	The land-use field with the zero normal gradient at the boundary.

Calls: Subroutine BNDRY calls no other subroutines.

Called by: Subroutine TERDRV.

Comments: By setting the values at the boundary equal to the values one grid point inside the boundary, the normal gradient at the boundary becomes zero. Most of numerical models take this provision to avoid computational trouble near the boundary.

In the MM5 modeling system, the terrain height and land-use are defined at the cross points with the dimensions of IX-1 and JX-1. The value of 10^{36} is assigned to the last rows and columns of the array HT (HT(I,JX) and HT(IX,J)) and 10^{10} to the arrays LAND (LAND(I,JX) and LAND(IX,J)) and LNDOUT (LNDOUT(I,JX) and LNDOUT(IX,J)).

Subroutine CRLND(LAND, LNDOUT, XLAT, XLON, IX, JX, III, JJJ, N, NI, NOBS, GRDLNMN, GRDLTMN, ZLON, ZLAT, ZLAND)

Purpose: Create the land-use categories field on the mesoscale grids by using the overlapping parabolic interpolation method with the source land-use data (percentage) on the regular latitude-longitude mesh as input.

On Entry:	XLAT(IX,JX):	Latitudes of the mesoscale grid points.
	XLON(IX,JX):	Longitudes of the mesoscale grid points.
	IX,JX:	The dimensions of the mesoscale domain N.

	III,JJJ:	The dimensions of the fields with regular latitude-longitude intervals (see Section 5.1.2).
	N:	The ID number of the mesoscale domain.
	NI:	The ID number of source data file. The land-use source data ID should not exceed 3 (see subroutine ANAL2).
	NOBS:	The number of input source data points.
	GRDLNMN:	The minimum longitude of the source data.
	GRDLTMN:	The minimum latitude of the source data.
	ZLON(NOBS):	Longitudes of the input source data.
	ZLAT(NOBS):	Latitudes of the input source data
	ZLAND(13,NOBS):	The 13 categories of the source land-use data. 1. Urban land 2. Agricultural land 3. Range land/grassland 4. Deciduous forest 5. Coniferous forest 6. Mixed forest including wetland 7. Water 8. Non-forested marsh or wetland 9. Desert 10. Tundra 11. Permanent ice 12. Nonseasonal tropical or subtropical forest 13. Savannah
On Exit:	LAND(IX,JX):	The land-use categories on the mesoscale domain N. This is an integer array.
	LNDOUT(IX,JX):	The land-use categories on the mesoscale domain N. This is a real array.

Calls: Subroutine CRLND calls function BINT and FUDGER.

Called by: Subroutine TERDRV.

Comments: In addition to the list of arguments, this subroutine gets the rest of its information from the common blocks /LTDATA/, /FUDG/, and /OPTION/. Based on the resolution and the minimum latitude and longitude of the source data, the one-dimensional source data are put into a 2-dimensional array LNDIN(III,JJJ). Then the indices of the mesoscale grid points in the latitude-longitude mesh are computed from XLAT and XLON, and the function BINT is called to interpolate the percentages of each of the categories to the mesoscale grids. The category with the highest percentage is set to be the land-use value at that grid point (see section 3.4 for more details).

If IFFUG = .TRUE., the subroutine FUDGER is called to fudge the land-use with the information from common block /FUDG/.

Subroutine CRTER(HT, IX, JX, N, DSS, ITER, JTER, ZLON, ZLAT, HTOB, NI, XLAT, XLON, INCR, NOBS, GRDLNMN, GRDLTMN, IMX, JMX)

Purpose: Create the terrain height field on the mesoscale grids by using the overlapping parabolic interpolation or Cressman-type objective analysis method with the source terrain height data on the regular latitude-longitude mesh as input.

On Entry:	IX,JX:	The dimensions of the domain N.
	N:	The ID number of the mesoscale domain.
	DSS:	The grid size of the domain N.
	ITER,JTER:	The dimensions of the fields with regular latitude and longitude intervals (see Section 5.1.2).
	ZLON(NOBS):	Longitudes of the input source data.
	ZLAT(NOBS):	Latitudes of the input source data.
	HTOB(NOBS):	The input terrain height source data.
	NI:	The ID number of the source data file.
	XLAT(IX,JX):	Latitudes of the mesoscale grid points.
	XLON(IX,JX):	Longitudes of the mesoscale grid points.
	INCR:	The source data resulotion (in $^\circ$).
	NOBS:	The number of the input source data points.
	GRDLNMN:	The minimum longitude of the source data.
	GRDLTMN:	The minimum latitude of the source data.
	IMX,JMX:	The dimensions of the coarse domain, including the extended increments if the coarse domain is expanded (IEXP = .TRUE.).
On Exit:	HT(IX,JX):	The terrain height field on the mesoscale domain N.

Calls: Subroutine CRTER calls the subroutines INTERP, XYOBSLL, and ANAL2.

Called by: Subroutine TERDRV.

Comments: In addition to the list of arguments, this subroutine gets more information about the domain settings and the options from the common blocks /NESTDMN/ and /OPTION/. There are two options to create the terrain height field: overlapping parabolic interpolation (IFANAL = .FALSE.) and Cressman-type objective analysis (IFANAL = .TRUE.). For overlapping parabolic interpolation, the one-dimesional input source data are put into a 2-dimensional array HTIN(ITER,JTER) prior to the call to subroutine INTERP. For Cressman-type objective analysis, subroutine XYOBSLL is called first to convert the latitudes and longitudes of the source data to the mesoscale grid indices, then subroutine ANAL2 is called to generate the terrain height field HT(IX,JX) on the mesoscale domain from the source data HTOB(NOBS).

If IPRINT = .TRUE., the minimum longitude and latitude and the maximum and minimum heights of the input source data are printed.

Subroutine EXAINT(TA, IYY, JXX, TAN, IYYN, JXXN, ISOUTH, JWEST, ICRSDOT, MIXX, MJXX)

Purpose: Gets the field on the fine grid from the field on its mother grid by using the semi-Lagrangian interpolation scheme.

On Entry:

TA(IYY,JXX):	The field on the mother grid.
IYY,JXX:	The dimension of the mother domain.
IYYN,JXXN:	The dimensions of the fine domain.
ISOUTH:	The mother domain grid index which defines the southern boundary of the fine domain.
JWEST:	The mother domain grid index which defines the western boundary of the fine domain.
ICRSDOT:	The flag indicating whether TA is a cross (=1) or dot (=0) points field.
MIXX,MJXX:	The maximum dimensions of the *maxnes* domains, including the expanded coarse domain if IEXP = .TRUE..

On Exit: TAN(IYYN,JXXN): The interpolated field on the fine grid.

Calls: Subroutine EXAINT calls the subroutines SINT.

Called by: Subroutine TFUDGE.

Comments: The nest ratio of the mother grid to the fine grid is allowed to be only 3 in this subroutine. The MIXX and MJXX are used to define the working arrays PSCA, XIG, XJG, IGO, and JGO (Grell and Dudhia, 1993). If the mother domain is domain 1 (the coarse domain), the expanded increments must be added to ISOUTH and JWEST if the coarse domain is expanded (IEXP = .TRUE.).

Subroutine FINPRT(IMAX, JMAX, NM)

Purpose: Reads the terrain height and land-use for the domain NM, and prints and plots these fields.

On Entry: IMAX,JMAX: The dimensions of the domain NM (non-expanded grid dimension even if IEXP = .TRUE.).

NM: The ID number of the mesoscale domain.

On Exit: The terrain height and land-use for the domain NM are printed on the mesoscale mesh.

The plots for the domain NM (see subroutine PLTTER).

Calls: Subroutine FINPRT calls the subroutine OUTPT and PLTTER.

Called by: The main program TERRAIN.

Comments: The common block /OPTION/ is used in this subroutine.

For the coarse domain (NM=1), the common header information is printed out. As for other domains, only the domain header information is printed out.

Subroutine FUDGER(N, NPFUG, IFUG, JFUG, LNDFUG, LAND, LNDOUT, IX, JX)

Purpose:	Assigns the fudge values for the domain N.	
On Entry:	N:	The ID number of the mesoscale domain.
	NPFUG:	The number of fudge points.
	IFUG(NPFUG):	The I indices of the fudge points.
	JFUG(NPFUG):	The J indices of the fudge points.
	LNDFUG(NPFUG):	The land-use category at the fudge points.
	LAND(IX,JX):	The land-use field, integer array.
	LNDOUT(IX,JX):	The land-use field, real array.
	IX,JX:	The dimensions of the mesoscale domain N.
On Exit:	LAND(IX,JX):	The land-use field after fudge, integer array.
	LNDOUT(IX,JX):	The land-use field after fudge, real array.
Calls:	Subroutine FUDGER calls no other subroutines.	
Called by:	Subroutine CRLND.	
Comments:	The I and J indices of the coarse domain are in reference to the expanded grid. The TERRAIN program takes care of this in subroutine SETUP. From the namelist input, the I and J provided by the user are referred to the non-expanded coarse domain indices.	

Subroutine INTERP(IMAX, JMAX, XLON, XLAT, HTGRID, HTIN, NI, GRDLTMN, GRDLNMN, III, JJJ)

Purpose:	Creates the terrain height fields on mesoscale grids by using the overlapping parabolic interpolation method with terrain source data on the regular latitude-longitude grid as input.	
On Entry:	IMAX,JMAX:	The dimensions of the mesoscale domain.
	XLON(IX,JX):	Longitudes of the mesoscale grid points.
	XLAT(IX,JX):	Latitudes of the mesoscale grid points.

	HTIN(III,JJJ):	The source terrain height field with the regular latitude and longitude intervals (see Section 5.1.2).
	NI:	The ID number of the source data file.
	GRDLTMN:	The minimum latitude of the source data.
	GRDLNMN:	The minimum longitude of the source data.
	III,JJJ:	The dimensions of the array HTIN.
On Exit:	HTGRID(IMAX,JMAX):	The interpolated terrain height field on the mesoscale domain.

Calls: Subroutine INTERP calls function BINT.

Called by: Subroutine CRTER.

Comments: In this subroutine, the FLAG is always set to be .FALSE. This means the 16-point overlapping parabolic interpolation is used. If one of the 16 points is missing, the interpolation has no meaning and the program is stopped. Since the 30-second terrain height data (NI = 5) is a local dataset, this situation may occur and INTERP will not work.

Subroutine IPACK(LL, NN, NP)

Purpose: Pack the NP integers NN(NP) to an integer.

On Entry:	NN(NP):	The integer array.
	NP:	The dimension of array NN.
On Exit:	LL:	The packed integer.

Calls: Subroutine IPACK calls the functions SHIFTL and OR.

Called by: Subroutine NPACK.

Comments: Actually the subroutines IPACK and NPACK are not used in the TERRAIN program. These two subroutines are used to pack the data in the format that TERRAIN uses.

The CRAY Fortran Intrinsic Functions, SHIFTL and OR, etc., are used here. They are boolean functions and the definitions are shown below.

Function name	Definition	Argument range
SHIFTL(x1,x2)	x1 shifts left x2 positions leftmost positions lost rightmost positions set to zero	$0 \leq x2 < 64$
SHIFTR(x1,x2)	x1 shift right x2 positions rightmost positions lost leftmost positions set to zero	$0 \leq x2 < 64$
OR(x1,x2)	bit-by-bit sum of x1 and x2	
AND(x1,x2)	bit-by-bit product of x1 and x2	

The above intrinsic functions should be found in other versions of the Fortran library on other machines. The CRAY is a 64-bit word machine; If you want to move the subroutine IPACK and NPACK to a 32-bit word machine, only minor modifications are needed (see Appendix C).

Subroutine IUNPACK(LL, M1, NP)

Purpose: Unpack the integer LL to NP integer variables M1(NP).

On Entry: LL: The integer.

NP: The dimension of array M1.

On Exit: M1(NP): The NP integer variables.

Calls: Subroutine IUNPACK calls the functions SHIFFTL, SHIFTR and AND.

Called by: Subroutine NUNPACK.

Comments: This subroutine is used to unpack the data. For the definition of functions SHIFTL, SHIFTR, and AND, see subroutine IPACK for details.

Subroutine LAKES(IX, JX, HT, LAND, XLAT, XLON)

Purpose: Corrects the terrain heights over the Great Lakes.

On Entry:

IX,JX:	The dimensions of the mesoscale domain.
HT(IX,JX):	Terrain height field.
LAND(IX,JX):	Land-use field.
XLAT(IX,JX):	Latitudes of the mesoscale grid.
XLON(IX,JX):	Longitudes of the mesoscale grid.

On Exit: HT(IX,JX): The terrain height field after correction over the Great Lakes.

Calls: Subroutine LAKES calls no other subroutines.

Called by: Subroutine CRTER.

Comments: Based on the land-use category 7 (water) and the latitudes and longitudes of the mesoscale grid, TERRAIN locates the Great Lakes region and re-sets the terrain height for the Great Lakes.

Lake name	Height(m)
Erie	174.0
Huron	177.0
Ontario	75.0
Superior	183.0
Michigan	177.0

Subroutine LATLON(IYY, JXX, XLAT, XLON, DS, XICNS, XJCNS, N, IRATIO, IP)

Purpose: Calculates the latitudes and longitudes of the mesoscale grids for domain N (refer to Section 3.1 for details).

On Entry:

IYY,JXX:	The dimensions of the mesoscale domain N.
DS:	The grid size of the domain N.
XICNS,XJCNS:	The I and J location of the domain N grid point (1,1) in the coarsest domain (domain 1). They are real numbers.

	N:	The ID number of the mesoscale domain.
	IRATIO:	The ratio of the grid distances between domain 1 and domain N.
	IP:	The switch for printing the XLAT and XLON fields.
On Exit:	XLAT(IYY,JXX):	Latitudes of the grid points for domain N.
	XLON(IYY,JXX):	Longitudes of the grid points for domain N.

Calls: Subroutine LATLON calls the subroutine OUTPT and the intrinsic functions EXP, COS, SQRT, SIN, TAN, ATAN, and ATAN2.

Called by: Subroutine CRTER.

Comments: In addition to the list of arguments, subroutine LATLON gets more information from the common blocks /MAPS/ and /OPTION/. Because terrain height and land-use are defined at the cross points, XLAT and XLON are the latitudes and longitudes at the cross points. Their ranges are from $-90°$ to $90°$ and from $-180°$ to $180°$, respectively. The location of the lower left corner and the grid size ratio are relative to the coarse domain, not its mother domain. This information is calculated in the subroutine SETUP. If IP = .TRUE., the latitude and longitude fields are printed out for verification.

Subroutine LLXY(XLAT, XLON, X, Y, IEND, JEND, DSKM)

Purpose: Calculates the X and Y given the latitude and longitude.

On Entry:	XLAT:	The latitude.
	XLON:	The longitude.
	IEND,JEND:	The number of domain grid points in the I and J directions in the coarse domain with the grid distance DSKM.
	DSKM:	The domain grid distance.
On Exit:	X:	The coordinate in the X (J)-direction.
	Y:	The coordinate in the Y (I)-direction.

Calls: Subroutine LLXY calls the intrinsic functions SIN, COS and TAN.

Called by: Subroutine PLTTER

Comments: In addition to the list of arguments, subroutine LLXY gets the map information (the central latitude (PHIC) and longitude (XLONC), the cone constant XN, etc.) from the common block /MAPS/. Since PHIC and XLONC are set with respect to the coarse mesh, the X and Y values that come out of the subroutine LLXY are the coarse domain grid indices that define the station location. The TERRAIN program then coverts the X and Y values back to the nested domain indices.

Subroutine MXMNLL(IYY, JXX, YLAT, YLON, XI1, XI2, XJ1, XJ2, XEXTND, XMAXLAT, XMIMLAT, XMAXLON, XMINLON, IPOLE, INSET)

Purpose: Finds the maximum and minimum latitude and longitude of the search area for the domain INEST+1 (refer to Section 3.2 for more details).

On Entry: IYY,JXX: The dimensions of the domain INEST+1.

YLAT(IYY,JXX):	Latitudes of the mesoscale grids.
YLON(IYY,JXX):	Longitudes of the mesoscale grids.
XI1:	The location of the southern boundary of the domain INEST+1 (see Section 3.2.1).
XI2:	The location of the northern boundary of the domain INEST+1 (see Section 3.2.1).
XJ1:	The location of the western boundary of the domain INEST+1 (see Section 3.2.1).
XJ2:	The location of the eastern boundary of the domain INEST+1 (see Section 3.2.1).
XEXTND:	The increment to expand the search area (see Section 3.2.1).
IPOLE(20):	A flag array to indicate the pole is inside (= 1) or outside (=0) the domain.
INEST:	The domain ID number - 1. For the coarse domain, INEST=0. For the nest domain, INSET > 0.

On Exit:

XMAXLAT:	The maximum latitude of the search area.
XMINLAT:	The minimum latitude of the search area.

XMAXLON:	The maximum longitude of the search area.
XMINLON:	The minimum longitude of the search area.
IPOLE(INEST+1):	IPOLE(INEST+1) = 1: pole is inside the domain, IPOLE(INEST+1) = 0: pole is outside the domain.

Calls: Subroutine MXMNLL calls the subroutine NESTLL.

Called by: Subroutine TERDRV.

Comments: The method to determine the XMAXLAT, XMINLAT, XMAXLON, and XMINLON is described in Section 3.2. In addition to the list of arguments, subroutine MXMNLL gets the central latitude PHIC, the central J-index XJC, etc., from the common block /MAPS/ and the switch IPRINT from the common block /OPTION/.

Subroutine NESTLL(IYY, JXX, YLAT, YLON, PHIC, XMAXLAT, XMINLAT, XMAXLON, XMINLON, XEXTND, IPOLE, INEST)

Purpose: Finds the maximum and minimum latitude and longitude of the search area for the NEST domains when the pole is inside the coarse domain for polar stereographic projection (refer to Section 3.2.3 for more details).

On Entry: IYY,JXX: The dimensions of the domain INEST+1.

YLAT(IYY,JXX):	Latitudes of the mesoscale grids.
YLON(IYY,JXX):	Longitudes of the mesoscale grids.
PHIC:	The central latitude of the coarse domain.
XEXTND:	The increment to expand the search area (see Section 3.2.1).
INEST:	The domain ID number - 1. For the coarse domain, INEST=0. For the nest domain, INEST $>$ 0.

On Exit:

XMAXLAT:	The maximum latitude of the search area.
XMINLAT:	The minimum latitude of the search area.
XMAXLON:	The maximum longitude of the search area.
XMINLON:	The minimum longitude of the search area.

IPOLE(INEST+1): IPOLE(INEST+1) = 1: pole is inside the domain,
IPOLE(INEST+1) = 0: pole is outside the domain.

Calls: Subroutine NESTLL calls no other subroutines.

Called by: Subroutine MXMNLL.

Comments: This subroutine is only called when the north or south poles are inside the coarse domain.

Subroutine NPACK(INA, IPA, NW, NP)

Purpose: Pack the integer array INA(NW) into the integer array IPA.

On Entry: INA(NW): The original input integer array.

NW: The dimension of the array INA.

NP: The packing density:

NP=2: pack 2 integers(32 bits) into one 64-bit word;
NP=4: pack 4 integers(16 bits) into one 64-bit word;
NP=8: pack 8 integers(8 bits) into one 64-bit word.

On Exit: IPA: The array holds the packed data with the size of NW/NP+2.

INA: A constant that is added to the original values, so all the values are greater than or equal to zero and the minimum must be zero.

Calls: Subroutine NPACK calls the subroutine IPACK.

Called by: No other subroutine in TERRAIN.

Comments: This subroutine is not used in the program TERRAIN. It is used to pack the source data into the format TERRAIN uses. The size of array IPA is NW/NP+2. The first element IPA(1) of the array IPA holds a constant, which is added to all elements of the array INA so that they are greater than or equal to zero prior to the packing. The last element of IPA is a residual one and is set to be zero. For a 32-bit machine, only minor modifications are needed to make this subroutine portable (see Appendix C).

Subroutine NUNPACK(INA, IPA, NW, NP)

Purpose: Unpacks the integer array IPA into the integer array INA(NW).

On Entry: IPA(NW/NP+2): The array holds the packed data.

NW: The dimension of the unpacked data array INA.

NP: The packing density (see subroutine NPACK).

On Exit: INA(NW): The integer array holds the unpacked data.

Calls: Subroutine NPACK calls the subroutine IUNPACK.

Called by: Subroutine RDLDTR.

Comments: This is a sister subroutine of the subroutine NPACK which is used to pack the data in data reconstruction. See subroutine NPACK for details. For a 32-bit machine, only minor modifications are needed to make it portable (see Appendix C).

Function ONED(X, A, B, C, D)

Purpose: Perform a one-dimensional interpolation to a point ξ from four flanking points, two on either side.

On Entry: X: Grid distance of interpolation location from the point B (See Fig. 5.3), and $0.0 \leq X \leq 1.0$.

A,B,C,D: Values at the four flanking grid points, from which the fifth value is interpolated. The grid distance equals 1.0 here.

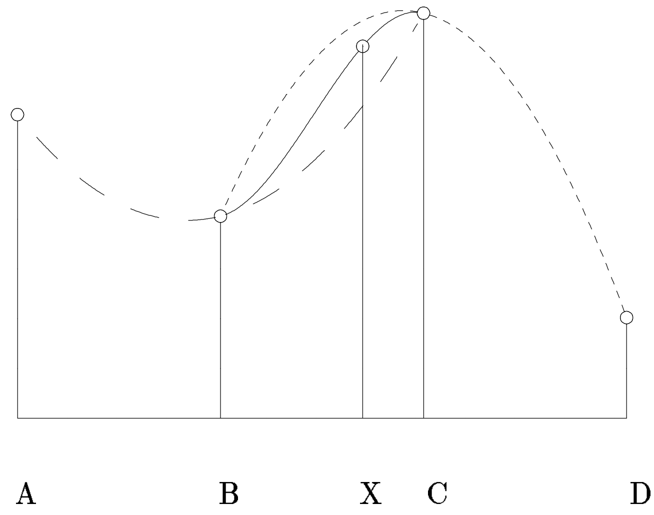

Figure 5.3 A biparabolic fit to point X from four points A, B, C, and D.

On Exit: ONED: The interpolated value at the interpolation location.

Calls: Function ONED calls no other subprograms.

Called by: Function BINT.

Comments: In this function ONED, the maximum number of data points available is 4. But the function ONED is still working in the case of less than 4 data points available. There are 4 kinds of interpolation formulas depended on which data points are available:

(1) for data points B and C only

$$f(X) = B \times (1 - X) + C \times X,$$

(2) for data points of A, B, and C

$$f_l(X) = B + X \times (0.5 \times (C - A) + X \times (0.5 \times (C + A) - B)),$$

(3) for data points of B, C, and D

$$f_r(X) = C + (1 - X) \times (0.5 \times (B - D) + (1 - X) \times (0.5 \times (B + D) - C)),$$

(4) and for data points of A, B, C, and D

$$f(X) = (1 - X) \times f_l(X) + X \times f_r(X).$$

Subroutine OUTPT(FLD, IYY, IA, IB, INY, JXX, JA, JB, JNX, KSIGT, NAME)

Purpose: Prints out a sample of the fields (on mesoscale grid).

On Entry: FLD(IYY,JXX): Array that holds the field on the mesoscale grid.

IA, IB: Initial and final sampling points in the I direction.

JA, JB: Initial and final sampling points in the J direction.

INY, JNX: Sampling intervals in the I and J directions.

KSIGT: Number of significant digits to be printed for each data point.

NAME: Hollerith identifier of variable to print.

Calls: Subroutine VTRAN.

Called by: Subroutine LATLON, TFUDGE, FINPRT.

Comments: The number of digits used to represent each printed value is assigned from KSIGT. The field is automatically scaled so that the largest value (absolute) will have KSIGT digits. Values more than KSIGT orders of magnitude less than the largest absolute value will be printed as a positive or negative zero. Data points may be excluded from the scaling process by setting the absolute values greater than 1.0×10^{30}. As the scaled data values are printed, labels showing the I and J indices are printed at the boundaries of the field. A caption is written above the data to identify the variable with a character string and to give the scaling power applied to the field. Subroutine VTRAN (not discussed) is used to transpose the data field array FLD, before and after the scaling and printing operations of OUTPT.

Subroutine PLTTER(GRIDIN, LNDXX, IMX, JMX, IYY, JXX, N)

Purpose: Plots the color-filled or non-color-filled map background, terrain height, and land-use fields, the mesoscale mesh, the rawinsonde stations, and the nested domain settings.

On Entry:

GRIDIN(IMX,JMX): The terrain height field. For the expanded coarse domain (IEXP = .TRUE.), it has the expanded grid dimension.

LNDXX(IMX,JMX): The land-use field. For the expanded coarse domain (IEXP = .TRUE.), it has the expanded grid dimension.

IMX,JMX: The dimension of the domain. If IEXP = .TRUE., they are the expanded grid dimensions for the coarse domain.

IYY,JXX: The dimension of the (non-expanded) domain.

N: The ID number of the domain.

On Exit: gmeta file TER.PLT.

Calls: Subroutine PLTTER calls subroutine MAPDRV, CONDRV, LABEL and many of the NCAR GKS subroutines.

Called by: Subroutine FINPRT.

Comments: In addition to the list of arguments, subroutine PLTTER gets additional information from the common blocks /MAPS/, /NESTDMN/, and /OPTION/. There is no rawinsonde station plot generated for Mercator projection. The following table shows the type of plots for the domains on the different nest levels.

Plot and Domain Level	1	2 to maxlvl-1	maxlvl
Map background	Yes	Yes	Yes
Terrain height	Yes	Yes	Yes
Land-use	Yes	Yes	Yes
Grid mesh	Yes	Yes	Yes
Rawinsonde stations	Yes	Yes	Yes
Nested domains setting	Yes	Yes	No

If IFILL = .TRUE., the color-filled map background, terrain height and land-use fields will be generated. The total number of the nest levels is $maxlvl$, the coarse domain is on level 1, and the finest domains are on the level $maxlvl$.

Subroutine RDLDTR(ID1, ID2, NREC, LREC, XLATI, XLONI, CENTER, GRDLTMN, GRDLNMN, XMAXLAT, XMINLAT, XMAXLON, XMINLON, NI, NIOLD, LND, NUNIT, LNDNAME, TER, IUNIT, TERNAME)

Purpose: Reads in the packed terrain height and land-use source data, unpacks and stores the data in core memory positioned by the PTRD11, PTRD12, PTRD13, and PTRD21 pointers in the common block /SPACE/ (see Section 5.1.2).

On Entry:

ID1: The number of land-use categories (=13).

ID2: The number of data points in a latitude circle.

NREC: The number of records in the dataset, one record contains one latitude line of data.

XLATI: The first integer latitude in the dataset.

	XLONI:	The first integer longitude in each latitude.
	CENTER:	The data resolution (°).
	XMAXLAT:	The maximum latitude of the search area.
	XMINLAT:	The minimum latitude of the search area.
	XMAXLON:	The maximum longitude of the search area.
	XMINLON:	The minimum longitude of the search area.
	NI:	The ID number of the source data file.
	NIOLD:	The previous NI value.
	LND:	A flag requesting to read in the land-use data (=1) or not (=0).
	NUNIT:	Read unit number for the land-use source data.
	LNDNAME:	The land-use source data MSS name.
	TER:	A flag requesting to read in the terrain height data (=1) or not (=0).
	IUNIT:	Read unit number for the terrain height source data.
	TERNAME:	The terrain height source data MSS name.
On Exit:	LREC:	The total number of the source data points within the search area.
	GRDLTMN:	The minimum longitude of the source data.
	GRDLNMN:	The minimum latitude of the source data.
	ZLON(IBLOCK):	Longitudes of the source data points within the search area.
	ZLAT(IBLOCK):	Latitudes of the source data points within the search area.
	HTOB(IBLOCK):	Terrain heights of the source data points within the search area.
	ZLAND(13,IBLOCK):	Land-use of the source data points within the search area.

Calls: Subroutine RDLDTR calls the subroutines NUNPACK.

Called by: Subroutine TERDRV.

Comments: In the subroutine TERDRV, the program dynamically allocates the core memory for arrays ZLON, ZLAT, HTOB, and ZLAND for the current working domain before entering this subroutine. In this subroutine, the common block /SPACE/ and the POINTER statement defines the four arrays. After exiting from this subroutine, these four arrays hold all the necessary source data to create the terrain height and land-use fields on the mesoscale grids (Section 5.1.2).

The program reads in one latitude of data at a time and checks if this latitude is within the search area. If not, the subroutine reads in the next latitude of data. If yes, the subroutine compares the longitude with XMAXLON and XMINLON to determine whether it should keep the data. Note that if the domain is across the date line (XMINLON $<$ -180), the conversion mentioned in Section 3.2.2 must be completed before the longitude comparison. The 1-degree, 30-, and 10-minute data (for both terrain height and land-use) are valid at the center of latitude-longitude box. Therefore, a shift equal to half of the resolution CENTER is incorporated with in the latitude and longitude before the comparison. For 5-minute and 30-second terrain-only data, the source data are located at the upper left corner of the latitude and longitude box. Therefore, no shift is needed.

During the process of reading the data, the minimum latitude and longitude, GRDLTMN and GRDLNMN, of the source data are found. There are two internal *msread* commands completed with CALL ISHELL in this subroutine to acquire the necessary source data files from the MSS onto the CRAY disk. If NI=NIOLD, it means that the data files have already been read in from the MSS, and the *msread* commands are skipped. With this internal *msread*, we may save some disk space. To make the TERRAIN program portable to another CRAY computer system, however, the NCAR internal *msread* commands are bypassed here and the source data files are read in from the job script. Therefore, all the arguments related to the internal *msread*, such as NIOLD, NUNIT, LNDNAME, IUNIT, TERNAME, in this subroutine would not be used with a version of the TERRAIN code that runs on a CRAY other than NCAR's.

If the users have their own source data and want to use it as the input to TERRAIN, the first step is to reconstruct the source data to the format described in Section 2.2. After the data is in the proper format, only minor modifications are needed to provide the necessary information such as the number of records, the first latitude and longitude of the data, the data resolution, etc., to the program (see the example using the Air Force 5-minute data in Appendix C).

Subroutine REPLACE(IYY, JXX, IINC, IYYF, JXXF, IINF, PFLAG)

Purpose:	Replaces the mother domain terrain heights at the overlapping grid points with its nest values.
On Entry:	IYY,JXX: The dimensions of the mother domain.
	IINC: The mother domain ID.
	IYYF,JXXF: The dimensions of the nest domain.
	IINF: The nest domain ID.
	PFLAG: The print switch. If .TRUE., print the terrain height fields after the replacement. If .FALSE., do not print.
On Exit:	The replaced terrain height and land-use for the mother domain are written out to disk with the same unit number.
Calls:	Subroutine REPLACE calls the subroutine OUTPT.
Called by:	The main program TERRAIN.
Comments:	The common block /OPTION/ is used in this subroutine. The locations of the south, north, west and east nest domain boundaries are determined based on the information from the header records of both "mother" and nest domains after reading in the data files from the disk. Then the values at the interior overlapping grid points of the nest domain are assigned to the corresponding grid points of the mother domain. The land-use values of the mother domain at those points whose heights are less than zero and greater than −1 are set to be 7 (water). Finally the terrain height and land-use for the mother domain are written back to the disk. This feedback procedure is needed only for two-way nested model applications (see Section 3.5.2).

Subroutine RFLP(YLAT, YLON, R, FLP)

Purpose:	Computes the distance from the pole to the latitude and longitude (YLAT,YLON) location and the angle between the central longitude and the longitude YLON (see Section 3.1).
On Entry:	YLAT: Latitude.
	YLON: Longitude.

On Exit: R: the distance from the pole to the latitude and longitude (YLAT,YLON) location.

FLP: the angle between the central longitude and the longitude YLON.

Calls: The subroutine calls the intrinsic functions SIN, COS, TAN, ALOG.

Called by: Subroutine XYOBSLL.

Comments: The common block /MAPS/ provides the information about the map background in this subroutine. In the northern hemisphere ($\phi_c < 0$), for Lambert conformal and polar stereographic projections, R < 0. For Mercator projection, R > 0. In the southern hemisphere ($\phi_c < 0$), it is vice versa.

Subroutine SETUP

Purpose: Initializes the constants and variables which can be determined based on the input namelist variables.

Calls: Subroutine SETUP calls the intrinsic functions SIN, COS, TAN, ALOG, and ALOG10.

Called by: The main program TERRAIN.

Comments: The common blocks /MAPS/, /NESTDMN/, /FUDG/, /TRFUDGE/, /OPTION/, /LTDATA/, and /HEADER/ are used in this subroutine. There are five namelists /MAPBG/, /DOMAIN/, /OPTN/, /FUDGE/, and /FUDGET/ in this subroutine. This subroutine has no dummy arguments. Please see Section 5.1.1 and 5.2.1 for details.

Subroutine SINT(XF, N, M, IFCT, N1STAR, N1END, N2STAR, N2END, IOR, N1OS, N2OS, XIG, XJG, IG0, JG0)

Purpose: Performs the semi-Lagrangian interpolation (Smolarkiewicz and Grell, 1992).

On Entry: XF(N,M,NF): 3-D array contains the mother domain 2-D fields (NF=9).

$$XF(N,M,1) = XF(N,M,2) = ... = XF(N,M,9)$$

	N:	Mother domain dimension in X-direction.
	M:	Mother domain dimension in Y-direction.
	IFCT:	Is set to be 1.
	N1STAR:	The start grid point in X-direction.
	N1END:	The end grid point in X-direction.
	N2STAR:	The start grid point in Y-direction.
	N2END:	The end grid point in Y-direction.
	IOR:	The nest ratio between the mother grid and the fine grid.
	N1OS:	The first dimension for the work arrays (=N).
	N2OS:	The second dimension for the work arrays (=M).
	XIG(NF):	9 numbers: 1/3, 0, -1/3, 1/3, 0, -1/3, 1/3, 0, -1/3
	YIG(NF):	9 numbers: 1/3, 1/3, 1/3, 0, 0, 0, -1/3, -1/3, -1/3
	IG0(N,M):	The index in X-direction; the values are not changed with the second dimension.
	JG0(N,M):	The index in Y-direction; the values are not changed with the first dimension.
On Exit:	XF(N,M,NF):	Holds the interpolated fine grid fields. The 9 values with the same first and second dimensions correspond to the 9 interpolated values at the 9 fine grid points in a mother grid box for the IOR = 3.

Calls: Subroutine SINT calls no other subroutines.

Called by: Subroutine EXAINT.

Comments: This subroutine can be used only with nest ratio equal to 3 (IOR = 3). Variable NF is set to be 9 in the parameter statement. The nested domain must be located at least 5 rows and columns inside its mother domain.

Subroutine SMTH121(TF, IYYF, JXXF, NP)

Purpose: Performs the 1-2-1 smoothing to remove primarily the $2\delta X$ waves from the field TF.

On Entry: TF(IYYF,JXXF): 2-D field to be smoothed.

IYYF,JXXF: The dimensions of the field TF.

NP: (NP-1) rows and columns near the boundary are not to be smoothed.

On Exit: TF(IYYF,JXXF): The smoothed field.

Calls: Subroutine SMTH121 calls no other subroutines.

Called by: Subroutine TERDRV.

Comments: See Section 3.4 for details.

Subroutine SMTHER(SLAB, IS1, IS2, NPASS, POINT)

Purpose: Spatially smoothes the 2-D field SLAB to damp the shorter wavelength components.

On Entry: SLAB(IS1,IS2): 2-D field to be smoothed.

IS1,IS2: The dimensions of the field SLAB.

NPASS: The number of smoother-desmoother passes.

POINT: POINT = 5HCROSS means the cross-point field, otherwise dot-point field.

On Exit: SLAB(IS1,IS2): The smoothed field.

Calls: Subroutine SMTHER calls no other subroutines.

Called by: Subroutine SMTHTR.

Comments: See Section 3.4 for more details.

Subroutine SMTHTR(SLAB1, IS1, IS2)

Purpose: Smoothes terrain height fields.

On Entry: SLAB1(IS1,IS2): The 2-D field to be smoothed.

	IS1,IS2:	The dimensions of the field SLAB.
On Exit:	SLAB1(IS1,IS2):	The smoothed terrain height field.

Calls: Subroutine SMTHTR calls the subroutine SMTHER.

Called by: Subroutine TERDRV.

Comments: In this subroutine, the ocean points (height < 0) are counted and set to be zero. After the two passes smoother-desmoother, set the terrain height to zero at the grid points with negative values and set the ocean points to -0.001.

Subroutine TERDRV(IX, JX, N)

Purpose: This is a driver routine to create the terrain height and land-use data on the mesoscale grids (Section 5.2.2 and Fig. 5.1b).

On Entry:	IX,JX:	The dimensions of the mesoscale domain. For the coarse domain IX and JX are the expanded grid dimension if IEXP = .TRUE..
	N:	The ID number of the mesoscale domain.

On Exit: The header record, terrain height, and land-use files for mesoscale domain N are created and written out to the disk with unit number 29+N.

Calls: Subroutine TERDRV calls the subroutines LATLON, MXMNLL, RDLDTR, CRTER, CRLND, SMTH121, SMTHTR, LAKES, BNDRY, and the CRAY dependent routines HPALLOC and HPDEALLC (see Section 5.1.2).

Called by: The main program TERRAIN.

Comments: In addition to the three arguments, the driver subroutine TERDRV gets most of the information from the common blocks /MAPS/, /NESTDMN/, /OPTION/, /LTDATA/, and /TRFUDGE/. The common block /SPACE/ and POINTER statement serve to allocate the core memory for the source data. The common block /HEADER/ is used to store the information related to the domains.

Subroutine TFUDGE(IYY, JXXX, IINC, IYYF, JXXF, IINF, PFLAG, MIXX, MJXX)

Purpose: Resets the boundary terrain height and land-use values for the nested domains (Section 3.5.1 and 5.3.3).

On Entry:

IYY,JXX:	The dimensions of the mother domain.
IINC:	The mother domain ID.
IYYF,JXXF:	The dimensions of the nest domain.
IINF:	The nest domain ID.
PFLAG:	The print switch. If .TRUE., print the terrain height fields after the adjustment. If .FALSE., do not print.
MIXX,MJXX:	The maximum dimensions of all the domains.

On Exit: The adjusted terrain height and land-use fields are written out to the disk again with the same unit numbers (29+N) for the domain N.

Calls: Subroutine TFUDGE calls the subroutines EXAINT, OUTPT.

Called by: The main program TERRAIN.

Comments: There is a check after reading in the data for the mother and nested domains from disk which is the necessary condition to perform the calculation in this subroutine. If the nest is too close to the boundaries (4 columns or 5 rows) of the mother domain, the program stops the execution. The maximum dimensions MIXX, MJXX are used in calling the subroutine EXAINT for the declaration of working arrays.

Subroutine VTRAN(FYX, IMAX, JMAX)

Purpose: Gets rid of lib=imslib on the segldr card, doing a simple vector slab transpose.

On Entry:

FYX:	The original 2-D field.
IMAX,JMAX:	The dimensions of the field FYX.

On Exit: FYX: The transpose of the original field.

Calls: Subroutine VTRAN calls no other subroutines.

Called by: Subroutine OUTPT.

Comments: The array is declared as an one-dimensional array FYX(IMAX*JMAX).

Subroutine XYOBSLL(YOBS, XOBS, HT, NOBS, IX, JX, DS, ICN, JCN, IRATIO, INEST, ICROSS)

Purpose: Calculates the indices (I,J) on the mesoscale grid from the latitudes and longitudes (see Section 3.1).

On Entry:

YOBS(NOBS):	Latitudes.
XOBS(NOBS):	Longitudes.
HT(NOBS):	Terrain height in meters.
NOBS:	The number of the data points.
IX,JX:	The dimensions of the coarse domain; they are the expanded grid dimensions if IEXP = .TRUE..
DS:	The coarse domain grid distance (km).
ICN,JCN:	The location of the lower left corner of the nested domain in the coarse domain. They are real numbers.
IRATIO:	The ratio of the coarse grid size to the nest grid size.
INEST:	Equal to domain ID number - 1. For coarse domain INEST=0, and for nested domain INEST > 0.
ICROSS:	=0 for dot point field, =1 for cross point field.

On Exit:

YOBS(NOBS):	The I indices in Y-direction.
XOBS(NOBS):	The J indices in X-direction.
HT(NOBS):	Terrain heights in 100 meters.

Calls: Subroutine XYOBSLL calls the subroutine RFLP and the intrinsic function SIN and COS.

Called by: Subroutine TERDRV.

Comments: In addition to the list of arguments, subroutine XYOBS gets information from the common blocks /MAPS/ and /OPTION/. Note that IRATIO is the ratio of the coarse grid size to the nested grid size, not the mother grid size to the nested grid size, and ICN and JCN are the location of nested grid point (1,1) in the coarse domain, not in the mother domain, so they may be real numbers.