14 MM5 Version 3

14.1 Introduction

Version 3 of the MM5 modeling system is being released in the summer of 1999. This version has come about as the result of several large changes that affect the modeling system. (I) The addition of a land-surface model that requires several more input datasets than standard model applications. (II) The use of a new, more efficient, file format for all the modeling system files. (III) The change of representation of date in the modeling system in order to be Y2K compliant as well as enabling date manipulation down to seconds on 32-bit machines. (IV) The removal of the hydrostatic option, and consequent streamlining of the code. (V) The re-writing of the solver routine for better efficiency. (VI) The need to move to a new set of pre-processing programs that make use of generalized data input formats, and many of which now use Fortran 90.

14.2 Version 3 File Format

The Version 1 header has been in use since 1994. During the period since, several ways to improve on it became apparent, and these are put into effect with Version 3. Firstly, Version 1 files are unnecessarily long because before every time period there are about 3.5 Mbytes of header record, only a small fraction of which is used. Version 1 had a first effort at a format that was self-describing, in that the header contains information about all the fields in the file. However it was made a little awkward to use because this information on each field is buried in the header. When adding a new field, the location and order in the header had to be consistent, as well as updating the total number of fields, so the files were difficult to manipulate.

Version 3 improves in both respects: the files are shorter, and it becomes easier to add or retrieve selected fields. The length of the header is reduced. The header still has 20 sections (second index which indicates program name), but now only 50 integers, and 20 reals, together with their 80-character descriptions, are in each section. This makes the header size a little over 100 kbytes. Moreover, there is only one header which is at the beginning of the file, but there is enough generality in the format to allow more headers at other times such as at the beginning of a restart run or when a nest moves. The header still contains information about the preprocessor options, and domain characteristics and location. However it no longer contains 1-dimensional fields, such as sigma or pressure levels, nor information about what is in the rest of the file. Version 3 introduces the concept of a sub-header, a 1-record description directly ahead of each field. This description includes information on the name, dimensionality, index order, index range, size and time of the following field. Flags in the file indicate whether to read a “big header’’, or sub-header and field, or whether it is the end of a time period. It can be seen that it is easy to insert a field as long as it is accompanied by a relevant flag and sub-header. It is also easy to search for a given field by reading sub-headers until a match is found then reading the following field.

An MM5 Version 3 modeling system output file contains the following records:

(first time period)
big header flag (integer value of 0)
big header
sub-header flag (integer value of 1)
sub-header
field
sub-header flag (integer value of 1)
sub-header
field
sub-header flag (integer value of 1)
sub-header
field
....
....
end-of-time-period flag (integer value of 2)
(second time period)
sub-header flag (integer value of 1)
sub-header
field
sub-header flag (integer value of 1)
sub-header
field
sub-header flag (integer value of 1)
sub-header
field
....
....
end-of-time-period flag (integer value of 2)
(and so on ....)

No particular order of fields is assumed, other than that they are chronologically grouped. When reading files in the modeling system, each field has to be read and matched to an expected 8-character name before being assigned to a variable in the program. Note that 1D, 2D, and 3D fields could be mixed, but that the sub-header gives enough information to assign an appropriate array to the read statement.

Thus, a simple read program would look like:

```
10      continue
        read (input_unit, end=900) flag
        if (flag.eq.0) then
        read (input_unit) big header
        go to 10
        else if (flag.eq.1) then
```

```
          read (input_unit) sub header
          read (input_unit) field
          go to 10
          else if (flag.eq.2) then
          print *, 'end of time period'
          go to 10
          end if
900       continue
```

In addition, the boundary condition file in V1/V2 system is separated into two files with one containing only the lateral boundary arrays, and the other the substrate temperature array (lower boundary condition file). Both files are brought to the same file structure as the rest of modeling system output files.

Sigma-level data in V3 are no longer coupled (e.g. multiplied by p*).

The units in V3 header are mostly MKS, e.g. pressure unit is Pascal, grid distance is in m, etc..

14.2.1 Big header

The big header has four 2-D arrays similar to that in the V1/V2 system, which we refer to in the V3 modeling system programs as

BHI,BHR,BHIC,BHRC

and the dimensions of these arrays are

BHI(50,20),BHR(20,20),BHIC(50,20),BHRC(20,20)

where BHI is an integer array, and BHIC is the companion array that contains the description of what is in BHI. Similarly BHR is a real array, and BHRC contains the description of what is in BHR.

The first value in the header, BHI(1,1), still represents data types. But there are some changes as shown below:

BHI(1,1)	**Data Types**
1	Terrain
2	Regrid
3	Rawins
4	Rawins' surface analysis
5	Model initial condition file
6	Model lower boundary condition file (substrate temp array)
7	Model lateral boundary condition file
8	Interpolated model output on pressure levels
11	Model output

MM5 model output actually occupies header locations 11 through 16.

14.2.2 Sub header

A sub-header contains the following information:

ndim, start_index(4), end_index(4), xtime, staggering, ordering, current_date, name, units, description

where

ndim:	integer	dimension of the field (integer)
start_index:	integer(4)	starting indices of the field array (generally 1's)
end_index:	integer(4)	ending indices of the field array (generally IX, JX, KX, and 1) (the fourth dimension is not yet used)
xtime:	real	the integration or forecast time for this field
staggering:	char(4)	whether the field is at dot or cross point (character C or D)
ordering:	char(4)	the order of the field array dimension (4-character string with the following values: YX: 2-D field, with array dimensioned by (IX,JX) with IX in Y direction YXP: 3-D field, pressure data dimensioned by (IX,JX,KXP) YXS: 3-D field, sigma data dimensioned by (IX,JX,KXS) YXW: 3-D field, sigma data dimensioned by (IX,JX,KXS+1) (e.g. vertical motion in MM5) XSB: 3-D field, containing north and south boundary arrays, dimensioned by (JX,KXS,5) YSB: 3-D field, containing west and east boundary arrays, dimensioned by (IX,KXS,5) P: 1-D field, pressure level array S: 1-D field, sigma level array
current_date:	char(24)	24-character representation of date valid for this field
name:	char(9)	8-character field name (kept the same as in Version 1/2 system)
unit:	char(25)	25-character unit description
description:	char(46)	field description (kept mostly the same as in Version 1/2 system)

14.3 Version 3 Date String

The new header and modeling system have eliminated the use of the 8-digit MDATE (yymmddhh) representation of time that is a standard part of the previous MM systems, primarily to be Y2K compliant. In its place is a 19- or 24-character date string, referred to as CDATE in the model, of the form "yyyy-mm-dd_hh:mm:ss.xxxx" where the last 5 characters are optional and represent ten thousandths of a second if such accuracy is needed. For example, the MDATE value of

99061812

is now

1999-06-18_12:00:00.0000

This allows for accurate and compact representations of the time that can easily be converted to integers by standard Fortran read statements. The charater string representation of date also allows the modeling system programs to manipulate output files at time intervals less than an hour. For example, programs Interp and Graph can now process and plot output from the model at time intervals down to seconds.

14.4 Land-Surface Model

The introduction of the Oregon State University / NCEP Eta Land-Surface Model (LSM) in MM5 (Chen and Dudhia 1999) has required that the modeling system be enhanced to allow for additional inputs and outputs. The land-surface model is capable of predicting soil moisture and temperature in four layers (10, 30, 60 and 100 cm thick), as well as canopy moisture and water-equivalent snow depth. It also outputs surface and underground run-off accumulations. The LSM makes use of vegetation and soil type in handling evapotranspiration, and has effects such as soil conductivity and gravitational flux of moisture. In MM5 it may be called instead of the SLAB model in the MRF PBL, taking surface-layer exchange coefficients as input along with radiative forcing, and precipitation rate, and outputting the surface fluxes for the PBL scheme. This scheme uses a diagnostic equation to obtain a skin temperature, and the exchange coefficients have to allow for this by use of a suitable molecular diffusivity layer to act as a resistance to heat transfer. In the future we expect to add several more land-surface models, such as BATS, PLACE, RUC's LSM and CCM3's LSM, and to couple them to more surface-layer/ boundary-layer schemes.

14.4.1 LSM Requirements in Model Pre-Processing System

To operate with the LSM, MM5 requires several additional input fields. The Version 3 Terrain program provides an annual-mean surface temperature adjusted to terrain elevation, a monthly climatological vegetation fraction, dominant soil type, and dominant vegetation type in each grid cell.

The Regrid program provides soil moisture and temperature at various depths, water-equivalent snow depth, sea ice, and optionally canopy moisture. Currently used sources of such fields are the NCEP/NCAR Reanalysis, and the Eta AWIP analyses (for the US only). These input data are illustrated in Fig. 14.1.

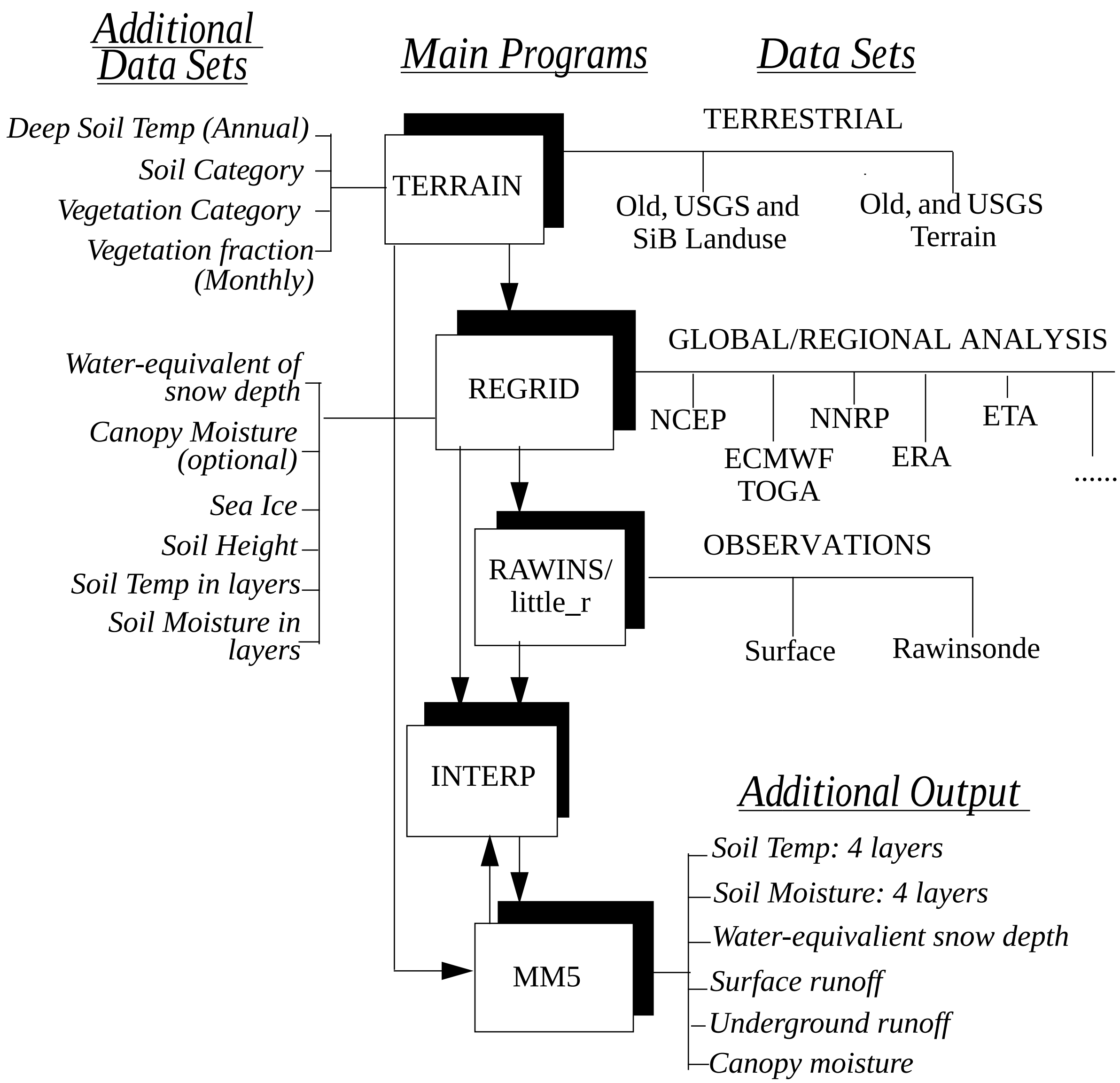

Fig 14.1 The MM5 Version 3 modeling system flow chart with input data to and output data from LSM shown.

14.5 Major changes in Version 3 Pre-Processing Programs

14.5.1 Terrain

Program Terrain has been significantly updated for MM5 Version 2 Release-2-12, and 13. Most of the changes are described in Chapter 4 of this volume. The additional changes for V3 are

- Constant model terrestrial fields, such as latitude/longitude, map scale factors, Coriolis parameter, are now computed and output in Terrain program instead in Version 2's Regrid/ Datagrid programs.
- Option to use high-resolution vegetation dataset generated land-water mask to set coastlines (IFEZFUG=.F.), instead of NCAR Graphics (IFEZFUG=.T.).
- Use of script variable LandSurface to choose whether to create additional data for use in MM5 LSM.

14.5.2 Regrid

Some user-level changes in V3 REGRID are listed below:

- The "_nml" is gone from the namelist variables
- The dates used to be character strings, with the interval in hours.
  The dates are now separate integers, with the interval in seconds.

```
start_year                      =  1996
start_month                     =    07
start_day                       =    12
start_hour                      =    12
end_year                        =  1996
end_month                       =    07
end_day                         =    13
end_hour                        =    12
interval                        = 21600 /
```

- The terrain file is named in the namelist file:

```
terrain_file_name               = '/headache2/gill/DATA_V3/terrainv3.d2'
```

- For RAWINS users, there is a print flag to aid in setting up the parameter statements:

```
print_f77_info                  = .TRUE.
```

14.5.3 Rawins

The Version 3 Rawins is very similar to the latest release of V2 Rawins which is becoming a one-source program for both Cray and workstations.

14.5.4 Interp

The V3 INTERP program will be divided into three separate programs according to its functions.

Front-end Interp *(interpf)*

The V3 INTERP program is a f90 source code that handles only the standard front-end tasks on a single domain for model input generation. The code has been modified by NOAA/FSL, AFWA and by NCAR. The direct access files have been replaced by dynamically allocated memory, which implies that INTERP uses significantly more memory than previous versions. Because the INTERP program dynamically allocates the required memory, there is no need to re-compile the code for different sized horizontal configurations, input vertical levels or output vertical levels. The MM5 v3 code expects three types of input files: initial condition of the meteorological fields, lateral boundary conditions of the meteorological fields and the lower boundary condition (just the reservoir temperature currently). These files are named MMINPUT_DOMAINx, BDYOUT_DOMAINx and LOWBDY_DOMAINx, where "x" is the grid identifier bhi(13,1). The INTERP code must be run multiple times to generate more than a single domain of input for the model.

The files from the INTERP program are all in the v3 format, including the BDYOUT_DOMAINx file. The MMINPUT_DOMAINx file does not have coupled output, but for convenience inside the model, the BDYOUT_DOMAINx file is coupled with p* (kPa). The boundary file no longer has p* or p* tendencies in the data set.

There is no hydrostatic option in INTERP, consistent with removing this capability from the model in the v3 release. The user may have the output for the initial condition MMINPUT_DOMAINx contain only a specified number of time slices to permit analysis nudging in the model. An option for diagnosing mixing ratio with respect to water only or water and ice is available. The choice of using surface data in the vertical interpolation is now located in the namelist. Users may also spread the surface data through several sigma layers in the vertical interpolation.

All of the date computations are Y2K compliant. The date information input to the program through the namelist is split into multiple integers (year, month, day, hour). The only input file other than the namelist is the first-guess file that is specified in the namelist.

The code is built with a 2-level Makefile. In the top directory, the user types # make XXX, where "XXX" is cray, dec, hp, ibm or sgi. The code does not currently work on the Sun architecture.

The INTERP namelist file is named "namelist.input" and must be in the user's current working directory. The namelist has comments (anything after a "!" on a line). This is part of the f95 standard. SGI and Cray machines do not yet support this capability, so the comments need to be removed when running on one of those platforms. IBM and Compaq support comments.

The namelist is broken into several records. All of the user defined information to run the INTERP program is provided by namelist options.

The new INTERP program accepts only a single input file and does no horizontal interpolation. Only a first-guess pressure-level data set may be input to INTERP. In the MM5 system, this file is either from regridder or from RAWINS/little_r. The file name may contain directory information.

```
&record0
 input_file     = '/mmmtmp/mesouser/v3/rawins.out.v3' /    ! Input file name
```

The dates are specified as components: a 4-digit year, 2-digit month, 2-digit day and 2-digit hour.

Both the starting and ending time of the INTERP processing are required. The time interval is specified in seconds.

```
&record1
 start_year     =  1993                                   ! The starting and
 start_month    =    03                                   ! ending dates to
 start_day      =    13                                   ! process
 start_hour     =    00                                   !
 end_year       =  1993                                   !
 end_month      =    03                                   !
 end_day        =    14                                   !
 end_hour       =    00                                   !
 interval       = 43200 /                                 ! time difference (s)
```

The next record deals loosely with vertical information. The sigma data are to be filled in as full levels, in the bottom-up orientation. The user must have 1.0 as the first value, 0.0 as the last value, and a monotonically decreasing distribution between these two anchoring points.

The pressure at the model lid is given in Pa. The switch "isfc" allows the user to adjust how many layers are required to use the surface data from the input. Isfc=0 forces the traditional technique where the two surrounding pressure levels are interpolated to the sigma surface. Isfc=1 directly injects the surface data as the lowest sigma level. Isfc=N, where N > 1, uses the surface data for all interpolations to the N sigma layers closest to the ground.

```
&record2
 sigma_f_bu     = 1.00,0.99,0.98,0.96,0.93,0.89,          ! full sigma, bottom-up,
                  0.85,0.80,0.75,0.70,0.65,0.60,          ! start with 1.0, end
                  0.55,0.50,0.45,0.40,0.35,0.30,          ! with 0.0
                  0.25,0.20,0.15,0.10,0.05,0.00           !
 ptop           = 10000                                   ! in Pa
 isfc           = 0 /                                   ! # sigma levels to spread
                                                          ! surface information
```

The MM5 base state is built in INTERP. The user supplies the reference sea-level pressure (1.e5 Pa), the temperature lapse rate (50 d(T)/d(ln P)), and the reference sea-level temperature (K).

```
&record3
 p0             = 1.e5                         ! base state sea-level pres (Pa)
 tlp            = 50                          ! base state lapse rate d(T)/d(ln P)
 ts0            = 275 /                        ! base state sea-level temp (K)
```

Several user options are available. Options exists to remove the integrated mean divergience, to force the interpolation scheme to use the meteorological surface field (u, v, T, RH), and the choice of computing specific humidty with respect to water (T) or ice (F).

```
&record4
 removediv     = .TRUE.                     ! T/F remove integrated mean divergence
 usesfc         = .TRUE.                     ! T/F use surface data
 wrth2o         = .TRUE. /                   ! T/F specific humidity wrt H2O
```

For users not doing any grid point nudging, the option is available to simply output a single time for the initial condition. Ifdatim=N says to output N time periods of initital condition data. This option does not effect the period through which the lateral boundary file or the lower boundary file are valid. This option simply reduces the amount of data that is written to the initial condition file. Without analysis FDDA, ifdatim=1 is adeqate for the model.

```
&record5
```

```
 ifdatim        = 3 /                    ! # of IC time periods to output
```

One-Way Interp *(nestdown)*

This is a backend option in V2 Interp. This is rewritten in Fortran 90, and the program will create one-way nest input files for MM5.

Back-end Interp II: *(interpb)*

This is another backend option in V2 Interp. This will be rewritten in Fortran 90, and the program will do simple one-domain vertical interpolation from sigma level to pressure level, including generating data in the format of Regrid output (the ‘First-guess’ option).

This backend program will be available at a later time

14.5.5 Graph

The V3 Graph is compatible with the rest of the V3 modeling system programs. It can now plot output data from the V3 modeling system at time intervals of either hours, minutes or seconds. The only change the user sees is in the *g_plots.tbl* is the time line. This in the old *g_plots.tbl*

```
TIME LEVELS (MDATE FORMAT YYMMDDHH): FROM 93031300 TO 93031312 BY 6
```

is changed to

```
TIME LEVELS: FROM 1993-03-13_00 TO 1993-03-13_12 BY 6
```

One can also specify

```
TIME LEVELS: FROM 1993-03-13_00:00 TO 1993-03-13_12:00 BY 360
```

i.e. specifying the time in terms of minutes. Or

```
TIME LEVELS: FROM 1993-03-13_00:00:00 TO 1993-03-13_12:00:00 BY 21600
```

i.e. specifying the time in terms of seconds.

14.6 Other Changes in MM5 Model

Apart from adding the LSM module into V3 MM5, there are other changes made in the model. These are:

Use Fortran OPEN to access input files and write output files. This has resulted in a great simplification in using the deck. There are no script variables to set in a mm5.deck (Cray deck may be an exception). Now to run a model interactively, the model expects to see the following files in the Run directory:

MMINPUT_DOMAIN1

BDYOUT_DOMAIN1
LOWBDY_DOMAIN1
TERRAIN_DOMAIN2 or MMINPUT_DOMAIN2 (if running a nest)
RESTART_DOMAIN1 (if it is a restart run)
RESTART_DOMAIN2 (if it is a restart run, and if running a nest)
MM5OBS_DOMAIN1 (if doing observational nudging on domain 1)
....

The output from the model, also appearing in the Run directory, has a name, rather fort.41, and fort.42:

MMOUT_DOMAIN1
MMOUT_DOMAIN2 (if you run two domains)
SAVE_DOMAIN1 (if IFSAVE = .T.)
SAVE_DOMAIN2 (if IFSAVE = .T., and if running a nest)

Most files use the same Fortran unit number as they did in V2. The new lower boundary file makes use of Fortran unit number 8, which was no longer needed because the removal of *ehtran* file among the input files.

- The *ehtran* is removed from the tar file, and the array is now initialized in the model.

- There are a few changes in the namelist. For example, the forecast length TIMAX is now at the top of the namelist for easy editing.

- A new namelist variable, BUFFRQ, is added. This defines the time interval (in minutes) the output files are splitted. For example, if BUFFRQ=720., the model will create an output file every 12 hours (applies to history file only). The splitted output file will have file extension _00 for hour 0, and _01 for 0 - 12 hours, and _02 for 13 - 24 hours, etc.. To turn the option off, set BUFFRQ < TAPFRQ. In addition, the output files from a restart run will also have an file extension (whether you choose to split files or not): _01 for first restart run, and _02 for the second, etc..

14.7 Utility Programs

A number of utility programs will be made available to users.

V2-to-V3 converter:

This utility will convert all V2 modeling system output to that of V3, including boundary condition file. (Currently available.)

V3-to-V2 converter:

This utility will convert all V3 modeling system output to that of V2.

readv3.f

This utility will read all V3 output files, print the header, and a value from each field. (Currently available.)