7 INTERP

7.1 Purpose

The INTERP program handles the data transformation required to go from the analysis programs to the mesoscale model. This can entail vertical and horizontal interpolation, diagnostic computation, and simple data reformatting. When INTERP takes DATAGRID or RAWINS as input to generate a model initial and boundary file, this is referred to as a front-end job. Using INTERP to place forecast data on pressure surfaces and horizontally interpolate forecast data to finer mesh for a 1-way run is a back-end job.

The INTERP program can run on both a Cray and workstations.

7.2 Standard Front-end Job

- input RAWINS or DATAGRID
- pressure level Qv for Psfc
- interpolate variables from pressure coordinates to hydrostatic σ
- u, v, RH: linear in pressure
- theta: linear in *ln* pressure
- remove integrated mean divergence
- compute substrate temperature

- compute base state
- compute w
- re-interpolate u, v, t, Qv
- compute perturbation pressure

- save current data for boundary file
- output interpolated data for initial and boundary conditions

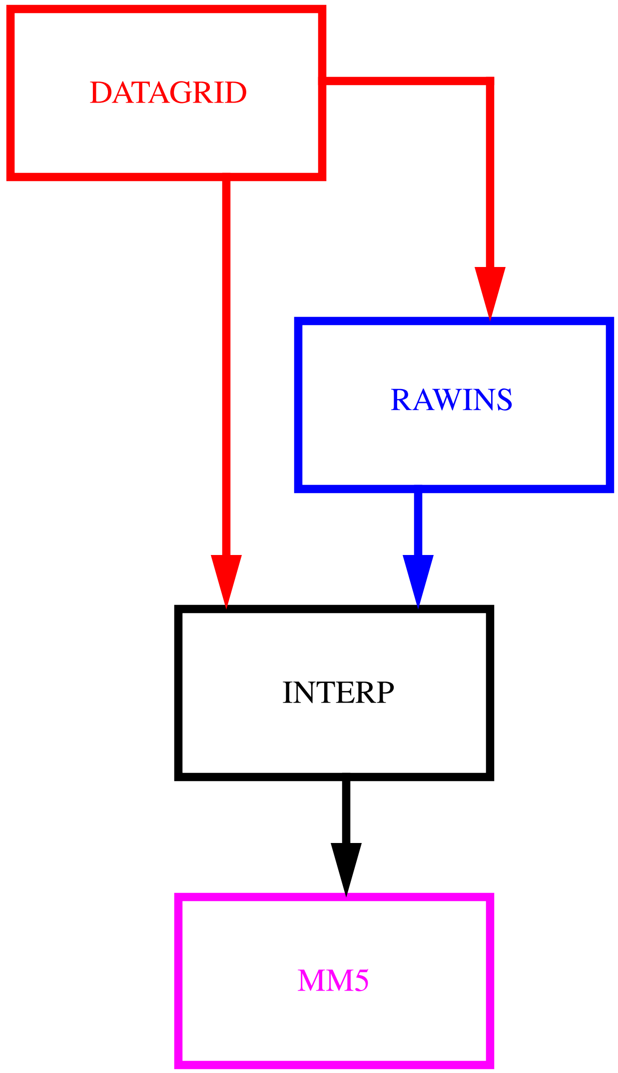

Fig 7.1 MM5 modeling system flow chart for standard front-end job.

7.3 Surface Pressure Computation

1. first guess for representative T, p 100 hPa above surface

$$P_{sfc} = P_{slv}\left(\frac{P_{slv}}{850}\right)^{-TER/H_{850}} \tag{7.1}$$

2. extrapolate Tslv

$$T_{slv} = T_{100-up}\left(\frac{P_{slv}}{P_{100-up}}\right)^{\gamma} \tag{7.2}$$

$$\gamma = \frac{ln\frac{T_{850}}{T_{700}}}{ln\frac{850}{700}}, \text{ if } 700 \le P_{100-up} \le 850 \tag{7.3}$$

3. corrected Tsfc

$$T_{sfc} = T_{slv} - \gamma_s \times \mathrm{TER} \tag{7.4}$$

4. use mean temperature underground to estimate surface pressure

$$P_{sfc} = P_{slv} exp\left\{\frac{\text{-TER} \times g}{R} / \left[\frac{1}{2} \times (T_{sfc} + T_{slv})\right]\right\} \tag{7.5}$$

7.4 Hydrostatic Vertical Interpolation

The process of going from pressure levels to the σ coordinate requires only strictly bounded interpolation. Since the σ coordinate is defined to be contained within the maximum and minimum pressure, no extrapolations are required. A generated surface field for DATAGRID is available as a coding option inside INTERP. Vertical interpolation uses linear techniques exclusively, typically linear in pressure or linear in *ln* pressure.

Hydrostatic pressure is defined as

$$P_{ijk} = \sigma_k \times p^*_{ij} + P_{top} \tag{7.6}$$

where σ is a 1-D vertical coordinate, σ=1 at the ground, σ=0 at the model lid; p* is a 2-D field of surface pressure minus a constant (Ptop); and Ptop is the constant pressure at the model lid.

$$\alpha_\sigma = \frac{\alpha_{P_A}(P_B - P_\sigma) + \alpha_{P_B}(P_\sigma - P_A)}{P_B - P_A} \tag{7.7}$$

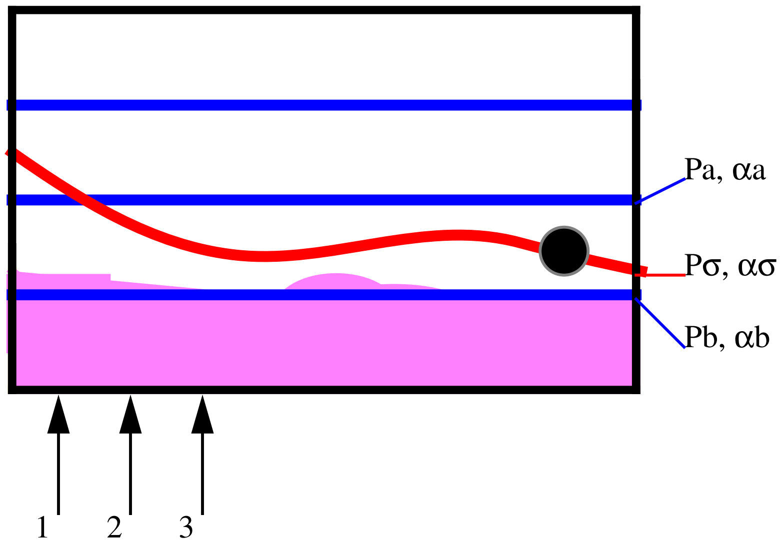

Fig. 7.2 A vertical profile of a σ surface cutting through several isobaric layers. The heavy dot is the location on the σ surface for which a vertical interpolation is requested. The arrows (labeled 1 through 3) represent consecutive grid points that use three separate surrounding layers along a σ surface.

7.5 Integrated Mean Divergence Removal

Removing the integrated mean divergence allows the model to begin with a smaller amount of initial condition noise that the analysis contains. Given the average upper-air station separation, the reasonableness of the high frequency, column-averaged vertical motion is spurious at best.

1. pressure weighted u, v on each σ

$$PU_{ijk} = p^*_{ij} \times u_{ijk},\ PV_{ijk} = p^*_{ij} \times v_{ijk} \tag{7.8}$$

2. vertically average p*u, p*v

$$U_{integ_{ij}} = \sum_k PU_{ijk} \times \Delta\sigma_k,\ V_{integ_{ij}} = \sum_k PV_{ijk} \times \Delta\sigma_k \tag{7.9}$$

3. divergence of vertically-averaged pressure-weighted wind [m is the map scale factor for dot (D) and cross (X)]

$$DIV_{ij} = m_X^2\left[\frac{\Delta(U_{integ_{ij}}/m_D)}{\Delta x} + \frac{\Delta(V_{integ_{ij}}/m_D)}{\Delta y}\right] \tag{7.10}$$

4. solve for the velocity potential, with assumed boundary conditions

$$\nabla^2\chi_{ij} = DIV_{ij},\ \chi_{ij_{boundary}} \equiv 0 \tag{7.11}$$

5. mean divergent wind components

$$U_{DIV_{ij}} = \frac{m_D}{p^*}\frac{\Delta\chi}{\Delta x},\ V_{DIV_{ij}} = \frac{m_D}{p^*}\frac{\Delta\chi}{\Delta y} \tag{7.12}$$

6. vertical weighting

$$require:\ \sum_k w_k \times \Delta\sigma_k = 1, \tag{7.13}$$

$$presently:\ w_k = 2(1-\sigma_k) \tag{7.14}$$

7. corrected wind components

$$U_{corrected_{ijk}} = u_{ijk} - U_{DIV_{ij}} \times w_k,\ V_{corrected_{ijk}} = v_{ijk} - V_{DIV_{ij}} \times w_k \tag{7.15}$$

7.6 Base State Computation

The base state for the MM5 model is constructed from several constants prescribing a surface level temperature and pressure, a temperature profile, and analytic expressions for a reference pressure and the height of the non-hydrostatic σ surfaces. Only the constants are required by the modeling system as user input.

1. constants

- P_{00}: reference sea level pressure
- T_{s0}: reference sea level temperature
- A: reference temperature lapse rate

- P_{TOP}: reference pressure at model top

2. reference p*

$$P_{s0} = P_{00}\ exp\left\{\frac{-T_{s0}}{A} + \left[\left(\frac{T_{s0}}{A}\right)^2 - 2g\frac{TER}{A \times R}\right]^{\frac{1}{2}}\right\} - P_{TOP} \tag{7.16}$$

3. reference pressure 3-D

$$P_0 = P_{s0} \times \sigma + P_{TOP} \tag{7.17}$$

4. reference temperature 3-D

$$T_0 = T_{s0} + A\,ln\frac{P_0}{P_{00}} \tag{7.18}$$

5. reference height

$$z = -\left[\frac{R \times A}{2g}\left(ln\frac{P_0}{P_{00}}\right)^2 + \frac{R \times T_{s0}}{g}ln\frac{P_0}{P_{00}}\right] \tag{7.19}$$

This provides a fixed (in time) height for each σ surface, since each i,j,k location is a function of the fixed σ values and the terrain elevation.

7.7 Initialization of Nonhydrostatic Model

INTERP first generates a hydrostatic input file on the hydrostatic sigma levels which are based on actual surface pressure, not reference pressure. To initialize the nonhydrostatic model a further small vertical interpolation is needed to the nonhydrostatic sigma levels. This involves first calculating the heights of the hydrostatic levels, then doing a height interpolation of u, v, T and q to the nonhydrostatic levels.

While sea-level pressure, u, v, T and q are known from the input datasets, the nonhydrostatic model requires two more variables to be initialized.

- Vertical velocity (w) is simply calculated from the pressure velocity (ω) obtained by integrating horizontal velocity divergence vertically while still on the hydrostatic sigma levels. Divergence removal has already ensured that this integration will give no vertical motion at the top or bottom of the model domain. This ω is then interpolated to the nonhydrostatic levels and converted to w ($w=-\omega/\rho g$). In practice, the results are not sensitive to whether w is initialized this way or equal to zero.

- Pressure perturbation (p′) has to be initialized to give a hydrostatic balance. Once virtual temperature is known on the nonhydrostatic model levels, the model's vertical velocity equation in finite difference form is used with the acceleration and advection terms set to zero. This leaves a relation between $T_v(z)$ and the vertical gradient of p′. Given the sea-level pressure, p′ at the lowest sigma level can be estimated, and then given the profile of virtual temperature vertical integration gives p′ at the other levels. This balance ensures that the initial vertical acceleration is zero in each model column.

7.8 Shell Variables

All of the MM5 system job decks for Cray are written as C-shell executables. Strict adherence to C-shell syntax is required in this section.

Table 7.1: INTERP deck shell variables.

C-shell Variable Name	Options and Use
ExpName	location of MSS files, keep same as used for deck generating input file for this program
RetPd	time in days to retain data on MSS after last access
HYDROsw	0 (hydrostatic) or 1 (non-hydrostatic), this is used for information inside INTERP as well as directory structure on the MSS
ForBsw	FRONT (front-end) or BACK (back-end) job
NESTsw	NEST or NoNEST depending on whether INTERP is supposed to generate more than a single domain for MM5 with this run
Host	full rcp path to a remote machine where your own INTERP source code can be acquired
ID	Domain identifier for the processed grid
InData	MSS path name for the input analysis/forecast
InTer	MSS name for the nest TERRAIN input

7.9 Parameter Statements

The PARAMETER statements used in the deck supporting the INTERP code are inserted directly into the FORTRAN source program. Standard syntax for FORTRAN 77 is required.

Table 7.2: INTERP deck domain parameter statements.

PARAMETER	Explanation
MAXNES	integer, total number of domains for INTERP to process
NLNES	integer, maximum depth of domain nesting, most coarse grid processed has NLNES=1, the next level of subdomain has NLNES=2, and each succeeding deeper nest level increases NLNES similarly

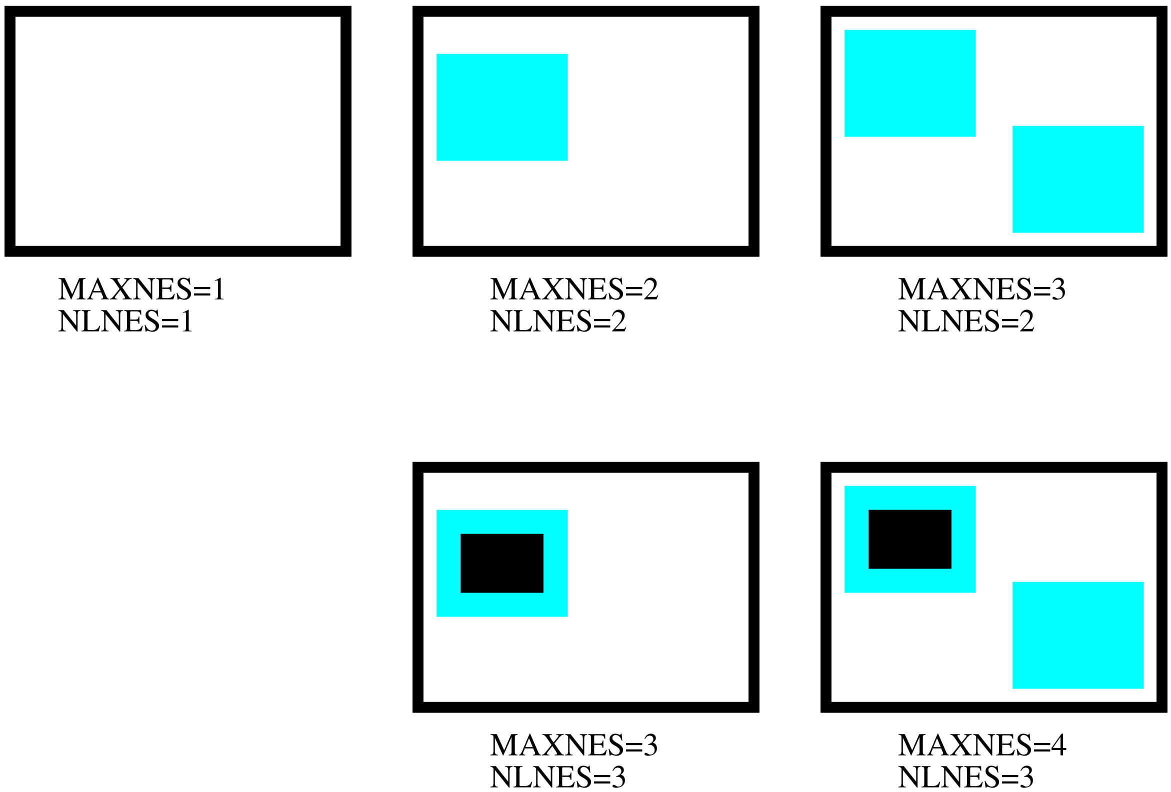

Fig. 7.3 The MM5 modeling system's definition of MAXNES and NLNES. The total number of domains (the rectangualr areas) is equal to MAXNES. The most deeply nested domain (there may be more than one domain at the same level) prescribed NLNES.

Table 7.3: INTERP deck horizontal and vertical dimension parameter statements.

PARAMETER	Explanation
IMAX, JMAX	integer, maximum horizontal domain size (grid points) where the first node is defined as (1,1), IMAX refers to the I-direction (y-axis) and JMAX refers to the J-direction (x-axis), these values include the size for the expanded domain if DATAGRID files are input
KXS	integer, number of half-σ layers (same as vertical dimension in MM5), this is one less than the number of full-σ levels specified in the INTERP namelist
KXP	integer, number of pressure levels (EXCLUDING the surface and sea level) of the DATAGRID or RAWINS data, count only isobaric surfaces on which an analysis was generated
KMAX	integer, max (KXS , KXP)

Table 7.4: INTERP deck machine-dependent parameter statements.

PARAMETER	Explanation
IMACHWORD	integer, used for opening direct access files
SMALLRES	floating, used to define criteria for convergence

7.10 FORTRAN Namelist Input File

Most of the available options for the INTERP code that do not require compile-time attention are handled through the namelist input file. All of the MM5 system programs have this structure, the files are typically referred to as the local input files. Since this file is a FORTRAN namelist (which is not FORTRAN 77 standard), syntax is very architecture specific. There are three namelist records: LOCMIF, NHYDMIF, and NONSTAN. Only LOCMIF is required to be filled out for every job submission. NHYDMIF is required for front-end jobs generating a non-hydrostatic data set for MM5. NONSTAN is used when modifying some INTERP defaults or creating non-standard data sets.

Table 7.5: INTERP deck LOCMIF namelist values.

Namelist Variable	Description
NEWCOORD	array of reals; for a front-end job this is a bottom-up array of full σ-levels (starting with 1.0 and ending with 0.0), there should be one more level than is specified by parameter KXS
STANDARD	logical; if this is DATAGRID or RAWINS going into MM5 then TRUE
BEGTIME	double precision; 10 digit (YYMMDDHHmm) date and time identifier specifying when the INTERP processing is to initiate
ENDTIME	double precision; 10 digit (YYMMDDHHmm) date and time identifier specifying when the INTERP processing is to complete
TIMINT	real; time period in minutes specifying increment between analysis times

Table 7.6: INTERP deck LOCMIF namelist values.

Namelist Variable	Description
NESTIX	integer; domain size in the I-direction (y-axis), each size is specified in grid units specific to the domain, the lower left point of each domain has the origin of the grid defined as (1,1) each domain is separated by a comma, up to 9 domains may be used
NESTJX	integer; domain size in the J-direction (x-axis)
NESTI	integer; starting location of the lower left grid point in the I-direction of each domain in the mother domain (given in mother domain grid units)
NESTJ	integer; starting location in the J-direction

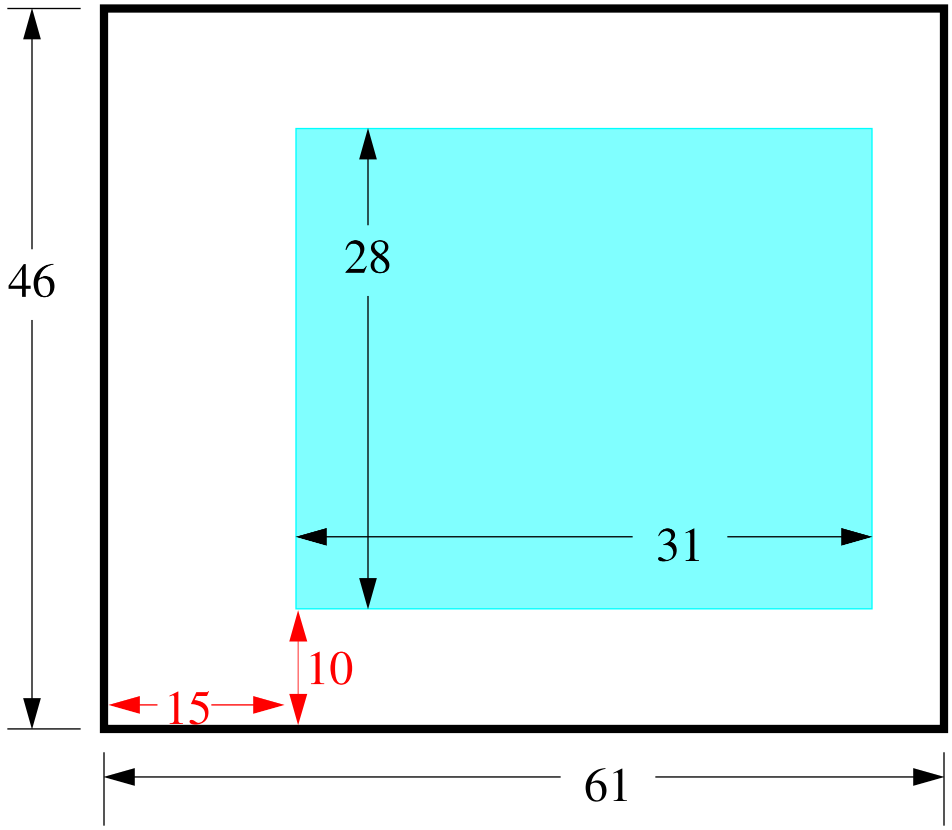

Fig 7.4 The relationship between a coarse and fine domain, and the grid information specified for the starting location. All MM5 system sub-domains are located with the fine grid's lower left corner defined on a mother domain's "dot" point. The most coarse grid has a starting location of (1,1) as a defaults.

- NESTIX = 46 , 28
- NESTJX = 61 , 31
- NESTI = 1 , 10
- NESTJ = 1 , 15

Table 7.7: INTERP deck LOCMIF namelist values.

Namelist Variable	Description
LEVIDN	integer; level identifier of each domain, level of most coarse domain processed defined to be 0
NUMNC	integer; domain identifier for the mother of the current domain, mother of most coarse domain processed is defined to be 1

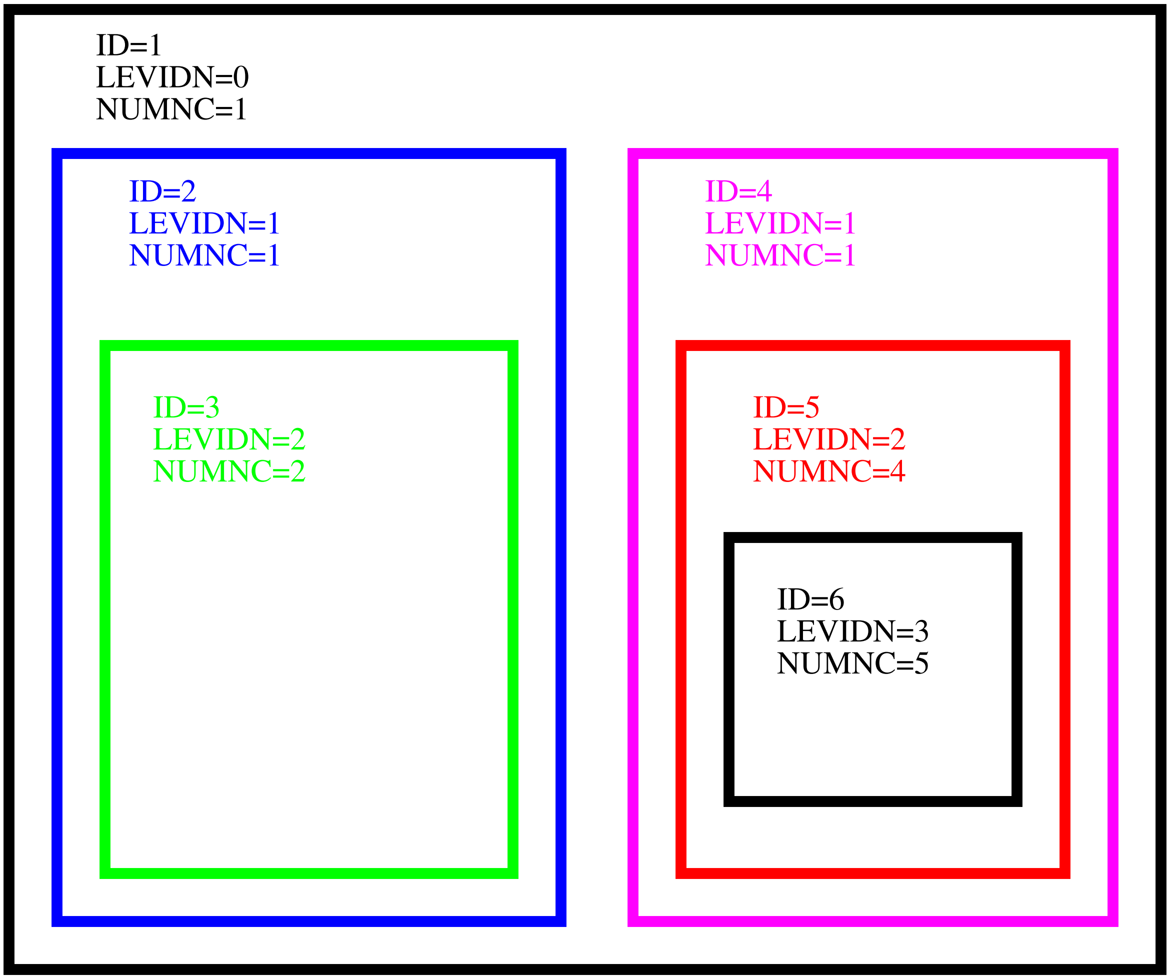

Fig. 7.5 A more complicated nest configuration than is typically used. By definition, the most coarse grid is domain #1 (ID=1), has a nestlevel of 0 (LEVIDN=0, note that this is different from NLNES), and has a mother whose ID=1 (NUMNC=1). The nesting level (LEVIDN) and the ID of the mother domiain of the domain (NUMNC) are required to exactly specify the nesting configuration requested by the user.

Table 7.8: INTERP deck LOCMIF namelist values.

Namelist Variable	Description
OLDTER	logical; if the subdomain terrain elevation data is to be interpolated from coarse grid files then TRUE, if the subdomain terrain elevation is to be provided by a seperate terrain file (created from TERRAIN program) then FALSE
IRATIO	integer; coarse grid to fine grid ratio, a two-way nest requires a three to one ratio (ie 3)

7.10.1 Namelist Record NHYDMIF

Table 7.9: INTERP deck NHYDMIF namelist values.

Namelist Variable	Description
TS0	real; reference sea level temperature
TLP	real; reference temperature lapse rate
P0	real; reference sea level pressure

7.10.2 Namelist Record NONSTAN

Table 7.10: INTERP deck NONSTAN namelist values.

Namelist Variable	Description
ICUT1, ICUT2	integer; beginning and ending location in the I-direction (y-axis) of the subdomain for the first-guess (for 1-way nest, PARAMETER nesting options are utilized to allow for IRATIO other than 1)
JCUT1, JCUT2	integer; beginning and ending location in the J-direction (x-axis) of the subdomain for the first-guess or 1-way nest options
IUSE	integer; 0 = use newcoord and/or IDIV option, 1 = generate an additional data set with MAKEDATA
MAKEDATA	integer; 1 = generate a first-guess file from MM5 output for RAWINS re-analysis, 2 = turn MM5 forecast into model IC and BC directly (no vertical interpolation), 3 = make input for the VERIFY processor
IDIV	integer; 0 = remove integrated mean divergence, 1 = do not modify the wind components

For a standard namelist for front-end job, please see *Templates/lif_front* in the interp.tar file.

7.11 Standard Back-end Job

Most of the standard job submission for the INTERP code involve front-end runs. The necessity to generate large amounts of pressure level data from the MM5 forecast is continuing to be reduced. Much of the post-analysis performed with MM5 data is done interactively with visualization tools that can handle simple vertical coordinate transformations on-the-fly.

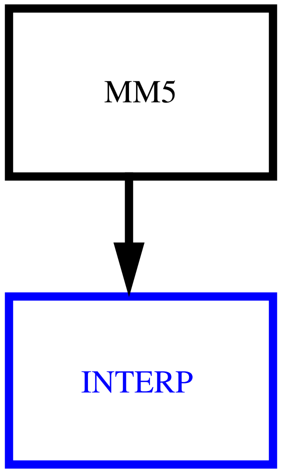

Fig. 7.6 All back-end jobs involve ingesting σ-level model output . On a standard back-end job, this data is interpolated to requested isobaric surfaces by INTERP.

- input MM5 data
- compute sea level pressure
- compute RH, Z on σ surfaces
- extrapolate below 1000 hPa
  - u, v, w, RH, Qc, Qr, Qi, Qs: underground is defined as the lowest σ surface
  - T: -6.5 K/km lapse rate from lowest σ level
  - Z: linear in *ln* pressure for Pslv > 1001 hPa
- interpolate to selected pressure levels
  - u, v, w, RH, Qc, Qr, Qi, Qs: linear in pressure
  - θ, P', Z: linear in *ln* pressure
- output interpolated variables

For the namelist for a standard back-end job, please see *Templates/lif_back* in the interp.tar file.

7.12 Sea Level Pressure Computation

1. Find two surrounding σ levels 100 hPa above the surface, compute T at this level

$$T = \frac{T_{\sigma_A} \ln \frac{P_{\sigma_B}}{P} + T_{\sigma_B} \ln \frac{P}{P_{\sigma_A}}}{\ln \frac{P_{\sigma_B}}{P_{\sigma_A}}} \tag{7.20}$$

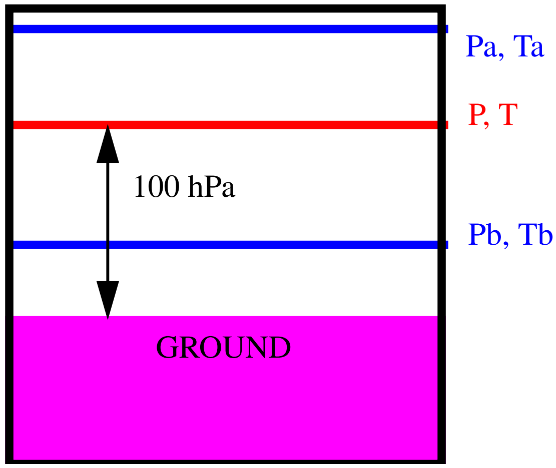

Fig. 7.7 To minimize the diurnal effects on the sea-level pressure computation, a pressure and temperature 100 hPa above the surface is used to compute a “surface” pressure and “surface” temperature.

2. Find T_s (surface temperature), T_m (mean temperature in layer above ground), Z at level 100 hPa above surface, and Tslv (sea level temperature)

$$T_S = T\left(\frac{P_{SFC}}{P}\right)^{\frac{\gamma_S}{g}} \tag{7.21}$$

$$T_m = \frac{T_S + T}{2} \tag{7.22}$$

$$Z = TER - \frac{R}{g}\ln\frac{P}{P_{SFC}} \times T_m \tag{7.23}$$

$$T_{SLV} = T + \gamma_S Z \tag{7.24}$$

3. Then sea level pressure is calculated as

$$P_{SLV} = P_{SFC}\, exp\left[\frac{g \times TER}{R\,\frac{(T_S + T_{SLV})}{2}}\right] \quad (7.25)$$

7.13 Back-end Interpolation/Extrapolation

Extrapolation is required near the surface when

$$p^*_{ij}\sigma_{k = KX} + P_{TOP} + P'_{ijk} < P_{int-bot} \quad (7.26)$$

where Pint-bot is typically 1000 hPa. This is handled in a subroutine specifically to allow vectorization of expensive inner loops for the vertical interpolation scheme. Extrapolation is required near the top of the model when

$$p^*_{ij}\sigma_{k = 1} + P_{TOP} + P'_{ijk} > P_{int-top} \quad (7.27)$$

where Pint-top is typically Ptop. Every column of σ level data has a fictitious level inserted in the column, below the 1000 hPa level (the chosen value is 1001 hPa). This is the technique used to allow removal of an if test in a FORTRAN loop.

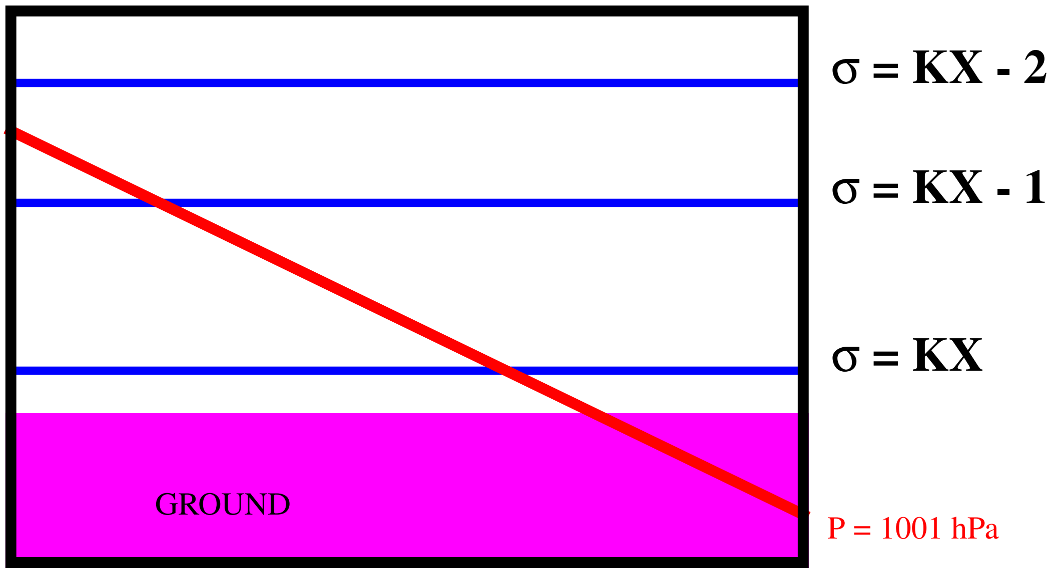

Fig. 7.8 Extrapolation is required on standard back-end jobs when the requested pressure is below the lowest σ level. A fictitious level is generated (1001 hPa) so that the 1000 hPa level is always available without extrapolation.

7.13.1 Interpolation (non-hydrostatic)

Similar to the front-end interpolation, the back-end interpolation is handled as either linear in pressure, linear in *ln* pressure, linear in p^{κ}. The vertical interpolation on the back-end may not be entirely contained within the bounds of valid data, resulting in extrapolation. The non-hydrostatic pressure from the forecast data is given as

$$P_{\sigma_{ijk}} = p^{*}_{ij}\sigma_k + P_{top} + P'_{ijk} \tag{7.28}$$

- $P_{\sigma ijk}$: 3-D pressure at each (i,j,k) of the σ-level variable
- p^*_{ij} : 2-D field of reference surface pressure minus a constant (P_{top})
- σ_k : 1-D vertical coordinate
- P_{top} : reference pressure at model lid
- P'_{ijk} : 3-D pressure perturbation from reference state

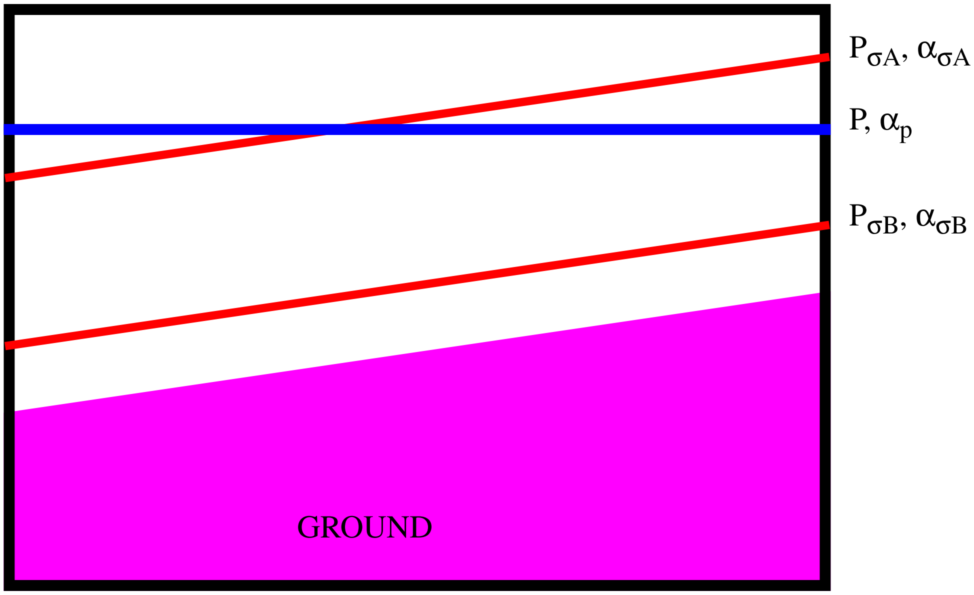

Fig. 7.9 For back-end INTERP jobs, most of the data placed on the isobaric surface is interpolated between the nearest two surrounding σ levels.

$$\alpha_P = \frac{\alpha_{\sigma_A}(P_{\sigma_B} - P) + \alpha_{\sigma_B}(P - P_{\sigma_A})}{P_{\sigma_B} - P_{\sigma_A}} \tag{7.29}$$

7.14 Non-standard Back-end: 1-way Nest

The INTERP program can turn coarse resolution MM5 forecast data into either a similar or higher resolution model initial and boundary condition. Hydrostatic and non-hydrostatic data types remain fixed during this process. Advantages with this method come from an assumed satisfactory previous MM5 model run which can supply high temporal-resolution boundary conditions, and during the process of generating new input files to MM5, higher resolution terrain fields may be used. Disadvantages come from the lack of additional data to be utilized from observations.

An example of the namelist file for setting up the 1-way option is provided in directory *Templates/lif_1way* based on the SESAME test case. Also see Tables 7.11 to 7.14.

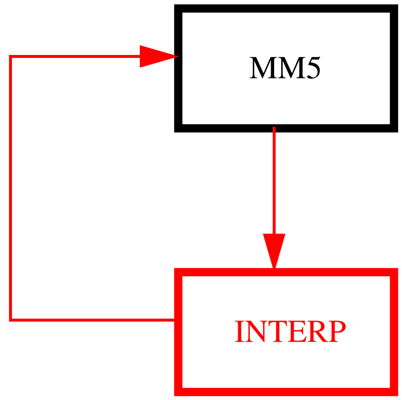

Fig. 7.10 A schematic diagram of a non-standard back-end INTERP job. A one-way nest is model output that has not been vertically interpolated. The INTERP program is used to allow the model to generate additional finer domains, possibly with high temporal resolution for the boundary conditions.

- input MM5 data, and higher resolution terrain fields if desired
- generate sub-domain or horizontal interpolation
- IF hydrostatic
  - Pslv from old terrain
  - Qv to RH
  - new Psfc with new terrain
  - RH to Qv
- IF non-hydrostatic
  - base state with new terrain
  - Qv to RH on old terrain
  - Tn from reference temperature difference between new and old base states
  - new u,v from vertical gradient
  - RH to Qv with new T, P
- re-couple, output IC and BC for MM5

7.14.1 Temperature correction

The ground temperature is not modified, regardless of the elevation change. The air temperature adjustment is based upon the difference in the reference temperature change at the particular (i,j,k) location. The “N” subscript is the new variable with a new terrain elevation, the “O” subscript denotes the old terrain values, and the “R” refers to the reference temperature.

$$T_{N_{ijk}} = T_{O_{ijk}} + (T_{R_O} - T_{R_N}) \tag{7.30}$$

7.14.2 Horizontal-wind correction

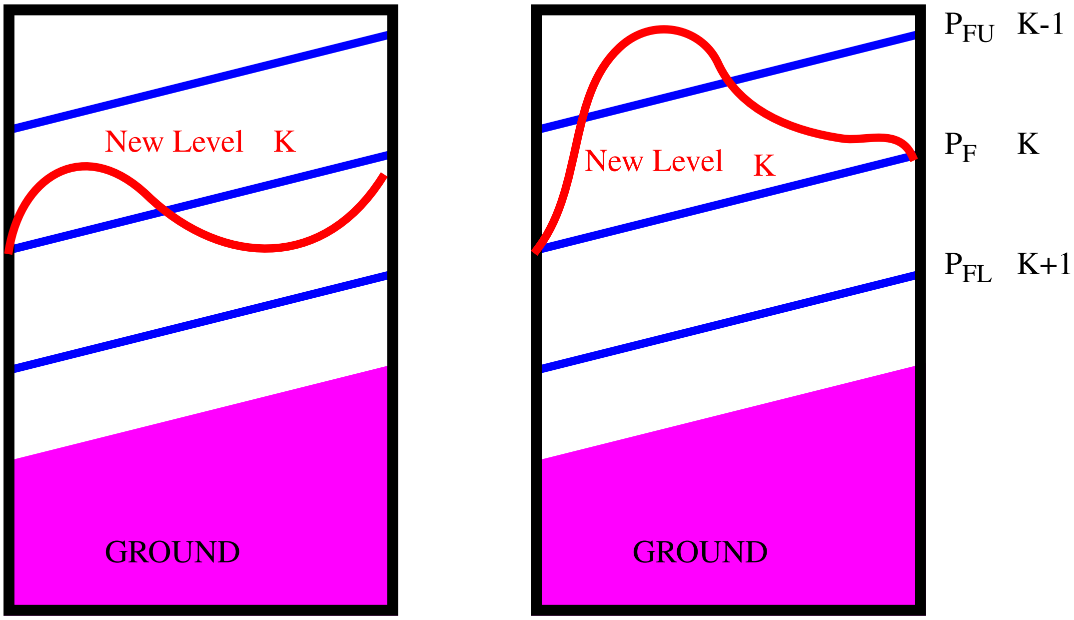

Fig. 7.11 With the introduction of new terrain data in a one-way nest, the σ levels (fixed in height) are adjusted. To keep surface winds on the new domain from picking up characteristics from the free atmosphere, a limit on the depth the vertical gradient is enforced.

1. restrict the vertical extrapolation to only 1 level at most

$$\Delta P = \max(\min((P_{F_N} - P_F), (P_{FL} - P_F)), (P_{FU} - P_F)) \tag{7.31}$$

2. vertical gradient of horizontal wind components

$$\frac{\delta F}{\delta P} = \frac{F_{K+1} - F_{K-1}}{P_{FL} - P_{FU}} \tag{7.32}$$

3. new wind component

$$F_N = F + \Delta P \frac{\delta F}{\delta P} \tag{7.33}$$

Table 7.11: INTERP deck shell variables.

C-shell Variable Name	Options and Use
Jobsw	InterpOnly; with possibly so many times generated for BC, look at the initial time and a few selected time periods interactively
ForBsw	BACK
NESTsw	NEST, NoNEST; whether you anticipate including additional terrain information into MM5, this is recommended

Table 7.12: INTERP deck parameter statements.

PARAMETER	Explanation
MAXNES	2
NLNES	2
IMAX, JMAX	largest I, J dimensions for the two domains
KXS	number of half σ layers in previous MM5 run
KXP	not required
KMAX	set to KXS

Table 7.13: INTERP deck LOCMIF namelist values.

Namelist Variable	Description
NEWCOORD	must be same as previous front-end INTERP job
STANDARD	FALSE
TIMINT	may be very high-temporal frequency
NESTIX, NESTJX	horizontal domain size in grid units
NESTI, NESTJ	starting location of the sub-domain
LEVIDN	most coarse grid processsed is level = 0
NUMNC	most coarse grid processed is domain = 1

Table 7.13: INTERP deck LOCMIF namelist values.

Namelist Variable	Description
OLDTER	T/F; depending on the inclusion of finer resolution terrain elevation
IRATIO	note that is not = 3, then the horizontal interpolation uses a different technique

Table 7.14: INTERP deck NONSTAN namelist values.

Namelist Variable	Description
ICUT1, ICUT2, JCUT1, JCUT2	not used
IUSE	1
MAKEDATA	2
OUTPUTONCE	T/F; whether to output IC time only or all time periods
MMRESTEMP	T/F; whether to use coarse domain substrate temperature

7.15 Non-standard Back-end: First-guess (Forecast Re-analysis)

To provide a method to build a better initial condition, an MM5 forecast may be used to supply the RAWINS program with the first-guess (background) fields which are to be re-analyzed. The traditional data sources range from 2.5 x 5.0 degrees of latitude and longitude through 1.25 x 1.25 degree boxes of latitude and longitude, with only mandatory vertical levels. While it can be argued that a successful MM5 forecast could be used to produce a superior initial condition, the negative impact of the double vertical interpolation can not be discounted (σ to pressure to σ).

1-way nest data can be used directly by MM5, first-guess data must be re-analyzed by RAWINS, and then the traditional IC and BC files must be generated for MM5 with INTERP. Typically, only the initial condition is re-analyzed.

An example of the namelist file for setting up the 1-way option is provided in directory *Templates/lif_firstg* based on the tutorial assignment case. Also see Tables 7.15 to 7.18.

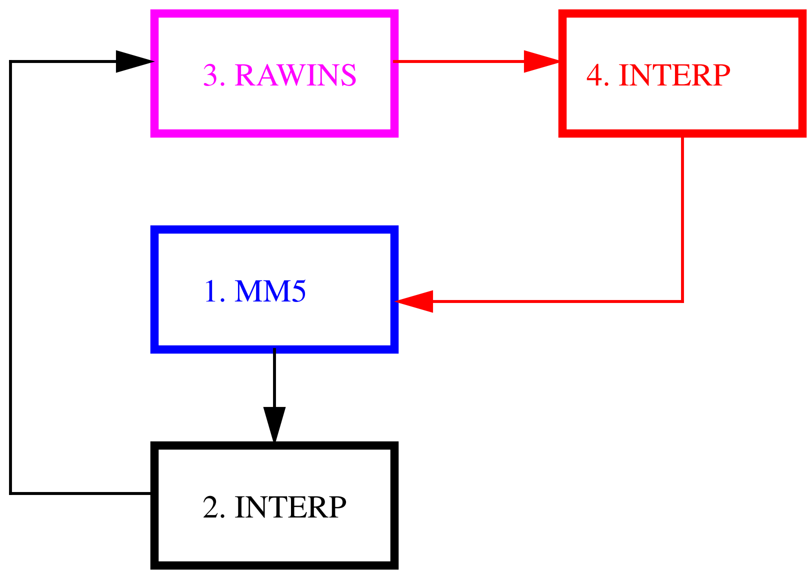

Fig. 7.12 A schematic diagram of the required steps for a first-guess generated from model data. the non-standard INTERP job that creates the first-guess data for RAWINS is shown as box 2. Note that after a re-analysis by RAWINS, that a standard front-end INTERP job is necessary.

- input MM5 data
- cut down to a sub-domain
- standard back-end processing to generate pressure level data on sub-domain
- output sub-domain data in RAWINS input format

Table 7.15: INTERP deck shell variables.

C-shell Variable Name	Options and Use
ForBsw	BACK
NESTsw	NoNEST

Table 7.16: INTERP deck parameter statements.

PARAMETER	Explanation
MAXNES	1
NLNES	1
IMAX, JMAX	I, J dimensions for the original domain
KXS	number of half σ layers in previous MM5 run
KXP	number of pressure levels in previous RAWINS run (not including the surface level)
KMAX	max (KXS , KXP)

Table 7.17: INTERP deck LOCMIF namelist values.

Namelist Variable	Description
NEWCOORD	levels used in previous RAWINS analysis (not including the surface)
STANDARD	FALSE
BEGTIME, ENDTIME	typically the same 10-digit date
NESTIX, NESTJX	horizontal domain size in grid units
NESTI, NESTJ	not used
LEVIDN,NUMNC	only one domain is processed, treat it as most coarse domain
OLDTER	F
IRATIO	1

Table 7.18: INTERP deck NONSTAN values.

Namelist Variable	Description
ICUT1, ICUT2	starting and ending location of the sub-domain in the I-direction
JCUT1, JCUT2	starting and ending location of the sub-domain in the J-direction
IUSE	1
MAKEDATA	1

7.16 How to Run Interp

1) Obtain the source code tar file from one of the following places:

Anonymous ftp:
ftp://ftp.ucar.edu/mesouser/MM5V2/Interp/interp.tar.Z

On NCAR MSS:
/MESOUSER/MM5V2/INTERP/INTERP.TAR.Z

2) gunzip the file, untar it, and edit configure.make file to select appropriate RUNTIME_SYSTEM and compile options. Comment out the ones not appropriate to your machine.

3) Type ‘make interp.deck’ to create a job deck for your platform.

4) Edit interp.deck to select script option, set parameter statements, and select namelist options.

5) Type interp.deck to compile and execute the program. It is usually a good practice to pipe the output to an output file so that if the program fails, you can take a look at the log file. To do so, type: interp.deck >& interp.log, for example.

Interp expects the following input file(s):

Output file from either Regrid, or Rawins/little_r.
Output file from Terrain for the nest (domain 2, 3, etc.), if you use NEST option

Output files from fron-end Interp (input files for MM5):

mminput_domain1
bdyout_domain1
rawins_domain2[,3,...] - if using NEST option

Output files from back-end Interp:

mmoutp_domain1[,2,3,...] - if input to Interp is mmout_domain1[,2,3,...]

mminput_1way_domain3 - if input to Interp is mmout_domain2, and use 1-way option
bdyout_1way_domain3

7.17 INTERP didn’t Work! What Went Wrong?

- Most of the errors from INTERP that do not end with a "segmentation fault", "core dump", or "floating point error" are accompanied with a simple print statement. Though the message itself may not contain enough substance to correct the problem, it will lead you to the section of the code that failed, which should provide more diagnostic information. The last statement that INTERP prints during a controlled failed run is the diagnostic error.

- To see if INTERP completed successfully, first check to see if the "STOP 99999" statement appears. Also check to see that INTERP processed each of the requested times from the namelist. On the front-end jobs, the initial condition file should be written-to after each analysis time. The boundary condition file is written-to after each analysis time, beginning with the second time period. The last boundary time should extend 15 minutes beyond the data to allow MM5 to gracefully exit.

- Since INTERP deals with multiple domains (non-expanded!) that need to be located consistently with any user supplied terrain, it is common to have the various domain dimensions or locations improperly specified. INTERP will ask you to modify the I or J dimension or starting point. INTERP will also ask you to provide the correct number of vertical levels, should they be incorrectly entered.

- When INTERP tells you that it "DID NOT CONVERGE IN 5000 ITERATIONS", you are probably doing an idealized run, with non-divergent winds. Set the flag (IDIV=1 in the NONSTAN namelist) so that you are not doing the mean divergence removal. It can also occur when IMACHWORD in the parameter statement is not set correctly.

- Remember that to generate a single boundary condition file, you must have at least two time periods, so that a lateral boundary tendency may be computed. Even if you are not going to run a long forecast, it is advantageous to provide a lengthy lateral boundary condition file, as this file contains the infinite slab reservoir temperature.

- When INTERP runs into an interpolation error that it did not expect (i.e. forced to do an extrapolation when none should be required), INTERP will stop and print out the offending (I,J,K) and pressure values. If this is not simply a fix by amending the provided σ or pressure surfaces, it is usually a bit more tricky and implies that the analysis data is possibly in error.

- When INTERP is ported to a different architecture, the length of the records for the direct access files must be specified in bytes (4 or 8 per word) or words. This information is found in the src/dadlens.incl file. This is recognized by the INTERP program not being able to write or read from a direct access unit.

7.18 File I/O

The interpolation program has input, output and scratch files that are ingested and created during an INTERP run. The binary input files and all of the output files are unformatted FORTRAN write statements (binary, sequential access). One of the input files is a human-readable namelist formatted file of run-time options. The scratch array are a mix of sequential access and direct-access files.

The following tables are for the scratch, input, and output units, respectively. When a UNIX file name exists, it is given (the FORTRAN writes to a unit number, the Cray assign command links the unit with the local file name). For the description, F refers to a standard front-end job, B refers to a standard back-end job, and N refers to a non-standard job. When multiple file names/unit numbers are possible (such as multiple input domains), the file names are given in C-shell glob format, while the file names are given as a list of available unit numbers. More than nine domains will cause MM5 and its pre-processors difficulties.

Table 7.19: INTERP deck (INTERP program) scratch files.

File Name	Unit	Description
bndry	31	F: accumulates ground temperature from each time period
bnd2d	32	F: 2-D values and tendencies of p*
bnd3d	33	F: 3-D values and tendencies of all coupled fields
int2d	90	F, B, N: direct access for 2-D fields for interpolation
int3d	91	F, B, N: direct access for 3-D fields for interpolation
dad2dc	92	F, B, N: direct access for coarse grid for 2-D fields
dad3dc	93	F, B, N: direct access for coarse grid for 3-D fields
dad2dn	94	F, B, N: direct access for fine grid for 2-D fields
dad3dn	95	F, B, N: direct access for fine grid for 3-D fields

Table 7.20: INTERP deck (INTERP program) input files.

File Name	Unit	Description
inlif	36	F, B, N: namelist
terrain[A-Z]	40-49	F, N:additional terrain elevation and land use for higher resolution information
interpin[A-Z]	50-59	F: DATAGRID or RAWINS, multiple files denotes multiple domains to process B: MM5 forecast, single domain input only N: DATAGRID or RAWINS (for VERIFY input) or MM5 forecast (for 1-way nest, first guess, or VERIFY input)

Table 7.21: INTERP deck (INTERP program) output files.

File Name	Unit	Description
bndtgd	30	F, N: boundary condition for MM5 (standard and 1-way nest)
firstg	34	N: for RAWINS, first guess
	21-29	F: initial condition for MM5 B: pressure level data for GRAPH

Table 7.21: INTERP deck (INTERP program) output files.

File Name	Unit	Description
	62-69	F: fine grid "RAWINS" interpolated for GRAPH N: 1-way nest initial condition for MM5
	80-89	N: VERIFY pressure level input

7.19 INTERP tar File

The interp.tar file contains the following files and directories:

CHANGES	Description of changes to the Graph program
Diff/	Contain difference files between consecutive releases
Makefile	Makefile to create Interp executable
README	General information about the Interp directory
README.namelist	Explanation for Interp namelist file
Templates/	Script files for different workstatons
configure.make	Compile options for various platforms
src/	Interp source code

Templates/:

configure.make.f90	Compile options for Cray
incldiff.sh	
interp.deck.cray	
interp.deck.dec	
interp.deck.hp	
interp.deck.ibm	
interp.deck.sgi	
interp.deck.sun	
lif_1way	
lif_back	
lif_firstg	
lif_front	
no_comment.sed	

7.20 interp.deck

```
#!/bin/sh
#
# temp files should be accessible
#
umask 022
#
# Sections
# 1. parameter statements
# 2. Options for namelist ("inlif")
# 3. Options for I/O
# 4. Compile
# 5. Running...
#
#--------------------------------------------------------------------------
# 1. parameter statement ("inlif")
#    Please referring to Templates/inlif_jobtype for parameter and
#      namelist setting
#--------------------------------------------------------------------------
#
cat > ./src/param.incl.tmp << EOF
      PARAMETER (MAXNES=2)         ! TOTAL NUMBER OF DOMAINS
      PARAMETER (NLNES=2)          ! NUMBER OF LEVELS OF NEST (SAME AS MM5)
      PARAMETER (IMAX = 34, JMAX =  37) ! MAX0(IX,IXN1,...)
      PARAMETER (KXS  = 23)        ! VERT COORD: # HALF SIGMA LAYERS
      PARAMETER (KXP  = 21)        ! VERT COORD: # 1000->PTOP
      PARAMETER (KXT  =  1)        ! IGNORE, BUT DON'T DELETE
      PARAMETER (KMAX = 23)        ! MAX0(KXS,KXP,KXT)
EOF
cat > ./src/dadlens.incl << EOF
C
C     ... SCALE FACTOR FOR DIRECT ACESS FILES
C
C     PARAMETER (IMACHWORD=8) ! Cray
      PARAMETER (IMACHWORD=4) ! SGI
C
C     ... "SMALL" NUMBER FOR RELAXATION AND PP CONVERGENCE
C
C     PARAMETER (SMALLRES=1.0E-9) ! 64 BIT FLOATING POINT
      PARAMETER (SMALLRES=1.0E-5) ! 32 BIT FLOATING POINT
C
EOF
#--------------------------------------------------------------------------
# 2. Options for namelist ("inlif")
#    Please read README.namelist for explanations of the namelist variables
#--------------------------------------------------------------------------
cat > ./inlif << EOF
 &LOCMIF ;--------   LOCAL MIF FOR PROGRAM INTERP ---------
 ;
 ;       FULL SIGMA LEVELS FOR FRONT END -- USUALLY REQUIRED
 ;       BOTTOM-UP (SURFACE TO PTOP) ORIENTATION
  NEWCOORD=1.,.99,.98,.96,.93,.89,.85,.8,.75,.7,.65,.6,.55,.5,.45,.4,
  .35,.3,.25,.2,.15,.1,0.05,0.0,;
 ;
 ;
 INHYD=.TRUE., ; FRONT END ONLY.  0: HYDROSTATIC,
 ;                         1: NONHYDROSTATIC MODEL INPUT
 STANDARD=.TRUE.,        ; IF TRUE, IGNORE NONSTAN NAMELIST
 ;STANDARD=.FALSE.,      ; IF FALSE, USE NONSTAN NAMELIST ALSO
 BEGTIME = 79041012.,    ; BEGINNING TIME FOR INTERP PROCESSING
 ENDTIME = 79041112.,    ; ENDING TIME FOR INTERP PROCESSING
 ;                         USE 10-DIGIT MDATE FOR V1 DATA AND
 ;                              8-DIGIT MDATE FOR V2 DATA AND ON WORKSTATION
 TIMINT  = 720.,         ; TIME INTERVAL IN MINUTES BETWEEN PROCESS TIME
 NESTIX  = 25,  34,  46,  46,  46,  46,  46,  46,  46,  46, ; DOMAIN SIZE I
 NESTJX  = 28,  37,  61,  61,  61,  61,  61,  61,  61,  61, ; DOMAIN SIZE J
 NESTI   =  1,   8,   1,   1,   1,   1,   1,   1,   1,   1,
 ;          START LOCATION I
 NESTJ   =  1,   9,   1,   1,   1,   1,   1,   1,   1,   1,
 ;          START LOCATION J
 LEVIDN  = 0,   1,   2,   0,   0,   0,   0,   0,   0,   0,
 ;          LEVEL OF NEST FOR EACH DOMAIN
```

```
 NUMNC    = 1,   1,   2,   1,   1,   1,   1,   1,   1,   1,
 ;          ID OF MOTHER DOMAIN FOR EACH NEST
 OLDTER   =.FALSE.         ; IF FALSE, USE NESTED TERRAIN FILE
 ;OLDTER   =.TRUE.         ; IF FALSE, USE NESTED TERRAIN FILE
 IRATIO   = 3,             ; NEST RATIO
 &END
 &NHYDMIF  ;----   ONLY REQUIRED FOR FRONT-END NONHYDROSTATIC INTERP JOB --
 TS0      = 290.,          ; SEA-LEVEL TEMPERATURE OF REF STATE (K)
 TLP      = 50.,           ; LAPSE RATE d(T)/d(ln P) OF REF STATE (K)
 P0       = 1.E+05         ; SEA-LEVEL PRESSURE (Pa)
 &END
 &NONSTAN  ;----   ONLY REQUIRED FOR NON-STANDARD INTERP JOBS   ----
 ;          use only if IUSE=1
 ICUT1    = 1,             ; I=1 LOCATION OF SUBDOMAIN IN BIG GRID
 ICUT2    = 25,            ; I=IMAX LOCATION OF SUBDOMAIN IN BIG GRID
 JCUT1    = 1,             ; J=1 LOCATION OF SUBDOMAIN IN BIG GRID
 JCUT2    = 28,            ; J=JMAX LOCATION OF SUBDOMAIN IN BIG GRID
 ;
 ;            *** REPLACE DEFAULTS WITH THESE VALUES ***
 IUSE     = 1,              ; 0 = INTERP USES NEWCOORD + REPLACE DEFAULTS
                           ; 1 = OUTPUT DATA IN SPECIAL FORMAT
 OUTPUTONCE=.FALSE.,       ; TRUE  = OUTPUT INITIAL CONDITION TIME PERIOD ONLY
                           ;     WHEN GENERATING 1-WAY MMINPUT FILE
                           ; FALSE = OUTPUT ALL TIMES
 MMRESTEMP=.FALSE.,        ; TRUE  = USE COARSE DOMAIN SUBSTRATE TEMP
                           ; FALSE = RECALCULATE
 MAKEDATA= 2,              ; IF IUSE=1, THEN EXTRA DATA TO DISPOSE
                           ;     1 = MM AS FIRST GUESS TO RAWINS
                           ;     2 = 1 WAY NEST
                           ;     3 = PRESSURE LEVEL INPUT FOR VERIFY
 IDIV     = 0,             ; FRONT END ONLY
                           ;     0 = REMOVE INTEGRATED MEAN DIVERGENCE
                           ;     1 = DO NOT REMOVE INTEGRATED MEAN DIVERGENCE
 &END
EOF
#
#-----------------------------------------------------------------------------
# 3. Options for I/O
#    Names of input/output files
#
#    Input files:
#    1) datagrid:       datagrid_domain1
#    2) rawins:         rawins_domain1
#    3) model:          mmout_domain1 [mmout_domain2, ....]
#
#    Output files
#    Front-end:
#    4) model input:                 fort.21 [fort.22, ....]
#       boundary file:               fort.30
#    Back-end:
#    5) interpolated model output:   fort.21
#    6) 1-way model input:           fort.61
#       boundary file for 1-way run: fort.30
#    7) first guess:                 fort.34
#-----------------------------------------------------------------------------
#
#
ForBsw=FRONT
#ForBsw=BACK
echo "ForBsw   = $ForBsw"
#
NESTsw=NEST
#NESTsw=NoNEST
echo "NESTsw   = $NESTsw"
#
#      input files
#
if [ $ForBsw = FRONT ]; then
      InData="rawins_domain1"
else
      if [ $ForBsw = BACK ]; then
    InData="mmout_domain1"
      fi
fi
```

```
echo "InData   = $InData"
#
if [ $NESTsw = NEST ]; then
      InTerr="terrain_domain2"
fi
#-----------------------------------------------------------------------------
#      Create a namelist without comments
#-----------------------------------------------------------------------------
#
sed -f Templates/no_comment.sed inlif | grep "[A-Z,a-z]" > inlif.tmp
mv inlif.tmp inlif
#
ForUnit=fort.
#
#-----------------------------------------------------------------------------
# 4. Compiling...
#-----------------------------------------------------------------------------
#
#    check to see if param.incl is different from existing one
#
./Templates/incldiff.sh src/param.incl.tmp src/param.incl
#
make
#
ln -s src/interp.exe interp.exe
#
#-----------------------------------------------------------------------------
# 5. Running...
#    This section should not have to be modified by the user!!
#-----------------------------------------------------------------------------
#
rm -rf ${ForUnit}*
#
# namelist file
#
ln -s inlif   fort.36
#
# input data
#
NUMd=49
  for i in $InData
  do
        NUMd=`expr $NUMd + 1`
          InData=$i
          echo "InData[$NUMd] = $InData"
          if [ ! -r ${ForUnit}$NUMd ]; then
            ln -s $InData ${ForUnit}$NUMd
          fi
          echo "ln -s $InData ${ForUnit}$NUMd"
  done
#
if [ $NESTsw = NEST ]; then
   NUMd=39
   for i in $InTerr
   do
      NUMd=`expr $NUMd + 1`
      InTer=$i
      echo "InTerr[$NUMd] = $InTer"
      if [ ! -r ${ForUnit}$NUMd ]; then
        ln -s $InTer ${ForUnit}$NUMd
      fi
      echo "ln -s $InTer ${ForUnit}$NUMd"
   done
fi
#
#   output data
#
if [ $ForBsw = FRONT ]; then
   ln -s bdyout_domain1  ${ForUnit}30
   echo "ln -s bdyout_domain1  ${ForUnit}30"
   ln -s mminput_domain1 ${ForUnit}21
   echo "ln -s mminput_domain1 ${ForUnit}21"
   if [ $NESTsw = NEST ]; then
     NUMd=1
```

```
     for i in $InTerr
     do
        NUMd=`expr $NUMd + 1`
        if [ ! -r ${ForUnit}2$NUMd ]; then
          ln -s mminput_domain$NUMd ${ForUnit}2$NUMd
        fi
        if [ ! -r ${ForUnit}6$NUMd ]; then
          ln -s rawins_domain$NUMd ${ForUnit}6$NUMd
        fi
        echo "ln -s mminput_domain$NUMd ${ForUnit}2$NUMd"
        echo "ln -s rawins_domain$NUMd ${ForUnit}6$NUMd"
     done
   fi
else
   if [ $ForBsw = BACK ]; then
      ln -s mmoutp_domain1 ${ForUnit}21
      if [ $NESTsw = NEST ]; then
         for i in $InTerr
         do
           NUMd=`echo $i | grep [0-9]$ | sed 's/.*\(.\)/\\1/'`
           echo "Input terrain domain is $NUMd"
           ln -s mminput_1way_domain$NUMd  ${ForUnit}62
           echo "ln -s mminput_1way_domain$NUMd  ${ForUnit}62"
           ln -s bdyout_domain$NUMd  ${ForUnit}30
           echo "ln -s bdyout_domain$NUMd  ${ForUnit}30"
         done
      fi
   fi
fi
#
#       run interp
#
date
echo "time interp.exe > interp.print.out "
time interp.exe > interp.print.out 2>&1
```