Appendix A:

MM5 Flow Charts

A flow chart for MM5 showing the calling structure of its subroutines is shown in Figure A1 with the flow charts (expansion) of major subroutines CHKNST, NSTLEV1, and SOLVE3 shown in Figures A2 and A3. Subroutines appended with a pound sign (#) are expanded elsewhere in the Figure, and those appended with an asterisk (*) are expanded in a different Figure. A few subroutines (ADDRX1C, ADDRX1N, DOTS, XTDOT, SKIPF, and EQUATE) are not shown in the flow charts. Information concerning which subroutines call them can be found in section 5.

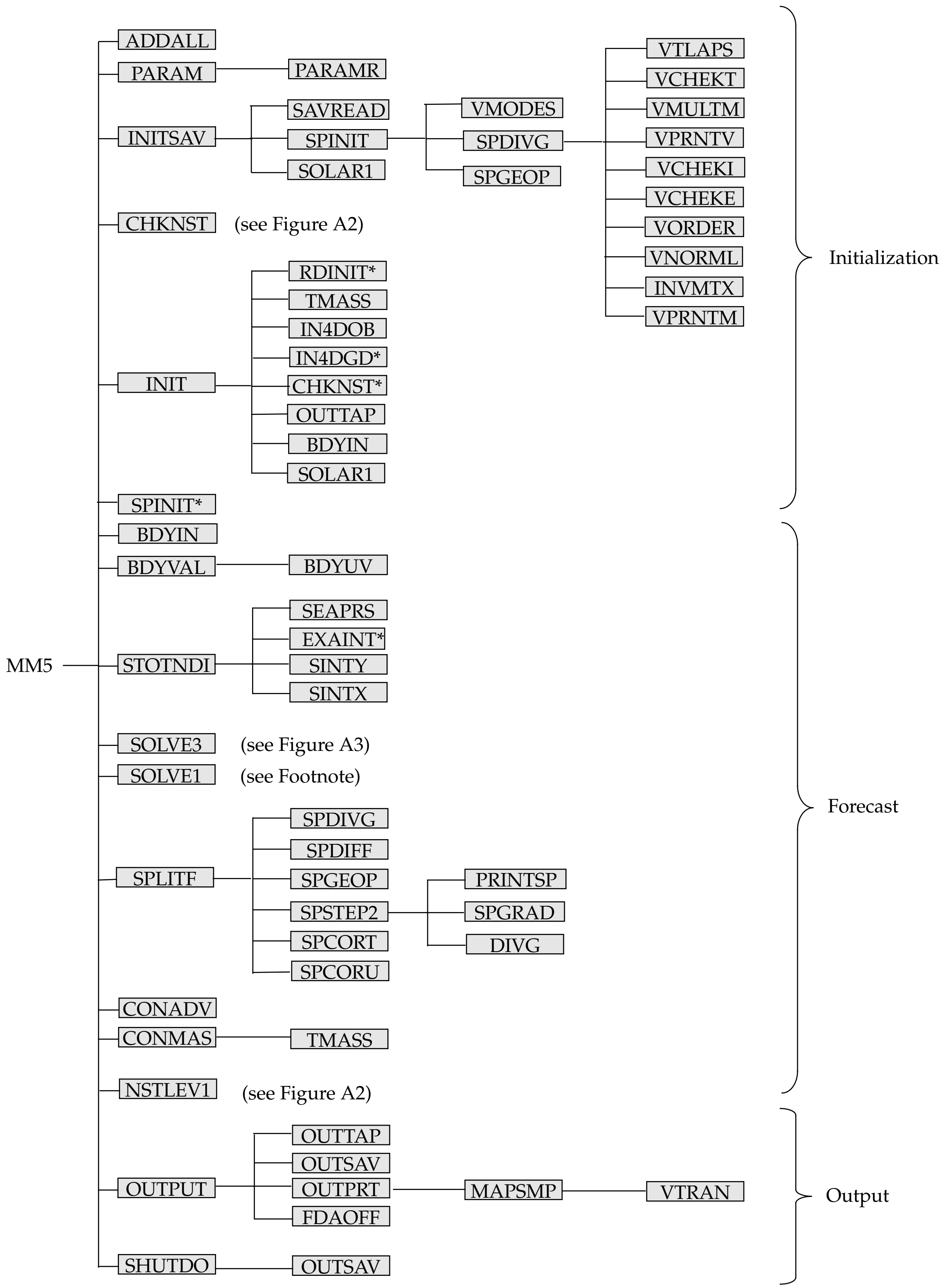

Figure Al. Flow chart of MM5. The expansions of the subroutines appended with an asterisk are shown in Figure A2.

Note: The flow chart structure of subroutine SOLVE1 is very similar to that of SOLVE3 except that subroutine SOUND is not called but subroutine SPONGE is called.

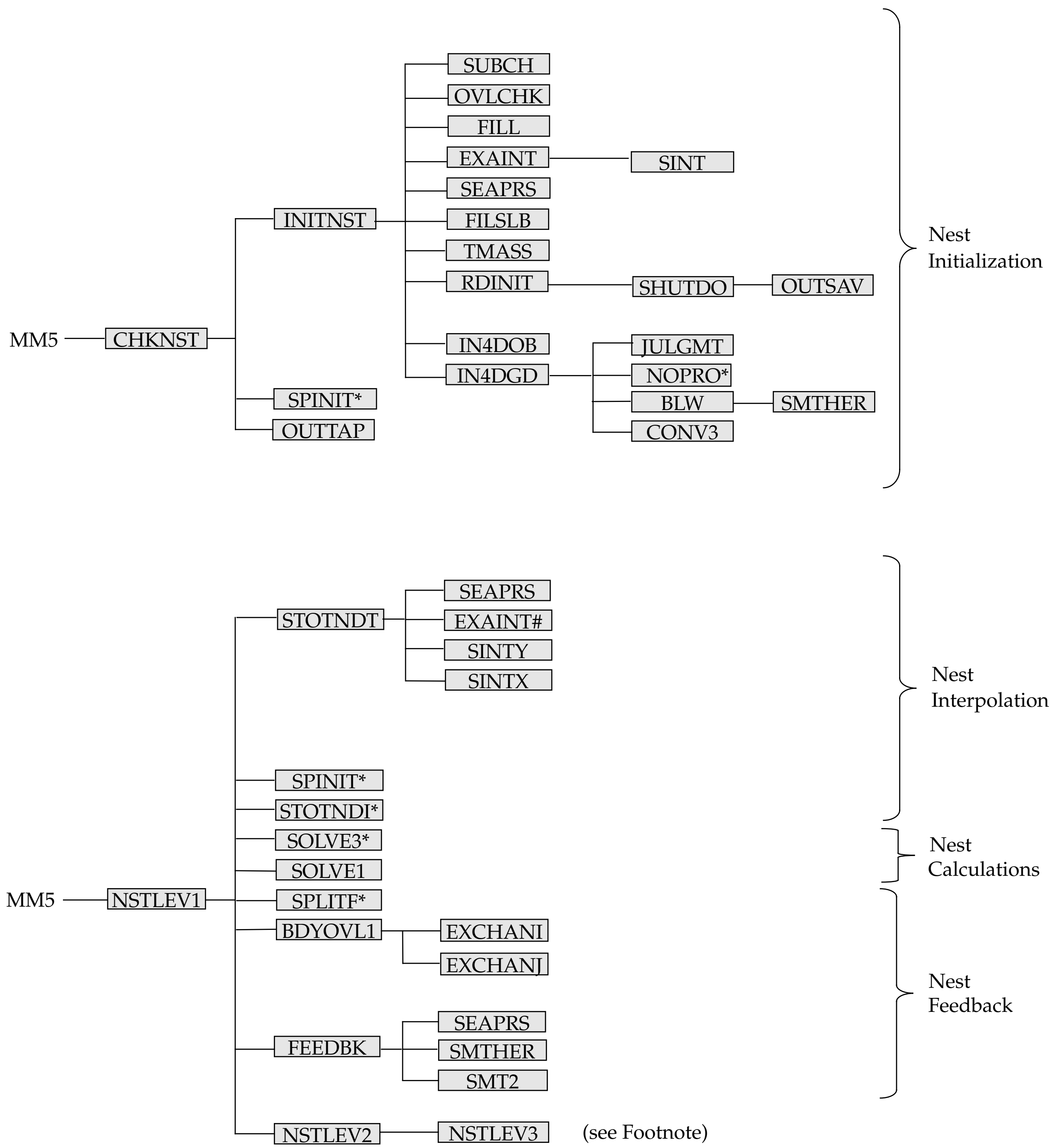

Figure A2. Flow chart for major subroutines CHKNST and NSTLEV1. The expansions of the subroutines appended with an asterisk are shown in Figure A1.

Note: The flow chart structure of subroutines NSTLEV2 and NSTLEV3 is nearly identical to that of NSTLEV1.

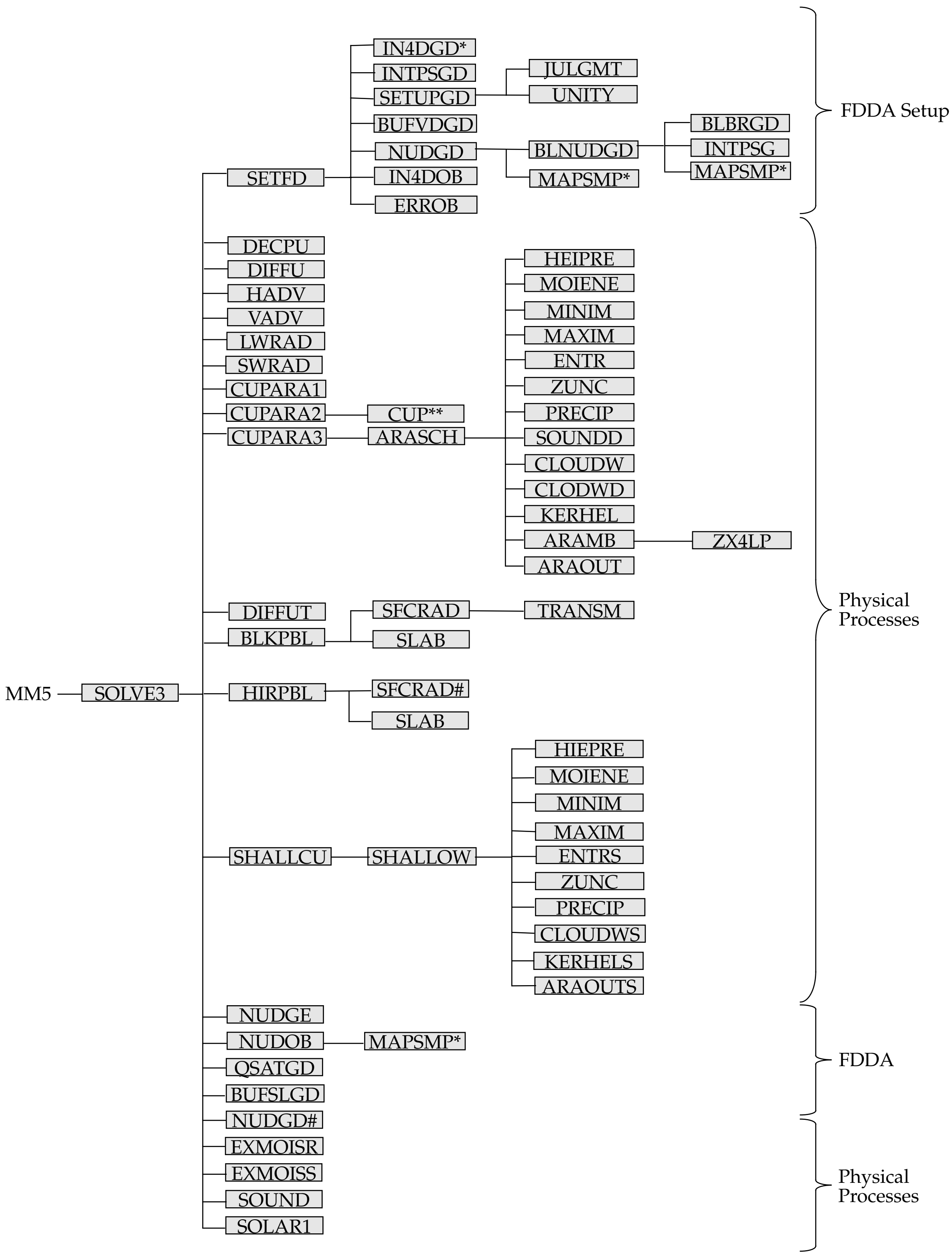

Figure A3. Flow chart for major subroutine SOLVE3. The expansions of the subroutines appended with an asterisk are shown in Figures A1 (for MAPSMP) and A2 (for IN4DGD). The routine appended with a double asterisk calls two subroutines not shown (MAXIMI and MINIMI).

Appendix B:

MM5 Job Deck

This appendix gives an example of the UNICOS script required to submit an MM5 job. It contains 1) UNICOS QSUB directives, 2) input/output Mass Store filenames, 3) UPDATE directives to define the model dimensions and other major PARAMETERs, and 4) the local input file with the NAMELISTs. It also includes the commands for compiling, loading, and running the model.

```
# QSUB -r MM5v1                         # request name
# QSUB -q reg                           # job queue class
# QSUB -eo                              # stdout and stderr together
# QSUB -lM 8Mw                          # maximum memory
# QSUB -lT 5000                         # time limit
# QSUB                                  # no more qsub commands
#
ja
cd $TMPDIR
batchname mm5.out
#
#          ********************************************
#          *******     mm5 batch C shell      *******
#          ********************************************
#
#
#   how many CRAY CPUs to use to run the model, set to 1 if not multitasking
#
setenv NCPUS 1
#
#     this should be the user's case or experiment (used in MS name)
#
 set ExpName = MM5/TEST
 set RetPd   = 365          # MSS retention period in days
#
 set Mods = ( )              # do not change if no user's own mm5 mods
#
# host computer name to rcp user's mm5 mods
 set Host    = username@host.domain:/usr/tmp/username
#
#     type of mm5 job
#
  set recompile = yes       # if yes, recompiled the mm5 code
#   set recompile = no      # if yes, recompiled the mm5 code
#
  set CaseName = CTL        # MSS pathname for this sensitivity test
#
  set STARTsw   = NoReStart   # start model run at hour 0
#  set STARTsw   = ReStart     # restart model run
#
 set FDDAsw    = NoFDDA        # no FDDA input files
#  set FDDAsw    = Anly        # gridded FDDA input files
#   set FDDAsw    = Obs        # obs FDDA input files
#  set FDDAsw    = Both        # gridded and obs FDDA input files
#
#  set HYDROsw   = Hydro       # hydrostatic input files
  set HYDROsw    = NonHydro    # nonhydrostatic input files
#
  set NumDomInp = 1            # number of initial condition input files
  set DomIDInp  = ( 1 )     # domain ID no. for the initial condition inputs
#
  if ( $HYDROsw == NonHydro ) then
     set InitName  =  $ExpName/NH
  else
     set InitName  =  $ExpName/HY
  endif
#
#
#     boundary file
#
```

```
     set InBdy  =  $InitName/BDYOUT_DOMAIN1
#
#     initial conditions
#
     set InMM    = ( $InitName/MMINPUT_DOMAIN1 )
#     set InMM    = ( $InitName/MMINPUT_DOMAIN1 \
#                  $InitName/MMINPUT_DOMAIN2 )
#
#     the input restart file
#
  if ( $STARTsw == ReStart ) then
     set InRst    = ( $InitName/${CaseName}/SAVE_DOMAIN1 \
                  $InitName/${CaseName}/SAVE_DOMAIN2 )
  endif
#
#     4dda surface analyses
#
if (( $FDDAsw == Anly ) || ( $FDDAsw == Both )) then
  set In4DSfc  = ( $ExpName/RW4DDA_DOMAIN1 \
               $ExpName/RW4DDA_DOMAIN2  )
endif
#
#     4dda observations
#
if (( $FDDAsw == Obs  ) || ( $FDDAsw == Both )) then
  set In4DObs  = ( $ExpName/MM5OBS_T12 )
endif
#
#
#     MSS directory name for history, save, shut down and print out files
#
if ( $STARTsw == ReStart ) then
  set OutMM       = ${InitName}/${CaseName}_RES
else
  set OutMM       = ${InitName}/$CaseName
endif
#
#      update for mm5, data dimensions
#
if ( $recompile == yes ) then
cat >! mm5.mods << EOF
*/
*/     ************************************************************
*/     ************************************************************
*/            PARAMETER VALUES TO MODIFY FOR DOMAIN SIZES
*/     ************************************************************
*/     ************************************************************
*/
*ID USERDEF
*/
*/  MAXNES          : TOTAL NUMBER OF DOMAINS IN THE FORECAST,
*/                  INCLUDING THE COARSE GRID
*/
*/  NLNES           : NUMBER OF LEVELS OF NEST.  IF THE COARSE GRID
*/                  HAS A FINE GRID, NLNES=2
*/
*/  MIX,MJX,MKX     : MIX (MJX) IS THE MAXIMUM NUMBER OF
*/                  VALUES ON DOT POINTS IN THE Y (X) DIRECTION
*/                  FOR ALL DOMAINS.
*/                  MKX IS THE NUMBER OF HALF SIGMA LAYERS.
```

```
*/
*/
*D PARAME.2,20
C
C
C--- ADDITIONAL MEMORY REQUIREMENTS FOR FDDA RUNS (IFDDA=1),
C--- NONHYDROSTATIC RUNS (INHYD=1), HIGHER ORDER PBL RUNS (INAV=1),
C--- EXPLICIT MOISTURE SCHEME (IEXMS=1), ARAKAWA-SCHUBERT
C--- CONVECTIVE PARAMETERIZATION (IARASC=1),AND ATMOSPHERIC
C--- RADIATION (IRDDIM=1).
C
C
     PARAMETER (IFDDA=0,INHYD=0,INAV=0,IICE=0,
    1          IEXMS=0,IARASC=0,IRDDIM=0)
C
C--- NUMBER OF DOMAINS (MAXNES) AND NESTLEVELS (NLNES)
C
     PARAMETER (MAXNES=1,NLNES=1)
C
C--- MAXIMUM DIMENSIONS OF ANY DOMAIN
C
     PARAMETER (MIX=1,MJX=1,MKX=1)
*/ COMMENT the following line, if running in extended mt class
*D MM5.170,178
EOF
endif
#
#       local namelist values
#
cat >! mmlif << EOF
 &OPARAM                                   ;        <-- MOD2
 ;-------------------------------------------------------------------
 ;      YOU CAN REMOVE THE UNWANTED DATA FROM THE FOLLOWING LISTING
 ;      AND USE THE DEFAULT VALUES DEFINED IN SUBROUTINE 'PARAM'.
 ;-------------------------------------------------------------------
 IFREST = F,      ;RESTART
   IXTIMR  =720,
 LEVIDN  = 0,1,1,1,1,1,1,1,1,1, ; level of nest for each domain
 NUMNC   = 1,1,1,1,1,1,1,1,1,1, ; ID of mother domain for each nest
 IFSAVE = T,       ; SAVE DATA FOR RESTART
   SAVFRQ =360.,  ;   ... in minutes.
 IFTAPE = 1,       ; OUTPUT FOR GRIN backend
   TAPFRQ =60.,   ;   ... in minutes.
 IFPRT = 0,        ; 1: print output
 PRTFRQ = 720.,    ; Print output frequency in minutes
 MASCHK = 60,      ; MASS CONSERVATION CHECK (minutes)
 & ;-----------------------------------------------------------------
 &LPARAM
  iactiv=1,0,0,0,0,0,0,0,0,0, ; in case of restart: was this domain active?
 ;
 ;
 ;************* start physics options ****************
 ;
 IFRAD     = 0 ,   ;RADIATION COOLING OF ATMOSPHERE - 0, 1, 2
 RADFRQ    = 30.,  :RADIATION FREQUENCY IN MINUTES
 ICUSTB    = 0,    ;STABILITY CHECK FOR CUMULUS PARAM. - 0, 1
 IEXICE    = 0,    ;ICE-PHYSICS IN EXPLICIT SCHEME - 0, 1
 IFDRY     = 0,    ;FAKE-DRY RUN - 0, 1
 IMVDIF    = 0,    ;MOIST VERTICAL DIFFUSION IN CLOUDS - 0, 1
 IBMOIST   = 0,    ;BOUNDARY AND INITIAL WATER/ICE SPECIFIED - 0, 1
```

```
ICOR3D    = 0,    ;3D CORIOLIS FORCE (FOR NH RUN ONLY) - 0, 1
IFUPR     = 0,    ;UPPER RADIATIVE BOUNDARY CONDITION (NH) - 0, 1
;
IBOUDY = 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, ;BOUNDARY CONDITIONS
;       (FIXED, RELAXATION, TIME-DEPENDENT, TIME AND INFLOW/OUTFLOW DEPENDENT,
;        SPONGE - 0, 1, 2, 3, 4)
IBLTYP = 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, ;PBL TYPE
;       (FRICTIONLESS, BULK PBL, MULTI-LEVEL PBL - 0, 1, 2)
IDRY   = 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ;MOIST OR DRY CASE - 0, 1
IMOIST = 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ;NON-EXPLICIT,  EXPLICIT, -  1, 2
ICUPA  = 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ;NONE,KUO,GRELL,A-S - 1,2,3,4
ISFFLX = 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ;SURFACE FLUXES - 0, 1
ITGFLG = 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ;SURFACE TEMPERATURE - 1, 3
ISFPAR = 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ;SURFACE CHARACTERISTICS - 0, 1
ICLOUD = 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ;CLOUD EFFECTS ON RADIATION - 0, 1
ICDCON = 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ;CONSTANT DRAG COEFFICIENTS - 0, 1
IFSNOW = 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ;SNOW COVER EFFECTS - 0, 1
IMOIAV = 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ;VARIABLE MOISTURE AVAILABILITY - 0, 1
IVMIXM = 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ;VERTICAL MIXING OF MOMENTUM - 0, 1
HYDPRE = 1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,;HYDRO EFFECTS OF LIQ WATER - 0., 1.
IEVAP  = 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ;EVAP OF CLOUD/RAINWATER - <0, 0, >0
ISHALLO= 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ;SHALLOW CONVECTION - 0, 1
;
;************* end physics options ******************
;
;
;
;************* start nesting options ****************
;
nestix =  46,  46,  46,  46,  46,  46,  46,  46,  46,  46, ; domain size I
nestjx =  61,  61,  61,  61,  61,  61,  61,  61,  61,  61, ; domain size J
nesti  =   1,   1,  1,   1,   1,   1,   1,   1,   1,   1, ; start location I
nestj  =   1,   1,  1,   1,   1,   1,   1,   1,   1,   1, ; start location J
xstnes =   0.,  0., 0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,; domain initiation
xennes = 1440.,1440.,1440.,720.,720.,720.,720.,720.,720.,; domain completion
ioverw =   1,   0,   0,   0,   0,   0,   0,   0,   0,   0, ; overwrite domain
;
;************* end nesting options ******************
;
;
;
;************* start moving options ******************
;
imove  =   0,   0,   0,   0,   0,   0,   0,   0,   0,   0, ; move domain 0,1
imovco =   1,   1,   1,   1,   1,   1,   1,   1,   1,   1, ; 1st move #
; imoveI(J,K)=L, ; I-INCREMENT MOVE (DOMAIN J, MOVE NUMBER K) IS L
imovei =   0,   0,   0,   0,   0,   0,   0,   0,   0,   0, ; I move #1
           0,   0,   0,   0,   0,   0,   0,   0,   0,   0, ; I move #2
           0,   0,   0,   0,   0,   0,   0,   0,   0,   0, ; I move #3
           0,   0,   0,   0,   0,   0,   0,   0,   0,   0, ; I move #4
           0,   0,   0,   0,   0,   0,   0,   0,   0,   0, ; I move #5
imovej =   0,   0,   0,   0,   0,   0,   0,   0,   0,   0, ; J move #1
           0,   0,   0,   0,   0,   0,   0,   0,   0,   0, ; J move #2
           0,   0,   0,   0,   0,   0,   0,   0,   0,   0, ; J move #3
           0,   0,   0,   0,   0,   0,   0,   0,   0,   0, ; J move #4
           0,   0,   0,   0,   0,   0,   0,   0,   0,   0, ; J move #5
imovet =   0,   0,   0,   0,   0,   0,   0,   0,   0,   0, ; time of move #1
           0,   0,   0,   0,   0,   0,   0,   0,   0,   0, ; time of move #2
           0,   0,   0,   0,   0,   0,   0,   0,   0,   0, ; time of move #3
           0,   0,   0,   0,   0,   0,   0,   0,   0,   0, ; time of move #4
```

```
      0,   0,   0,   0,   0,   0,   0,   0,   0,   0, ; time of move #5
;
;************* end moving options ******************
;
& ;-------------------------------------------------------------------
&PPARAM
TIMAX  = 1440.,   ; IN MINUTES. 720=12h, 1440=24h, 2160=36h, 2880=48h
ZZLND  = 0.1,    ; ROUGHNESS LENGTH OVER LAND IN METERS
ZZWTR  = 0.0001, ; ROUGHNESS LENGTH OVER WATER IN METERS
ALBLND = 0.15,   ; ALBEDO
THINLD = 0.04,   ; SURFACE THERMAL INERTIA
XMAVA  = 0.3,    ; MOISTURE AVAILABILITY OVER LAND AS A DECIMAL
;                 FRACTION OF ONE
CONF   = 1.0,    ; NON-CONVECTIVE PRECIPITATION SATURATION
TISTEP = 270.,   ; COARSE DOMAIN DT IN MODEL, USE 3*DX
ifeed  = 3,      ; OLD FEEDBACK, NO/LIGHT SMOOTHING IN FEEDBK - 1,2,3
iabsor = 0,      ; SPONGE ON UPPER BOUNDARY (HYD) - 0,1
& ;-------------------------------------------------------------------
&FPARAM
;
;
;        ****************************************************
;        ****************************************************
;
;
;    THE FIRST DIMENSION (COLUMN) IS THE DOMAIN IDENTIFIER:
;        COLUMN 1 = DOMAIN #1, COLUMN 2 = DOMAIN #2, ETC.
;
;      START TIME FOR FDDA (ANALYSIS OR OBS) FOR EACH DOMAIN
;      (IN MINUTES RELATIVE TO MODEL INITIAL TIME)
FDASTA=0.,0.,0.,0.,0.,0.,0.,0.,0.,0.,
;
;      ENDING TIME FOR FDDA (ANALYSIS OR OBS) FOR EACH DOMAIN
;      (IN MINUTES RELATIVE TO MODEL INITIAL TIME)
FDAEND=780.,0.,0.,0.,0.,0.,0.,0.,0.,0.,
;
;
;        ****************************************************
;        **************** ANALYSIS NUDGING ******************
;        ****************************************************
;
;
;    THE FIRST DIMENSION (COLUMN) OF THE ARRAYS DENOTES THE
;    DOMAIN IDENTIFIER:
;        COLUMN 1 = DOMAIN #1, COLUMN 2 = DOMAIN #2, ETC.
;    THE SECOND DIMENSION (ROW OR LINE) EITHER REFERS TO THE 3D VS
;    SFC ANALYSIS OR WHICH VARIABLE IS ACCESSED:
;        LINE 1 = 3D, LINE 2 = SFC   OR
;        LINE 1 = U, LINE 2 = V, LINE 3 = T, LINE 4 = Q
;
;      IS THIS A GRID 4DDA RUN? 0 = NO; 1 = YES
I4D=0,0,0,0,0,0,0,0,0,0,      ; 3D ANALYSIS NUDGING
    0,0,0,0,0,0,0,0,0,0,      ; SFC ANALYSIS NUDGING
;
;      SPECIFY THE TIME IN MINUTES BETWEEN THE INPUT (USUALLY
;      FROM INTERP) USED FOR GRID FDDA
DIFTIM=720.,720.,0.,0.,0.,0.,0.,0.,0.,0.,      ; 3D ANALYSIS NUDGING
       180.,180.,0.,0.,0.,0.,0.,0.,0.,0.,      ; SFC ANALYSIS NUDGING
;
;      GRID NUDGE THE WIND FIELD? 0 = NO; 1 = YES
```

```
IWIND=1,1,0,0,0,0,0,0,0,0,    ; 3D ANALYSIS NUDGING
      1,1,0,0,0,0,0,0,0,0,    ; SFC ANALYSIS NUDGING
;
;     NUDGING COEFFICIENT FOR WINDS ANALYSES
GV=2.5E-4,1.0E-4,0.,0.,0.,0.,0.,0.,0.,0.,      ; 3D ANALYSIS NUDGING
   2.5e-4,1.0E-4,0.,0.,0.,0.,0.,0.,0.,0.,      ; SFC ANALYSIS NUDGING
;
;     GRID NUDGE THE TEMPERATURE FIELD? 0 = NO; 1 = YES
ITEMP=1,1,0,0,0,0,0,0,0,0,    ; 3D ANALYSIS NUDGING
      1,1,0,0,0,0,0,0,0,0,    ; SFC ANALYSIS NUDGING
;
;     NUDGING COEFFICIENT FOR TEMPERATURE ANALYSES
GT=2.5e-4,1.0E-4,0.,0.,0.,0.,0.,0.,0.,0.,      ; 3D ANALYSIS NUDGING
   2.5e-4,1.0E-4,0.,0.,0.,0.,0.,0.,0.,0.,      ; SFC ANALYSIS NUDGING
;
;     GRID NUDGE THE MIXING RATIO FIELD? = 0 ; NO = 1 ; YES
IMOIS=1,1,0,0,0,0,0,0,0,0,    ; 3D ANALYSIS NUDGING
      1,1,0,0,0,0,0,0,0,0,    ; SFC ANALYSIS NUDGING
;
;     NUDGING COEFFICIENT FOR THE MIXING RATIO ANALYSES
GQ=1.E-5,1.E-5,0.,0.,0.,0.,0.,0.,0.,0.,        ; 3D ANALYSIS NUDGING
   1.E-5,1.E-5,0.,0.,0.,0.,0.,0.,0.,0.,        ; SFC ANALYSIS NUDGING
;
;     GRID NUDGE THE ROTATIONAL WIND FIELD? 0 = NO; 1 = YES
IROT=0,0,0,0,0,0,0,0,0,0,     ; 3D ANALYSIS NUDGING
;
;     NUDGING COEFFICIENT FOR THE ROTATIONAL COMPONENT OF THE WINDS
GR=5.E6,5.E6,0.,0.,0.,0.,0.,0.,0.,0.,          ; 3D ANALYSIS NUDGING
;
;      IF GRID NUDGING (I4D(1,1)=1) AND YOU WISH TO EXCLUDE THE
;      BOUNDARY LAYER FROM FDDA OF COARSE GRID THREE DIMENSIONAL
;      DATA (USUALLY FROM INTERP),
;          0 = NO,  INCLUDE BOUNDARY LAYER NUDGING
;          1 = YES, EXCLUDE BOUNDARY LAYER NUDGING
INONBL =0,0,0,0,0,0,0,0,0,0,0,0,         ; U WIND
       0,0,0,0,0,0,0,0,0,0,0,0,          ; V WIND
       0,0,0,0,0,0,0,0,0,0,0,0,          ; TEMPERATURE
       0,0,0,0,0,0,0,0,0,0,0,0,          ; MIXING RATIO
;
;      RADIUS OF INFLUENCE FOR SURFACE ANALYSIS (KM).
;      IF I4D(2,1)=1 OR I4D(2,2)=1, ETC, DEFINE RINBLW (KM) USED
;      IN SUBROUTINE BLW TO DETERMINE THE HORIZONTAL VARIABILITY
;      OF THE SURFACE-ANALYSIS NUDGING AS A FUNCTION OF SURFACE
;      DATA DENSITY.  OVER LAND, THE STRENGTH OF THE SURFACE-
;      ANALYSIS NUDGING IS LINEARLY DECREASED BY 80 PERCENT AT
;      THOSE GRID POINTS GREATER THAN RINBLW FROM AN OBSERVATION
;      TO ACCOUNT FOR DECREASED CONFIDENCE IN THE ANALYSIS
;      IN REGIONS NOT NEAR ANY OBSERVATIONS.
RINBLW=250.,
;
;      SET THE NUDGING PRINT FREQUENCY FOR SELECTED DIAGNOSTIC
;      PRINTS IN THE GRID (ANALYSIS) NUDGING CODE (IN CGM
;      TIMESTEPS)
NPFG=50,
;
;       ****************************************************
;       *************** OBSERVATION NUDGING ***************
;       ****************************************************
;
;
```

```
;    INDIVIDUAL OBSERVATION NUDGING.  VARIABLES THAT ARE ARRAYS
;    USE THE FIRST DIMENSION (COLUMN) AS THE DOMAIN IDENTIFIER:
;        COLUMN 1 = DOMAIN #1, COLUMN 2 = DOMAIN #2, ETC.
;
;      IS THIS INDIVIDUAL OBSERVATION NUDGING? 0 = NO; 1 = YES
I4DI   =0,0,0,0,0,0,0,0,0,0,
;
;      OBS NUDGE THE WIND FIELD FROM STATION DATA? 0 = NO; 1 = YES
ISWIND =1,0,0,0,0,0,0,0,0,0,
;
;      NUDGING COEFFICIENT FOR WINDS FROM STATION DATA
GIV  =4.E-4,4.E-4,0.,0.,0.,0.,0.,0.,0.,0.,
;
;      OBS NUDGE THE TEMPERATURE FIELD FROM STATION DATA? 0 = NO; 1 = YES
ISTEMP=1,0,0,0,0,0,0,0,0,0,
;
;      NUDGING COEFFICIENT FOR TEMPERATURES FROM STATION DATA
GIT  =4.E-4,4.E-4,0.,0.,0.,0.,0.,0.,0.,0.,
;
;      OBS NUDGE THE MIXING RATIO FIELD FROM STATION DATA? 0 = NO; 1 = YES
ISMOIS=1,0,0,0,0,0,0,0,0,0,
;
;      NUDGING COEFFICIENT FOR THE MIXING RATIO FROM STATION DATA
GIQ  =4.E-4,4.E-4,0.,0.,0.,0.,0.,0.,0.,0.,
;
;      THE OBS NUDGING RADIUS OF INFLUENCE IN THE
;      HORIZONTAL IN KM FOR CRESSMAN-TYPE DISTANCE-WEIGHTED
;      FUNCTIONS WHICH SPREAD THE OBS-NUDGING CORRECTION
;      IN THE HORIZONTAL.
RINXY=240.,
;
;      THE OBS NUDGING RADIUS OF INFLUENCE IN THE
;      VERTICAL IN SIGMA UNITS FOR CRESSMAN-TYPE DISTANCE-
;      WEIGHTED FUNCTIONS WHICH SPREAD THE OBS-NUDGING
;      CORRECTION IN THE VERTICAL.
RINSIG=0.001,
;
;      THE HALF-PERIOD OF THE TIME WINDOW, IN MINUTES, OVER
;      WHICH AN OBSERVATION WILL AFFECT THE FORECAST VIA OBS
;      NUDGING. THAT IS, THE OBS WILL INFLUENCE THE FORECAST
;      FROM TIMEOBS-TWINDO TO TIMEOBS+TWINDO.  THE TEMPORAL
;      WEIGHTING FUNCTION IS DEFINED SUCH THAT THE OBSERVATION
;      IS APPLIED WITH FULL STRENGTH WITHIN TWINDO/2. MINUTES
;      BEFORE OR AFTER THE OBSERVATION TIME, AND THEN LINEARLY
;      DECREASES TO ZERO TWINDO MINUTES BEFORE OR AFTER THE
;      OBSERVATION TIME.
TWINDO=40.0,
;
;      THE NUDGING PRINT FREQUENCY FOR SELECTED DIAGNOSTIC PRINT
;      IN THE OBS NUDGING CODE (IN CGM TIMESTEPS)
NPFI=20,
;
;      FREQUENCY (IN CGM TIMESTEPS) TO COMPUTE OBS NUDGING WEIGHTS
IONF=2,
 & ;-----------------------------------------------------------------
EOF
#
########################################################################
###################                     #######################
###################    END USER MODIFICATION    #######################
```

```
####################               #######################
########################################################################
#
#       this is INTERACTIVE or BATCH
#
if ( $?ENVIRONMENT ) then
  echo "environment variable defined as $ENVIRONMENT"
else
  setenv ENVIRONMENT INTERACTIVE
  echo "environment variable defined as $ENVIRONMENT"
endif
#
#       initializations, no user modification required
#
set LETTERS = (A B C D E F G H I J K L M N O P Q R S T U V W X Y Z)
set MesoUser = /u1/mesouser/MM5V1/MM5
#
#       get boundary conditions from MS
#
msread bdyout $InBdy
#
#       loop over how many files of domains to acquire
#
set NUM = 0
while ( $NUM < $NumDomInp )
  @ NUM ++
#
#       initial conditions
#
  msread fort.1$DomIDInp[$NUM] $InMM[$NUM]
#
#       input restart conditions
#
  if ( $STARTsw == ReStart ) then
    msread fort.9$DomIDInp[$NUM] $InRst[$NUM]
  endif
#
#       get analyses for nudging
#
  if (( $FDDAsw == Anly ) || ( $FDDAsw == Both )) then
    cp fort.1$DomIDInp[$NUM] fort.3$DomIDInp[$NUM]
    msread fort.7$DomIDInp[$NUM] $In4DSfc[$NUM]
    cp fort.7$DomIDInp[$NUM] fort.8$DomIDInp[$NUM]
  endif
#
#       observations if OBS nudging
#
  if (( $FDDAsw == Obs ) || ( $FDDAsw == Both )) then
    msread fort.6$DomIDInp[$NUM] $In4DObs[$NUM]
  endif
end

#
#       set up fortran input files for MM5
#
if ( -e assign.mm5 ) rm assign.mm5
setenv FILENV assign.mm5
assign -a mmlif                    fort.7
assign -a ${MesoUser}/ehtran       fort.8
assign -a bdyout                   fort.9
```

```
if ( $recompile == yes ) then
#
#       mm5 mod decks to include in update
#
  set NUM = 0
  while ( $NUM < ${#Mods} )
     @ NUM ++
     echo "using $Mods[$NUM] mod deck"
     rcp $Host/${Mods[$NUM]} .
     cat $Mods[$NUM] >> mm5.mods
     rm $Mods[$NUM]
  end

  if ( -e m5_stand.mods ) then
     echo "using local copy of m5_stand.mods"
     cat m5_stand.mods >> mm5.mods
  else if ( -e ~/m5_stand.mods ) then
     echo "using MY copy of m5_stand.mods"
     cat ~/m5_stand.mods >> mm5.mods
  else if ( -e ${MesoUser}/m5_stand.mods ) then
     echo "using standard copy of m5_stand.mods"
     cat ${MesoUser}/m5_stand.mods >> mm5.mods
  else
     echo "not using any copy of m5_stand.mods"
  endif
#
  if ( -e m5_my.mods ) then
     echo "using local copy of m5_my.mods"
     cat m5_my.mods >> mm5.mods
  else if ( -e ~/m5_my.mods ) then
     echo "using MY copy of m5_my.mods"
     cat ~/m5_my.mods >> mm5.mods
  else
     echo "not using any copy of m5_my.mods"
  endif
#
#       make an MM5 source code
#
  if ( ! -e mm5.v1 ) then
     echo "acquiring the mm5.v1 source code"
     cp ${MesoUser}/mm5v1.s mm5.v1
  endif
  nupdate -i mm5.v1 -n mmzin
  nupdate -p mmzin -m 1 -f -i mm5.mods -o "id sq" -c mm5
  set toast = $status
  if     ( $toast != 0 ) then
     echo "error in the update, stopping"
     exit(1)
  else if ( $ENVIRONMENT != BATCH ) then
     echo -n "update complete, continue?    (yes)   "
     set ans = "$<"
     if (( $ans == "n" ) || ( $ans == "no" )) then
        exit (1)
     endif
  endif
#
#       compile MM5
#
  cf77 -c -Wf"-o aggress" mm5.f
  set toast = $status
```

```
   if     ( $toast != 0 ) then
      echo "error in the compile, stopping"
      exit(1)
   else if ( $ENVIRONMENT != BATCH ) then
      echo -n "compile complete, continue?    (yes)    "
      set ans = "$<"
      if (( $ans == "n" ) || ( $ans == "no" )) then
         exit (1)
      endif
   endif
#
#      load MM5
#
   segldr -L /lib,/usr/lib,/usr/local/lib -l etime,imslcnv,imsl -o mm5.exe mm5.o
   set toast = $status
   if     ( $toast != 0 ) then
      echo "error in the segldr, stopping"
      exit(1)
   else if ( $ENVIRONMENT != BATCH ) then
      echo -n "segldr complete, continue?     (yes)    "
      set ans = "$<"
      if (( $ans == "n" ) || ( $ans == "no" )) then
         exit (1)
      endif
   endif
endif

if ( $recompile == no ) then
   msread mm5.exe ${InitName}/${CaseName}/mm5.exe
   chmod +x mm5.exe
else
   mswrite mm5.exe ${InitName}/${CaseName}/mm5.exe
endif

execute:
ls -l
#
#      run MM5
#
date
mm5.exe >&! mm5.print.out
set toast = $status
if ( $toast != 0 ) then
   echo "error in the forecast, stopping"
   debug -s mm5.exe
   if ( $ENVIRONMENT != BATCH ) exit(1)
endif
#
#      if interactive, probably do not want to dispose files
#
if ( $ENVIRONMENT != BATCH ) then
   echo -n "test mm5 run complete, continue?     (no)     "
   set ans = "$<"
   if (( $ans != "y" ) && ( $ans != "yes" )) then
      exit (0)
   endif
endif
#
#      print and save the print output
#
```

```
ja -chls >! acct
cat acct >> mm5.print.out
if ( $ENVIRONMENT == BATCH ) cat mm5.print.out
#
#      history output   41-49
#      save file output 51-59
#      shutdown output  61-69
#
ls -ls
set Tens  = (   4     5     6   )
set Name  = ( MMOUT SAVE SHUTDO )
foreach OutType ( 1 2 3 )
   set OutFileType = $Name[$OutType]
   foreach Units ( 1 2 3 4 5 6 7 8 9 )
      if ( -e fort.$Tens[$OutType]${Units} ) then
         echo ls -l fort.$Tens[$OutType]${Units} >! hold
         set test = `source hold`
         if ( $test[4] < 400000000 ) then
            mswrite -t $RetPd fort.$Tens[$OutType]$Units \
                    $OutMM/${OutFileType}_DOMAIN${Units}
         else
          if ($Tens[$OutType] == 4) then
            if ( ! -e split.deck ) then
            cp ${MesoUser}/split.deck .
            chmod +x split.deck
            endif
            mv fort.$Tens[$OutType]${Units} ${OutFileType}_DOMAIN${Units}
            split.deck ${OutFileType}_DOMAIN${Units}

            set Numb = 0
            foreach fil ( `ls mmtmp*` )
            @ Numb ++
            echo "mswrite $fil \
                 $OutMM/${OutFileType}_DOMAIN${Units}_$LETTERS[$Numb] "
            mswrite -t $RetPd $fil \
                 $OutMM/${OutFileType}_DOMAIN${Units}_$LETTERS[$Numb]
            end

            rm mmtmp*
          else

            bsplit -400000000 fort.$Tens[$OutType]${Units} \
                   small.$Tens[$OutType]${Units}.
            set NUM = 1
            echo ls small.$Tens[$OutType]${Units}.* >! hold
            set test = `source hold`
            foreach split ( $test )
               mswrite -t $RetPd $split \
                  $OutMM/${OutFileType}_DOMAIN${Units}_BSPLIT_$LETTERS[$NUM]
               @ NUM ++
            end
          endif

         endif
      endif
   end
end
#
# tar the namelist, mods, source code, executable, and output together
# save the MM5 tar file on MSS
```

```
#
tar -cvf mm5.tar mm5.f mmlif mm5.mods mm5.print.out
echo " mswrite -t $RetPd mm5.tar $ExpName/mm5.tar "
mswrite -t $RetPd mm5.tar $OutMM/mm5.tar
ls -ls
```

Appendix C:

Unit Number Allocations

Model input and output of various types are assigned to FORTRAN logical units as follows. Where the unit number has two digits, the second digit always refers to the domain number. All units are binary unless ASCII is specified. Units designated (I) are input, but are only used internally. Units 1-6 and those ending in zero are not used.

UNIT	I/O	DESCRIPTION
7	I	mmlif: model input namelist file (ASCII)
8	I	ehtran: look-up table for transmissivities
9	I	bdyout: coarse-mesh boundary conditions
11-19	I	Initial conditions and FDDA analyses
21-29		Not used
31-39	(I)	Analyses for FDDA (dummy file copied from 11-19)
41-49	O	Model output file for GRIN
51-59	O	Save-file output for restarts
61-69	I	Individual observations for FDDA
61-69	O	Shut-down save-file output for restarts
71-79	I	Surface analyses for FDDA
81-89	(I)	Surface analyses for FDDA (dummy file copied from 71-79)
91-99	I	Input restart file