Appendix B: Pack and Unpack Subroutines

The following are the four subroutines for packing and unpacking the terrain height and land-use source data. Detailed discussions of these subroutines can be found in Section 5.3. The modifications to these subroutines for a 32-bit word machine can be found in Appendix C under Section "(2) Use of the IEEE format data".

(1) Subroutine to pack NP integers to one integer

```
        SUBROUTINE IPACK(LL,NN,NP)
C
C   PURPOSE :  PACK THE NP INTEGERS TO 1 INTEGER
C
C   On entry :    INTEGER ARRAY NN(NP)
C   On exit  :    INTEGER LL
C
        DIMENSION NN(NP)
C
        MV = 64/NP
        LL = 0
C
        DO 20 K = 1,NP
        KK = (K-1)*MV
c  LOGICAL SHIFT: NN(K) SHIFTS LEFT KK
        MM = SHIFTL(NN(K),KK)
c  LOGICAL SUM:
        LL = OR(LL,MM)
20      CONTINUE
C
        RETURN
        END
```

(2) Subroutine to unpack an integer to NP integers

```
       SUBROUTINE IUNPACK(LL,M1,NP)
C
C  PURPOSE : UNPACK THE INTEGER LL TO NP INTEGERS M1(NP)
C
C  On entry   :  LL
C  On exit    :  M1(NP)
C
       DIMENSION M1(NP)
C
       MV = 64/NP
       IF (NP.EQ.2) MO = 4294967295   !     2^32 - 1
       IF (NP.EQ.4) MO = 65535        !     2^16 - 1
       IF (NP.EQ.8) MO = 255          !     2^8 - 1
C
       DO 30 K = 1,NP
       KK = (K-1)*MV
       MM = SHIFTL(MO,KK)
c LOGICAL PRODUCT:
       N1 = AND(LL,MM)
c LOGICAL SHIFT: N1 SHIFTS RIGHT KK
       N1 = SHIFTR(N1,KK)
       M1(K) = N1
30     CONTINUE
c
       RETURN
       END
```

(3) Subroutine to pack an integer array with dimension NW to another smaller array

```
      SUBROUTINE NPACK(INA,IPA,NW,NP)
C
C PURPOSE   : PACK THE INTERGER ARRAY
C
C On entry  :
C     INA(NW)         AN ARRAY OF INTEGER NUMBERS
C     NW              ITS LENGTH
C
C On exit   :
C     IPA(.)                   THE PACKED DATA
C
      DIMENSION II(8),INA(NW),IPA(1)
C
      IF (NP.EQ.2) NU = 4294967295   !    2^32 - 1
      IF (NP.EQ.4) NU = 65535        !    2^16 - 1
      IF (NP.EQ.8) NU = 255          !    2^8 - 1
C
      MAX = -40000
      MIN =  40000
      DO 1 I = 1,NW
      IF (INA(I).GT.MAX) MAX = INA(I)
      IF (INA(I).LT.MIN) MIN = INA(I)
1     CONTINUE
      IF ((MAX-MIN) .GT. NU) THEN
       PRINT *,'(MAX-MIN)=',(MAX-MIN),' > ',NU,', THE NP=',
     1           NP,' IS TOO BIG.'
       STOP 77777
      ENDIF
C
      IPA(1) = 0
      IF (MIN.LT.0) IPA(1) = -MIN
      DO 2 I = 1,NW
2       INA(I) = INA(I) + IPA(1)
C
      N = NW/NP
      IF  (MOD(NW,NP) .NE. 0) N = N+ 1
      DO 10 I = 1,N
        DO 11 J = 1,NP
        II(J) = NU
        JJ = (I-1)*NP + J
        IF (JJ.LE.NW) II(J) = INA(JJ)
11      CONTINUE
        CALL IPACK(LL,II,NP)
        IPA(I+1) = LL
10    CONTINUE
C
      RETURN
      END
```

(4) Subroutine to unpack an integer array to an array with dimension NW

```
      SUBROUTINE NUNPACK(INA,IPA,NW,NP)
C
C PURPOSE : UNPACK THE INTEGER ARRAY
C
C On entry:
C     IPA(.)                   THE PACKED ARRAY PRODUCED BY NPACK
C     NW              THE NUMBER OF ELEMENTS WANTED
C
C On exit :
C     INA(NW)         THE ARRAY OF INTEGER NUMBERS
C
      DIMENSION M1(8),INA(NW),IPA(1)
C
      N = NW/NP
      IF  (MOD(NW,NP) .NE. 0) N = N+ 1
      DO 10 I = 1,N
        LL = IPA(I+1)
        CALL IUNPACK(LL,M1,NP)
        DO 11 J = 1,NP
        JJ = (I-1)*NP + J
        IF (JJ.LE.NW) INA(JJ) = M1(J) - IPA(1)
11      CONTINUE
10    CONTINUE
C
      RETURN
      END
```

Appendix C: Modifications for the Air Force and IEEE Data

(1) Use of Air Force 5-minute terrain height data

- The modifications to the TERRAIN program:

```
*/
*/  Air Force 5 minutes data information:
*/  -- the number of latitude records, initial latitude, MSS file name:
*D SETUP.34
     2       nrec/   180,   360,  1080,   804,  3361/,
*D SETUP.37
     5      xlati/   90.,   90.,   90.,   75.,   51./,
*D SETUP.49
     8             ' /MESOUSER/TERRAIN/DATA/AFC05MIN',
*/
*/   -- The data located at the center of lat.-lon.  box,
*/       keep the CSHIFT = CENTER/2.
*D RDLDTR.133
*/
*/   -- Missing and ocean depth values are -999.9 < -999.8.
*I ANAL2.72
      IF (NI.EQ.4 .AND. ASTA(KK) .LE. -998.9) GO TO 80
*I CRTER.65
      IF (NI.EQ.4 .AND. HTOB(II) .LE. -998.9) HTOB(II) = -.001
*/   -- Out of data range --> STOP.
*I TERDRV.93
      IF (NI.EQ.4 .AND. (XMAXLAT.GT.75.0 .OR. XMINLAT.LT.8.))
     1     THEN
        PRINT *,'+++ 1/12 deg.  data cover area from 8 to 75 ',
     1  'north,  the domain specified above exceeded ',
     2  'the data coverage area.+++'
        STOP
      ENDIF
```

- The changes in the job script

The command line:

```
tar -xf terln.tar
```

is changed to 3 command lines:

```
tar terln.tar
rm ter.05
msread ter.05  /MESOUSER/TERRAIN/DATA/AFC05MIN
```

(2) Use of the IEEE format data

- The modifications to the TERRAIN program:

```
*/   Using IEEE format data:
*/
*/    -- MSS file names:
*D LTDATA.11
      character LNDNAME(3)*40, TERNAME(5)*37
*D SETUP.43,50
      data lndname/' /MESOUSER/IEEE/TERRAIN/LANDUSE/GLB60MIN',
     1           ' /MESOUSER/IEEE/TERRAIN/LANDUSE/GLB30MIN',
     2           ' /MESOUSER/IEEE/TERRAIN/LANDUSE/GLB10MIN'/
     5     tername/' /MESOUSER/IEEE/TERRAIN/DATA/GLB60MIN',
     6           ' /MESOUSER/IEEE/TERRAIN/DATA/GLB30MIN',
     7           ' /MESOUSER/IEEE/TERRAIN/DATA/GLB10MIN',
     8           ' /MESOUSER/IEEE/TERRAIN/DATA/GLB05MIN',
     9           ' /MESOUSER/IEEE/TERRAIN/DATA/USA30SEC'/
*D RDLDTR.127
      character lndname*40, tername*37, msread*7, remove*3, cmnd*70
*/
*/    -- Internal read format:
*D RDLDTR.153,154
 10   FORMAT(A7,1X,A8,2X,A40)
 11   FORMAT(A7,1X,A7,2X,A37)
*/
*/    -- Buffer array dimension:
*D RDLDTR.119,120
      dimension landout1(id1*id2/4+2),landin(13,2160),
     1                inht1(id2/2+2),inter1(8401)
*/
*/    -- Read in data and unpacking:
*D RDLDTR.178,180
      READ(NUNIT,ERR=100) LANDOUT1
      CALL NUNPACK (LANDIN,LANDOUT1,ID,4)
*D RDLDTR.184,186
      READ(IUNIT,ERR=200) inht1
      CALL NUNPACK (inter1,inht1,id2,2)
*/
*/    -- Mods for pack and unpack subroutine (32-bit word):
*D NPACK.42,44
C       IF (NP.EQ.2) NU = 4294967295   !      2^32 - 1
      IF (NP.EQ.2) NU = 65535          !      2^16 - 1
      IF (NP.EQ.4) NU = 255            !      2^8 - 1
*D NPACK.69
        IF (JJ.LE.NW) THEN
           II(J) = INA(JJ)
        ELSE
           II(J) = 0
        ENDIF
*D IPACK.43
      MV = 32/NP
*D IUNACK.21,24
      MV = 32/NP
C       IF (NP.EQ.2) MO = 4294967295   !    2^32 - 1
      IF (NP.EQ.2) MO = 65535          !    2^16 - 1
      IF (NP.EQ.4) MO = 255            !    2^8 - 1
```

** If the internal *msread* is not used, the first 4 mods should be removed.

• The changes in the job script

The command line:

```
msread terln.tar /MESOUSER/MM5V1/TERRAIN/ter.lndu.tar
```

is changed to the command line:

```
msread terln.tar /MESOUSER/MM5V1/TERRAIN/IEEE/terln.tar
```

and the command lines:

```
assign -a landu.60              fort.20
assign -a ter.60                fort.21
assign -a landu.30              fort.22
assign -a ter.30                fort.23
assign -a landu.10              fort.24
assign -a ter.10                fort.25
assign -a ter.05                fort.27
assign -a ter.30s               fort.29
```

are changed to the command lines:

```
assign -a landu.60   -Ff77 -Nieee fort.20
assign -a ter.60     -Ff77 -Nieee fort.21
assign -a landu.30   -Ff77 -Nieee fort.22
assign -a ter.30     -Ff77 -Nieee fort.23
assign -a landu.10   -Ff77 -Nieee fort.24
assign -a ter.10     -Ff77 -Nieee fort.25
assign -a ter.05     -Ff77 -Nieee fort.27
assign -a ter.05     -Ff77 -Nieee fort.29
```

Appendix D: FTP'ing VAX Files to and from shavano

If the user's local machine is a VAX, then modification of the TERRAIN shell script is needed to send the TERRAIN plot file from shavano back to the user's local VAX. Before the user modifies the TERRAIN shell script, a file called ftp.remote needs to be in the user's login directory. Section 2.3 (p.2-9) of the documentation "A User's Guide to the Penn State/NCAR Mesoscale Modeling System" by Dave Gill (1992) has a detailed description on how to set up this file. Below are the sections of the terrain.deck that the user needs to modify.

Change

```
#
#  send terrain plots to various plotting device
#
rcp TER.PLT $Host/ter.plt
```

To

```
#
#  send terrain plots to various plotting device
#
   if ( -e ftp.put ) rm ftp.put
   cat > ftp.put << EOF
   put TER.PLT ter.plt
   quit
EOF
   cp ~/ftp.remote ftp.temp
   chmod 600 ftp.temp
   cat ftp.put >> ftp.temp
   ftp -n < ftp.temp
   rm ftp*
   echo "put terrain plot file ter.plt to the front-end machine"
```

To send a file to shavano from the local machine, change the "put TER.PLT ter.plt" to "get flnm1 flnm2" in the above script, where flnm1 is the local machine filename and flnm2 is the shavano filename.

Appendix E: TERRAIN Output Format

The TERRAIN program outputs the header, terrain height, and land-use categories for programs DATAGRID, GRIN, or MM5. Each TERRAIN output file contains information on only one domain. The header contains domain information which is passed to other programs and information which defines the contents and the format of subsequent records in the TERRAIN output file. The header record contains four two-dimensional arrays: MIF, MIFC, MRF, and MRFC. The first array index for these arrays range from 1 to 1000. The coarse domain information is located at array index 1 to 100. The domain (coarse or nested) information is located at array index 101 to 200. For array index 201 to 1000, the subsequent record information is defined such as the number of 3-D, 2-D, 1-D, and 0-D arrays and their units. The second array index for the header record arrays represents the type of the output files. Now we have seven types of the output files as follows,

1: from program TERRAIN

2: from program DATAGRID

3: from RAWINS

4: surface FDDA fields from RAWINS

5: from the front end of INTERP

6: from MM5 model

7: from the back end of INTERP

Below the user will find a sample FORTRAN subroutine to read the TERRAIN output and a brief description of the variables.

```
      SUBROUTINE RDTER (MIF,MRF,MIFC,MRFC,TER,
     1                  LAND,IX,JX,IUNIT)
C
C  PURPOSE : THIS SUBROUTINE READS THE TERRAIN OUTPUT
C
C    IX    : NUMBER OF GRID POINTS IN I (Y) DIRECTION
C    JX    : NUMBER OF GRID POINTS IN J (X) DIRECTION
C    IUNIT : UNIT NUMBER TO READ THE TERRAIN FILE
C
C  OUTPUT  :
C    MIF   : 2-D INTEGER ARRAY CONTAINS INTEGER VARIABLES
C            INFORMATION
C    MIFC  : 2-D CHARACTER ARRAY WHICH DESCRIBES THE ARRAY
C            MIF
C    MRF   : 2-D REAL ARRAY CONTAINS REAL VARIABLE
C            INFORMATION
C    MRFC  : 2-D CHARACTER ARRAY WHICH DESCRIBES THE ARRAY
C            MRF
C    TER   : 2-D REAL ARRAY CONTAINS THE TERRAIN HEIGHT IN
C            METER
C    LAND  : 2-D INTEGER ARRAY CONTAINS THE LAND-USE CATEGORIES
C
      REAL TER(IX,JX)
      INTEGER LAND(IX,JX)
      INTEGER      MIF(1000,20)
      REAL         MRF(1000,20)
      CHARACTER*80 MIFC(1000,20),
     1             MRFC(1000,20)
C
C  read header record
C
      READ (IUNIT) MIF,MIFC,MRF,MRFC
C
C  print out header record information
C
      DO 10 I=1,1000
        IF(MIF(I,1).EQ.-999) GO TO 20
        WRITE(6,100) I,MIF(I,1),MIFC(I,1)
10    CONTINUE
20    CONTINUE
      DO 30 I=1,1000
        IF(MRF(I,1).EQ.-999.)  GO TO 40
        WRITE(6,110) I,MRF(I,1),MRFC(I,1)
30    CONTINUE
40    CONTINUE
C
C  read terrain height and land-use fields
C
      READ (IUNIT) TER
      READ (IUNIT) LAND
100   FORMAT('MIF(',I3,',1) = ',I7,5X,A50)
110   FORMAT('MRF(',I3,',1) = ',F7.2,5X,A50)
      RETURN
      END
```

Description of Subroutine RDTER Variables

The MIFC and MRFC character arrays describe the variables in the MIF (integer) and MRF (real) arrays for program TERRAIN. Their current definitions are

MIFC(1,1)	program name : TERRAIN
MIFC(2,1)	coarse domain grid dimension in I (N-S) direction
MIFC(3,1)	coarse domain grid dimension in J (E-W) direction
MIFC(4,1)	map projection
	1: lambert conformal
	2: polar stereographic
	3: Mercator
MIFC(5,1)	coarse domain expansion
	1: yes
	0: no
MIFC(6,1)	expanded coarse domain grid dimension in I direction
MIFC(7,1)	expanded coarse domain grid dimension in J direction
MIFC(8,1)	grid offset in I direction due to the coarse grid expansion
MIFC(9,1)	grid offset in J direction due to the coarse grid expansion
MIFC(10,1)	type of the vertical coordinate
	0: no vertical coordinate defined
	1: Pressure
	2: Sigma
	3: MAPS Generalized
	4: Isentropic
MIFC(101,1)	domain id
MIFC(102,1)	mother domain id
MIFC(103,1)	nest level (0: coarse mesh)
MIFC(104,1)	domain grid dimension in I direction
MIFC(105,1)	domain grid dimension in J direction
MIFC(106,1)	I location in the mother domain of the domain grid point (1,1)
MIFC(107,1)	J location in the mother domain of the domain grid point (1,1)
MIFC(108,1)	domain grid size ratio with respect to the coarse domain
MIFC(109,1)	domain grid size ratio with respect to the mother domain
MIFC(110,1)	nest type
	1: one-way
	2: two-way
MIFC(111,1)	smoother
	1: 1-2-1
	2: smoother-desmoother

MIFC(201,1)	number of 3-D fields in output
MIFC(202,1)	number of 2-D fields in output
MIFC(203,1)	number of 1-D fields in output
MIFC(204,1)	number of 0-D fields in output
MIFC(205,1)	terrain elevation m cross nocoupled
MIFC(206,1)	land use category cross nocoupled

MRFC(1,1)	coarse domain grid distance (km)
MRFC(2,1)	coarse domain center latitude (degree)
MRFC(3,1)	coarse domain center longitude (degree)
MRFC(4,1)	cone factor
MRFC(5,1)	true latitude 1 (degrees)
MRFC(6,1)	true latitude 2 (degrees)
MRFC(7,1)	pole position in degrees latitude
MRFC(8,1)	approximate expansion (km)
MRFC(101,1)	domain grid distance (km)
MRFC(102,1)	I location in the coarse domain of the domain point (1,1)
MRFC(103,1)	J location in coarse domain of the domain point (1,1)
MRFC(104,1)	I location in the coarse domain of the domain point (IX,JX)
MRFC(105,1)	J location in the coarse domain of the domain point (IX,JX)
MRFC(106,1)	terrain data resolution (in degrees)
MRFC(107,1)	land-use data resolution (in degrees)
TER	domain terrain elevation (m)
LAND	domain land-use categories (integers from 1 - 13)
IUNIT	FORTRAN logical unit number assigned to the terrain file

Appendix F: TERRAIN Input and Output Units

Below the user will find the description of the input and output units for program TERRAIN.

Input Unit	Local File Name	Description
15	mif	TERRAIN namelist
16	raobsta	global raob station list
20	landu.60	one deg. source land-use file
21	ter.60	one deg. source terrain file
22	landu.30	30 min source land-use file
23	ter.30	30 min source terrain file
24	landu.10	10 min source land-use file
25	ter.10	10 min source terrain file
27	ter.05	5 min source terrain file
29	ter.30s	30 sec source terrain file

Output Unit	Local File Name	Description
31	fort.31	domain 1 (coarse domain) TERRAIN output
32	fort.32	domain 2 TERRAIN output
33	fort.33	domain 3 TERRAIN output
34	fort.34	domain 4 TERRAIN output
35	fort.35	domain 5 TERRAIN output
36	fort.36	domain 6 TERRAIN output
37	fort.37	domain 7 TERRAIN output
38	fort.38	domain 8 TERRAIN output
39	fort.39	domain 9 TERRAIN output