Atmospheric Modeling to Refine Lichen-based Critical Loads for the North American Marine West Coast Forests Ecological Region

Pierre Wong, Joe Vaughan, and Brian Lamb

Laboratory for Atmospheric Research at Washington State University

September 30, 2010

TASK AGREEMENT NO.: J8W07090010

COOPERATIVE AGREEMENT NO. H8W07060001

EFFECTIVE DATES: 06/05/2009 - 09/30/2010

Introduction

This study, funded by the National Park Service and in collaboration with the USDA Forest Service, supports investigation of the lichen response to atmospheric nitrogen and thereby contributes to establishment of critical loads for nitrogen deposition. In this work The Laboratory for Atmospheric Research at WSU has provided the NPS with AIRPACT-3 simulation results for nitrogen deposition in the Pacific Northwest region generally for 28 months, and for 23 forest research sites in Washington and Oregon for specific periods of measurement.

This project provides model results for nitrogen deposition for the Pacific Northwest from the air-quality modeling system AIRPACT-3. Modeled deposition results are output by species and deposition mechanism and the results are thus separable as contributing to nitrogen dry deposition or nitrogen wet deposition. The dry and wet deposition results are reported as aerosol loadings (kg/ha) by the model for individual nitrogen-bearing molecules (species) on an hourly basis throughout the model domain, the 12-km grid shown in Figure 1. This project sums these species and accumulates their deposition over time, to facilitate comparison with field observations and better represent the model results in terms of ecosystem inputs of nitrogen. The nitrogen deposition model results were extracted and summed over time for the specific monitoring sites operated from June 2006 through September 2008, which were identified by National Park Service (NPS) and National Forest Service (NFS) collaborators.

WSU provides the following deliverables:

1) Nitrogen deposition in units of N kg/Ha for twelve (12) deposition variables along with four (4)

additional meteorological variables (for 16 variables, as listed in Appendix A) for each monitoring site for its specific period, as well as monthly totals. These deliverables are being shared with project correspondents using the Dropbox facility (dropbox.com) via folders shared by invitation.

2) Regional maps of monthly dry and wet nitrogen deposition in units of kg N/ha have been constructed and are being made available via the AIRPACT-3 website at: www.atmos.washington.edu/~empact/airpact_monthly_depo/AP_monthly_depo.php and examples are shown in Figure 2.

Methodology

1. Data Source

All reported data are from the AIRPACT-3 air-quality forecasting system operated by LAR. The data reported are specifically from the CMAQ model with some meteorological (precipitation) results from the MCIP-processed WRF results. *No observational (field) data are represented in these modeling results,* only results from numerical simulations.

AIRPACT-3 is an air-quality forecast system for the Pacific Northwest reporting to the public daily via the web (http://lar.wsu.edu/AIRPACT-3). The AIRPACT system combines air chemistry and meteorology using community modeling software including the Weather Research Forecast (WRF) meteorological model (Mesoscale Model 5 (MM5) prior to April, 2008), the SMOKE processing system, and the Community Multi-scale Air Quality Model (CMAQ). WRF output fields are obtained from the University of Washington mesoscale meteorological forecast operations on a daily basis (http://www.atmos.washington.edu/mm5rt/, Mass et al., 2003). The WRF meteorological fields are processed using the MCIP program prior to use in CMAQ. Details about MCIP and the governing equations of CMAQ can be found in Byun and Schere (2006), which describes the calculations for advection, diffusion, chemical reactions, photolysis, cloud mixing, aerosol dynamics, and deposition. The AIRPACT-3 domain (shown in Figure 1) uses 95 x 95 grid of 12 km x 12 km grid cells (9025 total) with 21 vertical layers increasing in layer thickness from the surface to the tropopause. Further details describing AIRPACT-3 and recent evaluation results are given in Chen et al. (2008). The AIRPACT-3 forecast results, along with automated evaluation results based upon AIRNOW monitoring data, are provided on a daily

basis on the AIRPACT web site (http://lar.wsu.edu/airpact-3). The AIRPACT-3 simulation results for nitrogen deposition (wet and dry) are output on the 12-km grid (described above and shown in Figure 1) and those data are used in this project. In addition, a diagnostic file is output by CMAQ, representing that part of the wet deposition that is associated with convective storm events. This convective wet deposition output is generated by CMAQ because meteorological modeling for scales such as associated with the 12-km grid used in AIRPACT-3 is well known to have problems predicting location for convective storms.

AIRPACT-3 results have been archived on a variety of media over the decade of its development and operation, from cartridge tapes to DVDs and most recently hard drives. The backups were stored on DVD for 2004 through 2007 and then on mountable external hard drives for 2008 and thereafter. Checking for data to cover the 28-month period of interest, June 2006 through September 2008, we determined that significant archive periods on DVD were unreadable and LAR therefore, the AIRPACT-3 system was re-run to generate data for missing periods, particularly the seven months from June 2007 to December 2007. This involved requesting that University of Washington restore the meteorology data (MM5 for that period) and rerunning AIRPACT.

For each day's CMAQ run, the previous day's local midnight pollutant concentration is used as the current day's initial condition. Since the AIRPACT-3 archive of May 2007 was missing, AIRPACT was rerun for a week to initialize the model domain chemistry to provide initial conditions to start the first day of June 2007. Overall, the process of rerunning for missing results took most of the time and much of the effort in this project; a CMAQ simulation for one day takes about 3 hours clock time plus the file transfer time between Washington State University and University of Washington.

2. Data Processing

The processing of the AIRPACT-3 meteorology and CMAQ data was accomplished in three stages:

- Summation of nitrogen species deposited into ten (12) deposition variables, for output for the entire grid, as hourly netCDF files. This process was run for both wet and dry deposition output files separately.. See the output species listed in Appendix A and the DepSum code in Appendix B.
- Extraction of hourly summed deposition variables for grid cells containing research sites (see Appendix C - CheckSites) resulting in hourly data by site as a text file.

- Reprocessing the deposition as reported by hour (see Appendix D - Trimming) for reporting as research site period and monthly values.

DepSum: This program, written in Fortran, sums deposited nitrogen species to an hourly basis. The list of accumulated nitrogen deposition species plus those nitrogen species reported individually is shown in Appendix A. This program sums the nitrogen-related species into the total nitrogen deposition and writes out the specified nitrogen-bearing species and total nitrogen deposition. For the accumulated nitrogen deposition, three more nitrogen-related species (PAN2, PBZN, and MA_PAN) are added to the total nitrogen deposition suggested in the proposal. The main reason is to include all nitrogen-deposited species for the accumulated nitrogen deposition. The input file from CMAQ (wet or dry deposition) result contains 58 chemical species with 24 time-steps of 1 hour. The output file is in netCDF format with 9 new variables in hourly basis. For the nitrate flux, NO and NO_2 are removed from the list appearing in the proposal scope of work because there is some thought that NO_2 might react to form HONO at the surfaces layer. NO and NO_2 are reported as individual species, and the data is available to add into nitrate flux if desired.

CheckSites: The script is modified to extract nitrogen deposition and the meteorology data for specific latitude and longitude. First, the program situates the site location provided by the project sponsors within the AIRPACT-3 domain. Then, the nitrogen deposition and meteorology data for the model grid cell within which that site is located are extracted. The output files for each site are written as ASCII format on an hourly basis.

Trimming: The bash script is written to winnow the hourly nitrogen deposition data for a specific site to the period of site operation. In the process the deposition data is also accumulated for reporting on monthly basis, as well as for the period of site operation. The output is in ASCII format and also MS Excel format. These data are being shared via the Dropbox website (www.dropbox.com) in lieu of FTP (file transfer protocol) as FTP support has been terminated at LAR/WSU.

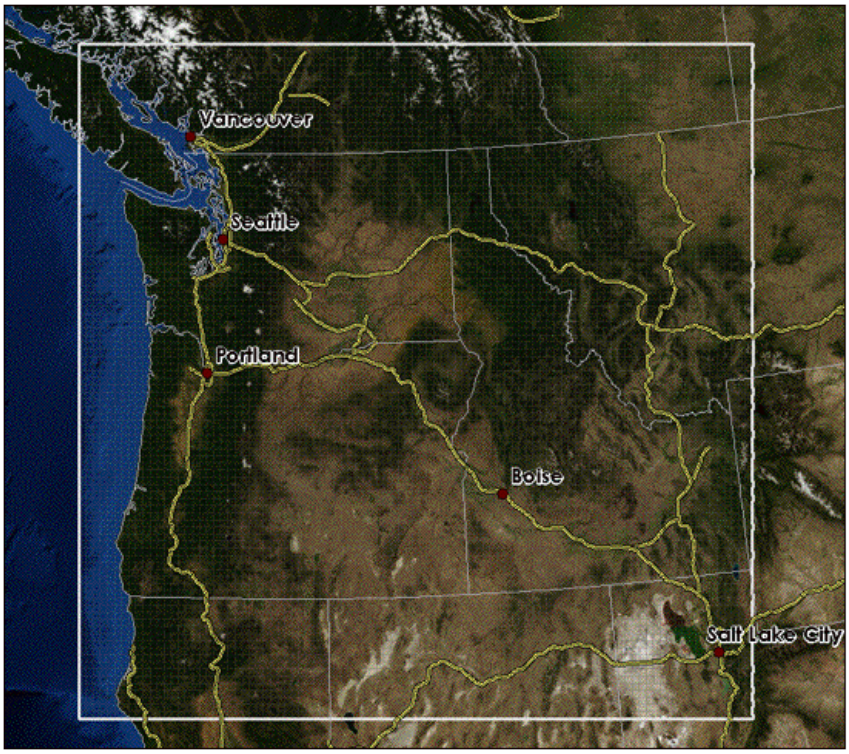

Figure 1. The AIRPACT -3 domain is indicated by the white border, and is represented as a grid of 95 by 95 cells of 12 km by 12 km. Vertical layers number 21 and vary in thickness, becoming deeper with increasing height above the surface and effectively including the entire troposphere.

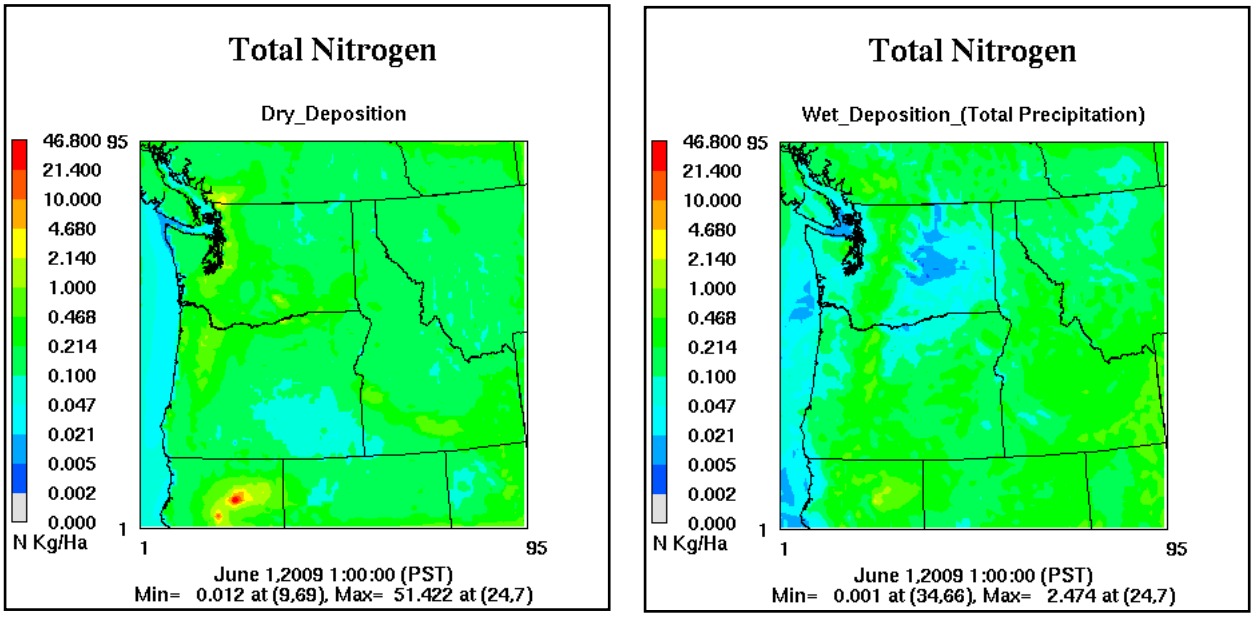

Figure 2. Examples for June 2009 showing AIRPACT-3 dry nitrogen deposition and wet nitrogen deposition.

References

Byun, D. and Schere, K. L.: Review of the Governing Equations, Computational Algorithms, and other components of the models-3 community multiscale air quality (CMAQ) modeling system, *Applied Mechanics Review*, Vol. **59**, March 2006, 51-77.

Chen, J., Vaughan, J., Avise, J., O'Neill, S., and Lamb, B.: Enhancement and evaluation of the AIRPACT ozone and PM2.5 forecast system for the Pacific Northwest, *Journal of Geophysical Research*, VOL. **113**, D14305, doi: 10.1029/2007JD009554, 2008

Mass, C. F., M. Albright, D. Ovens, R. Steed, M. MacIver, E. Grimit, T. Eckel, B. Lamb, J. Vaughan, K. Westrick, P. Storck, B. Colman, C. Hill, N. Maykut, M. Gilroy, S. A. Ferguson, J. Yetter, J. M. Sierchio, C. Bowman, R. Stender, R. Wilson and W. Brown, 2003: Regional Environmental Prediction over the Pacific Northwest, *The Bulletin of the American Meteorological Society*, **84**:10, 1353-1366.

Appendix A: Species documentation for site-specific text and MS Excel files.

This numbered list shows a descriptive name (optional, use for combined species), a short name based on the most important (or sole) nitrogen-bearing molecule (ion) , the units, followed by the AIRPACT CMAQ species contributing. In the CMAQ species definitions, "I" and "J" represent difference size modes of aerosol (ie., NH4I and NH4J); for deposition of aerosol these modes were summed in all cases.

1) Nitrogen Dry Deposition
NDRYDEP (N kg/ha) $NO_2 + NO + NO_3 + NH_3 + N_2O_5 + HNO_3 + HONO + RNO_3$ + PAN [1] + PAN2 [2] + PBZN [3] + MA_PAN [4]

2) Nitrogen Total Wet Deposition
NWET-Total (N kg/ha) $NO_2 + NO + NO_3 + NH_3 + N_2O_5 + HNO_3 + HONO + RNO_3$ + PAN + PAN2 + PBZN + MA_PAN

3) Nitrogen Wet Deposition from Convective Precipitation
NWET-Convective (N kg/ha) $NO_2 + NO + NO_3 + NH_3 + N_2O_5 + HNO_3 + HONO + RNO_3$ + PAN + PAN2 + PBZN + MA_PAN

4) Nitrate Dry Deposition
NO_3^+ (N kg/ha) $HNO_3 + NO + NO_2 + RNO_3 + ANO_3I + ANO_3J$

5) Ammonia Dry Deposition
NH_4^+ (N kg/ha) $NH_3 + ANH_4I + ANH_4J$

Other nitrogen-bearing species of a single molecule.

6) HNO_3 (N kg/ha) HNO_3
7) NO_2 (N kg/ha) NO_2
8) NO (N kg/ha) NO
9) NH_3 (N kg/ha) NH_3
10) $RHNO_3$ (N kg/ha) $RHNO_3$

11) Nitrate Wet Deposition
NO_3 (N kg/ha) $HNO_3 + NO + NO_2 + RNO_3 + ANO_3I + ANO_3J$

12) Ammonia Wet Deposition
NH_4^+ (N kg/ha) $NH_3 + ANH_4I + ANH_4J$

Meteorology Parameters
13) RC (cm) Convective Precipitation
14) RN (cm) Non-convective Precipitation
15) QC (kg/kg) Cloud Water Mixing Ratio [kg liquid water per kg air]
16) QI (kg/kg) Ice Mixing Ratio [kg ice per hg air]

[1] PAN: Peroxyacetyl Nitrate $CH_3C(O)\text{-}OONO_2$
[2] PAN2: Peroxypropionyl Nitrate $CH_3CH_2C(O)\text{-}OONO_2$
[3] PBZN: Peroxybenzoyl Nitrate $C_6H_5C(O)\text{-}OONO_2$
[4] MA_PAN: Methyl Peroxyacetyl Nitrate $CH_2C(CH_3)C(O)\text{-}OONO_2$

Appendix B: DepSum.F

```
        IMPLICIT NONE

! This is sum of the Deposition in hour basis.
! Pierre Wong April 19,2010

!=======[Includes]=====================================================

        INCLUDE 'PARMS3.EXT'    ! IOAPI Constants
        INCLUDE 'FDESC3.EXT'    ! IOAPI File Description Data Structure
        INCLUDE 'IODECL3.EXT'   ! IOAPI Function Declarations

!=======[External Functions]===========================================

        INTEGER     TRIMLEN
        real        getreal
        EXTERNAL    TRIMLEN, nextime, getreal

!=======[Local Variables]==============================================

        CHARACTER*160   MESG, endString
        CHARACTER*8     inFile, sumFile, progName

        INTEGER LOGDEV
        LOGICAL outIsNew

        INTEGER nxm, nym, nx, ny, nz
        INTEGER concDt, concTm
        INTEGER stdDt, stdTm
        INTEGER INnvars
        REAL dxkm, dykm

        INTEGER allocateStat
        REAL, DIMENSION(:,:,:), ALLOCATABLE :: aSumDep
        REAL, DIMENSION(:,:,:), ALLOCATABLE :: NPS_NO3, NPS_NH4
        REAL, DIMENSION(:,:,:), ALLOCATABLE :: NHNO3, NNO2, NNO
        REAL, DIMENSION(:,:,:), ALLOCATABLE :: NNH3, NRNO3

        REAL, DIMENSION(:,:,:), ALLOCATABLE :: NPS_NO3T, NPS_NH4T
        REAL, DIMENSION(:,:,:), ALLOCATABLE :: NHNO3T, NNO2T, NNOT
        REAL, DIMENSION(:,:,:), ALLOCATABLE :: NNH3T, NRNO3T

        INTEGER h, i, j, k, species

        INTEGER newnvars3d
        CHARACTER*16, PARAMETER :: NO2='NO2', NO= 'NO', NO3='NO3'
        CHARACTER*16, PARAMETER :: HNO3='HNO3', HONO='HONO', HNO4='HNO4'
        CHARACTER*16, PARAMETER :: RNO3='RNO3', PAN='PAN', PAN2='PAN2'
        CHARACTER*16, PARAMETER :: PBZN='PBZN', MA_PAN='MA_PAN'
        CHARACTER*16, PARAMETER :: NH3='NH3', NPHE='NPHE', ANH4J='ANH4J'
        CHARACTER*16, PARAMETER :: ANH4I='ANH4I', ANO3J='ANO3J'
```

```
        CHARACTER*16, PARAMETER :: ANO3I='ANO3I', SO2='SO2', N2O5='N2O5'
        CHARACTER*16, PARAMETER :: SULF='SULF', ASO4J='ASO4J'
        CHARACTER*16, PARAMETER :: ASO4I='ASO4I', ASO4K='ASO4K'
        CHARACTER*16, PARAMETER :: HG='HG', HGIIGAS='HGIIGAS'
        CHARACTER*16, PARAMETER :: APHGI='APHGI', APHGJ='APHGJ'
        CHARACTER*16, PARAMETER :: O3='O3', NA= 'NA', CL='CL'
        CHARACTER*16, PARAMETER :: ANAJ='ANAJ', ANAK='ANAK'
        CHARACTER*16, PARAMETER :: ACLJ='ACLJ', ACLK='ACLK'

        CHARACTER*16, PARAMETER :: NUNIT ='N Kg/Ha '

!=======[Main Program]==================================================

          progName = 'Dep_NPS'
          endString = "."

!!  Get the input and output file names.

          CALL GETARG(1, inFile)
          CALL GETARG(2, sumFile)

          LOGDEV = INIT3()

        write(6,*) "Opening the File"
!!  Open the input file from the AIRRUN Directories.
        IF(.NOT. OPEN3(inFile, FSREAD3, progName)) THEN
             MESG = 'Could not open file ' //  inFile
             CALL M3EXIT(inFile, 0, 0, MESG, 2)
        ELSE
            write(6,*) " Opened the inFile"
        END IF

        IF(.NOT. DESC3(inFile)) THEN
             MESG = 'Could not get file desc for ' // inFile
             CALL M3EXIT(inFile, 0, 0, MESG, 2)
        ELSE
             write(6,*) 'The inFile is read as'
             write(6,*) ' sdate3d: ', sdate3d
             write(6,*) ' stime3d: ', stime3d
             write(6,*) ' fdesc3d: ', fdesc3d(1)
             write(6,*) ' fdesc3d: ', fdesc3d(2)
             write(6,*) ' fdesc3d: ', fdesc3d(3)
             write(6,*) ' gdnam3d: ', gdnam3d
             write(6,*) ' xcell3d: ', xcell3d
             write(6,*) ' ycell3d: ', ycell3d
             write(6,*) ' ncols3d: ', ncols3d
             write(6,*) ' nrows3d: ', nrows3d
             write(6,*) ' tstep3d: ', tstep3d
             write(6,*) ' nvars3d: ', nvars3d
             write(6,*) ' nlays3d: ', nlays3d
             write(6,*) ' mxrec3d: ', mxrec3d

             concDt = SDATE3D
             concTm = STIME3D
```

```
            stdDt = SDATE3D
            stdTm = STIME3D
            INnvars = nvars3d
            nx = ncols3d
            ny = nrows3d

        ENDIF

           write(6,*) ' Got the description of inFile'

!!  Allocate memory for the reading and summing operations
!!  add new variables here
          write(6,*) "Allocating the variables1"

          ALLOCATE ( aSumDep (nx, ny, 1), stat = allocatestat )
              IF (allocatestat /= 0) STOP "Error allocating Array"
              aSumDep = 0.0

          write(6,*) "Allocating the variables2"
          ALLOCATE ( NPS_NO3 (nx, ny, 1), stat = allocatestat )
              IF (allocatestat /= 0) STOP "Error allocating Array"

          ALLOCATE ( NPS_NH4 (nx, ny, 1), stat = allocatestat )
              IF (allocatestat /= 0) STOP "Error allocating Array"

          ALLOCATE ( NHNO3 (nx, ny, 1), stat = allocatestat )
              IF (allocatestat /= 0) STOP "Error allocating Array"

          ALLOCATE ( NNO2 (nx, ny, 1), stat = allocatestat )
              IF (allocatestat /= 0) STOP "Error allocating Array"

          ALLOCATE ( NNO (nx, ny, 1), stat = allocatestat )
              IF (allocatestat /= 0) STOP "Error allocating Array"

          ALLOCATE ( NNH3 (nx, ny, 1), stat = allocatestat )
              IF (allocatestat /= 0) STOP "Error allocating Array"

          ALLOCATE ( NRNO3 (nx, ny, 1), stat = allocatestat )
              IF (allocatestat /= 0) STOP "Error allocating Array"

          ALLOCATE ( NPS_NO3T (nx, ny, 24), stat = allocatestat )
              IF (allocatestat /= 0) STOP "Error allocating Array"

          ALLOCATE ( NPS_NH4T (nx, ny, 24), stat = allocatestat )
              IF (allocatestat /= 0) STOP "Error allocating Array"

          ALLOCATE ( NHNO3T (nx, ny, 24), stat = allocatestat )
              IF (allocatestat /= 0) STOP "Error allocating Array"

          ALLOCATE ( NNO2T (nx, ny, 24), stat = allocatestat )
              IF (allocatestat /= 0) STOP "Error allocating Array"

          ALLOCATE ( NNOT (nx, ny, 24), stat = allocatestat )
              IF (allocatestat /= 0) STOP "Error allocating Array"
```

```
          ALLOCATE ( NNH3T (nx, ny, 24), stat = allocatestat )
              IF (allocatestat /= 0) STOP "Error allocating Array"

          ALLOCATE ( NRNO3T (nx, ny, 24), stat = allocatestat )
              IF (allocatestat /= 0) STOP "Error allocating Array"


        write(6,*) "Finish allocating"


!!  over 24 hours
        DO h = 1, 24
             write(6,*) 'reading in  hour: ', h
          NPS_NO3 = 0
          NPS_NH4 = 0
          NHNO3 = 0
          NNO2 = 0
          NNO = 0
          NNH3 = 0
          NRNO3 = 0

!!  over species
          DO species = 1, INnvars
             write(6,*) ' DOING SPECIES: ',VNAME3D(species)
             IF(.NOT.
READ3(inFile,VNAME3D(species),1,concDt,concTm,aSumDep)) THEN
                    MESG = 'Failed to read inFile: ' // inFile
                    CALL M3EXIT(inFile, concDt, concTm, 2)
             ENDIF

          SELECT CASE (VNAME3D(species))
            !N GAS
              CASE (HNO3)
                  NPS_NO3  = NPS_NO3  + aSumDep*14.0067/63.0128
                  NHNO3  = NHNO3  + aSumDep*14.0067/63.0128
              CASE (NO2)
                  NNO2  = NNO2  + aSumDep*14.0067/46.0055
              CASE (NO)
                  NNO  = NNO  + aSumDep*14.0067/30.0061
              CASE (NH3)
                  NPS_NH4  = NPS_NH4  + aSumDep*14.0067/17.0304
                  NNH3  = NNH3  + aSumDep*14.0067/17.0304
              CASE (RNO3) !organic nitrates
                  NPS_NO3  = NPS_NO3  + aSumDep*14.0067/133
                  NRNO3  = NRNO3  + aSumDep*14.0067/133

            !N AEROSOL
              CASE (ANH4J)
                  NPS_NH4  = NPS_NH4  + aSumDep*14.0067/18.0383
              CASE (ANH4I)
                  NPS_NH4  = NPS_NH4  + aSumDep*14.0067/18.0383
              CASE (ANO3J)
                  NPS_NO3  = NPS_NO3  + aSumDep*14.0067/63.0128
```

```
            CASE (ANO3I)
                NPS_NO3  = NPS_NO3  + aSumDep*14.0067/63.0128

        END SELECT

        ENDDO  ! Finish species 1-58


          write(6,*) 'Allocaling in  hour: ', h
          DO i = 1, nx
          DO j = 1, ny
            NPS_NO3T(i, j, h) = NPS_NO3(i, j, 1)
            NPS_NH4T(i, j, h) = NPS_NH4(i, j, 1)
            NHNO3T(i, j, h) = NHNO3(i, j, 1)
            NNO2T(i, j, h) = NNO2(i, j, 1)
            NNOT(i, j, h) = NNO(i, j, 1)
            NNH3T(i, j, h) = NNH3(i, j, 1)
            NRNO3T(i, j, h) = NRNO3(i, j, 1)
          ENDDO
          ENDDO


        call nextime(concDt,concTm,TSTEP3D)
      ENDDO !h = 1, 24

      write(6,*) "Done of reading the inFile"



!!  New Setting for New Species
                nvars3d = 7
                concDt = stdDt
                concTm = stdTm

!!  New Species
      write(6,*) "Setting the New Species"
        VTYPE3D(1) = M3REAL
        VNAME3D(1) = 'NPS_NO3_Flux'
        UNITS3D(1) = NUNIT
        VDESC3D(1) = 'HNO3 + R-NO3 + Aerosol NO3'

        VTYPE3D(2) = M3REAL
        VNAME3D(2) = 'NPS_NH4_Flux'
        UNITS3D(2) = NUNIT
        VDESC3D(2) = 'Gasous NH3 + Aerosol NH4'

        VTYPE3D(3) = M3REAL
        VNAME3D(3) = 'HNO3'
        UNITS3D(3) = NUNIT
        VDESC3D(3) = 'HNO3'

        VTYPE3D(4) = M3REAL
        VNAME3D(4) = 'NO2'
```

```
          UNITS3D(4) = NUNIT
          VDESC3D(4) = 'NO2'

          VTYPE3D(5) = M3REAL
          VNAME3D(5) = 'NO'
          UNITS3D(5) = NUNIT
          VDESC3D(5) = 'NO'

          VTYPE3D(6) = M3REAL
          VNAME3D(6) = 'NH3'
          UNITS3D(6) = NUNIT
          VDESC3D(6) = 'Gasous NH3'

          VTYPE3D(7) = M3REAL
          VNAME3D(7) = 'RNO3'
          UNITS3D(7) = NUNIT
          VDESC3D(7) = 'RNO3'

!!  Attempt to open the sumput file for RDWR.
!!  else Open as NEW and call DESC3.

          IF(.NOT. OPEN3(sumFile, FSRDWR3, progName)) THEN

            IF(.NOT. OPEN3(sumFile, FSNEW3, progName)) THEN
                write(6,*) 'sumFile could not be opened as NEW'
                write(6,*) 'sumFile is existed'
                MESG = 'Could not open new file ' //  sumFile
                CALL M3EXIT(sumFile, 0, 0, MESG, 2)
            ELSE
                write(6,*) 'File opened as NEW'
                write(6,*) 'File starts at the firs day of month'
            ENDIF !For OPEN3 as new

          ELSE
                write(6,*) 'sumFile opened as NEW'
          ENDIF !For OPEN3 as read and write

          IF(.NOT. DESC3(sumFile)) THEN
                write(6,*) 'Could not get file desc for ' // sumFile
                MESG = 'Could not fet file desc for sum file ' // sumFile
                CALL M3EXIT(sumFile, 0, 0, MESG, 2)
          ELSE
                write(6,*) 'SumFile Records == ', mxrec3d
                write(6,*) ' sdate3d: ', sdate3d
                write(6,*) ' stime3d: ', stime3d
                write(6,*) ' fdesc3d: ', fdesc3d(1)
                write(6,*) ' fdesc3d: ', fdesc3d(2)
                write(6,*) ' fdesc3d: ', fdesc3d(3)
                write(6,*) ' gdnam3d: ', gdnam3d
                write(6,*) ' xcell3d: ', xcell3d
                write(6,*) ' ycell3d: ', ycell3d
                write(6,*) ' ncols3d: ', ncols3d
                write(6,*) ' nrows3d: ', nrows3d
```

```
                nvars3d = 7

                write(6,*) ' tstep3d: ', tstep3d
                write(6,*) ' nvars3d: ', nvars3d
                write(6,*) ' nlays3d: ', nlays3d
                write(6,*) ' mxrec3d: ', mxrec3d
            ENDIF !For DECS3



!!  Writing for 24 hr
          DO h = 1, 24
                write(6,*) 'writing in hour: ', h

                NPS_NO3 = 0
                NPS_NH4 = 0
                NHNO3 = 0
                NNO2 = 0
                NNO = 0
                NNH3 = 0
                NRNO3 = 0

              DO i = 1, nx
              DO j = 1, ny
                NPS_NO3(i, j, 1) = NPS_NO3T(i, j, h)
                NPS_NH4(i, j, 1) = NPS_NH4T(i, j, h)
                NHNO3(i, j, 1) = NHNO3T(i, j, h)
                NNO2(i, j, 1) = NNO2T(i, j, h)
                NNO(i, j, 1) = NNOT(i, j, h)
                NNH3(i, j, 1) = NNH3T(i, j, h)
                NRNO3(i, j, 1) = NRNO3T(i, j, h)

              ENDDO ! i = 1, nx
              ENDDO !j = 1, ny

                write(6,*) 'WHATPROB NHNO3 ',h, NHNO3(50, 50, 1)
                write(6,*) 'WHATPROB  NNO2 ',h, NNO2(50, 50, 1)
                write(6,*) 'WHATPROB  NNO ',h, NNO(50, 50, 1)


            DO i = 1, 7
                SELECT CASE (i)
                    CASE (1)
                        aSumDep = NPS_NO3
                    CASE (2)
                        aSumDep = NPS_NH4
                    CASE (3)
                        aSumDep = NHNO3
                    CASE (4)
                        aSumDep = NNO2
                    CASE (5)
                        aSumDep = NNO
                    CASE (6)
                        aSumDep = NNH3
```

```
                   CASE (7)
                       aSumDep = NRNO3
               END SELECT


               write(6,*) ' DOING SPECIES: VNAME3D(i)',VNAME3D(i)
               write(6,*) ' DOING SPECIES: VTYPE3D(i)',VTYPE3D(i)
               write(6,*) ' DOING SPECIES: UNITS3D(i)',UNITS3D(i)
               write(6,*) ' DOING SPECIES: VDESC3D(i)',VDESC3D(i)

               IF( .NOT. WRITE3(sumFile, VNAME3D(i), concDt, concTm,
aSumDep)) THEN
                    MESG = 'Failed to write sumFile ' // sumFile
                    write(6,*) ' Calling M3EXIT() due to: ', MESG
                    CALL M3EXIT(sumFile, concDt, concTm, 2)
              ELSE
                    write(6,*) 'Successfully Wrote sumFile for ' //
VNAME3D(i)
               ENDIF

          ENDDO   !for looping i=1,7

          call nextime(concDt,concTm,TSTEP3D)
          ENDDO ! on over 24 hours



          IF (.NOT. SHUT3()) THEN
               write(6,*) 'Critical Error, IOAPI Unable to Shutdown'
          ENDIF

      END
```

Appendix C: CheckSite.F

```
      PROGRAM CheckSITES

!      PWong March 16, 2010
!      Read the sumFiles from DepSum.F
!      Then, convert the SITE from (lat/long) to grid cell
!      Write out the files for specific site in ASCII format
      IMPLICIT NONE

!...........   INCLUDES:

      INCLUDE 'PARMS3.EXT'      ! I/O API constants
      INCLUDE 'FDESC3.EXT'      ! I/O API file description data structure
      INCLUDE 'IODECL3.EXT'     ! I/O API function declarations

!...........   EXTERNAL FUNCTIONS and their descriptions:

      INTEGER   TRIMLEN, MMDDYY
      real      getreal
      EXTERNAL  TRIMLEN, nextime, getreal, MMDDYY

!...........   LOCAL VARIABLES and their descriptions:

      character*16  ::  PROGNAME = 'ChecksSITES.F   '
      character*22  ::  FILENAME
      character*160  MESG, HEADER1
      character*8    sumFile, sumFile1, sumFile2
      character*8    NPSD, NPSWT
      character*8    METCRO2D, METCRO3D, METDOT3D
      character*16   GRIDCRO2D
      integer    LOGDEV
      integer    YYYY, DDD, GregDate, YY, MM, DD, HH, MN, SS
      integer    MXSITES
      parameter (MXSITES = 500)
COLS+&   +    5    2    5    3    5    4    5    5    5    6    5    712
      integer allocateStat, iost
      integer i, j, k, l, m, n, ihr
      integer N_SITES
      integer readDate, readTime
      integer writeDate, writeTime
      integer stdDate, stdTime
      real*8  Lati, Longi, HT
      real    NDRY, NWET1, NWET2
      real    DNO3F, DNH4F, WNO3F, WNH4F
      real    WriteHNO3, WriteNO2, WriteNO
      real    WriteNH3, WriteRNO3
      real    WriteRC, WriteRN, WriteQC, WriteQI
      real, dimension(:,:,:), allocatable :: CONC_IN
      real, dimension(:,:,:), allocatable :: NDRY_IN
      real, dimension(:,:,:), allocatable :: NWET1_IN
      real, dimension(:,:,:), allocatable :: NWET2_IN
      real, dimension(:,:,:), allocatable :: DNO3F_IN
```

```
      real, dimension(:,:,:), allocatable :: DNH4F_IN
      real, dimension(:,:,:), allocatable :: WNO3F_IN
      real, dimension(:,:,:), allocatable :: WNH4F_IN
      real, dimension(:,:,:), allocatable :: HNO3_IN
      real, dimension(:,:,:), allocatable :: NO2_IN
      real, dimension(:,:,:), allocatable :: NO_IN
      real, dimension(:,:,:), allocatable :: NH3_IN
      real, dimension(:,:,:), allocatable :: RNO3_IN
      real, dimension(:,:,:), allocatable :: RC_IN
      real, dimension(:,:,:), allocatable :: RN_IN
      real, dimension(:,:,:), allocatable :: QC_IN
      real, dimension(:,:,:), allocatable :: QI_IN
      character*9,  dimension(MXSITES) :: SiteCode
      integer,      dimension(MXSITES) :: ROW
      integer,      dimension(MXSITES) :: COL
      real,         dimension(MXSITES) :: LAT
      real,         dimension(MXSITES) :: LONG
      character*16, parameter :: NTOT='NTOT'
      character*16, parameter :: NPS_NO3Flux='NPS_NO3_Flux'
      character*16, parameter :: NPS_NH4Flux='NPS_NH4_Flux'
      character*16, parameter :: RN= 'RN', RC='RC'
      character*16, parameter :: QI='QI', QC= 'QC'
      character*16, parameter :: HNO3= 'HNO3', NO2='NO2'
      character*16, parameter :: NO='NO', NH3= 'NH3', RNO3= 'RNO3'


COLS+&   +    5    2    5    3    5    4    5    5    5    6    5
7123456789$
COLS+&   +    5    2    5    3    5    4    5    5    5    6    5    712
!*********************************************************************

!.... Initialize the I/O API:
      LOGDEV = INIT3()        !  initialization returns unit # for log

! Other initialization tasks

      CALL GETARG(1, METCRO2D)
      CALL GETARG(2, METCRO3D)
      CALL GETARG(3, sumFile)
      CALL GETARG(4, sumFile1)
      CALL GETARG(5, sumFile2)
      CALL GETARG(6, GRIDCRO2D)
      CALL GETARG(7, NPSD)
      CALL GETARG(8, NPSWT)


      Lati=  40.095
      Longi= -125.220
      write(6,*) 'Southwest corner is ', Lati,Longi
      call ll2cell(Lati, Longi, k, n, ht)
      write(6,*) 'Done!!! with southwest corner!!'

      Lati=  50.569
      Longi= -126.088
```

```
      write(6,*) 'Northwest corner is ', Lati,Longi
      Call ll2cell(Lati,Longi,k,n,ht)

      Lati=  50.158
      Longi= -109.657
      write(6,*) 'Northeast corner is ', Lati,Longi
      Call ll2cell(Lati,Longi,k,n,ht)

      Lati= 39.754
      Longi= -111.543
      write(6,*) 'Southeast corner is ', Lati,Longi
      Call ll2cell(Lati,Longi,k,n,ht)

      write(6,*) 'Done with four boundary corncers'

!!!!!!Openning the files
      write(6,*) '###### Openning sumFile1###### '

      IF(.NOT. OPEN3(sumFile1, FSREAD3, PROGNAME)) THEN
          MESG = 'Could not open file ' //  sumFile1
          CALL M3EXIT(sumFile1, 0, 0, MESG, 2)
      END IF

      write(6,*) '###### Openning sumFile2###### '
      IF(.NOT. OPEN3(sumFile2, FSREAD3, PROGNAME)) THEN
          MESG = 'Could not open file ' //  sumFile2
          CALL M3EXIT(sumFile2, 0, 0, MESG, 2)
      END IF

      write(6,*) '###### Openning NPSD###### '
      IF(.NOT. OPEN3(NPSD, FSREAD3, PROGNAME)) THEN
          MESG = 'Could not open file ' //  NPSD
          CALL M3EXIT(NPSD, 0, 0, MESG, 2)
      END IF

      write(6,*) '###### Openning NPSD###### '
      IF(.NOT. OPEN3(NPSWT, FSREAD3, PROGNAME)) THEN
          MESG = 'Could not open file ' //  NPSWT
          CALL M3EXIT(NPSWT, 0, 0, MESG, 2)
      END IF

      write(6,*) '###### Openning METCRO2D###### '
      IF(.NOT. OPEN3(METCRO2D, FSREAD3, PROGNAME)) THEN
          MESG = 'Could not open file ' //  METCRO2D
          CALL M3EXIT(METCRO2D, 0, 0, MESG, 2)
      END IF
      IF(.NOT. DESC3(METCRO2D)) THEN
          MESG = 'Could not get file desc for ' // METCRO2D
          CALL M3EXIT(METCRO2D, 0, 0, MESG, 2)
      END IF

      write(6,*) '###### Openning METCRO3D###### '
      IF(.NOT. OPEN3(METCRO3D, FSREAD3, PROGNAME)) THEN
          MESG = 'Could not open file ' //  METCRO3D
```

```
          CALL M3EXIT(METCRO3D, 0, 0, MESG, 2)
        END IF
        IF(.NOT. DESC3(METCRO3D)) THEN
          MESG = 'Could not get file desc for ' // METCRO3D
          CALL M3EXIT(METCRO3D, 0, 0, MESG, 2)
        END IF

        write(6,*) '###### Openning sumFile###### '
        IF(.NOT. OPEN3(sumFile, FSREAD3, PROGNAME)) THEN
          MESG = 'Could not open file ' //  sumFile
          CALL M3EXIT(sumFile, 0, 0, MESG, 2)
        END IF
        IF(.NOT. DESC3(sumFile)) THEN
          MESG = 'Could not get file desc for ' // sumFile
          CALL M3EXIT(sumFile, 0, 0, MESG, 2)
        ELSE
          write(6,*) 'The inFile is read as'
          write(6,*) ' sdate3d: ', SDATE3d
          write(6,*) ' stime3d: ', STIME3d
          readDate = SDATE3D
          readTime = STIME3D
          writeDate = SDATE3D
          writeTime = STIME3D
          stdDate = SDATE3D
          stdTime = STIME3D
        END IF

!!!!!!read in the sites file, storing and counting
10      OPEN(11, IOSTAT=iost,FILE='SITE', STATUS='old')
        read(11,'(A80)',IOSTAT=iost) HEADER1
        write(6,*) HEADER1
        l = 1
40      read(11,*,END=45) SiteCode(l), LAT(l), LONG(l)
        write(6,*)'read from unit 11', l, SiteCode(l),LAT(l),LONG(l)
           Lati       = LAT(l)
           Longi      = LONG(l)
           Call ll2cell(Lati, Longi, k, n, ht)
           COL(l) = k
           ROW(l) = n

           if ((ROW(l).GE.1).AND.(ROW(l).LE.NROWS3D).AND.(COL(l).GE.1)
     &          .AND.(COL(l).LE.NCOLS3D)) THEN
             write(6,29) l, SiteCode(l),LAT(l),LONG(l),COL(l),ROW(l)
29           FORMAT(i3,xx,a9,' lat: ',f9.3,xx,' long: ',f9.3,xx,' col: ',
     &             i3.3,xx,' row: ',i3.3)
             l = l + 1
           else
             write(6,*) ' vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv '
             write(6,*) ' ERROR!! THIS SITE IS NOT IN DOMAIN!!SKIP IT !!'
             write(6,29) l, SiteCode(l),LAT(l),LONG(l),COL(l),ROW(l)
             write(6,*) ' ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^ '
           endif
        goto 40
45      N_SITES = l - 1
```

```
      write(6,*) ' SITES FOUND: ' , N_SITES

!!!!!!allocate variables
      write(6,*) '###### Allocating the variables###### '

      ALLOCATE ( CONC_IN  (95, 95, 1), stat = allocateStat)
      ALLOCATE ( NDRY_IN  (N_SITES, 24, 1), stat = allocateStat )
      ALLOCATE ( NWET1_IN (N_SITES, 24, 1), stat = allocateStat )
      ALLOCATE ( NWET2_IN (N_SITES, 24, 1), stat = allocateStat )
      ALLOCATE ( DNO3F_IN (N_SITES, 24, 1), stat = allocateStat )
      ALLOCATE ( DNH4F_IN (N_SITES, 24, 1), stat = allocateStat )
      ALLOCATE ( WNO3F_IN (N_SITES, 24, 1), stat = allocateStat )
      ALLOCATE ( WNH4F_IN (N_SITES, 24, 1), stat = allocateStat )
      ALLOCATE ( HNO3_IN (N_SITES, 24, 1), stat = allocateStat )
      ALLOCATE ( NO2_IN (N_SITES, 24, 1), stat = allocateStat )
      ALLOCATE ( NO_IN (N_SITES, 24, 1), stat = allocateStat )
      ALLOCATE ( NH3_IN (N_SITES, 24, 1), stat = allocateStat )
      ALLOCATE ( RNO3_IN (N_SITES, 24, 1), stat = allocateStat )
      ALLOCATE ( RC_IN (N_SITES, 24, 1), stat = allocateStat )
      ALLOCATE ( RN_IN (N_SITES, 24, 1), stat = allocateStat )
      ALLOCATE ( QC_IN (N_SITES, 24, 1), stat = allocateStat )
      ALLOCATE ( QI_IN (N_SITES, 24, 1), stat = allocateStat )



!!!!!!Start to read the variables from sumFile and write out for each site
      write(6,*) '###### Reading the sumfiles###### '

      write(6,*) ' Started date ', readDate
      write(6,*) ' Started time ', readTime
      write(6,*) ' MXREC3D ',      MXREC3D

      DO ihr = 1, MXREC3D

        write(6,*) ' Next    date ', readDate
        write(6,*) ' Next    time ', readTime

!!!DRYDEP - Total Nitrogen Deposition
        IF (.NOT.READ3(sumFile, NTOT, 1, readDate, readTime, CONC_IN))
THEN
             MESG = 'Failed Reading NDRYDEP from File'
             write(6,*) ' CALLING M3EXIT due to ',MESG
        ELSE
           DO l = 1, N_SITES   ! loop for each site
             NDRY_IN(l, ihr, 1) = CONC_IN(COL(l),ROW(l),1)
           ENDDO   !loop for each site
           CONC_IN = 0
        ENDIF

!!!WETDEP_T - Total Nitrogen Deposition
        IF (.NOT.READ3(sumFile1, NTOT, 1, readDate, readTime, CONC_IN))
THEN
             MESG = 'Failed Reading NWETDEP1 from File'
             write(6,*) ' CALLING M3EXIT due to ',MESG
```

```
         ELSE
            DO l = 1, N_SITES   ! loop for each site
              NWET1_IN(l, ihr, 1) = CONC_IN(COL(l),ROW(l),1)
            ENDDO   !loop for each site
            CONC_IN = 0
         ENDIF

!!!WETDEP_C - Total Nitrogen Deposition
         IF (.NOT.READ3(sumFile2,NTOT,1,readDate,readTime,CONC_IN)) THEN
              MESG = 'Failed Reading NWETDEP2 from File'
              write(6,*) ' CALLING M3EXIT due to ',MESG
         ELSE
            DO l = 1, N_SITES   ! loop for each site
              NWET2_IN(l, ihr, 1) = CONC_IN(COL(l),ROW(l),1)
            ENDDO   !loop for each site
            CONC_IN = 0
         ENDIF


!!!NPSD - DRY Nitrate_Flux Deposition
         IF (.NOT.READ3(NPSD, NPS_NO3Flux, 1, readDate, readTime,
     &      CONC_IN)) THEN
              MESG = 'Failed Reading NDRYDEP from File'
              write(6,*) ' CALLING M3EXIT due to ',MESG
         ELSE
            DO l = 1, N_SITES   ! loop for each site
              DNO3F_IN(l, ihr, 1) = CONC_IN(COL(l),ROW(l),1)
            ENDDO   !loop for each site
            CONC_IN = 0
         write(6,*) ihr, '###### Finish reading NPS_NO3-Flux ',
     &                        'from the NPSD######'
         ENDIF

!!!NPSD - DRY Ammonia_Flux Deposition
         IF (.NOT.READ3(NPSD, NPS_NH4Flux, 1, readDate, readTime,
     &      CONC_IN)) THEN
              MESG = 'Failed Reading NDRYDEP from File'
              write(6,*) ' CALLING M3EXIT due to ',MESG
         ELSE
            DO l = 1, N_SITES   ! loop for each site
              DNH4F_IN(l, ihr, 1) = CONC_IN(COL(l),ROW(l),1)
            ENDDO   !loop for each site
            CONC_IN = 0
         write(6,*) ihr, '###### Finish reading NPS_NH4-Flux ',
     &                        'from the NPSD######'
         ENDIF

!!!NPSD - DRY HNO3 Deposition
         IF (.NOT.READ3(NPSD, HNO3, 1, readDate, readTime,
     &      CONC_IN)) THEN
              MESG = 'Failed Reading NDRYDEP from File'
              write(6,*) ' CALLING M3EXIT due to ',MESG
         ELSE
            DO l = 1, N_SITES   ! loop for each site
```

```
              HNO3_IN(l, ihr, 1) = CONC_IN(COL(l),ROW(l),1)
            ENDDO   !loop for each site
            CONC_IN = 0
          write(6,*) ihr, '###### Finish reading HNO3 ',
     &                       'from the NPSD######'
          ENDIF

!!!NPSD - DRY NO2 Deposition
          IF (.NOT.READ3(NPSD, NO2, 1, readDate, readTime,
     &      CONC_IN)) THEN
              MESG = 'Failed Reading NDRYDEP from File'
              write(6,*) ' CALLING M3EXIT due to ',MESG
          ELSE
            DO l = 1, N_SITES   ! loop for each site
              NO2_IN(l, ihr, 1) = CONC_IN(COL(l),ROW(l),1)
            ENDDO   !loop for each site
            CONC_IN = 0
          write(6,*) ihr, '###### Finish reading NO2 ',
     &                       'from the NPSD######'
          ENDIF

!!!NPSD - DRY NO Deposition
          IF (.NOT.READ3(NPSD, NO, 1, readDate, readTime,
     &      CONC_IN)) THEN
              MESG = 'Failed Reading NDRYDEP from File'
              write(6,*) ' CALLING M3EXIT due to ',MESG
          ELSE
            DO l = 1, N_SITES   ! loop for each site
              NO_IN(l, ihr, 1) = CONC_IN(COL(l),ROW(l),1)
            ENDDO   !loop for each site
            CONC_IN = 0
          write(6,*) ihr, '###### Finish reading NO ',
     &                       'from the NPSD######'
          ENDIF

!!!NPSD - DRY NH3 Deposition
          IF (.NOT.READ3(NPSD, NH3, 1, readDate, readTime,
     &      CONC_IN)) THEN
              MESG = 'Failed Reading NDRYDEP from File'
              write(6,*) ' CALLING M3EXIT due to ',MESG
          ELSE
            DO l = 1, N_SITES   ! loop for each site
              NH3_IN(l, ihr, 1) = CONC_IN(COL(l),ROW(l),1)
            ENDDO   !loop for each site
            CONC_IN = 0
          write(6,*) ihr, '###### Finish reading NH4 ',
     &                       'from the NPSD######'
          ENDIF

!!!NPSD - DRY RNO3 Deposition
          IF (.NOT.READ3(NPSD, RNO3, 1, readDate, readTime,
     &      CONC_IN)) THEN
              MESG = 'Failed Reading NDRYDEP from File'
              write(6,*) ' CALLING M3EXIT due to ',MESG
```

```
         ELSE
            DO l = 1, N_SITES   ! loop for each site
              RNO3_IN(l, ihr, 1) = CONC_IN(COL(l),ROW(l),1)
            ENDDO   !loop for each site
            CONC_IN = 0
         write(6,*) ihr, '###### Finish reading RNO3 ',
     &                       'from the NPSD######'
         ENDIF

!!!NPSWT - WET Nitrate_Flux Deposition
         IF (.NOT.READ3(NPSWT, NPS_NO3Flux, 1, readDate, readTime,
     &      CONC_IN)) THEN
              MESG = 'Failed Reading NDRYDEP from File'
              write(6,*) ' CALLING M3EXIT due to ',MESG
         ELSE
            DO l = 1, N_SITES   ! loop for each site
              WNO3F_IN(l, ihr, 1) = CONC_IN(COL(l),ROW(l),1)
            ENDDO   !loop for each site
            CONC_IN = 0
         write(6,*) ihr, '###### Finish reading NPS_NO3-Flux ',
     &                       'from the NPSWT######'
         ENDIF

!!!NPSWT - WET Ammonia_Flux Deposition
         IF (.NOT.READ3(NPSWT, NPS_NH4Flux, 1, readDate, readTime,
     &      CONC_IN)) THEN
              MESG = 'Failed Reading NDRYDEP from File'
              write(6,*) ' CALLING M3EXIT due to ',MESG
         ELSE
            DO l = 1, N_SITES   ! loop for each site
              WNH4F_IN(l, ihr, 1) = CONC_IN(COL(l),ROW(l),1)
            ENDDO   !loop for each site
            CONC_IN = 0
         write(6,*) ihr, '###### Finish reading NPS_NH4+Flux ',
     &                       'from the NPSWT######'
         ENDIF

         write(6,*) ihr, '###### Finish reading Deposition from sumFile',
     &                       '######'

         call nextime(readDate,readTime,TSTEP3D)

       ENDDO   !END Looping for reading the sumDep files in ihr


!!!!!!Start to read the variables from METFile and write out for each site
!!!!!!METFile has 25 time steps and begin at 8am,
!!!!!!Therefore, call nextime to set the time beginning at 9am.

      write(6,*) '###### Reading the metfiles###### '


         readDate = stdDATE
         readTime = stdTIME
```

```
!      write(6,*) ' Started date ', readDate
!      write(6,*) ' Started time ', readTime
!      write(6,*) ' MXREC3D ',      MXREC3D
!      write(6,*) '#%^%#^#^%^%# Before resetting #%^%#^#%^#%^ '

!        call nextime(readDate,readTime,TSTEP3D)

       write(6,*) ' Started date ', readDate
       write(6,*) ' Started time ', readTime
       write(6,*) ' MXREC3D ',      MXREC3D
       write(6,*) '#%^%#^#^%^%# After resetting #%^%#^#%^#%^ '


       DO ihr = 1, MXREC3D

         write(6,*) ' Next    date ', readDate
         write(6,*) ' Next    time ', readTime

         IF (.NOT.READ3(METCRO2D,RN,1,readDate,readTime,CONC_IN)) THEN
              MESG = 'Failed Reading RN from File'
              write(6,*) ' CALLING M3EXIT due to ',MESG
         ELSE
            DO l = 1, N_SITES   ! loop for each site
              RN_IN(l, ihr, 1) = CONC_IN(COL(l),ROW(l),1)
            ENDDO   !loop for each site
            CONC_IN = 0
         ENDIF
         write(6,*) ihr, '###### Finish reading RN from METCRO2D',
      &                       'DDDDDDDDDD######'

         IF (.NOT.READ3(METCRO2D,RC,1,readDate,readTime,CONC_IN)) THEN
              MESG = 'Failed Reading RC from File'
              write(6,*) ' CALLING M3EXIT due to ',MESG
         ELSE
            DO l = 1, N_SITES   ! loop for each site
              RC_IN(l, ihr, 1) = CONC_IN(COL(l),ROW(l),1)
            ENDDO   !loop for each site
            CONC_IN = 0
         ENDIF
         write(6,*) ihr, '###### Finish reading RC from METCRO2D',
      &                       'DDDDDDDDDD######'

         IF (.NOT.READ3(METCRO3D,QC,1,readDate,readTime,CONC_IN)) THEN
              MESG = 'Failed Reading QC from File'
              write(6,*) ' CALLING M3EXIT due to ',MESG
         ELSE
            DO l = 1, N_SITES   ! loop for each site
              QC_IN(l, ihr, 1) = CONC_IN(COL(l),ROW(l),1)
            ENDDO   !loop for each site
            CONC_IN = 0
         ENDIF
         write(6,*) ihr, '###### Finish reading QC from METCRO3',
      &                       '333333333333DDDDDDDDDD######'
```

```
        IF (.NOT.READ3(METCRO3D,QI,1,readDate,readTime,CONC_IN)) THEN
             MESG = 'Failed Reading QI from File'
             write(6,*) ' CALLING M3EXIT due to ',MESG
        ELSE
           DO l = 1, N_SITES   ! loop for each site
             QI_IN(l, ihr, 1) = CONC_IN(COL(l),ROW(l),1)
           ENDDO   !loop for each site
           CONC_IN = 0
        ENDIF
        write(6,*) ihr, '###### Finish reading QI from METCRO3',
     &                      '333333333333DDDDDDDDDD######'

        call nextime(readDate,readTime,TSTEP3D)

       ENDDO   !END Looping for reading the file in ihr


!!!!!!Write out the above data into ASCII format

      write(6,*) '###### Writing the files###### '

       DO l = 1, N_SITES   ! loop for each site

         writeDate = stdDate
         writeTime = stdTime

         write(6,*) 'Now is process for SiteCode:', SiteCode(l)
         write(6,*) 'The Starting Time is:', writeDate, writeTime


             DO ihr = 1, 24

70           write(6,*) 'IHR: ',ihr
             write(6,*) ' Julian Date and Time: ',writeDate, writeTime
             YYYY = INT(writeDate / 1000)
             DDD = writeDate - (YYYY * 1000)
             call DAYMON( writeDate, MM, DD)
             YY = imod(YYYY, 100)
             HH = int ( writeTime/10000)
             MN = int ( (writeTime - HH * 10000)/100 )
             SS = writeTime - HH * 10000 - MN * 100  ! = 00
49           FORMAT(' Gregorian Date and Time: ',i4.2,
     &            i2.2,'/',i2.2,'/',i2.2,xx,i2.2,':',i2.2,':',i2.2, )
             write(6,49) YYYY, YY, MM, DD, HH, MN, SS

             IF (ihr == 1) THEN
                write(6,*) 'Open a new file for: ', SiteCode(l)
                FILENAME = '                    '
50              FORMAT(i2.2, '.', i4.4, 2i2.2)
                write(FILENAME,50) l, YYYY, MM, DD  ! l.YYYYMMDD
                OPEN(20,FILE=FILENAME,FORM='FORMATTED',STATUS='UNKNOWN')
                close(20)
                FILENAME = '                    '
```

```
51              FORMAT(i2.2, '.', i4.4, 2i2.2,'-', A9)
                write(FILENAME,51) l, YYYY, MM, DD, SiteCode(l)  !
l.YYYYMMDD-SiteCode
                OPEN(20,FILE=FILENAME,FORM='FORMATTED',STATUS='UNKNOWN')

                write(20,19)
19              format('MM/DD/YY|HH:MN UTZ|SiteCode|NDRYDEP',
     &                 '|NWET-Total|NWET-Convective|Dry NO3- Flux',
     &                 '|Dry NH4+ Flux|Dry HNO3|Dry NO2',
     &                 '|Dry NO|Dry NH3|Dry RNO3',
     &                 '|Wet NO3- Flux|Wet NH4+ Flux',
     &                 '| RC | RN | QI | QC  ')
                write(20,18)
18              format('Month/Day/Year| Time   | Location | N KG/Ha ',
     &                 '| N KG/Ha | N KG/Ha | N KG/Ha ',
     &                 '| N KG/Ha | N KG/Ha | N KG/Ha ',
     &                 '| N KG/Ha | N KG/Ha | N KG/Ha ',
     &                 '| N KG/Ha | N KG/Ha ',
     &                 '| CM | CM | KG/KG | KG/KG ')
              ENDIF  !End of ihr==1 loop

              NDRY = NDRY_IN(l,ihr,1)
              NWET1 = NWET1_IN(l,ihr,1)
              NWET2 = NWET2_IN(l,ihr,1)
              DNO3F = DNO3F_IN(l,ihr,1)
              DNH4F = DNH4F_IN(l,ihr,1)
              WNO3F = WNO3F_IN(l,ihr,1)
              WNH4F = WNH4F_IN(l,ihr,1)
              WriteHNO3 = HNO3_IN(l,ihr,1)
              WriteNO2 = NO2_IN(l,ihr,1)
              WriteNO = NO_IN(l,ihr,1)
              WriteNH3 = NH3_IN(l,ihr,1)
              WriteRNO3 = RNO3_IN(l,ihr,1)
              WriteRC = RC_IN(l,ihr,1)
              WriteRN = RN_IN(l,ihr,1)
              WriteQC = QC_IN(l,ihr,1)
              WriteQI = QI_IN(l,ihr,1)

21            format(2(i2.2,'/'),i2.2,' | ',i2.2,':',i2.2,' | ',A9,
     &               16(' | ',e10.3))
              write(20,21) MM, DD, YY, HH, MN, SiteCode(l), NDRY,
     &                     NWET1, NWET2, DNO3F,
     &                     DNH4F, WriteHNO3, WriteNO2,
     &                     WriteNO, WriteNH3, WriteRNO3,
     &                     WNO3F, WNH4F,
     &                     WriteRC, WriteRN, WriteQI, WriteQC

              call nextime(writeDate,writeTime,TSTEP3D)
              ENDDO ! END Looping for reading the file in ihr

          write(6,*) l, "#%^%#^#^%^%# NEXTSITE #%^%#^#%^#%^ "
        ENDDO ! END Looping for the sites

        write(6,*)  "%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%"
```

```
       write(6,*)  "%%%%%%%%%%",yy,DD,"%%%%%%%%%%%%%%%%%%%%%%"
       write(6,*)  "%%%%%%%CheckSite is over!!!!%%%%%%%"
       write(6,*)  "%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%"
       MESG = "This run looks fine"
       CALL M3EXIT ( PROGNAME , readDate, readTime, MESG, 0 )
       STOP
       END PROGRAM
!=====================================
        INCLUDE 'll2cell_MCIP_sub.F'
```

Appendix D: Trimming.F

```
      IMPLICIT NONE

!...........   EXTERNAL FUNCTIONS and their descriptions:

      INTEGER   TRIMLEN, MMDDYY
      real      getreal
      EXTERNAL  TRIMLEN, nextime, getreal, MMDDYY

!...........   LOCAL VARIABLES and their descriptions:

      character*16  ::  PROGNAME = 'TTTT.f   '
      character*22  ::  FILENAME
      character*180  MESG, HEADER1
      character*180  HEADER2
      character*180  HEADER3
      character*180  HEADER4
      integer    BeginDt, EndDt
      character*8    NPSD, NPSWT
      character*8    METCRO2D, METCRO3D, METDOT3D
      character*16   GRIDCRO2D
      integer    LOGDEV
      integer    YYYY, DDD, GregDate, YY, MM, DD, HH, MN, SS
      integer    MXSITES, MONTH
      parameter (MXSITES = 40000)
      parameter (MONTH = 240)
COLS+&   +    5    2    5    3    5    4    5    5    5    6    5    712
      integer allocateStat, iost
      integer i, j, k, l, m, n, ihr
      integer N_SITES
      character*9    location
!      real, dimension(:,:,:), allocatable :: QI_IN
      character*9,  dimension(MXSITES) :: SiteCode
!      integer,  dimension(MXSITES) :: bDate
!      integer,  dimension(MXSITES) :: eDate
!      integer,  dimension(MXSITES) :: rDate
      integer       bDate
      integer       eDate
      integer       rDate
!      character*9,  dimension(MXSITES) :: wDate
      integer,      dimension(MXSITES) :: bYY, bMM, bDD
      integer,      dimension(MXSITES) :: eYY, eMM, eDD

      integer,      dimension(MXSITES) :: readYY, readMM, readDD
      integer,      dimension(MXSITES) :: readHH, readMN
      real,         dimension(MXSITES) :: readNDRY, readNWET1, readNWET2
      real,         dimension(MXSITES) :: readHNO3, readNO2, readNO
      real,         dimension(MXSITES) :: readNH3, readRNO3
      real,         dimension(MXSITES) :: readRC, readRN, readQC, readQI
      real,         dimension(MXSITES) :: readDNO3F, readDNH4F
      real,         dimension(MXSITES) :: readWNO3F, readWNH4F
```

```
      real,         dimension(MONTH) :: writeNDRY, writeNWET1
      real,         dimension(MONTH) :: writeNWET2
      real,         dimension(MONTH) :: writeHNO3, writeNO2
      real,         dimension(MONTH) :: writeNO
      real,         dimension(MONTH) :: writeNH3, writeRNO3
      real,         dimension(MONTH) :: writeRC, writeRN
      real,         dimension(MONTH) :: writeQC, writeQI
      real,         dimension(MONTH) :: writeDNO3F, writeDNH4F
      real,         dimension(MONTH) :: writeWNO3F, writeWNH4F

      real writeNDRY_T, writeNWET1_T
      real writeNWET2_T
      real writeHNO3_T, writeNO2_T
      real writeNO_T
      real writeNH3_T, writeRNO3_T
      real writeRC_T, writeRN_T
      real writeQC_T, writeQI_T
      real writeDNO3F_T, writeDNH4F_T
      real writeWNO3F_T, writeWNH4F_T

      integer temp_MM, temp_YY, temp_DD
      real writeNDRY_M, writeNWET1_M
      real writeNWET2_M
      real writeHNO3_M, writeNO2_M
      real writeNO_M
      real writeNH3_M, writeRNO3_M
      real writeRC_M, writeRN_M
      real writeQC_M, writeQI_M
      real writeDNO3F_M, writeDNH4F_M
      real writeWNO3F_M, writeWNH4F_M

!      integer,      dimension(MXSITES) :: ROW
!      integer,      dimension(MXSITES) :: COL
      real,         dimension(MXSITES) :: LAT
      real,         dimension(MXSITES) :: LONG
!      character*16, parameter :: NO='NO', NH3= 'NH3', RNO3= 'RNO3'
      logical :: ex


COLS+&   +    5    2    5    3    5    4    5    5    5    6    5
7123456789$
COLS+&   +    5    2    5    3    5    4    5    5    5    6    5    712
!*****************************************************************

!!!!!!read in the sites file, storing and counting
      OPEN(12, IOSTAT=iost,FILE='SITE', STATUS='old')
      read(12,'(A80)',IOSTAT=iost) HEADER1
      write(6,*) HEADER1
      l = 1
20    format(A9,1x,f8.5,1x,f11.5,2(1x,i2.2,1x,i2.2,1x,i2.2))
40    read(12,20,END=45) SiteCode(l),LAT(l),LONG(l),bMM(l),bDD(l),bYY(l)
     &                   ,eMM(l),eDD(l),eYY(l)
      write(6,*) l, SiteCode(l),LAT(l),LONG(l),
```

```
     &           bYY(l),bMM(l),bDD(l), eYY(l),eMM(l),eDD(l)
            l = l + 1
      goto 40

45    N_SITES = l - 1
      write(6,*) ' SITES FOUND: ' , N_SITES
      close(12)


      OPEN(50,FILE="SUMMARY",FORM='FORMATTED',STATUS='NEW')
      write(50,19)
19    format('Period |SiteCode |NDRYDEP',
     &       '|NWET-Total|NWET-Convective|Dry NO3- Flux',
     &       '|Dry NH4+ Flux|Dry HNO3|Dry NO2',
     &       '|Dry NO|Dry NH3|Dry RNO3',
     &       '|Wet NO3- Flux|Wet NH4+ Flux',
     &       '| RC | RN | QI | QC  ')
      write(50,18)
18             format('Month/Day/Year| Location | N KG/Ha ',
     &                '| N KG/Ha | N KG/Ha | N KG/Ha ',
     &                '| N KG/Ha | N KG/Ha | N KG/Ha ',
     &                '| N KG/Ha | N KG/Ha | N KG/Ha ',
     &                '| N KG/Ha | N KG/Ha ',
     &                '| CM | CM | KG/KG | KG/KG ')


      DO k = 1, N_SITES
      INQUIRE (file= SiteCode(k), EXIST=ex)

      if (ex) then

      write(6,*) "Test begin here!!!!!!!!!!"
      write(6,*) 'working on this location:  ', SiteCode(k)
!20    format(i2.2,1x,i2.2,1x,i2.2)
      write(6,*) '$^$%#&% OPEN the First Location ^$%&^&^%  '
      OPEN(11, IOSTAT=iost,FILE=SiteCode(k), STATUS='old')
      read(11,'(A180)',IOSTAT=iost) HEADER2
      read(11,'(A180)',IOSTAT=iost) HEADER3
21    format(2(i2,1x),i2,3x,i2,1x,i2,3x,A9,
     &       16(3x,e10.3))

      DDD = 1
34    read(11,21,END=35) readMM(DDD), readDD(DDD), readYY(DDD),
     &            readHH(DDD), readMN(DDD),
     &            location,  readNDRY(DDD), readNWET1(DDD),
     &            readNWET2(DDD), readDNO3F(DDD),readDNH4F(DDD),
     &            readHNO3(DDD), readNO2(DDD), readNO(DDD),
     &            readNH3(DDD), readRNO3(DDD),
     &            readWNO3F(DDD), readWNH4F(DDD),
     &            readRC(DDD), readRN(DDD), readQI(DDD), readQC(DDD)
      DDD = DDD + 1
      goto 34

35    close(11)
```

```
      DDD = DDD - 1
      write(6,*) 'DDD', DDD

      write(6,*) '$^$%Setting the initial variables'
      bDate = bYY(k)*10000 + bMM(k)*100 + bDD(k)
      eDate = eYY(k)*10000 + eMM(k)*100 + eDD(k)
      temp_MM = bMM(k)
      temp_YY = bYY(k)
      temp_DD = bDD(k)
      writeNDRY_M = 0
      writeNWET1_M = 0
      writeNWET2_M = 0
      writeDNO3F_M = 0
      writeDNH4F_M = 0
      writeHNO3_M = 0
      writeNO2_M = 0
      writeNO_M = 0
      writeNH3_M = 0
      writeRNO3_M = 0
      writeWNO3F_M = 0
      writeWNH4F_M = 0
      writeRC_M = 0
      writeRN_M = 0
      writeQI_M = 0
      writeQC_M = 0

      writeNDRY_T = 0
      writeNWET1_T = 0
      writeNWET2_T = 0
      writeDNO3F_T = 0
      writeDNH4F_T = 0
      writeHNO3_T = 0
      writeNO2_T = 0
      writeNO_T = 0
      writeNH3_T = 0
      writeRNO3_T = 0
      writeWNO3F_T = 0
      writeWNH4F_T = 0
      writeRC_T = 0
      writeRN_T = 0
      writeQI_T = 0
      writeQC_T = 0

      write(6,*) '$^$%#&% Read and Write the Info  '
51    FORMAT('OUT-', A9)
      write(FILENAME,51) SiteCode(k)  ! SiteCode.txt
      OPEN(20,FILE=FILENAME,FORM='FORMATTED',STATUS='NEW')

      write(6,*) 'bDate', bDate
      write(6,*) 'eDate', eDate

      write(20,'(A180)') HEADER2
      write(20,'(A180)') HEADER3
```

```
      DO i = 1, DDD
             rDate = readYY(i)*10000 + readMM(i)*100 + readDD(i)

             if((rDate.GE.bDate).AND.(rDate.LE.eDate)) THEN
22                format(i2.2,"/", i2.2, "/", i2.2, ' | ', i2.2, ':'
     &                ,i2.2,  ' | ', A9,16(' | ',e10.3) )
                  write(20,22) readMM(i), readDD(i), readYY(i),
     &                         readHH(i), readMN(i),
     &                         location,  readNDRY(i), readNWET1(i),
     &                         readNWET2(i), readDNO3F(i),
     &                         readDNH4F(i),
     &                         readHNO3(i), readNO2(i), readNO(i),
     &                         readNH3(i), readRNO3(i),
     &                         readWNO3F(i), readWNH4F(i),
     &                         readRC(i), readRN(i), readQI(i),
     &                         readQC(i)

                  if((temp_MM<readMM(i)).OR.(readYY(i)>temp_YY))THEN
!     &                .OR.((readYY==eYY(k).AND.readMM==eMM(k).AND.
!     &                 readDD(i)==eDD(k).AND.readHH(i)==24))) THEN
                  write(6,*) temp_MM," <", readMM(i), "/", readDD(i)
                  write(6,*) "TEST statment is True"
23                    format(i2.2,"/", i2.2, "/", i2.2, ' to ',
     &                       i2.2,"/", i2.2, "/", i2.2, ' | ',
     &                       A9,16(' | ',e10.3) )

                  n = i - 1
                       write(50,23) temp_MM, temp_DD, temp_YY,
     &                               readMM(n), readDD(n), readYY(n),
     &                               location,  writeNDRY_M, writeNWET1_M,
     &                               writeNWET2_M, writeDNO3F_M,
     &                               writeDNH4F_M,
     &                               writeHNO3_M, writeNO2_M, writeNO_M,
     &                               writeNH3_M, writeRNO3_M,
     &                               writeWNO3F_M, writeWNH4F_M,
     &                               writeRC_M, writeRN_M, writeQI_M,
     &                               writeQC_M
                       writeNDRY_T = writeNDRY_T + writeNDRY_M
                       writeNWET1_T = writeNWET1_T + writeNWET1_M
                       writeNWET2_T = writeNWET2_T + writeNWET2_M
                       writeDNO3F_T = writeDNO3F_T + writeDNO3F_M
                       writeDNH4F_T = writeDNH4F_T + writeDNH4F_M
                       writeHNO3_T = writeHNO3_T + writeHNO3_M
                       writeNO2_T = writeNO2_T + writeNO2_M
                       writeNO_T = writeNO_T + writeNO_M
                       writeNH3_T = writeNH3_T + writeNH3_M
                       writeRNO3_T = writeRNO3_T + writeRNO3_M
                       writeWNO3F_T = writeWNO3F_T + writeWNO3F_M
                       writeWNH4F_T = writeWNH4F_T + writeWNH4F_M
                       writeRC_T = writeRC_T + writeRC_M
                       writeRN_T = writeRN_T + writeRN_M
                       writeQI_T = writeQI_T + writeQI_M
                       writeQC_T = writeQC_T + writeQC_M
                  temp_DD = readDD(i)
```

```
                    temp_MM = readMM(i)
                    temp_YY = readYY(i)
                         writeNDRY_M = 0
                         writeNWET1_M = 0
                         writeNWET2_M = 0
                         writeDNO3F_M = 0
                         writeDNH4F_M = 0
                         writeHNO3_M = 0
                         writeNO2_M = 0
                         writeNO_M = 0
                         writeNH3_M = 0
                         writeRNO3_M = 0
                         writeWNO3F_M = 0
                         writeWNH4F_M = 0
                         writeRC_M = 0
                         writeRN_M = 0
                         writeQI_M = 0
                         writeQC_M = 0
               else
                    write(6,*) temp_MM,">", readMM(i), "/", readDD(i)
                    write(6,*) "TEST statment is False*(^*(%*%(%^*%%*^"
               endif  !Test for month to month accumlation
                         writeNDRY_M = writeNDRY_M + readNDRY(i)
                         writeNWET1_M = writeNWET1_M + readNWET1(i)
                         writeNWET2_M = writeNWET2_M + readNWET2(i)
                         writeDNO3F_M = writeDNO3F_M + readDNO3F(i)
                         writeDNH4F_M = writeDNH4F_M + readDNH4F(i)
                         writeHNO3_M = writeHNO3_M + readHNO3(i)
                         writeNO2_M = writeNO2_M + readNO2(i)
                         writeNO_M = writeNO_M + readNO(i)
                         writeNH3_M = writeNH3_M + readNH3(i)
                         writeRNO3_M = writeRNO3_M + readRNO3(i)
                         writeWNO3F_M = writeWNO3F_M + readWNO3F(i)
                         writeWNH4F_M = writeWNH4F_M + readWNH4F(i)
                         writeRC_M = writeRC_M + readRC(i)
                         writeRN_M = writeRN_M + readRN(i)
                         writeQI_M = writeQI_M + readQI(i)
                         writeQC_M = writeQC_M + readQC(i)
               endif
          ENDDO !for the write-in file loop (DDD)

          close(20)
! Adding up for the last month
                         writeNDRY_T = writeNDRY_T + writeNDRY_M
                         writeNWET1_T = writeNWET1_T + writeNWET1_M
                         writeNWET2_T = writeNWET2_T + writeNWET2_M
                         writeDNO3F_T = writeDNO3F_T + writeDNO3F_M
                         writeDNH4F_T = writeDNH4F_T + writeDNH4F_M
                         writeHNO3_T = writeHNO3_T + writeHNO3_M
                         writeNO2_T = writeNO2_T + writeNO2_M
                         writeNO_T = writeNO_T + writeNO_M
                         writeNH3_T = writeNH3_T + writeNH3_M
                         writeRNO3_T = writeRNO3_T + writeRNO3_M
                         writeWNO3F_T = writeWNO3F_T + writeWNO3F_M
```

```
                    writeWNH4F_T = writeWNH4F_T + writeWNH4F_M
                    writeRC_T = writeRC_T + writeRC_M
                    writeRN_T = writeRN_T + writeRN_M
                    writeQI_T = writeQI_T + writeQI_M
                    writeQC_T = writeQC_T + writeQC_M
!Writing for the last month
                    write(50,23) temp_MM, temp_DD, temp_YY,
     &                           eMM(k), eDD(k), eYY(k),
     &                           location,  writeNDRY_M, writeNWET1_M,
     &                           writeNWET2_M, writeDNO3F_M,
     &                           writeDNH4F_M,
     &                           writeHNO3_M, writeNO2_M, writeNO_M,
     &                           writeNH3_M, writeRNO3_M,
     &                           writeWNO3F_M, writeWNH4F_M,
     &                           writeRC_M, writeRN_M, writeQI_M,
     &                           writeQC_M

24                  format("Total  | ", A9,16(' | ',e10.3) )

                    write(50,24) location, writeNDRY_T, writeNWET1_T,
     &                           writeNWET2_T, writeDNO3F_T,
     &                           writeDNH4F_T,
     &                           writeHNO3_T, writeNO2_T, writeNO_T,
     &                           writeNH3_T, writeRNO3_T,
     &                           writeWNO3F_T, writeWNH4F_T,
     &                           writeRC_T, writeRN_T, writeQI_T,
     &                           writeQC_T
                    write(50,*) "      "

      ELSE
      write(6,*) "Skip this location:   ", SiteCode(k)

      ENDIF !for determining the existence of the file
      ENDDO !for the N-SITE loop

      END PROGRAM
```

Appendix E: Forest Research Sites From Project Sponsors

Site_Code	Latitude	Longitude	StartDate	EndDate
C2CCRLA	41.28153	-122.2976	06-19-07	06-18-08
C2CBeaCk	44.4353	-123.41018	07-02-07	06-16-08
C2CHendx	44.036	-123.0577	07-03-07	06-28-08
C2CFinWR	44.3997	-123.349	07-05-07	06-21-08
C2CDunnF	44.6893	-123.2902	07-06-07	07-07-08
C2CHysF	44.6348	-123.189	07-10-07	06-27-08
C2CMaryP	44.5142	-123.554	07-11-07	07-07-08
C2CForPk	45.59765	-122.81422	07-12-07	07-24-08
C2CAlsea	44.38539	-123.61535	07-13-07	07-21-08
C2CMcKR	44.28216	-122.02939	07-16-07	07-03-08
C2CHJA	44.21189	-122.25462	07-17-07	07-02-08
C2CKeelM	44.51939	-122.62855	07-18-07	07-27-08
C2CCapeP	44.25635	-123.98633	07-19-07	07-12-08
C2CZoo	45.50886	-122.7092	07-20-07	07-24-08
C2CSewP	47.54988	-122.25373	07-23-07	07-23-08
C2CAlpLW	47.53925	-121.57313	07-25-07	07-23-08
WindR	45.8155	-121.9459	09-19-05	09-27-08
7MHPipo	45.63695	-121.2968	09-20-05	09-27-08
HermanCk	45.69167	-121.80783	10-04-05	09-27-08
MtZion	45.5694	-122.2082	10-24-06	09-27-08
MTRA	46.7582	-122.1243	05-24-05	05-26-09
NOCA	48.54056	-121.44528	05-25-05	05-26-09
OLYM	47.86	-123.93194	05-26-05	05-26-09