CMAQv5.3 User Manual

09/09/2019

Contents

1. Overview **7**

Disclaimer 7

1.1 Introduction 7

1.2 Features 8

1.3 Instrumented Models 10

1.4 New Features in CMAQv5.3 10

1.5 System Recommendations 11

1.6 CMAQ Support Resources 11

2. Program Structure **11**

2.1 Introduction 11

2.2 CMAQ Core Programs 12

2.3 Online Emissions Options 13

2.4 Post-processing Tools 13

2.5 Utilities for Developers 13

3. Preparing Compute Environment for CMAQ Simulations **14**

3.1 Introduction 14

3.2 Hardware Requirements 14

3.3 Software Requirements 15

3.3.1 Message Passing Interface (MPI) library 15

3.3.2 netCDF library 15

3.3.3 I/O API library 16
3.4 Optional Software 17
3.5 References: 18

4. Model Input Files **18**
4.1 Introduction 18
4.2 CMAQ Pre-processors 19
4.2.1 Meteorology-Chemistry Interface Processor (MCIP) 19
4.2.2 Initial Conditions Processor (ICON) 19
4.2.3 Boundary Conditions Processor (BCON) 20
4.2.4 External Software Programs for Preparing CMAQ Inputs 21
4.4 GRIDDESC and Species Namelist Files 27
4.7 Meteorological Inputs (Processed for the CMAQ System using MCIP) 31

5. Running a CMAQ Simulation **31**
5.1 Introduction 31
5.2 Getting the CMAQ Source Code 31
5.2.1 Git Installation 31
5.2.2 Zip file Installation 32
5.3 The CMAQ Repository Structure 32
5.4 Building CMAQ Outside of the Repository in a User-Specified Directory 34
5.5 Initialization of CMAQ Environment 35
5.6 Compiling CMAQ Chemistry-Transport Model (CCTM) 35
5.7 Running CCTM 37
5.7.1 CCTM Logfiles 37
5.7.2 CCTM Output files 43
5.7.3 Common errors causing the CCTM simulation to crash 44

6. Model Configuration Options **44**
6.1 Introduction 44
6.2 Numerical Approach 44
6.3 Grid Configuration 45
6.3.1 Horizontal Domains and Lateral Boundaries 46

6.3.2 Vertical Domains 46
6.4 Science Configurations 47
6.5 Advection 47
6.6 Horizontal Diffusion 48
6.7 Vertical Diffusion 49
6.8 Dry Deposition/Air-surface exchange 49
6.8.1 Dry Deposition - m3dry 50
6.8.2 Dry Depostion - STAGE 50
6.9 Emissions 52
6.9.1 Emission Streams 52
6.9.2 Online Emission Streams 55
6.10 Gas Phase Chemistry 61
6.10.1 Gas Phase Chemical Mechanisms 61
6.10.2 Solvers 62
6.10.3 Photolysis 63
6.10.4 Nitrous Acid (HONO) 63
6.11 Aerosol Dynamics and Chemistry 64
6.12 Aqueous Chemistry, Scavenging and Wet Deposition 66
6.13 Potential Vorticity Scaling 69
6.14 References 69

7. Model Output Files **73**
7.1 Introduction 73
7.2 CCTM Output Files 74
7.3 Restart Files 75
7.4 Diagnostic and Advanced CMAQ Output Files 76

8. Analysis Tools for CMAQ output **80**
8.1 Introduction 80
8.2 Aggregating and Transforming Model Species 81
8.3 Model-Observation Pairing for Model Evaluation 81
8.3.1 Spatial matching in sitecmp and sitecmp_dailyo3 82
8.3.2 Temporal matching in sitecmp and sitecmp_dailyo3 82

8.4 The Atmospheric Model Evaluation Tool (AMET) 83
8.5 Visualization Environment for Rich Data Interpretation (VERDI) 84

9. Process Analysis **85**
9.1 Introduction 85
9.2 Use of Process Analysis 86
9.3 Description of the PACM_INFILE 87
9.4 Parameters for IPR 89
9.5 Parameters for IRR 90
9.6 Example IRR applications 90
9.7 References 92

10. Decoupled Direct Method in Three Dimensions (CMAQ-DDM-3D) **93**
10.1 Introduction 93
10.2 CMAQ-DDM-3D Releases 93
10.3 References 94

11. Integrated Source Apportionment Method (CMAQ-ISAM) **94**
11.1 Introduction 94
11.2 Build Instructions 95
11.3 Run Instructions 95
11.3.1 ISAM control file (SA_IOLIST) 96
11.4 ISAM Benchmark data 97
11.5 References 98

12. Sulfur Tracking Method **98**
12.1 Introduction 98
12.2 Usage 98

Appendix A: Model options **101**
A.1 Configuration Options (config_cmaq.csh) 101
A.2 Compilation Options (bldit_cctm.csh) 103
A.3 Execution Options (run_cctm.csh) 107
MPI Configuration 108

Vertical extent . . . 108
Timestep Configuration . . . 108
CCTM Configuration Options . . . 109
Synchronization Time Step and Tolerance Options . . . 109
Science Options . . . 110
Process analysis options . . . 112
I/O Controls . . . 113
Aerosol Diagnostics Controls . . . 113
Diagnostic Output Flags . . . 114
Inline emissions configuration . . . 114
Lightning NOx configuration . . . 115
Windblown dust emissions configuration . . . 117

Appendix B: Emissions Input and Control **117**

B.1 Emissions Control with the Detailed Emissions Scaling, Isolation and Diagnostics Module (DESID) . . . 117
B.2 General Specs . . . 118
B.3 Emission Scaling Rules . . . 118
B.3.1 Default Rules . . . 119
B.3.2 Modifying Default rules . . . 120
B.4 Applying Masks for Spatial Dependence . . . 121
B.5 Aerosol Size Distributions . . . 122
B.6 Additional DESID Features . . . 124
B.6.1 Summary Output to Processor-Specific Logfiles . . . 124
B.6.2 Diagnostic Gridded Output Files . . . 125

Appendix C: Processing Spatial Data for CMAQ Inputs **126**

C.1 Geospatial Data . . . 126
C.2 Geodetic datum . . . 126
C.3 Spatial Data Projection . . . 127
C.4 Spatial Data Generation . . . 127

Appendix D: Parallel Implementation of CMAQ **129**

D.1 General Structure of data 129

D.2 Interprocessor Communication 130

D.3 Parallel I/O 132

D.4 Reference: 136

Appendix E: Configuring the Weather Research and Forecasting Model (WRF) for Use with Air Quality Models **136**

E.1 WRF version 4+ 136

E.2 WRF version 3.7 136

E.3 WRF with lightning assimilation 136

E.4 Reference: 137

1. Overview

Disclaimer

The information in this operational guidance document has been funded wholly or in part by the United States Environmental Protection Agency (EPA). The draft version of this document has not been subjected to the Agency's peer and administrative review, nor has it been approved for publication as an EPA document. The draft document has been subjected to review by the Community Modeling and Analysis System Center only; this content has not yet been approved by the EPA. Mention of trade names or commercial products does not constitute endorsement or recommendation for use.

1.1 Introduction

Under the authority of the Clean Air Act, the U.S. Environmental Protection Agency (EPA) has established National Ambient Air Quality Standards (NAAQS). These standards are designed to protect human health and the environment from high levels of criteria pollutants, such as ozone and particulate matter. Meeting the NAAQS often requires the use of controls on sources of air pollutants. The complex nature of air pollution scenarios requires control strategies to be effective for a variety of air pollutants, geographic regions, and scales. The design of these control strategies is guided by comprehensive air pollution modeling systems which are applied to assess the ability of various control strategies to improve air quality in a cost-effective manner.

Because some emission sources contribute to the ambient levels of more than one pollutant and can affect an entire region on various time scales, an integrated modeling approach capable of handling multiple air pollutants and spatiotemporal scales is needed to identify cost-effective control strategies that improve overall air quality. The EPA Community Multiscale Air Quality (CMAQ) modeling system was formulated and designed to facilitate extensions needed to examine emerging linked multi-pollutants air pollution issues. The source code for CMAQ is available through a publicly-accessible, version-controlled git repository on GitHub where interested parties may obtain the open-source software and contribute to enhancements of the model. CMAQ is designed for applications ranging from regulatory and policy analysis to probing and understanding the complex interactions of atmospheric chemistry and physics. It is a three-dimensional Eulerian (i.e., gridded) atmospheric chemistry and transport modeling system that simulates ozone, particulate matter (PM), toxic airborne pollutants, visibility, and acidic and nutrient pollutant species throughout the troposphere. Designed as a "one-atmosphere" model, CMAQ can address the complex couplings among several air quality issues simultaneously across spatial scales ranging from urban to hemispheric.

Air quality models integrate our understandings of the complex processes that affect the concentrations of pollutants in the atmosphere. Establishing the relationships among meteorology, emissions of chemical species, chemical transformations, and removal processes in the context of atmospheric pollutants is the fundamental goal of an air quality model (Seinfeld and Pandis, 1998). CMAQ uses detailed mathematical representations of coupled atmospheric dynamical, chemical, and physical processes to describe the fate of airborne pollutants. The model is formulated to conserve mass in the 3-D atmosphere within the modeled domain. The resultant partial differential governing equations are numerically solved over a 3-D grid discretizing the geographic domain of interest. A model grid is an

x-y-z array that is fixed in space and covers a prescribed domain (i.e., a geographic area of interest). CMAQ therefore belongs to the Eulerian class of mathematical models that calculate a mass balance over each discrete grid volume by accounting for transport across the boundaries of the grid volume and relevant source and sink terms within the grid volume over a given time period. As a mathematical framework for simulating the interactions of multiple complex atmospheric processes, CMAQ thus requires two primary types of inputs: meteorological information, and rates of emissions from a variety of anthropogenic and natural sources of primary pollutant species of interest or those that serve as precursors for formation of other pollutants of interest.

With weather conditions contributing the primary physical driving forces in the atmosphere (such as the changes in temperature, winds, cloud formation, and precipitation rates), representative gridded meteorology forms the basis of all 3-D air quality model simulations. The Weather Research and Forecasting (WRF) model - Advanced Research WRF (WRF-ARW) (Skamarock et al., 2005) is compatible with CMAQ in that both systems can be configured to use identical horizontal and vertical coordinate and grid structures and is commonly used to drive CMAQ. The meteorology inputs dictate the following CMAQ configuration parameters:

- Horizontal grid coordinate system (e.g., latitude-longitude, Lambert Conformal)
- Horizontal grid resolution (i.e., the size of the cells composing the grid)
- Maximum spatial coverage (horizontal geographic extent, i.e., *the domain*) of the grid
- Maximum vertical extends (model top) and vertical grid resolution
- Temporal extent (the starting and ending dates and times, and the meteorology update frequency)

To obtain inputs on emissions, CMAQ relies on an emissions processor to estimate the magnitude, location, and temporal variability of pollution sources. Open-source processors such as the Sparse Matrix Operator Kernel Emissions (SMOKE) processor (IE, 2008) are available for computing emissions inputs to CMAQ from emissions inventories. These emissions inputs must be specified on CMAQ's horizontal and vertical grid structure and cover at least the time period of the air quality model simulation. The emission inputs must also represent chemical species that conform with the gas and aerosol chemical mechanism employed in the CMAQ configuration; currently supported gas-phase mechanisms include recent versions of the Carbon Bond mechanism, the Statewide Air Pollution Research Center (SAPRC) mechanism, and the Regional Atmospheric Chemistry Mechanism (RACM). Additional details about the gas- and aerosol-phase chemistry in CMAQ are provided in Chapter 6.

1.2 Features

From inception, CMAQ was designed to facilitate community modeling. "Community modeling" is the concept that air quality model development should be a collective effort by a broad community of developers, thereby leveraging the cross-disciplinary expertise needed to keep the physical, numerical, and computational components of the modeling system at the state-of-the-science. By adopting a standardized modeling architecture, the air quality modeling community can focus its efforts on creating software enhancements and new science modules. CMAQ is designed to meet the needs of the multiple groups contained within the air quality modeling community: research and regulatory modelers, algorithm and science module developers, air quality forecasters, and planners and policy makers. While

each of these groups has distinct individual requirements for CMAQ, they also share a common need for an efficient, transparent, and scientifically credible tool to simulate the air pollution phenomena. To address these individual and common needs, CMAQ development and maintenance have the following goals:

1. *Scientific Integrity*. Ensure that the model remains state-of-the-science through subjecting it to regular peer reviews
2. *Community Development*. Utilize a design that encourages innovations and enhancements by all members of the air quality modeling community
3. *Multiscale Modeling*. Provide adequate technical formulations to address air quality issues on multiple spatial scales, from urban to hemispheric
4. *Multi-pollutant Design*. Provide robust and integrated science for modeling multiple, coupled air quality issues in a single simulation
5. *Modularity*. Maintain flexibility to add new, or select from existing, science modules to optimize model performance for specific applications
6. *Transparency*. Utilize programming practices that promote understanding of the model formulation at the source-code level
7. *Computational Efficiency*. Provide scientifically acceptable results without compromising the speed at which the results are generated
8. *Open-Source Design*. Enable no-cost distribution and application by the modeling community

Designed under a community-modeling paradigm, CMAQ is distributed as open-source software engineered with a modular code design to facilitate decentralized development. Built around a layered I/O API and netCDF code framework, CMAQ provides a flexible platform for testing new science algorithms, chemistry representations, and optimization techniques. CMAQ provides the following features to scientists interested in developing new algorithms or adding science to the model:

- All CMAQ source code is available through GitHub.
- Developed and distributed following open-source software conventions, CMAQ source code is easily accessible and free to obtain.
- Designed for modularity, CMAQ uses standardized input/output (I/O) routines to facilitate extensibility.
- The diverse and continually growing community of CMAQ developers provides an excellent forum for discussing development-related topics of all kinds.

The CMAQ modeling system is being developed and maintained under the leadership of the EPA Office of Research and Development in Research Triangle Park, NC. CMAQ represents nearly three decades of research in atmospheric modeling and has been in active development since the early 1990s. The first public release of CMAQ was in 1998 to enable use by air quality scientists, policy makers, and stakeholder groups to address multiscale, multipollutant air quality concerns. Since then, through a series of phased development activities, new versions of the CMAQ modeling system are periodically released for use by the growing user community.

1.3 Instrumented Models

In addition to the air pollutant concentration and deposition fields output by CMAQ, the modeling system can also be instrumented to compute and output additional diagnostic information that can be used to probe the workings of the atmosphere as well as inform and guide policy inferences. These instrumented configurations include:

1. *Integrated Source Apportionment Method (ISAM)*: Estimates source attribution information for user specified ozone and particulate matter precursors modeled in CMAQ. Such apportionment information could be used to gain insight on, for example, how much of the ozone in an urban area was formed due to nitrogen oxides emitted from motor vehicles in a neighboring state?
2. *Decoupled Direct Method in Three Dimensions (DDM-3D)*: A formal mathematical formulation that propagates sensitivity of CMAQ estimated concentrations and/or deposition to specified parameters (e.g., emissions) through the science modules in CMAQ. CMAQ-DDM-3D can be used for sensitivity to emission rates, boundary conditions, initial conditions, reaction rates, potential vorticity, or any combination of these parameters. Second order sensitivity calculations, or sensitivity of sensitivity, are also available.
3. *Sulfur Tracking Model (STM)*: Tracks sulfate production from gas- and aqueous-phase chemical reactions, as well as contributions from emissions and initial and boundary conditions. The additional diagnostic information enables users to better understand the relative contribution of various pathways for airborne sulfate, a dominant contributor to fine particulate matter.
4. *Integrated Process Rates (IPR)*: CMAQ can be configured to output the process rates for each of the modeled processes impacting change in ambient concentrations of modeled species. This essentially provides a breakdown of the various terms contributing to the overall species mass-balance and thus helps with species mass-budget analysis.
5. *Integrated Reaction Rates (IRR)*: This technique involves integrating the rates of individual chemical reactions represented in the gas-phase chemical mechanism employed by CMAQ. As an example, this information can then be used to infer the relative importance of various precursor species contributing to ozone production in a grid cell or region.

1.4 New Features in CMAQv5.3

Building on previous versions of the modeling system, numerous updates to the process science and model structure have been implemented in CMAQv5.3 including: 1. Advances in the science of modeling particulate matter composition, size distribution and optical properties. These updates are encapsulated in a new version of the CMAQ aerosol module named AERO7. 2. Improvements and updates to the representation of multi-phase chemistry in the CMAQ modeling system, including incorporation of new gas-phase chemical mechanisms, expansion of aqueous chemistry pathways, incorporation of improved data sets to estimate photolysis rates, and updates to numerical integration schemes to improve runtime efficiency. 3. Improvements in representation of land-atmosphere exchange processes, including consistent representation of deposition process for unidirectional and bidirectional species, explicit mapping of wet and dry deposited species to modeled chemical species and incorporation of land-use specific deposition output. 4. Improvements to coupling between meteorology and chemistry models, including updates to land surface fluxes using different land surface models in WRF (e.g.,

Noah), consistent coupling with new coordinate systems in WRF (e.g., hybrid vertical coordinate), incorporation of alternate output file formats. 5. Improvements to CMAQ model structural attributes, including a new interface for mapping and perturbing emission inputs, streamlining CMAQ output and log files to improve usability, improved documentation of chemical mechanism inputs and process analysis files. 6. Improvements in linkages of regional- and large-scale pollution through downscaling from global and hemispheric models and prototyping a variable resolution global chemistry-transport modeling system.

Collectively, these updates improve model capabilities for addressing local nonattainment issues, quantifying natural contributions versus anthropogenic enhancements, enabling examination of regional air pollution in context of changing global emissions, and for cross-media applications. Additionally, changes to model structure in CMAQv5.3 are oriented towards improving transparency of model assumptions, improved diagnostic tools for model probing, and improvements in overall numerical and computational efficiency of the modeling system.

1.5 System Recommendations

CMAQ is a comprehensive air pollution modeling system whose source code is written in Fortran. CMAQ execution is typically performed on Linux based systems. The hardware configuration of such a system depends on the domain size, grid resolution and simulation duration. Since typical input and output data sets for CMAQ entail three dimensional descriptions of the dynamical and chemical state of the simulated atmosphere, these data sets could require upwards of several gigabytes of disk storage per simulation day.

1.6 CMAQ Support Resources

Extensive information on the model's scientific basis, applications, publications, peer-review, and instructions to download the CMAQ modeling system are available at https://www.epa.gov/cmaq. To support the CMAQ user community, EPA currently funds the University of North Carolina at Chapel Hill to host the Community Modeling and Analysis System (CMAS) Center, which maintains a user help desk, provides new user training, and promotes the dissemination and use of the modeling system through exploration of new technologies and platforms (e.g., cloud-based). The CMAS Center offers an e-mail help desk and an online forum to allow users to connect with model developers and other model users around the world.

---

2. Program Structure

2.1 Introduction

The CMAQ system is a suite of software programs that work in concert to estimate ozone, particulate matter, toxic compounds, and acid deposition in addition to other atmospheric pollutants of interest.

As a framework for simulating the interactions of multiple complex atmospheric processes, CMAQ requires many types of inputs including meteorological information, primary pollutant emission rates, chemical properties and reactions, and land properties that are influential for exchange of pollutants with the atmosphere.

2.2 CMAQ Core Programs

Weather conditions such as the changes in temperature, winds, cloud formation, and precipitation rates are the primary physical driving forces for transport in the atmosphere. These conditions are represented in air quality model simulations using output from regional-scale numerical meteorology models, such as the Weather Research and Forecasting (WRF) Model. To obtain inputs on emissions, CMAQ relies on the open-source Sparse Matrix Operator Kernel Emissions (SMOKE) model to estimate the magnitude and location of pollution sources. Another open-source system, the Fertilizer Emission Scenario Tool for CMAQ (FEST-C) is used to run the Environmental Policy Integrated Climate (EPIC) model to generate agricultural land nitrogen and soil information needed for CMAQ bi-directional NH3 modeling.

The structure of the CMAQ system is illustrated in Fig. 2-1. The main CMAQ program, the CMAQ Chemistry Transport Model (CCTM), which is often referred to simply as CMAQ, contains the principal equations used for predicting pollutant concentrations given the inputs discussed above. These partial differential equations are designed for mass conservation and consider a myriad of important processes such as emissions, chemical reaction, uptake to clouds and precipitation, and dry deposition. An overview of the science configuration options in CMAQ can be found in Chapter 6. Instructions for compiling and running the CCTM are covered in Chapters 3 (Preparing Compute Environment), 4 (Model Inputs) and 5 (Running a CMAQ Simulation).

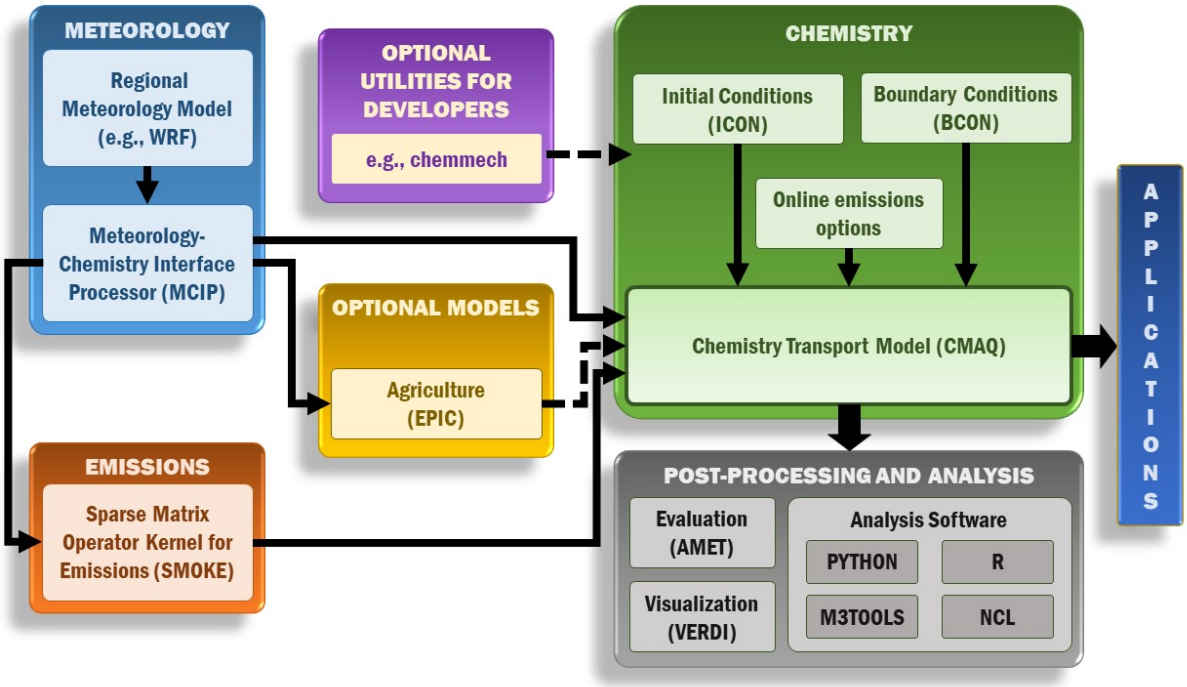


Figure 2□1. Overview of the CMAQ System

Several important tools are provided with the CMAQ system to handle the preparation of important input data. The meteorology data provided by the upstream meteorological model (e.g. WRF) is prepared for input to the CCTM by the Meteorology-Chemistry Interface Processor (MCIP). The CCTM also requires inputs for specifying the initial and boundary conditions of each chemical species treated by the model. These data may be processed and prepared for use by the ICON and BCON tools, respectively. Documentation on compiling and running MCIP, ICON and BCON is available under the PREP folder.

2.3 Online Emissions Options

CMAQ includes several "online" emissions options to support coupling between meteorology and chemistry processes, and to facilitate operational air quality forecast modeling. The emissions streams available for running online in CMAQ are: biogenics, wind-blown dust, sea spray, and lightning NO. One important advantage of incorporating these processes directly in a CCTM simulation is that emissions are meteorologically modulated at the synchronization (chemistry) time step rather than being linearly time-interpolated within each simulation hour. In addition, disk space may be saved, because a 3□D emissions file is no longer needed for elevated point sources.

2.4 Post-processing Tools

The CMAQ release includes a set of Fortran programs for manipulating CMAQ input and output files including matching model output to observed air quality data. Information on these post-processing tools is available under the POST folder. There are many additional resources available for visualizing and evaluating CMAQ input and output which are described in Chapter 8.

2.5 Utilities for Developers

The CMAQ release includes several optional utilities for model developers. These tools may be useful for advanced users who wish to use other chemical mechanisms and/or a different set of photolysis reaction input data. Chemical reaction data is processed by the Chemical Mechanism Compiler (*chemmech*) for all chemical reaction solver approaches. This tool needs chemical namelists (e.g. GC_NAMELIST, AE_NAMELIST, etc) in order to run, and these namelists can be modified directly with a text editor or converted to CSV with the namelist converter *nml*. After running chemmech, to then generate files specifically for the Euler Backward Iterative (EBI) solver approach, the *create_ebi* is provided. Finally the Photolysis Preprocessor (*inline_phot_preproc*) provides support for generating photolysis rate input to customize chemical mechanisms. In addition, the CMAQ repository includes software for generating Makefiles necessary for compiling the CCTM and other components. This *bldmake* utility is designed to account for user options, diagnose dependencies in source code and produce a Makefile ready to build executable files. Documentation for each utilitiy program is provided under the UTIL folder.

3. Preparing Compute Environment for CMAQ Simulations

3.1 Introduction

In this chapter the user will learn basic hardware and software requirements to run CMAQ. In addition, if the user does not have the required software, this chapter provides links to download the required software.

3.2 Hardware Requirements

The suggested hardware requirements for running the CMAQ Southeast Benchmark case on a Linux workstation are:

- 8 processors
- 4 GB RAM
- 400 GB hard drive storage

However, to use CMAQ in a production environment where multiple iterations of the model will be executed for different spatial domains and/or emissions control strategies, either a cluster of multiprocessor PCs on a high-end network or an expandable rack-mounted Linux server is recommended.

For example, the CMAQ team at the EPA uses a Dell cluster. The cluster consists 128 nodes and each node contain two Intel Xeon E5-2697A v4 16-core processors (with a total of 4096 processors), 256 GB memory (8 GB/core), EDR InfiniBand interconnect and runs on Red Hat Enterprise Linux 7 operating system.

Table 3-1 provides a general snapshot of three different CMAQ setups for a day of simulation conducted at the EPA. The output only included: the concentration file (CONC), the average concentration file (ACONC), 3-D average concentration file (CGRID), hourly dry deposition file (DRYDEP), and wet deposition from the clouds file (WETDEP1). The run time and domain size are dictated by the system hardware. Furthermore, the run time may vary due to compiler choice and system load.

Table 3□1. Example of job scenarios at EPA for a single day simulation

Domain	Domain size	Species Tracked	Input files size	Output files size	Run time (# cores)
2016 Southeast US	100 X 80 X 35	218	6.7GB	6.3GB	8 min/day (32); 47 min/day (4)
2016 CONUS	459 X 299 X 35	219	18GB	107GB	50 min/day (128); 90 min/day (32)
2016 N. Hemisphere	187 X 187 X 44	255	15GB	40GB	25 min/day (128)

3.3 Software Requirements

To build the CMAQ program suite, users must install these libraries in the order listed: MPI, netCDF and IOAPI. As always, we recommend using the latest release available at the time of your CMAQ installation. A table of the minimum required software versions is shown below:

Table 3□2. Minimum required software versions

Software	Version
Intel Compiler	17.0
GNU Compiler	6.1.0
PGI Compiler	17.4

NOTE: The CMAQ team recommends using a single compiler suite when building these libraries. Mixing compiler suites when building these libraries can cause unexpected behavior (e.g., mixing intel 18.0 to build netCDF C libraries and gcc 6.1.0 to build netCDF fortran libraries is not going to work).

3.3.1 Message Passing Interface (MPI) library

CMAQ is a MPI based program that runs on parallel programming platforms. Various flavour of MPI libraries are available for users to choose. CMAQ has been tested with the OpenMPI, MPICH, MVAPICH2, and the Intel MPI libraries. The choice of MPI library may affect model run time. For example, if you have Intel compiler suite available on your system, chooses Intel MPI or if your system is using InfiniBand (IB) interconnect, chooses MVAPICH2 which is tailored for IB.

Users can download the MPI library source code from one of these sites and follow provided procedures for proper installation. **Minimum versions: IntelMPI 2017.0 | MPICH 3.3.1 | MVAPICH2 2.3.1 | OpenMPI 2.1.0**

3.3.2 netCDF library

Most of the CMAQ input files and all output files are in netCDF format (the rest are in ASCII format). Hence the netCDF library is an essential component of the CMAQ model. The netCDF library is available for download at http://www.unidata.ucar.edu/software/netcdf/ and users should follow the instructions for proper installation. Users should install **classic shared netCDF C and Fortran libraries only without netCDF4, HDF5, HDF4, DAP client, PnetCDF, or zlib support.** In order to do so, users should provide the appropriate flags to build and install minimal netCDF-3 with no DAP client support, such as –disable-netcdf-4 and –disable-dap, at the configure stage for netCDF C. After successful installation, check the environment PATH & LD_LIBRARY_PATH to ensure that the paths have been updated to include the path of the netCDF C and Fortran libraries and bin. Note you may have to set these paths manually if not set, and these paths must be loaded every time you start a new shell. **Minimum versions: NetCDF-C 4.2 | NetCDF-Fortran 4.4.2**

3.3.3 I/O API library

The I/O API library provides an interface between the netCDF libraries and CMAQ to handle input and output (I/O) calls throughout the CMAQ code. The latest version of the I/O API library (version 3.2) is available for download at https://www.cmascenter.org/ioapi/documentation/all_versions/html/AVAIL.html#v32. Users should note that the I/O API library requires netCDF files to be adhere to a strict formatting guidelines that can be found in the I/O API documentation. For simplicity, files following the IOAPI-netCDF formatting guidelines will be called "IOAPI FILES" from now on. **Minimum version: IOAPI 3.1**

The general steps for installation of I/O API libraries on a Linux system (with C-shell and GNU compilers) are below. These instructions are an example and we recommend using the latest release available at the time of your CMAQ installation.

The following is a procedure to install "basic" I/O API libraries (this is based on gfortran compiler, for other compilers, look for corresponding Linux2_x86_64*):

```
mkdir ioapi_3.2
cd IOAPI_3.2

## Download IOAPI Libraries and untar downloaded source code in this directory
tar xvzf ioapi-3.2.tar.gz

### Set up your Linux system environment
setenv BIN Linux2_x86_64gfort
```

Edit the top level Makefile with the following steps:

1. comment out the line with BIN ="
2. Add explicit netCDF C and Fortran library paths in front of -lnetcdf -lnetcdff, respectively, the following is an example:

```
NCFLIBS = -L/usr/local/apps/netcdf-c-4.7.0/gcc-9.1.0/lib -lnetcdf -L/usr/local/apps/netcdf-fort
```

Edit the file in the ioapi folder called Makeinclude.Linux2_x86_64gfort to comment out all openMP options as CMAQ does not support openMP. Note: If users are using the ifort compiler you also need to remove -Bstatic flag within the ioapi/Makeinclude.Linux2_x86_64ifort file as well.

```
OMPFLAGS = # -fopenmp
OMPLIBS = # -fopenmp
```

In the top level IOAPI_3.2 directory run:

```
make configure
make
```

Other I/O API library configuration options are available, and users can see a list of these options within the I/O API documentation. For example, I/O API can be configured in a manner that allows the CMAQ model to be run with the parallel I/O (PIO) feature turned on called the "mpi" I/O API libraries (Wong et al. 2015). More information about how to enable PIO within CMAQ can be found in Appendix D. There is a also an I/O API version 3.2 "large" that is designed for applications with a large number of model output files (e.g. utilizing all of CMAQv5.3 optional diagnostic output files) and/or a large number of model variables (e.g. CMAQ-HDDM or CMAQ-ISAM applications). I/O API v3.2-large increases the MXFILE3 variable from 64 to 512 and increases the MXVARS3 varialbe from 2048 to 16384. This version is available as a zip file from the following address:

https://www.cmascenter.org/ioapi/download/ioapi-3.2-large.tar.gz

Installation instructions for I/O API v5.3-large are provided in README.txt in the .tar.gz file.

3.4 Optional Software

Table 3□3. Optional support software for CMAQ

Software	Description	Source
Evaluation and visualization tools		
VERDI	Visualization Environment for Rich Data Interpretation for graphical analysis of netCDF gridded data	<http://www.verdi-tool.org>
PAVE	Package for Analysis and Visualization of Environmental data for graphical analysis of netCDF gridded data	<http://www.cmascenter.org>
IDV	Integrated Data Viewer for 3-D graphical analysis of netCDF gridded data	<http://www.unidata.ucar.edu/software/idv/>
I/O API Tools	Postprocessing tools for manipulating data in the I/O API/netCDF format	<https://www.cmascenter.org/ioapi/>
netCDF Tools	Postprocessing tools for manipulating data in the netCDF format	<http://my.unidata.ucar.edu/content/software/netcdf/index.html>
Source code diagnostics		
GDB	Gnu Fortran debugger	<https://www.sourceware.org/gdb/>

Software	Description	Source
PGDBG	Portland Group Fortran debugger	<http://www.pgroup.com/>
PGPROF	Portland Group Fortran code profiler	<http://www.pgroup.com/>
IDB	Intel Fortran debugger	<https://software.intel.com/en-us/articles/idb-linux>

3.5 References:

Wong, D. C., Yang, C. E., Fu, J. S., Wong, K., and Gao, Y., "An approach to enhance pnetCDF performance in environmental modeling applications", Geosci. Model Dev., 8, 1033-1046, 2015.

4. Model Input Files

4.1 Introduction

This chapter provides basic information on the format and content of CMAQ input files. It also provides information on using the pre-processing tools provided in the repository for preparing initial and boundary conditions and meteorology inputs. Links are provided for the emissions processing tools that are released through their own repository or website. A list of CMAQ input files can be found in Table 4-1. Some CMAQ input files are in ASCII format while the majority of them are in the Network Common Data Form (netCDF) format. CMAQ input and output files are self-describing netCDF-format files in which the file headers have all the dimensioning and descriptive information needed to define the resident data. Users should download the latest code for the NetCDF from the NetCDF website. Compilation and configuration information for the NetCDF is available through the Unidata website.

All CMAQ input and output files are conformed to I/O API netCDF file format. Please refer to the I/O API User's Manual for details.

Full input datasets for 2016 over two domains are publically available to download from the CMAS Data Warehouse. The input files are stored on Google Drive with metadata organized through Dataverse.

Domain	Simulation Dates	Dataverse DOI
Southeast US	July 1 - 14, 2016	https://doi.org/10.15139/S3/IQVABD
CONUS	Jan 1 - Dec 31, 2016	https://doi.org/10.15139/S3/MHNUNE

4.2 CMAQ Pre-processors

Figure 2-1 shows the relationship between CMAQ pre-processors and the main CMAQ program, the CMAQ Chemistry Transport Model (CCTM). MCIP, ICON and BCON are included in the CMAQ repository and are used to create meteorological, initial conditions and boundary conditions inputs. SMOKE, FEST-C and Spatial Allocator Tools are external software packages required for creating emissions inputs for CMAQ. The following subsections provide more information on these software and point the user to additional sources of documentation.

4.2.1 Meteorology-Chemistry Interface Processor (MCIP)

MCIP processes meteorological fields output by the WRF model into files that are compatible with the CCTM and SMOKE (an emissions processor that computes emissions inputs for CMAQ). The output files generated by MCIP are used by ICON and BCON and various other programs in CMAQ, so MCIP must be the first program run after installing the CMAQ source codes and initializing CMAQ environment variables. Configuration options for MCIP include the time periods over which to extract data from the meteorological model output files, horizontal grid definitions for output, and control for optional 3D output variables. MCIP can either process the full horizontal domain from WRF or a user-defined subset of that domain (that is, a "window"). Unlike many of the programs in the CMAQ program suite MCIP is compiled with a Makefile and then run with a run script. Instructions on how to compile and run MCIP are provided in the README.md file in the PREP/mcip folder.

Most of the fields that are simulated by WRF are not modified by MCIP for the CCTM and the emissions model, and they are "passed through" to the output. In additions, fields that are required to transform to CMAQ's generalized coordinate system are calculated within MCIP. The dry deposition velocities are now calculated in the CCTM; MCIPv3.4 was the last version to calculate those velocities internally.

4.2.2 Initial Conditions Processor (ICON)

ICON generates a gridded netCDF file of the chemical conditions for all grid cells in the modeling domain for the initial time of a simulation. It can generate these initial conditions from either an existing CCTM output file or one of four ASCII files of vertically resolved concentration profiles distributed with CMAQ. Running ICON requires that the user already generated MCIP files for their target modeling domain. For both input file options, ICON will interpolate the data to the horizontal and vertical structure of the target domain as defined in the MCIP files. The species in the ICON output file are identical to those in the input (either CCTM output or ASCII profile) file.

Using an existing CCTM output file to generate initial conditions is applicable when interpolating initial conditions from a coarse to a fine grid domain, as may occur when setting up nested simulations (simulations with finer-resolution grids that cover part of coarser-resolution grids). This is the preferred mode of specifying initial conditions since the spatial concentration patterns derived from the coarser-resolution simulation can be considered a first approximation of the concentration fields over the finer-resolution subdomain at the beginning of the simulation.

The four ASCII files of vertically resolved concentration profiles distributed with CMAQ represent annual average concentrations at a grid cell over the Pacific derived from a simulation with the hemispheric version of CMAQv5.3 beta2 for the year 2016. As such, these concentration profiles are reflective of conditions in a remote marine environment. The simulation was performed with the cb6r3m_ae7_kmtbr chemical mechanism and profiles for racm_ae6_aq, saprc07tc_ae6_aq, and saprc07tic_ae7i_aq were derived using the species mapping approach described in Step 3 of the CMAQ Tutorial on creating Initial and Boundary Conditions from Seasonal Average Hemispheric CMAQ Output. If one of these ASCII profile files is used to generate initial conditions, the resulting concentration fields will be uniform over the modeling domain and will not be a realistic representation of conditions over the modeling domain. As a result, simulations initialized with profile-derived rather than CCTM-derived concentration fields may require longer spin-up periods before conditions simulated within the domain no longer are influenced by these unrealistic initial concentration fields.

The configuration options for ICON include choosing whether the initial conditions are generated from an existing CCTM output file or from an ASCII profile, and defining the horizontal and vertical grids and time for which initial conditions are to be generated. Information on configuring ICON for the different kinds of input data, environment variables, input and output files, compiling and running ICON are provided in the README.md file in the PREP/icon folder.

4.2.3 Boundary Conditions Processor (BCON)

BCON generates a netCDF file of the chemical conditions along the lateral boundaries of the modeling domain. BCON will generate an output file with chemical concentrations for all grid cells along the modeling domain's horizontal boundaries. It can generate these boundary conditions from either an existing CCTM output file or one of four ASCII files of vertically resolved concentration profiles distributed with CMAQ. Running BCON requires that the user already generated MCIP files for their target modeling domain. For both input file options, BCON will interpolate the data to the horizontal and vertical structure of the target domain as defined in the MCIP files. The species in the BCON output file are identical to those in the input (either CCTM output or ASCII profile) file. Depending on user specified options and/or input datasets, the boundary conditions generated by BCON can be time varying, time independent, and either spatially uniform or variable across the model boundaries.

Using an existing CCTM output file to generate boundary conditions is applicable when setting up windowed simulations (simulations with the same resolution that cover only a part of the outer domain) or nested simulations (simulations with finer-resolution grids that cover part of coarser-resolution grids). This is the preferred mode of specifying boundary conditions since the spatial concentration patterns derived from the coarser-resolution simulation will be spatially varying along the boundaries of the finer-resolution domain. Boundary conditions generated from CCTM output files will be either time varying or time independent.

The four ASCII files of vertically resolved concentration profiles distributed with CMAQ represent annual average concentrations at a grid cell over the Pacific derived from a simulation with the hemispheric version of CMAQv5.3 beta2 for the year 2016. As such, these concentration profiles are reflective of conditions in a remote marine environment. The simulation was performed with the cb6r3m_ae7_kmtbr chemical mechanism and profiles for racm_ae6_aq, saprc07tc_ae6_aq, and saprc07tic_ae7i_aq were derived using the species mapping approach described in Step 3 of the

CMAQ Tutorial on creating Initial and Boundary Conditions from Seasonal Average Hemispheric CMAQ Output. If one of these ASCII profile files is used to generate boundary conditions, the resulting concentration fields will be uniform along the boundaries of the modeling domain and will not vary in time. Therefore, they are not a realistic representation of conditions along the domain boundaries and should only be used in cases where boundary conditions are not expected to affect the interpretation of model results.

CMAQ can use boundary conditions derived from global chemistry models (GCMs). While BCON does not directly support processing of datasets from GCMs (other than the hemispheric version of CMAQ) in their native formats, users could develop their own custom codes to transform their GCM datasets into I/O API format, which would then allow these datasets to be input into BCON in the same way as an existing CCTM output file.

The configuration options for BCON include choosing whether the boundary conditions are generated from an existing CCTM output file or from an ASCII profile, and defining the horizontal and vertical grids and time period for which boundary conditions are to be generated. Information on configuring BCON for the different kinds of input data, environment variables, input and output files, compiling and running BCON are provided in the README.md file in the PREP/bcon folder.

4.2.4 External Software Programs for Preparing CMAQ Inputs

The SMOKE and FEST-C modeling systems and the Spatial Allocator tools are used to create CMAQ emissions and land surface inputs. These systems are maintained by EPA and CMAS developers and are hosted and supported by the CMAS Center. Links to documentation and software download for each system are provided below.

Emissions Processor (SMOKE) Sparse Matrix Operator Kerner Emissions (SMOKE) Modeling System is designed to create gridded, speciated, hourly emissions for input into CMAQ and other air quality models. SMOKE supports area, biogenic, mobile (both onroad and nonroad), and point source emissions processing for criteria, particulate, and toxic pollutants. For biogenic emissions modeling, SMOKE uses the Biogenic Emission Inventory System. SMOKE is also integrated with the on-road emissions model MOBILE6 and MOVES.

Fertilizer Emissions Processor (FEST-C) The Fertilizer Emission Scenario Tool for CMAQ (FEST-C) system is used to generate agricultural-land nitrogen and soil information for CMAQ bi-directional NH3 modeling. FEST-C contains three main components: Java interface, Environmental Policy Integrated Climate (EPIC) model, and SA Raster Tools. The interface guides users through generating required land use and crop data and EPIC input files and simulating EPIC, and extracting EPIC output for CMAQ.

FEST-C is used to create the E2C_LU, E2C_SOIL, and E2C_CHEM files discussed later in this chapter.

Processing Spatial Data with the Spatial Allocator (SA) The Spatial Allocator is a set of tools that helps users manipulate and generate data files related to emissions and air quality modeling. The tools perform functions similar to Geographic Information Systems (GIS), but are provided to the modeling community free of charge. In addition, the tools are designed to support some of the unique aspects of the file formats used for CMAQ, SMOKE and WRF modeling.

SA is used to generate the surf zone and open ocean file that is a required input for utilizing marine emissions and chemistry in CMAQ. This file is discussed later in this chapter under the OCEAN_1: Sea spray mask section.

Additional information on processing data for CMAQ inputs is provided in Appendix C.

4.3 CMAQ Input Files

Jump to Table of Input Files Jump to CCTM Output Files in Chapter 6

CMAQ requires a basic set of input files: initial condition file, which is created by ICON process or previous day output; boundary condition file, which is created by BCON process; emission files; and meteorological data created by MCIP using WRF and terrain data. Additional input files may be required based on specific run time options. CMAQ output files include a basic set of files with aerosol and gas-phase species concentrations, wet and dry deposition estimates, and visibility metrics, and an auxiliary set of output files for diagnosing model performance and in-line-calculated emissions. Model outputs are discussed in Chapter 6.

Rather than forcing the user to deal with hard-coded file names or hard-coded unit numbers, the I/O API netCDF file format utilizes the concept of logical file names. The modelers can define the logical names as properties of a program, and then at run-time the logical names can be linked to the actual file name using environment variables. For programming purposes, the only limitations are that logical file names cannot contain blank spaces and must be at most 16 characters long. When a modeler runs a program that uses the I/O API format, environment variables must be used to set the values for the program's logical file names. A complete list of CMAQ input is provided in Table 4-1.

This section describes each of the input files required by the various CMAQ programs. The section begins with a description of the grid definition file, GRIDDESC, which is used by several CMAQ programs, and then goes through a program-by-program listing of the CMAQ input file requirements. Table 4-1 lists the source, file type (e.g. ASCII, GRDDED3, BNDARY3, etc.), and temporal and spatial dimensions of each CMAQ input file. Typical time step is 1 hour; however a user can specify a finer one, e.g. 20 minutes. In addition, typical thickness of a boundary file is 1, i.e. NTHIK = 1 but it can be any positive integer.

Table 4-1. CMAQ input files. Note that when "Time-Dependence" is listed as "Hourly", it is shorthand for a time-varying file. It is recommended that CMAQ use a time increment that is no longer than one hour. However, CMAQ can be run with a Time Dependence that is shorter than hourly.

Environment Variable Name for File	File Type	Time-Dependence	Spatial Dimensions	Source	Required
General					
GRIDDESC	ASCII	n/a	n/a	MCIP	required
gc_matrix_nml	ASCII	n/a	n/a	CMAQ repo	required
ae_matrix_nml	ASCII	n/a	n/a	CMAQ repo	required
nr_matrix_nml	ASCII	n/a	n/a	CMAQ repo	required

Environment Variable Name for File	File Type	Time-Dependence	Spatial Dimensions	Source	Required
tr_matrix_nml	ASCII	n/a	n/a	CMAQ repo	required
Initial Conditions Inputs					
INIT_CONC_1	GRDDED3	Time-invariant	XYZ	ICON or CCTM	required
Boundary Conditions Inputs					
BNDY_CONC_1	BNDARY3	Hourly	PERIM*Z	BCON	required
MCIP					
GRID_CRO_2D	GRDDED3	Time-invariant	XY	MCIP	required
GRID_BDY_2D	BNDARY3	Time-invariant	PERIM*Z	MCIP	required
GRID_DOT_2D	GRDDED3	Time-invariant	(X+1)*(Y+1)	MCIP	required
MET_BDY_3D	BNDARY3	Hourly	PERIM*Z	MCIP	required
MET_CRO_2D	GRDDED3	Hourly	XY	MCIP	required
MET_CRO_3D	GRDDED3	Hourly	XYZ	MCIP	required
MET_DOT_3D	GRDDED3	Hourly	(X+1)*(Y+1)Z	MCIP	required
LUFRAC_CRO	GRDDED3	Time-invariant	XYL	MCIP	required
SOI_CRO	GRDDED3	Hourly	XYS	MCIP	optional (Contains soil moisture and soil temperature in layers. A two-layer representation of those fields is currently mirrored in MET_CRO_2D.)

Environment Variable Name for File	File Type	Time-Dependence	Spatial Dimensions	Source	Required
MOSAIC_CRO	GRDDED3	Hourly	XYM	MCIP	optional (Contains surface fields in mosaic land use categories if Noah Mosaic LSM was run in WRF. Can work with STAGE deposition in CCTM.)
mcip.nc	netCDF	varies by field	varies by field	MCIP	required if IOFORM=2 (Currently not compatible with rest of CMAQ system.)
mcip_bdy.nc	netCDF	varies by field	varies by field	MCIP	required if IOFORM=2 (Currently not compatible with rest of CMAQ system.)
Emissions Inputs					
GR_EMIS_XXX*	GRDDED3	Hourly	XYZ	SMOKE	required
STK_GRPS_XXX	GRDDED3	Time-invariant	XY	SMOKE	required

Environment Variable Name for File	File Type	Time-Dependence	Spatial Dimensions	Source	Required
STK_EMIS_XXX	GRDDED3	Hourly	XY	SMOKE	required
NLDN_STRIKES	GRDDED3	Hourly	XY	Must purchase data	optional for including NO from lightning
LTNGPARMS_FILE	GRDDED3	Time-invariant	XY	CMAS	required for including NO from lightning
Biogenic and Land Surface Inputs					
OCEAN_1	GRDDED3	Time-invariant	XY	Spatial Allocator	required
GSPRO	ASCII	Time-invariant	N/a	CMAQ repo	required for running CMAQ with online biogenics
B3GRD	GRDDED3	Time-invariant	XY	SMOKE	required for running CMAQ with online biogenics
BIOSEASON	GRDDED3	Time-invariant	XY	SMOKE	run-time option for running CMAQ with online biogenics

Environment Variable Name for File	File Type	Time-Dependence	Spatial Dimensions	Source	Required
E2C_LU	GRDDED3	Time-invariant	XY	EPIC	required for running CMAQ with bidirectional NH3
E2C_SOIL	GRDDED3	Time-invariant	XY	EPIC	required for running CMAQ with bidirectional NH3
E2C_CHEM	GRDDED3	Daily	XY	EPIC	optional
DUST_LU_1	GRDDED3	Time-invariant	XY	Spatial Allocator	optional when running CMAQ with wind-blown dust
DUST_LU_2	GRDDED3	Time-invariant	XY	Spatial Allocator	optional when running CMAQ with wind-blown dust
Photolysis					
OMI	ASCII	Daily	n/a	CMAQ repo or create_omi	required

*XXX - three-digit variable indicating emission stream number. Gridded and Inline Point emissions are numbered independently.

4.4 GRIDDESC and Species Namelist Files

GRIDDESC: Horizontal domain definition Return to Table 4-1 Used by: ICON, BCON, CCTM

The CMAQ grid description file (**GRIDDESC**) is an ASCII file that contains two sections: a horizontal coordinate section, and domain description section. The GRIDDESC file is generated automatically by MCIP; alternatively, GRIDDESC can be created using a text editor.

The horizontal coordinate section consists of text records that provide the coordinate-system name, the map projection, and descriptive parameters that define the projection. This section is used to provide projection information that is used by a family of nested domains, where the coordinate-system name is shared by each of the domains.

The grid description section consists of text records that indicate the grid name, related coordinate-system name (i.e., which GRIDDESC horizontal coordinate name that is defined in the previous section that is applied to this grid), and descriptive parameters for the coordinates of the lower-left corner of the grid, grid cell size, number of columns, and rows. There are at most 32 coordinate systems and 256 grids that can be listed in one of these files. These files are small enough to be archived easily with a study and have a sufficiently simple format that can easily be constructed "by hand." The elements of the GRIDDESC files are typically included with the metadata for the output files in the CMAQ system.

An example of a GRIDDESC file is shown below:

```
' '

'LAM_40N100W'

2 30.0 60.0 -100.0 -100.0 40.0

' '

'M_32_99TUT02'

'LAM_40N100W' 544000.0 -992000.0 32000.0 32000.0 38 38 1

' '
```

The horizontal coordinate section (first section) in this example GRIDDESC file defines a horizontal coordinate named "LAM_40N100W". The coordinate definition is for a Lambert conformal grid, keyed by the first column of the coordinate description line, which corresponds to the numeric code for the various I/O API-supported grid types (2 = Lambert). The next three parameters (P_ALP, P_BET, and P_GAM) have different definitions for different map projections. For Lambert conformal, P_ALP and P_BET are the true latitudes of the projection cone (30°N and 60°N in the example), and P_GAM (100°W in the example) is the central meridian of the projection. The last two parameters, XCENT and YCENT, are the reference longitude and latitude for the domain, which are 100°W and 40°N in the example.

The second section in the example describes a domain named "M_32_99TUT02". In this example, the coordinate named "LAM_40N100W" is referenced in the domain definition. The next two parameters in the domain definition (XORIG and YORIG) are the east-west and north-south offsets from XCENT and YCENT in meters. The next two parameters (XCELL and YCELL) are the horizontal grid spacing in meters for the X and Y directions (i.e., Δx and Δy). The next two parameters (NCOLS and NROWS) are the numbers of grid cells in the X and Y directions. The grid definition concludes with the number

of boundary cells, NTHIK, which is typically set to 1. Note that the number of boundary cells for CMAQ differs from that used by WRF.

Additional information about the parameters in the GRIDDESC file can be found in the I/O API Documentation.

{gc|ae|nr|tr}_matrix.nml: Species namelist files Return to Table 4-1

Used by: CCTM, CHEMMECH

Namelist look-up tables for different classes of simulated pollutants are used to define the parameters of different model species during the execution of the CMAQ programs. Gas-phase (gc), aerosol (ae), non-reactive (nr), and tracer (tr) species namelist files contain parameters for the model species that are included in these different classifications. The species namelist files are used to control how the different CMAQ programs and processes handle the model species. The namelist files define the following processes for each model species:

- Initial conditions – which initial condition species is the pollutant mapped to; if not specified, this will default to the species name.
- IC Factor – if the pollutant is mapped to an initial condition species, uniformly apply a scaling factor to the concentrations.
- Boundary conditions – which boundary condition species is the pollutant mapped to; if not specified, this will default to the species name.
- BC Factor – if the pollutant is mapped to a boundary condition species, uniformly apply a scaling factor to the concentrations.
- Deposition velocity – which (if any) deposition velocity is the deposition velocity for the pollutant mapped to; allowed velocities are specified within the model source code.
- Deposition velocity factor – if the pollutant is mapped to a deposition velocity, uniformly apply a scaling factor to this velocity.
- Scavenging - which (if any) species is the pollutant mapped to; Allowed scavenging surrogates are specified within the model source code ("hlconst.F").
- Scavenging factor - if the pollutant is mapped to a species for scavenging, uniformly apply a scaling factor to the scavenging rate.
- Gas-to-aerosol conversion – which (if any) aerosol chemistry species does the gas phase pollutant concentration go into for transformation from the gas-phase to the aerosol-phase. Allowed gas-to-aerosol surrogates are specified within the model source code ("PRECURSOR_DATA.F" and "SOA_DEFN.F")
- Gas-to-aqueous Surrogate – which (if any) cloud chemistry species does the gas pollutant concentration go into for simulating chemistry within cloud water. Allowed gas-to-aqueous surrogates are specified within the model source code and depends on the cloud model/aqueous chemistry being used (for example, for the acm_ae6, see "AQ_DATA.F").
- Aerosol-to-aqueous Surrogate – which (if any) cloud chemistry species does the aerosol pollutant concentration go into for simulating chemistry within cloud water. Allowed aerosol-to-aqueous surrogates are specified within the model source code and depends on the cloud model/aqueous chemistry being used (for example, for the acm_ae6, see "AQ_DATA.F").
- Transport – is the pollutant transported by advection and diffusion in the model?
- Dry deposition – Write the pollutant to the dry deposition output file?

- Wet deposition – Write the pollutant to the wet deposition output file?
- Concentration – Write the pollutant to the instantaneous concentration output file?

The namelist files contain header information that describe which class of species are contained in the file, the number of parameters contained in the file, headers describing the parameter fields, and then a series of rows with configuration parameters for every model species. Table 4-2 contains the namelist file format for the gas-phase (GC) species namelist file. The namelist files for the other species classifications (AE, NR, TR) are similar to the format shown in Table 4-2.

Table 4-2. GC species namelist file format

Line	Column	Name	Type	Description	Options for Syntax:
1		File Type	String	String to delineate Gas Phase (GC), Aerosol (AE), Non-reactive (NR) and Tracer (TR) species namelist	{&GC_nml, &AE_nml, &NR_nml, &TR_nml}
3		Header ID	String	String to define data structure relating to namelist	{GC_SPECIES_DATA=, AE_SPECIES DATA= , NR_SPECIES_DATA= ,TR_SPECIES_DATA = }
5	1	SPECIES	String	CMAQ Species name, i.e. NO, HNO3, PAR; dependent on chemical mechanism	-
	2	MOLWT	Integer	Species Molecular Weight	-
	3	IC	String	IC surrogate species name for the CMAQ Species	{‘Species name’, ’ ’}
	4	FAC	Integer	Scaling factor for the IC concentration	{Any real: default = -1 if IC is not specified}
	5	BC	String	BC surrogate species name for the CMAQ Species	{‘Species name’, ’ ’}
	6	FAC	Integer	Scaling factor for the BC concentration	{Any real: default = -1 if BC is not specified}
	7	DRYDEP SURR	String	Deposition velocity variable name for the CMAQ Species	{‘Species name’, ’ ’}
	8	FAC	Integer	Scaling factor for the deposition velocity	{Any real: default = -1 if SURR is not specified}
	9	WET-SCAV SURR	String	Wet Deposition Scavenging surrogate species	{‘Species name’, ’ ’}

Line	Column	Name	Type	Description	Options for Syntax:
	10	FAC	Integer	Scaling factor for Scavenging	{Any real: default = -1 if SURR is not specified}
	11	GC2AE SURR	String	Gas-to-aerosol transformation species	{'Species name', ' '}
	12	GC2AQ SURR	String	Gas-to-aqueous transformation species	{'Species name', ' '}
	13	TRNS	String	Transport Switch. *NOTE*: Instead of using one column labeled "TRNS" to turn/off both advection and diffusion for a pollutant, two separate columns labeled "ADV" and "DIFF" can be used to switch on/off advection and diffusion separately.	{YES/NO}
	14	DDEP	String	Dry deposition output file switch	{YES/NO}
	15	WDEP	Real	Wet deposition output file switch	{YES/NO}
	16	CONC	String	Concentration output file switch	{YES/NO}

The namelist files for the other pollutant classes have similar configurations as the gas-phase species configuration shown in Table 4-2. For an example see this link to the GC namelist species file for the cb06r3_ae7_aq mechanism.

4.5 Initial Conditions Input

INIT_CONC_1: Initial conditions Return to Table 4-1 Used by: CCTM

The initial concentrations of each species being modeled must be input to CMAQ. The initial conditions input file type is GRDDED3 and does not vary with time. The actual file data are organized in this manner: by column, by row, by layer, by variable. Initial conditions files have the same structure as concentration files, so the predicted concentrations from the last hour of day 1 can be used to initialize the following day's simulation. This gives CMAQ users the flexibility to segment simulations in any way they choose.

4.6 Boundary Conditions Input **BNDY_CONC_1: Boundary conditions** Return to Table 4-1

Used by: CCTM

CMAQ boundary condition data are of the BNDARY3 file type. Produced by the boundary condition processor, BCON, CCTM reads these data and correlates them with the interior data using a pointer system. This pointer system designates the beginning location of the data in memory that start a new side of the domain (i.e., south, east, north, or west). Consult I/O API User Guide for a pictorial description.

Each species being modeled should be in the BNDY_CONC_1 file. If some modeled species are not contained in this file, the boundary condition for these species will default to the value 1 × 10e-30. The perimeter of the CMAQ domain is NTHIK cell wide (typically NTHIK = 1), where the number of boundary cells = NTHIK*(2*NCOLS + 2*NROWS +4*NTHIK).

4.7 Meteorological Inputs (Processed for the CMAQ System using MCIP)

MCIP output files generated when IOFORM=1 (Models-3 I/O API) - GRIDDESC: Grid description used throughout the CMAQ System - GRID_CRO_2D: Time-invariant 2D fields (XY) at cell centers (cross points) - GRID_BDY_2D: Time-invariant fields from GRID_CRO_2D, but along domain lateral boundaries - GRID_DOT_2D: Time-invariant 2D fields (XY) at cell corners (dot points) and cell faces - MET_CRO_2D: Time-varying 2D fields (XY) at cell centers (cross points) - MET_CRO_3D: Time-varying 3D fields (XYZ) at cell centers (cross points) - MET_BDY_3D: Time-varying fields from MET_CRO_3D, but along domain lateral boundaries - MET_DOT_3D: Time-varying 3D fields (XYZ) at cell corners (dot points) and cell faces - LUFRAC_CRO: Time-invariant 3D fractional land use (XYL) at cell corners (cross points) - SOI_CRO: Time-varying 3D soil moisture and temperature (XYS) in model soil layers at cell centers - MOSAIC_CRO: Time-varying 3D surface fields by mosaic land use category (XYM) at cell centers

MCIP output files generated when IOFORM=2 (netCDF) - GRIDDESC: Grid description used throughout the CMAQ System - mcip.nc: All time-invariant and time-varying 2D and 3D fields (all dimensions) - mcip_bdy.nc: All required time-invariant and time-varying 2D and 3D fields along lateral boundaries

5. Running a CMAQ Simulation

5.1 Introduction

During this chapter the user will learn about how to obtain the CMAQ source codes and how to set-up their CMAQ environment to complete a CMAQ simulation. It should be noted that before you can configure your CMAQ Environment, consult the chapter “Preparing to run” to see you have the minimum requirement of hardware and software on your system.

5.2 Getting the CMAQ Source Code

CMAQ source code can be installed either using git or from tarballs downloaded from the git repository hosted by GitHub. Both options are described here.

5.2.1 Git Installation

In the directory where you would like to install CMAQ, issue the following command to clone the official EPA GitHub repository for CMAQv5.3:

```
git clone -b master https://github.com/USEPA/CMAQ CMAQ_REPO
```

Using the git clone option, CMAQ will install into the following directories:

```
CMAQ_REPO/CCTM
CMAQ_REPO/PREP
CMAQ_REPO/POST
CMAQ_REPO/UTIL
CMAQ_REPO/DOCS
```

5.2.2 Zip file Installation

Zip files of the CMAQ source code are available from the public GitHub repository. Click the button "Clone or download" from https://github.com/USEPA/CMAQ and select "Download ZIP" to download a Zip file of the CMAQv5.3 repository. Alternatively, you may download the Zip file from the EPA CMAQ website.

Reference input/output data for testing the installation of the software are available from the CMAS Center; *data are not available through GitHub*. You must register/login to access the source codes and data from the CMAS Center.

In the directory where you would like to install CMAQ, unzip the model distribution file:

```
unzip CMAQ-master.zip
```

The following directories will be created:

```
CMAQ-master/CCTM
CMAQ-master/PREP
CMAQ-master/POST
CMAQ-master/UTIL
CMAQ-master/DOCS
```

The Git and Zip file installation options will produce slightly different subdirectories on your Linux system. The base installation directory using the git clone command will be `CMAQ_REPO`; the directory from the Zip file will be `CMAQ-master`. The subsequent instructions in this guide will be based on the git clone installation. For Zip file installations, replace `CMAQ_REPO` with `CMAQ-master` in the instructions that follow. The differences in the directory names highlights the difference in functionality between the two options. Cloning the repository gives the user access to the full repository and its history, while downloading the Zip file will only give access to version 5.3.

5.3 The CMAQ Repository Structure

After downloading the source codes the user is encouraged to look through the repository to familiarize themselves with the structure. A summarized image of the repository is shown below:

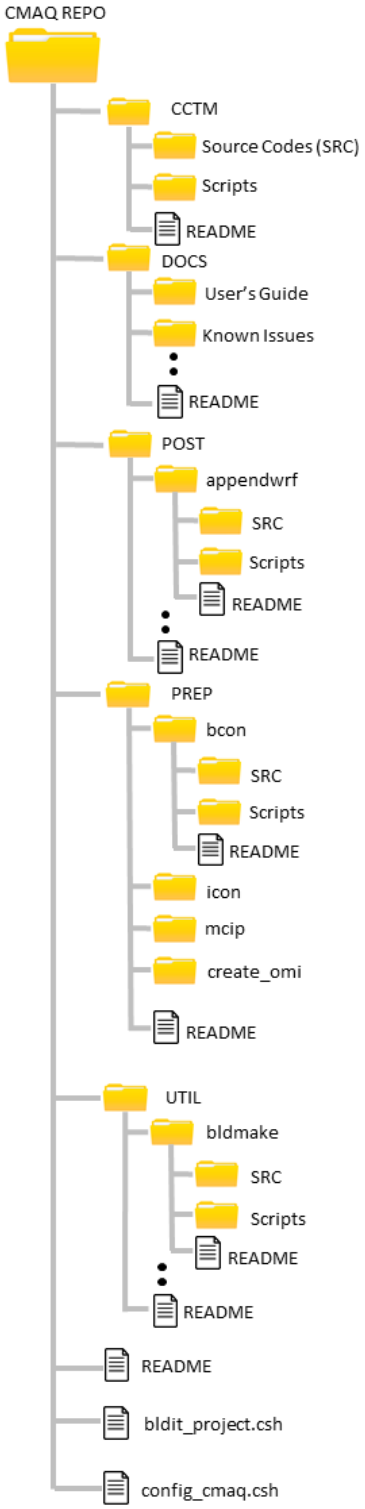

CMAQ REPO
CCTM
Source Codes (SRC)
Scripts
README
DOCS
User's Guide
Known Issues
README
POST
appendwrf
SRC
Scripts
README
README
PREP
bcon
SRC
Scripts
README
icon
mcip
create_omi
README
UTIL
bldmake
SRC
Scripts
README
README
README
bldit_project.csh
config_cmaq.csh

Figure 5□1. CMAQ repository structure

In this image it can be seen that there are four main sub folders within the CMAQ repository. The first folder, CCTM, houses all the source codes (i.e. fortran/C programs) and scripts that drive the CMAQ Chemistry Transport Model (CCTM).

The second folder, DOCS, contains all relevant documentation pertaining to the CMAQ program suite including the User Manual, Release Notes and Known issues associated with the current release and a Developers Guide for a general description of CMAQ's open-source collaboration workflow and step-by-step instructions for how to make code contributions through GitHub.

The third folder in the repository is the POST folder which contains several very useful tools for post-processing of the input/output data files. Each tool within the folder comes wth the source code, scripts and a README used to run the tool. A technical description of the tools within this folder can be found in Chapter 8.

The fourth folder in the repository is the PREP folder which contains several pre-processing programs that can be run before the CCTM to prepare meteorology, initial conditions and boundary conditions inputs. Similar to the POST tools, documentation on compiling and running the programs is provided within each subfolder under PREP.

The last folder within the repository is the UTIL folder which contains useful utilities relating to the CMAQ program suite. An example is the bldmake utility which is used to compile the source code into executables when you use any of the build scripts in the CMAQ repository. Also included in this repository is a top-level README file with an overview of the contents of the release and two additional C-Shell scripts, `bldit_project.csh` and `config_cmaq.csh`. `bldit_project.csh` allows the user to extract the build and run scripts and compile the model outside of the repository, while `config_cmaq.csh` helps enforce consistent environment setting for the CMAQ project. Both these scripts will be discussed in the following sections.

5.4 Building CMAQ Outside of the Repository in a User-Specified Directory

When cloning the repository or unpacking the tar file of the CMAQ distribution, the top-level directory is recognized by the default build and run scripts as `CMAQ_HOME` (formerly M3HOME prior to CMAQv5.2). This directory is an arbitrary base location of the CMAQ installation on your Linux system for a specific application. If the user will build and run CMAQ within the repository folder structure, then `CMAQ_HOME` does not need to be set explicitly in the `bldit_project.csh` script. If, on the other hand, the user wishes to extract the build and run scripts and compile the model outside of the repository, then `CMAQ_HOME` will need to be specified in `bldit_project.csh`. Executing `bldit_project.csh` will automatically perform this extraction and create a CMAQ folder structure under the location now specified by `CMAQ_HOME`. To perform this operation, modify the variable `CMAQ_HOME` in the `bldit_project.csh` script to identify the folder that you would like to install the CMAQ package under. For example:

```
set CMAQ_HOME = /home/username/CMAQ_v5.3
```

Now execute the script:

```
./bldit_project.csh
```

It should be noted that from now on, the other CMAQ directories are referenced relative to CMAQ_HOME and it is where your CMAQ project will be run from. While this directory structure is convenient for the benchmark case and most CMAQ applications, other configurations are possible.

5.5 Initialization of CMAQ Environment

Consistency of configuration variables is critical for building CMAQ itself, not just its libraries. Accordingly CMAQ includes the configuration script `config_cmaq.csh` to help enforce consistent environment settings for CMAQ and its associated libraries Appendix A lists the `config_cmaq.csh` variables defined for the build process and suggests values to which to set those variables.

Note that for multiprocessor applications it is recommended that the Fortran MPI wrapper script mpifort (for Intel compiler and for GNU and PGI fortran compiler, use mpifort) be specified for the Fortran compiler (myFC). Using this script, instead of a direct call to the Fortran compiler, will ensure that the full suite of MPI components (libraries and include files) for the compiler are included in the parallel build without anything provided by the user explicitly.

Use the following steps to initialize your CMAQ environment:

```
source config_cmaq.csh [compiler]
```

After running the above command, it should be noticed now under CMAQ_HOME, the data file directory has been created and serves as a container for the input and output data for the model, and the lib directory contains links to the compiled binary library files required to build the CMAQ executables. The CMAQ scripts use the following environment variables to alias the locations of these directories:

`CMAQ_LIB = $CMAQ_HOME/lib` (M3LIB before CMAQv5.2) `CMAQ_DATA = $CMAQ_HOME/data` (M3DATA before CMAQv5.2)

If you encounter errors about libraries not being found, check the settings of the `config_cmaq.csh` script variables IOAPI, NETCDF, or MPI to ensure that they correctly point to the locations of these libraries on your Linux system.

Sourcing the `config_cmaq.csh` script only needs to be invoked during a new installation of CMAQ to make sure the links to these libraries are working correctly. For every successive session, links to these libraries will automatically be created when you run any of the build or run scripts.

5.6 Compiling CMAQ Chemistry-Transport Model (CCTM)

After all required CMAQ inputs are generated using the preprocessors mentioned above the user is now ready to compile CCTM. CMAQ's current coding structure is based on a modularity level that distinguishes from each other CCTM's main driver, science modules, data estimation modules, and

control/utility subroutines. Also distinguished from each other are the science models (including sub-models for meteorology, emissions, chemistry-transport modeling) and the analysis and visualization subsystems.

In CCTM, the process modules that affect the pollutant concentration fields are classified as listed below. Each bullet contains a description of the process followed by module name in parentheses. These modules are discussed further in Chapter 6.

Science Modules:

- Horizontal advection (hadv)
- Vertical advection (vadv)
- Horizontal diffusion (hdiff)
- Vertical diffusion (vdiff)
- Emissions (emis)
- In-line BEIS3 biogenic emissions (biog)
- In-line plume rise (plrise)
- Gas-phase chemical reaction solver (gas)
- Aqueous-phase reactions and cloud mixing (cloud)
- Aerosol dynamics and size distributions (aero)
- Potential vorticity scaling for stratosphere/troposphere exchange (pv_o3)

The user has the ability to configure the model in a multitude of ways by selecting from different options for each scientific process. Model configuration is split into build time options and run time options. To modify any science options during build time, edit the `bldit_cctm.csh` script. The `bldit_cctm.csh` script also contains other information, such as the option to run in single or multiprocessor mode as well as debug mode. To modify any run time options, such as turn on in-line biogenic emission calculation or use in-line windblown dust emission, edit the run script, `run_cctm.csh`, and set the corresponding environment variable. To read more about build and run time configurations for specific scientific processes, see the next chapter (Chapter 6). To see a complete list configuration options reference Appendix A.

Once the `bldit_cctm.csh` script is configured to the user's preference, the user is ready to run the script to build the CCTM executable. To do this run the following commands:

```
cd $CMAQ_HOME/CCTM/scripts
source bldit_cctm.csh [compiler] [version] |& tee build_cctm.log
```

The bldit script invokes the CMAQ utility program bldmake, which extracts source code from your CMAQ GIT repository, constructs a Makefile based on your selected options, and compiles the executable automatically. Following normal termination of the script with the default configuration, the user will notice a BLD directory created. This is the location of the CCTM executable along with the relevant source codes and the Makefile needed to build the model. In this directory a few useful commands can be used to update the executable if any changes are made to the fortran source codes via the MakeFile. For example, if the user wants to recompile the source codes in debug mode instead of re-running the `bldit_cctm.csh` script the user can use the following commands:

```
cd BLD_CCTM_v53_[compiler][version]
make clean
make DEBUG=TRUE
```

In another example, if the user has made any changes to the source codes in the BLD directory and wanted to update the CCTM executable to reflect these changes the user can use the following commands:

```
cd BLD_CCTM_v53_[compiler][version]
make
```

The Make utility is smart enough to compile only the modified files and all associated file which are defined by the dependency of each source file in the Makefile.

5.7 Running CCTM

After setting up the CCTM executable the model is ready to be run. Much like the `bldit_cctm.csh` script, to modify any run time options edit the `run_cctm.csh` script referencing Appendix A for a complete list of optional settings. After these settings have been configured use the following commands to run the script:

```
cd $CMAQ_HOME/CCTM/scripts
run_cctm.csh |& tee run_cctm.log
```

5.7.1 CCTM Logfiles

The CCTM simulation will write two types of logfile, a master logfile (e.g. run_cctm.log) and processor-specific logfiles that have the name convention:

```
CTM_LOG_[ProcessorID].v53_[compiler]_[data_name]/_[RUNDATE].log
```

The master logfile contains extensive metadata and useful information about the details of your simulation. The following examples describe some of this information:

```
Start Model Run At  Tue Apr 9 08:18:06 EDT 2019
Compiler is set to intel
No compiler version given. Atmos system Detected. Assume Intel 18.0

Working Directory is ...
Build Directory is ...
Output Directory is ...
Log Directory is ...
```

```
Executable Name is CCTM_v53.exe

---CMAQ EXECUTION ID: CMAQ_CCTMv53_[userID]_YYYYMMDD_hhmmss_nanosecs ---

Set up input and output files for Day YYYY-MM-DD.

Existing Logs and Output Files for Day YYYY-MM-DD Will Be Deleted
/bin/rm: No match.

CMAQ Processing of Day 20140620 Began at Tue Apr  9 08:18:07 EDT 2019
```

This section documents the folder structure, username, and run date for the simulation, and is meant to aid in maintaining transparency of simulation results after runs have been completed. This section is followed by the CMAQ and I/O API headers, and a record of all environment variables and their values for this simulation.

Next, the program outputs a table describing the domain decomposition breakdown for the run.

```
      -=-  MPP Processor-to-Subdomain Map  -=-
             Number of Processors = 128
 ____________________________________________________
|                                                    |
|  PE    #Cols    Col_Range     #Rows    Row_Range  |
|____________________________________________________|
|                                                    |
|  0       12      1:  12         24      1:  24    |
|  1       12     13:  24         24      1:  24    |
|  2       12     25:  36         24      1:  24    |
|  3       12     37:  48         24      1:  24    |
|  4       12     49:  60         24      1:  24    |
|  5       12     61:  72         24      1:  24    |
|  6       12     73:  84         24      1:  24    |
|  7       12     85:  96         24      1:  24    |
|  8       12     97: 108         24      1:  24    |
|  9       12    109: 120         24      1:  24    |
| 10       12    121: 132         24      1:  24    |
| 11       11    133: 143         24      1:  24    |
| 12       11    144: 154         24      1:  24    |
| 13       11    155: 165         24      1:  24    |
| 14       11    166: 176         24      1:  24    |
| 15       11    177: 187         24      1:  24    |
| 16       12      1:  12         24     25:  48    |
| 17       12     13:  24         24     25:  48    |
| 18       12     25:  36         24     25:  48    |
| 19       12     37:  48         24     25:  48    |
```

```
| 20       12     49:  60         24     25:  48   |
| 21       12     61:  72         24     25:  48   |
| 22       12     73:  84         24     25:  48   |
| 23       12     85:  96         24     25:  48   |
| 24       12     97: 108         24     25:  48   |
| 25       12    109: 120         24     25:  48   |
| 26       12    121: 132         24     25:  48   |
| 27       11    133: 143         24     25:  48   |
| 28       11    144: 154         24     25:  48   |
| 29       11    155: 165         24     25:  48   |
| 30       11    166: 176         24     25:  48   |
| 31       11    177: 187         24     25:  48   |
| 32       12      1:  12         24     49:  72   |
| 33       12     13:  24         24     49:  72   |
| 34       12     25:  36         24     49:  72   |
| 35       12     37:  48         24     49:  72   |
| 36       12     49:  60         24     49:  72   |
| 37       12     61:  72         24     49:  72   |
| 38       12     73:  84         24     49:  72   |
| 39       12     85:  96         24     49:  72   |
| 40       12     97: 108         24     49:  72   |
| 41       12    109: 120         24     49:  72   |
| 42       12    121: 132         24     49:  72   |
| 43       11    133: 143         24     49:  72   |
| 44       11    144: 154         24     49:  72   |
| 45       11    155: 165         24     49:  72   |
| 46       11    166: 176         24     49:  72   |
| 47       11    177: 187         24     49:  72   |
| 48       12      1:  12         23     73:  95   |
| 49       12     13:  24         23     73:  95   |
| 50       12     25:  36         23     73:  95   |
| 51       12     37:  48         23     73:  95   |
| 52       12     49:  60         23     73:  95   |
| 53       12     61:  72         23     73:  95   |
| 54       12     73:  84         23     73:  95   |
| 55       12     85:  96         23     73:  95   |
| 56       12     97: 108         23     73:  95   |
| 57       12    109: 120         23     73:  95   |
| 58       12    121: 132         23     73:  95   |
| 59       11    133: 143         23     73:  95   |
| 60       11    144: 154         23     73:  95   |
| 61       11    155: 165         23     73:  95   |
| 62       11    166: 176         23     73:  95   |
| 63       11    177: 187         23     73:  95   |
| 64       12      1:  12         23     96: 118   |
```

```
| 65       12     13:  24         23     96: 118   |
| 66       12     25:  36         23     96: 118   |
| 67       12     37:  48         23     96: 118   |
| 68       12     49:  60         23     96: 118   |
| 69       12     61:  72         23     96: 118   |
| 70       12     73:  84         23     96: 118   |
| 71       12     85:  96         23     96: 118   |
| 72       12     97: 108         23     96: 118   |
| 73       12    109: 120         23     96: 118   |
| 74       12    121: 132         23     96: 118   |
| 75       11    133: 143         23     96: 118   |
| 76       11    144: 154         23     96: 118   |
| 77       11    155: 165         23     96: 118   |
| 78       11    166: 176         23     96: 118   |
| 79       11    177: 187         23     96: 118   |
| 80       12      1:  12         23    119: 141   |
| 81       12     13:  24         23    119: 141   |
| 82       12     25:  36         23    119: 141   |
| 83       12     37:  48         23    119: 141   |
| 84       12     49:  60         23    119: 141   |
| 85       12     61:  72         23    119: 141   |
| 86       12     73:  84         23    119: 141   |
| 87       12     85:  96         23    119: 141   |
| 88       12     97: 108         23    119: 141   |
| 89       12    109: 120         23    119: 141   |
| 90       12    121: 132         23    119: 141   |
| 91       11    133: 143         23    119: 141   |
| 92       11    144: 154         23    119: 141   |
| 93       11    155: 165         23    119: 141   |
| 94       11    166: 176         23    119: 141   |
| 95       11    177: 187         23    119: 141   |
| 96       12      1:  12         23    142: 164   |
| 97       12     13:  24         23    142: 164   |
| 98       12     25:  36         23    142: 164   |
| 99       12     37:  48         23    142: 164   |
|100       12     49:  60         23    142: 164   |
|101       12     61:  72         23    142: 164   |
|102       12     73:  84         23    142: 164   |
|103       12     85:  96         23    142: 164   |
|104       12     97: 108         23    142: 164   |
|105       12    109: 120         23    142: 164   |
|106       12    121: 132         23    142: 164   |
|107       11    133: 143         23    142: 164   |
|108       11    144: 154         23    142: 164   |
|109       11    155: 165         23    142: 164   |
```

```
|110       11    166: 176         23    142: 164   |
|111       11    177: 187         23    142: 164   |
|112       12      1:  12         23    165: 187   |
|113       12     13:  24         23    165: 187   |
|114       12     25:  36         23    165: 187   |
|115       12     37:  48         23    165: 187   |
|116       12     49:  60         23    165: 187   |
|117       12     61:  72         23    165: 187   |
|118       12     73:  84         23    165: 187   |
|119       12     85:  96         23    165: 187   |
|120       12     97: 108         23    165: 187   |
|121       12    109: 120         23    165: 187   |
|122       12    121: 132         23    165: 187   |
|123       11    133: 143         23    165: 187   |
|124       11    144: 154         23    165: 187   |
|125       11    155: 165         23    165: 187   |
|126       11    166: 176         23    165: 187   |
|127       11    177: 187         23    165: 187   |
|_____________________________________________________|
```

With this output, users will be able to trace issues that occur on specific processors to geographic regions of the model domain.

Then, as the time-dependent portion of the model begins, output is provided for every timestep with the following form:

```
Processing Day/Time [YYYYDDD:HHMMSS]: 2015274:000000
  Which is Equivalent to (UTC): 0:00:00  Thursday,  Oct. 1, 2015
  Time-Step Length (HHMMSS): 000500
            VDIFF completed...    6.2 seconds
           COUPLE completed...    0.0 seconds
             HADV completed...    0.3 seconds
             ZADV completed...    0.0 seconds
            HDIFF completed...    0.1 seconds
         DECOUPLE completed...    0.0 seconds
             PHOT completed...    0.6 seconds
          CLDPROC completed...    0.0 seconds
             CHEM completed...    0.4 seconds
             AERO completed...    0.4 seconds
       Master Time Step
       Processing completed...    8.0 seconds
```

This section documents the date and time the model is currently processing along with the time spent calculating every major sub-process. At the end of each simulation hour, the calculation time is also printed for the output process.

```
    Processing Day/Time [YYYYDDD:HHMMSS]: 2015274:005500
       Which is Equivalent to (UTC): 0:55:00  Thursday,  Oct. 1, 2015
       Time-Step Length (HHMMSS): 000500
                 VDIFF completed...    0.9 seconds
                COUPLE completed...    0.0 seconds
                  HADV completed...    0.2 seconds
                  ZADV completed...    0.0 seconds
                 HDIFF completed...    0.0 seconds
               DECOUPLE completed...    0.0 seconds
                  PHOT completed...    0.2 seconds
                CLDPROC completed...    0.3 seconds
                  CHEM completed...    0.4 seconds
                  AERO completed...    1.0 seconds
            Master Time Step
            Processing completed...    3.0 seconds

      =--> Data Output completed...    0.3 seconds
```

This procedure repeats for every hour of the output day until completion of that day.

```
     ================================================
     |>---   PROGRAM COMPLETED SUCCESSFULLY   ---<|
     ================================================
     Date and time 0:00:00   Oct. 2, 2015   (2015275:000000)

     The elapsed time for this simulation was     733.0 seconds.

real 734.70
user 0.07
sys 0.17

CMAQ Processing of Day 20151001 Finished at Fri Apr  5 11:21:20 EDT 2019

\\\\\=====\\\\\=====\\\\\=====\\\\\=====/////=====/////=====/////=====/////
```

After the final day has been completed, summary information is printed for the computation time of every executed day.

```
=================================
  ***** CMAQ TIMING REPORT *****
=================================
Start Day: 2015-10-01
End Day:   2015-10-14
Number of Simulation Days: 14
```

```
Domain Name:               WRF_CMAQ_2WAY
Number of Grid Cells:      1538636  (ROW x COL x LAY)
Number of Layers:          44
Number of Processes:       128
   All times are in seconds.

Num  Day        Wall Time
01   2015-10-01   727.67
02   2015-10-02   717.89
03   2015-10-03   709.40
04   2015-10-04   701.84
05   2015-10-05   703.34
06   2015-10-06   708.96
07   2015-10-07   708.07
08   2015-10-08   707.25
09   2015-10-09   706.42
10   2015-10-10   703.56
11   2015-10-11   707.74
12   2015-10-12   705.44
13   2015-10-13   712.43
14   2015-10-14   718.59
     Total Time = 9938.60
      Avg. Time = 709.90
```

The processor-specific logfiles provide detailed information on the operation of hundreds of model tasks from mapping variables to opening and reading input files. Warnings that may be important for users to be aware of are printed to these files. To confirm that the model ran to completion view the run.[data].log file. For MPI runs, you may check any of the CTM_LOG_[ProcessorID]*.log files. A successful run will contain the following line at the bottom of the log(s):

```
>>----> Program completed successfully <----<<
```

Note: The log file for each processor is also moved from the $CMAQ_HOME/CCTM/scripts directory to the data output directory:

```
$CMAQ_DATA/output_CCTM_v53_[compiler]/[data_name]
```

5.7.2 CCTM Output files

The output results will have been placed in the directory:

```
$CMAQ_DATA/output_CCTM_v53_[compiler]_[data_name]
```

and can include the following netCDF-type files: ACONC, APMDIAG, B3GTS_S, CGRID, CONC, DEPV, DRYDEP, DUSTEMIS, LTNGDIAG1, LTNGDIAG2, MEDIA_CONC, PMDIAG, PT3D_DIAG, RJ_1, RJ_2, RJ_3, SOILOUT, SSEMIS, VDIFF, VSED, WETDEP1, WETDEP2 and VEXT_1. The in-depth description about each of these files is described in Chapter 7.

5.7.3 Common errors causing the CCTM simulation to crash

Common errors in a CCTM simulation include the following:

- Incorrect paths to input files. Look in the CCTM screen output (captured in your log file) for an Error message about an input file not being found.
- Incorrect MPI implementation. A series of MPI errors at the end of the log file often indicate that the MPI job was not submitted correctly.

Check the last few lines of the CCTM output log for messages to help diagnose why the simulation did not complete.

6. Model Configuration Options

6.1 Introduction

As discussed in Chapter 1, CMAQ is a multipollutant, multiscale air quality modeling system that estimates the transport and chemistry of ozone, PM, toxic airborne pollutants, and acidic and nutrient pollutant species, as well as visibility degradation and deposition totals. CMAQ includes state-of-the-art technical and computational techniques to simulate air quality from urban to global scales. It can model complex atmospheric processes affecting transformation, transport, and deposition of air pollutants using a system architecture that is designed for fast and efficient computing. (See Appendix D for an introduction on how data-parallelism can be applied in the CMAQ system to increase computational efficiency.) This chapter presents a brief overview of the conceptual formulation of Eulerian air quality modeling and the science features in various components of the Chemistry-Transport Model (CTM) component of CMAQ, CCTM.

6.2 Numerical Approach

The theoretical basis for CMAQ's formulation is the conservation of mass for atmospheric trace species emissions, transport, chemistry, and removal in the atmosphere. The general form of a chemical species equation derives from this conservation, so that changes in atmospheric concentrations of a species, Ci, can mathematically be represented as

$$\frac{\partial C_i}{\partial t} = \left(\frac{\partial C_i}{\partial t}\right)_{adv} + \left(\frac{\partial C_i}{\partial t}\right)_{diff} + \left(\frac{\partial C_i}{\partial t}\right)_{cloud} + \left(\frac{\partial C_i}{\partial t}\right)_{dry} + \left(\frac{\partial C_i}{\partial t}\right)_{aero} + R_{gi} + E_i$$

where the terms on the right-hand side of the equation represent the rate of change in Ci due to advection, diffusion, cloud processes (mixing, scavenging, and aqueous-phase chemistry), dry deposition, and aerosol processes (phase partitioning, and aerosol dynamics). Rgi represents the rate of change due to gas and heterogeneous chemical reactions, while Ei is the emission rate for that species. The mass conservation for trace species and the moment dynamic equations for the various modes of the particulate size distribution in CMAQ are further formulated in generalized coordinates, where in the same formulation allows the model to accommodate the commonly used horizontal map projections (i.e., Lambert conformal, polar stereographic, and Mercator) as well as different vertical coordinates (see Chapters 5 and 6 in Byun and Ching, 1999). The governing equation for CMAQ is numerically solved using the time-splitting or process splitting approach wherein each process equation is solved sequentially, typically with the process with the largest time-scale solved first.

6.3 Grid Configuration

CMAQ is a three-dimensional Eulerian air quality model. To solve the governing partial differential equations, the modeling domain (that is, the volume of the atmosphere over a geographic region of interest) is discretized with three-dimensional cells. The grid cells and lateral boundaries of the domain must be rigorously and consistently defined across the scientific components of the model, including chemistry, emissions, meteorology, and other peripheral scientific processors. In other words, all components of the CMAQ system must use the same map projections and horizontal grid spacing to maintain scientific consistency across the modeling domain. The number of grid cells in the west-east dimension is typically counted in “columns” or “NCOLS”, and the number of grid cells in the south-north dimension is typically counted in “rows” or “NROWS”. The vertical discretization is typically counted in “layers” or “NLAYS”.

CMAQ uses a generalized coordinate system to map the physical space to the computational space; see Chapter 6 of Byun and Ching (1999). The generalized coordinates enable CMAQ to maintain mass consistency under different horizontal map projections (such as Lambert conformal, polar stereographic, and Mercator) and under different vertical coordinate systems (such as terrain-following “sigma”, height, and hybrid sigma-pressure). CMAQv5.3 supports modeling domains comprised of rectilinear cells, where the length of each *side* of the cells in projected space is the same (such as $\Delta x = \Delta y = 12$ km). By contrast, the vertical grid is generally irregular, such that the modeling layers are thinnest near the ground. The absolute dimensions of the horizontal grid (that is, the west-east and south-north extents of the computational domain) can differ.

In general, the characteristics of the CMAQ modeling domain (including the map projection, horizontal grid spacing, vertical grid type, and maximum areal coverage) are inherited from the meteorological model. Beginning with CMAQv5.3 and MCIPv5.0, the public release of CMAQ is only configured for meteorological data from the Weather Research and Forecasting (WRF) model. However, MCIP (which translates and prepares meteorological model data for CMAQ) can be expanded to process data from other meteorological models to be used within the CCTM.

6.3.1 Horizontal Domains and Lateral Boundaries

After determining the horizontal and vertical extent of the domain of interest, the meteorological model must be run for a horizontal domain slightly larger than the CMAQ domain. A larger meteorology domain is required so the boundary conditions in the WRF simulation will fall outside the CMAQ simulation domain. Because there is a blend of larger-scale driving data and scale-specific physics within the WRF lateral boundaries, these data are inappropriate to use in the CCTM, so they are usually removed in MCIP. The lateral boundaries for WRF are typically a "picture frame" of the outermost 5 cells of the WRF domain. These lateral boundaries are used to blend the influence of larger-scale meteorological driving data with the WRF simulation. In WRF, the lateral boundaries are calculated and included as part of the modeling domain. By contrast, the lateral boundaries for the CCTM are external to the modeling domain.

MCIP can be used to extract a subset of the WRF modeling domain (that is, a "window") to be used for the CCTM modeling domain. The window can be any rectangular area within the meteorological model's lateral boundaries, provided it contains a nominally large enough areal coverage.

Horizontal grids specifications for CMAQ are contained in the grid definition file (GRIDDESC), which is output by MCIP and can be edited by the user. Further details on grid configuration are available in the README.md file in the PREP/mcip folder. If several domains have been used within a group, the horizontal domain for a given CMAQ run can be defined at runtime by setting the GRIDDESC and GRID_NAME environment variables to point to an existing grid definition file and to one of the grids defined in the file, respectively.

6.3.2 Vertical Domains

CMAQ can support multiple vertical coordinate systems via the generalized coordinate. Most of the grid transformation to maintain mass consistency in CMAQ occurs through the mathematical term, Jacobian; see Chapter 6 of Byun and Ching (1999) and Otte and Pleim (2010). In the CMAQ system, the Jacobian is calculated in MCIP. The vertical processes in the CCTM (such as mixing within the planetary boundary layer and convective mixing) must also be cast in a flexible coordinate system.

There are two options for vertical coordinates in the WRF model: terrain-following ("sigma"), and hybrid sigma-pressure. In both vertical coordinate systems, there is a "model top" employed (often called PTOP, or pressure at the top of the model) to limit the vertical extent of the modeling domain. The model top is usually set within the lower stratosphere (for example, 50 hPa), but can be higher for some modeling applications. The sigma coordinate system allows the influence of the terrain to gradually diminish with height toward the model top. The sigma coordinate (technically called "eta" in the WRF system) has been used since WRF was initially released to the public in the late 1990s. The hybrid sigma-pressure coordinate was introduced in WRFv3.9 (released in 2017), and it uses a terrain-following coordinate in the lower part of the atmosphere which transitions to a constant pressure coordinate in the upper part of the atmosphere. The hybrid sigma-pressure coordinate is often used to reduce the presence of gravity waves in the model in steep and complex terrain, and to enable a higher model top to be used.

Beginning with CMAQv5.3 and MCIPv5.0, both the sigma and the hybrid sigma-pressure coordinates are supported. MCIPv5.0 was modified to calculate the Jacobian from the hybrid coordinate, and

CMAQv5.3 has some scientific processes recast more generically so that both the sigma coordinate and the hybrid coordinate can be properly represented. CMAQ prior to v5.3 (and MCIP prior to v5.0) is not compatible with the hybrid coordinate system introduced in WRF 3.9. If the hybrid coordinate is used in WRF (versions 3.9 or later), MCIPv5.0 must be used with CMAQv5.3. See Appendix E for notes on configuring WRF4.0 and later for use with CMAQv5.3.

6.4 Science Configurations

CCTM contains several science configurations for simulating transport, chemistry, and deposition. All the science configuration options in CCTM, such as the chemical mechanism to be used, are set when building the executable. The model grid and vertical layer structure for CCTM are set at execution. The important distinction between selecting the science configuration and the model grid/layer configuration is that CCTM does not need to be recompiled when changing model grids/layers but does need to be recompiled when new science options are invoked. The following sections describe how these science options can be utilized by configuring using the `bldit_cctm.csh` and `run_cctm.csh` scripts. For the remainder of this chapter these files will be referred to as simply BuildScript and RunScript.

6.5 Advection

In CCTM, the 3-dimensional transport by mean winds (or advection) is numerically represented by sequentially solving locally-one dimensional equations for the two horizontal and vertical components. CMAQ uses the piecewise parabolic method (PPM) (Colella and Woodward, 1984) for representing tracer advection in each of the three directions. This algorithm is based on the finite-volume sub-grid definition of the advected scalar. In PPM, the sub-grid distribution is described by a parabola in each grid interval. PPM is a monotonic and positive-definite scheme. Positive-definite schemes maintain the sign of input values, which in this case means that positive concentrations will remain positive and cannot become negative.

Mass consistency is a key desired attribute in tracer advection. Data consistency is maintained for air quality simulations by using dynamically and thermodynamically consistent meteorology data from WRF/MCIP. Mass inconsistencies can nevertheless arise either using different grid configurations (horizontal or vertical) or due to differing numerical advection schemes between the driving meteorological model and the CCTM. While inconsistencies due to the former can be eliminated through use of the same grid configurations (thus, layer collapsing is not recommended), some inconsistencies can still remain due to differing numerical representations for satisfying the mass-continuity equation between the driving meteorological model and the CCTM. These mass-inconsistencies manifest as first order terms (whose magnitude can often be comparable to tracer lifetimes if continuity is not satisfied with high accuracy) that can artificially produce or destroy mass during 3D tracer advection (e.g., Mathur and Peters, 1990).

CMAQ has two options that minimize mass consistency errors in tracer advection. In one scheme (designated “local_cons” in the BuildScript), first implemented in CMAQv4.5 and later improved for CMAQv4.7.1, CMAQ advects air density and re-diagnoses the vertical velocity field according to the layer-by-layer mass continuity equation which guarantees that the CCTM advected density matches

that derived from the driving meteorological inputs (e.g., Odman and Russell, 2000). Briefly, x- and y-advection are first performed (the order of these is reversed every step to minimize aliasing errors) to yield intermediate tracer and density fields. The intermediate density field is then subject to vertical advection with the PPM scheme such that it yields the WRF-derived density field at the end of the advection time-step. This scheme results in an estimated vertical velocity field that is minimally adjusted relative to the WRF derived field in the lower model layers but yields strict mass-consistent tracer advection in CMAQ. A drawback to this approach is that erroneous noise in the diagnosed vertical velocity field accumulates toward the top of the model with non-zero velocity and mass flux across the top boundary. The noise in the vertical velocity field causes excessive diffusion in upper layers. Therefore, since CMAQv5.0, a scheme designated "wrf_cons", that closely follows the vertical velocity calculation in WRF has been available. This scheme solves the vertically integrated mass continuity equation such that the column integrated horizontal mass divergence is balanced by the net change in column mass (Skamarock et al, 2019). An advantage of this scheme is that the diagnosed vertical velocity agrees more closely with the WRF vertical velocity field with zero velocity and mass flux across the upper model boundary. Thus, the spurious velocity noise and excessive diffusion in the upper layer are eliminated. The main drawback of this scheme is that mass conservation is not guaranteed so density must be updated from the meteorology inputs every timestep.

The **"WRF_CONS"** option is the recommended configuration for CMAQv5.3.

To invoke the "WRF_CONS" option in 3-D advection, set the following in the BuildScript within the CCTM Science Modules section:

```
set ModAdv = wrf_cons
```

To invoke the "LOCAL_CONS" option in 3-D advection, set the following in the BuildScript within the CCTM Science Modules section:

```
set ModAdv = local_cons
```

Note: The local_cons option is a legacy extension and can cause unexpected results when used.

6.6 Horizontal Diffusion

The lack of adequate turbulence measurements has limited the development of robust model parameterizations for horizontal turbulent diffusion, a scale and resolution dependent problem. With the advent of very accurate minimally diffusive numerical advection schemes and need for high resolution modeling, horizontal diffusion algorithms are needed to balance the numerical diffusion inherent in advection schemes relative to the physical horizontal diffusion in the atmosphere. Currently in CMAQ, horizontal diffusion fluxes for transported pollutants are parameterized using eddy diffusion theory. The horizontal diffusivity coefficients are in turn formulated using the approach of Smagorinsky (1963) which accounts for local horizontal wind deformation and are also scaled to the horizontal grid size.

6.7 Vertical Diffusion

The vertical diffusion model in CMAQ is the Asymmetrical Convective Model Version 2 (ACM2) (Pleim 2007a,b). The ACM2 is a combined local and non-local closure PBL scheme that is implemented in CMAQ and WRF for consistent PBL transport of meteorology and chemistry. Thus, it is recommended that the ACM2 option in WRF or MPAS also be used when preparing meteorology for CMAQ.

There are two options for the ACM2 model in the BuildScript that are compatible with either the M3Dry or STAGE dry deposition options.

When running m3dry dry deposition:

```
Set ModVdiff   = acm2_m3dry
```

When running STAGE dry deposition:

```
Set ModVdiff   = acm2_stage
```

6.8 Dry Deposition/Air-surface exchange

Exchange of pollutants between the atmosphere and Earth's surface can be modeled as unidirectional exchange, commonly referred to as dry deposition, or bidirectional exchange where the direction of the flux depends on the relative concentration of the pollutant in the atmosphere and the surface (e.g. soil, plant stomata). If the concentration in the atmosphere is greater than the concentration at the surface, then deposition occurs. If the concentration in the atmosphere is lower than the concentration at the surface, emission occurs. CMAQ contains algorithms for modeling either of these situations. The rate of exchange is controlled by surface characteristics such as vegetation type, leaf area index, and surface roughness as well as meteorological influences such as temperature, radiation, and surface wetness which are provided to CMAQ from the land surface model (LSM) in the driving meteorological model.

Currently, most chemicals in CMAQ are modeled as depositing only. However, ammonia and mercury can be both emitted from the surface and deposited and are therefore modeled as bidirectional. Estimates of the soil and stomatal compensation concentrations needed to compute the bidirectional ammonia flux in CMAQ are derived from input provided by the Environmental Policy Integrated Climate (EPIC) agricultural ecosystem model that is executed using the Fertilizer Emission Scenario Tool for CMAQ (FEST-C, https://www.cmascenter.org/fest-c) (Ran et al., 2011; Cooter et al., 2012). Information for surface concentrations of mercury are initially specified using land use specific tabular data and then by modeling the accumulation, transformation and evasion of mercury in the surface media (Bash 2010).

CMAQ v5.3 contains two options for calculating dry deposition/surface exchange which are invoked in the BuildScript as:

```
Set DepMod   = m3dry
```

or:

```
Set DepMod   = stage
```

Deetails of each module are provided in the sections below.

6.8.1 Dry Deposition - m3dry

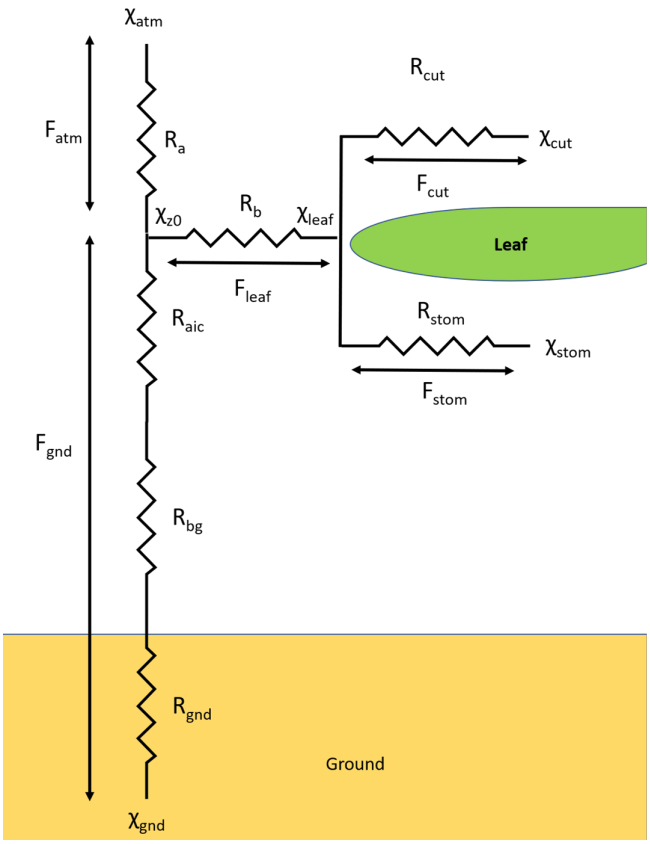
The m3dry option for dry deposition and ammonia bidirectional surface flux in CMAQv5.3 is the next evolution of the dry deposition model that has been in CMAQ since its initial release and was originally based on the dry deposition model developed for the Acid Deposition and Oxidant Model (ADOM) (Pleim et al., 1984). Dry deposition is computed by electrical resistance analogy where concentration gradients are analogous to voltage, flux is analogous to current, and deposition resistance is analogous to electrical resistance (Pleim and Ran, 2011). In m3dry, several key resistances, such as aerodynamic resistance and bulk stomatal resistance, and other related parameters, such as LAI, vegetation fraction, roughness length, friction velocity etc., are expected to be provided from the meteorological inputs. Use of common model elements and parameters with the land surface model in the meteorology model ensures consistency between chemical surface fluxes and meteorological surface fluxes (moisture, heat, momentum). While the m3dry dry deposition model was designed to be used with the PX LSM option in WRF, any LSM can be used if the necessary parameters are output and then provided for input into CMAQ. It features consideration of subgrid land-use fractions through aggregation of key model parameters, such as LAI, veg fraction, roughness length and minimum stomatal conductance, to the grid cell level.

Upgrades for version 5.3 include larger surface resistances for deposition to snow and ice and reduced resistance for deposition to bare ground for ozone with dependence on surface soil moisture content. The aerosol deposition has also been revised including a new dependence on LAI. The ammonia bidirectional surface flux from croplands has been substantially revised from earlier versions. The new version has close linkages with the EPIC agricultural ecosystem model. Daily values of all soil parameters needed to compute the available soil ammonia concentrations (soil ammonia content, soil moisture, soil texture parameters, soil pH, and Cation Exchange Capacity (CEC)) for each of 21 agricultural production types that are either rainfed or irrigated (42 types total) are input to CMAQ. Soil ammonia concentrations and soil pH are combined to derive the soil compensation concentration for the bidirectional flux calculation (Pleim et al., 2019).

6.8.2 Dry Depostion - STAGE

In CMAQ v5.3., a new tiled, land use specific, dry deposition scheme, the Surface Tiled Aerosol and Gaseous Exchange (STAGE), option has been developed to better estimate atmospheric deposition for terrestrial and aquatic ecosystem health and applications to evaluate the impact of dry deposition on ambient air quality. This new scheme explicitly supports Weather Research and Forecasting (WRF) simulations with a variety of land surface schemes (Noah, Pleim-Xiu, etc). The model resistance framework, Figure 6-1, parameterizes air-surface exchange as a gradient process and is used for both bidirectional exchange and dry deposition following the widely used resistance model of Nemitz et al. (2001).

Grid scale fluxes are estimated from sub-grid cell land use specific fluxes and are area weighted to the grid cell totals which are then output in the standard dry deposition file with positive values indicating deposition and negative values indicating evasion. The model resistances are largely estimated following Massad et al. (2010) with the following exceptions. Deposition to wetted surfaces considers the bulk accommodation coefficient, following Fahey et al. (2017), and can be a limiting factor for highly soluble compounds. The in-canopy resistance is derived using the canopy momentum attenuation parameterization from Yi (2008). Aerosol dry deposition includes parameterizations for deposition to water or bare ground surfaces (Giorgi 1986), and vegetated surfaces (Slinn 1982), using the characteristic leaf radius parameterization of Zhang et al. (2001). The ammonia bidirectional option follows the ammonia specific parameterizations of Massad et al. (2010). Mercury bidirectional exchange is also available and follows the parameterization of Bash (2010). In this modeling framework, it is possible to consider bidirectional exchange for any species by providing a parametrization or constant that sets the stomatal, cuticular, soil and/or water compensation point as a value greater than 0.



VARIABLE	DESCRIPTION
R_A	Aerodynamic Resistance
R_B	Leaf Quasi Boundary Layer Resistance
R_{CUT}	Cuticular Resistance
R_{STOM}	Stomatal Resistance
R_{AIC}	In Canopy Resistance
R_{BG}	Soil Quazi Boundary Layer Resistance
R_{GND}	Soil Resistance
X_{ATM}	Atmospheric Concentration
X_{Z0}	Canopy Compensation Point
X_{CUT}	Cuticular Compensation Point
X_{STOM}	Stomatal Compensation Point
X_{GND}	Soil Compensation Point
X_{LEAF}	Leaf Compensation Point
F_{ATM}	Atmosphere-surface flux
F_{LEAF}	Flux between χ_{z0} and χ_{leaf}
F_{CUT}	Flux between χ_{z0} and χ_{leaf}
F_{STOM}	Flux between χ_{z0} and χ_{stom}
F_{GND}	Flux between χ_{z0} and χ_{gnd}

Figure 6-1. STAGE resistance diagram (modified from Nemitz et al., 2001) with a table of variables descriptions.

STAGE options in the RunScript:

```
setenv CTM_MOSAIC Y
```

Sets output for land use specific dry deposition and dry deposition velocities. Note: To retrieve the grid cell average from these files it should be area weighted by the land use fraction by summing the product of the land use fraction and the dry deposition/deposition velocity for each grid cell.

```
setenv CTM_FST Y
```

Sets output for land use specific dry deposition to leaf stomata.

```
setenv PX_VERSION   Y
setenv CLM_VERSION Y
setenv NOAH_VERSION Y
```

Sets the correct soil hydrological properties and soil layer information needed to calculate soil NO emissions, NH3 bidirectional exchange and O3 deposition. These options are currently based on WRF 3.8.1 and earlier values for PX and CLM and WRF 4.0 for NOAH. If the land surface model is run with another look up table or parameterization, soil moisture will be constrained between saturation and residual water content from the parameterization in CMAQ. This is also the case for the m3dry deposition option, soil NO emissions, and wind blown dust.

6.9 Emissions

CMAQ introduces emissions of trace gases and aerosols from a variety of important sources (e.g. electric generating utilities, vehicles, fires, trees, dust storms, farms, etc.). Some emissions are applied in the surface layer of the model grid, while others are applied at higher altitudes if, for example, they originate from point source like an elevated stack, or a large forest fire. Many sources that are related to local meteorology may be calculated online in CMAQ. However, most sources, especially anthropogenic ones, are preprocessed using software like the Sparse Matrix Operator Kerner Emissions (SMOKE) Modeling System. Once these external tools have calculated the offline emissions, they may merge them into larger aggregated files. We refer to emissions that are either calculated online or read into CMAQ from a file as emission "streams".

Because CMAQ represents both primary and secondary pollutants, emissions are processed for a subset of the species CMAQ treats. The emissions chemical speciation must be compatible with the chemical mechanism chosen for CMAQ (e.g. cb6r3_ae7_aq) because different mechanisms represent large compounds like functionalized hydrocarbons with different surrogates. CMAQv5.3 has introduced new features that make the process of mapping emissions species to CMAQ species more transparent and flexible (see Appendix B: Emission Control with DESID). In fact, users can now toggle, modify, and augment emissions from all available streams in order to better tailor their simulations to the questions they are asking CMAQ to help answer. For tutorials covering specific tasks, please see the DESID tutorial page.

6.9.1 Emission Streams

Depending on the nature of any stream and the information used to quantify its emissions, it may be treated as one of three types:

Online Stream:

CMAQ will calculate the emission rates from this source using information about local meteorology, land characteristics, etc. The streams that can be run online in CMAQ are: biogenics (BEIS),wind-blown dust, sea spray, and lightning NO.

Gridded Stream (offline):

CMAQ will read emission rates from an input file, which is organized into an array that is identical in shape to the CMAQ model grid. Typically, these rates are stored at hourly time points and are then interpolated within CMAQ to each time step. These files may be 2D to represent just the surface layer emissions or they may be 3D. If 3D, the file may have the same number or fewer number of layers as the CMAQ grid. Some common examples of Gridded emissions include:

- Mobile sources such as passenger vehicles, trains, ships, scooters, etc.
- Low-level point source emissions that are not large enough to be treated individually
- Residential heating
- Consumer product use (e.g. adhesives, personal care products, pesticides, etc.)
- Agricultural (e.g. burning, dust, animal waste, etc.)
- Road, Construction and mechanically generated dust
- Biogenic VOCs (if not calculated online with BEIS)

Users add Gridded emissions to a simulation via the RunScript. First the variable N_EMIS_GR must be set to the number of Gridded Streams to be used:

```
setenv N_EMIS_GR 3
```

The RunScript must also specify the location of the input files using three-digit suffixes for the stream number:

```
setenv GR_EMIS_001 /home/user/path-to-file/emiss_stream_1_${DATE}.nc
```

and the short-name label to be used to refer to the Stream in logfiles:

```
setenv GR_EMIS_LAB_001 MOBILE
```

If N_EMIS_GR is set 0, then CMAQ will run with no Gridded emissions even if the values for GR_EMIS_XXX and GR_EMIS_LAB_XXX are all set.

Inline Stream (offline):

For these streams, emission rates and stack characteristics are provided for many individual sources on the same file. CMAQ uses the stack information to calculate important quantities online like the injection height which is influenced by local meteorology. A specific latitude/longitude pair is given for each source to locate it in the CMAQ grid. Sources outside the CMAQ grid domain are ignored by CMAQ; thus the same files may be used for a large domain and a nest within that domain. Some common examples of Inline emissions include:

- Stacks (electric generation units, industrial sources, manufacturing, etc.)
- Forest fires
- Large prescribed fire events

Users add Inline emissions to a simulation via the RunScript. First the variable N_EMIS_PT must be set to the number of Inline Streams to be used:

```
setenv N_EMIS_PT 3
```

The RunScript must also specify the location of the input files using three-digit suffixes for the stream number:

```
setenv STK_EMIS_002 /home/user/path-to-file/inline_emiss_stream_2_${DATE}.nc
```

The location to the "stack file" with static information about the properties of each source on the stream:

```
setenv STK_GRPS_002 /home/user/path-to-file/inline_stack_groups_2.nc
```

and the short-name label to be used to refer to the Stream in logfiles:

```
setenv STK_EMIS_LAB_002 POINT_FIRES
```

If N_EMIS_PT is set 0, then CMAQ will run with no Inline emissions even if the values for STK_EMIS_XXX, STK_GRPS_XXX and STK_EMIS_LAB_XXX are all set.

Plume Rise - Plume rise can be calculated inline within CMAQ using the Briggs solution as it is implemented in SMOKE and documented in the SMOKE user guide (https://www.cmascenter.org/smoke/documentation/4.6/
It is required that emission files have been processed to include the necessary stack parameters (e.g. exit velocity, diameter, stack gas temperature, stack height, etc.).

6.9.2 Online Emission Streams

Biogenics To calculate online biogenic emissions, CMAQ uses the Biogenic Emission Inventory System (BEIS). BEIS calculates emissions resulting from biological activity from land-based vegetative species as well as nitric oxide emissions produced by microbial activity from certain soil types.

This biogenic model is based on the same model that is included in SMOKE. Before using the CMAQ online version of BEIS users should confirm that biogenic emissions are not already included in their emissions files from SMOKE to avoid double counting biogenic emissions. User documentation for BEIS can be found in Chapter 6.17 of the SMOKE manual.

Speciation of biogenic emissions is controlled by gspro_biogenics.txt under CCTM/src/biog/beis.

Running CMAQ with online biogenics is controlled by the following RunScript flag:

```
setenv CTM_BIOGEMIS Y
```

Running CMAQ with online biogenic emissions requires a user-supplied, gridded normalized biogenic emissions input netCDF file, B3GRD. This file is created with the normbeis3 program in SMOKE prior to running the inline biogenic option in CMAQ and contains winter and summer normalized emissions and Leaf Area Indices. The location of the B3GRD file is set in the RunScript:

```
setenv B3GRD /home/user/path-to-file/b3grd.nc
```

For short simulations that span only summer months set the SUMMER_YN flag to Y and the BIOSW_YN flat to N in the RunScript so that biogenic emissions will be calculated using summer factors throughout the entire domain.

```
setenv BIOSW_YN N
```

```
setenv SUMMER_YN Y
```

For simulations that span only winter months, set the SUMMER_YN flag to N so that biogenic emissions will be calculated using winter factors throughout the entire domain.

For simulations of spring or fall, or simulations covering multiple seasons, a user must set the BIOSW_YN to Y and provide a BIOSEASON file to enable an appropriate mixture of winter and summer emission values across the domain and simulation period. The BIOSEASON file is created with the metscan program in SMOKE using the MCIP data for the modeling domain prior to running the inline biogenic option in CMAQ. It provides daily gridded values of an indicator variable derived from MCIP temperature fields to determine whether winter or summer biogenic emission values should be used for a given grid cell and day. To use the BIOSEASON file set the following two environment variables in the RunScript:

```
setenv BIOSW_YN Y
```

```
setenv BIOSEASON /home/user/path-to-file/bioseason.nc
```

Additionally, when using the inline biogenic option, the user must point to the SOILOUT file from one day's simulation as the SOILINP file for the next day. The user must also decide whether to write over SOILOUT files from previous days or create a uniquely named SOILOUT file for each day. The latter approach is recommended if the user wishes to retain the capability to restart simulations in the middle of a sequence of simulations.

The INITIAL_RUN variable in the RunScript to Y if this is the first time that biogenic NO soil emissions will be calculated. If there is a previously created file, set to N. When INITIAL_RUN is set to N, the directory path and file name of biogenic NO soil emissions file must be set in the RunScript:

```
setenv INITIAL_RUN N
```

```
setenv SOILNP /home/user/path-to-file/cctm_soilout.nc
```

Wind-Blown Dust The actual amount of dust emitted from an arid surface depends on wind speed, surface roughness, moisture content of the soil, vegetation coverage, soil type and texture, and air density. The main mechanism behind strong dust storms is called "saltation bombardment" or "sandblasting." The physics of saltation include the movement of sand particles due to wind, the impact of these particles to the surface that removes part of the soil volume, and the release of smaller dust particles. CMAQ first calculates friction velocity at the surface of the Earth. Once this friction velocity exceeds a threshold value, saltation, or horizontal movement, flux is obtained. Finally, the vertical flux of the dust is calculated based on a sandblasting efficiency formulation – a vertical-to-horizontal dust flux ratio.

CMAQ uses time-varying vegetation coverage, soil moisture and wind speed from the meteorological model, WRF. The vegetation coverage in WRF can vary depending on the configuration. In WRFv4.1+, the Pleim-Xiu land-surface model (PX LSM) was modified to provide CMAQ vegetation fraction (VEGF_PX in WRF renamed VEG in MCIP) from either the old fractional landuse weighting table lookup method (pxlsm_modis_veg = 0), or a new option where vegetation fraction is directly read from the monthly MODIS derived vegetation coverage (pxlsm_modis_veg = 1) found in the wrflowinp_d0* file(s). This was done because in recent years WRF has provided high resolution ~1 km monthly vegetation coverage that is more accurate than tables. Updates are backward compatible with older version of MCIP or WRF as long as VEG and VEGF_PX/VEGFRA are in those files. If users employ a different land surface model like the NOAH LSM, MCIP will assign the values of VEGFRA in WRF to VEG for CMAQ and the dust module will operate the same. Using the MODIS data in WRF via the new PX vegetation option provides the dust model a more accurate representation of vegetation in regions where windblown dust most occurs.

The CMAQ windblown dust module optionally requires utilizes additional land use information beyond the land use information contained in the MCIP files. This optional additional land use information is generally available for North American domains only and is provided by specifying either one (E2C_LU) or two (DUST_LU_1 and DUST_LU_2) additional input data files. See Chapter 4 for more information on these optional model input files. If these optional additional input files are not available (e.g. for a hemispheric modeling domain), the windblown dust module can function with only the land

use information contained in the MCIP files. See Appendix A on further information on how to specify the land use information for the windblown dust module.

The CMAQ windblown dust module is controlled by the following RunScript flag:

```
setenv CTM_WB_DUST Y
```

Note that if this flag is set to N to indicate zero wind-blown dust emissions, users should set the CTM_EMISCHK variable in the RunScript to FALSE to avoid crashing CMAQ when it cannot find species it is looking for from dust emissions.

Alternatively, users can also edit the emission control file by commenting out the coarse and fine species expected for the wind-blown dust module. The following species are emitted by the Dust module and may be referenced in the emission control file Table 6-1:

Table 6-1. Aerosol Species Predicted by the Wind-Blown Dust Module

Dust Surrogate Name	Default CMAQ Species	Description
PMFINE_SO4	ASO4	Fine-mode Sulfate
PMCOARSE_SO4	ASO4	Coarse-mode Sulfate
PMFINE_NO3	ANO3	Fine-mode Nitrate
PMCOARSE_NO3	ANO3	Coarse-mode Nitrate
PMFINE_CL	ACL	Fine-mode Chlorine
PMCOARSE_CL	ACL	Coarse-mode Chlorine
PMFINE_NH4	ANH4	Fine-mode Ammonium
PMFINE_NA	ANA	Fine-mode Sodium
PMFINE_CA	ACA	Fine-mode Calcium
PMFINE_MG	AMG	Fine-mode Magnesium
PMFINE_K	AK	Fine-mode Potassium
PMFINE_POC	APOC	Fine-mode Organic Carbon
PMFINE_PNCOM	APNCOM	Fine-mode Non-Carbon Organic Matter
PMFINE_LVPO1	ALVPO1	Fine-mode Low-Volatility hydrocarbon-like OA
PMFINE_LVOO1	ALVOO1	Fine-mode Low-Volatility Oxygenated OA
PMFINE_EC	AEC	Fine-mode Black or Elemental Carbon
PMFINE_FE	AFE	Fine-mode Iron
PMFINE_AL	AAL	Fine-mode Aluminum
PMFINE_SI	ASI	Fine-mode Silicon
PMFINE_TI	ATI	Fine-mode Titanium
PMFINE_MN	AMN	Fine-mode Manganese
PMFINE_H2O	AH2O	Fine-mode Water
PMCOARSE_H2O	AH2O	Coarse-mode Water
PMFINE_OTHR	AOTHR	Fine-mode Other
PMCOARSE_SOIL	ASOIL	Coarse-mode Non-Anion Dust
PMFINE_MN_HAPS	AMN_HAPS	Fine-mode Air toxics Manganese
PMCOARSE_MN_HAPS	AMN_HAPS	Coarse-mode Air toxics Manganese
PMFINE_NI	ANI	Fine-mode Nickel

Dust Surrogate Name	Default CMAQ Species	Description
PMCOARSE_NI	ANI	Coarse-mode Nickel
PMFINE_CR_III	ACR_III	Fine-mode Trivalent Chromium
PMCOARSE_CR_III	ACR_III	Coarse-mode Trivalent Chromium
PMFINE_AS	AAS	Fine-mode Arsenic
PMCOARSE_AS	AAS	Coarse-mode Arsenic
PMFINE_PB	APB	Fine-mode Lead
PMCOARSE_PB	APB	Coarse-mode Lead
PMFINE_CD	ACD	Fine-mode Cadmium
PMCOARSE_CD	ACD	Coarse-mode Cadmium
PMFINE_PHG	APHG	Fine-mode Mercury
PMCOARSE_PHG	APHG	Coarse-mode Mercury

Sea Spray Because sea spray particles are emitted during wave breaking and bubble bursting at the ocean surface, the main factor affecting the emission rate is the wind speed. The temperature of the ocean also affects bubble bursting and subsequent emission rate of sea spray particles. Wave breaking is enhanced near the surf zone just offshore, and CMAQ accounts for this by increasing sea spray particle emission rates in the surf zone.

The current open ocean sea spray particle emission rate in CMAQ, as described in Gantt et al. (2015), is based on Gong (2003) with a temperature dependence derived from Jaeglé et al. (2011) and Ovadnevaite et al. (2014) and an adjustment of Θ from 30 to eight to account for higher accumulation mode emissions. The current surf zone sea spray particle emission rate in CMAQ as described in Gantt et al. (2015) is based on Kelly et al. (2010) with a reduction of the assumed surf zone width from 50 to 25 meters. The CMAQ sea spray emissions module is controlled by the following RunScript flag:

```
setenv CTM_OCEAN_CHEM Y
```

Speciation of sea spray emissions is controlled by AERO_DATA.F under CCTM/src/aero. Note that CMAQ employing Carbon Bond 6 version r3 with DMS and marine halogen chemistry (cb6r3m_ae7_kmtbr) slightly modifies the speciation of Sea Spray emissions by including bromide from Sea Spray emissions.

Note that if the CTM_OCEAN_CHEM flag is set to N to indicate zero sea spray emissions, users should set the CTM_EMISCHK variable in the RunScript to FALSE to avoid crashing CMAQ when it cannot find species it is looking for from sea spray.

Alternatively, users can also edit the emission control file by commenting out the coarse and fine species expected for the sea spray module. The following species are emitted by the Dust module and may be referenced in the emission control file Table 6-2:

Table 6-2. Aerosol Species Predicted by the Sea-Spray Aerosol Module

Sea Spray Surrogate Name	Default CMAQ Species	Description
PMFINE_SO4	ASO4	Fine-mode Sulfate

Sea Spray Surrogate Name	Default CMAQ Species	Description
PMCOARSE_SO4	ASO4	Coarse-mode Sulfate
PMFINE_CL	ACL	Fine-mode Chlorine
PMCOARSE_CL	ACL	Coarse-mode Chlorine
PMFINE_NA	ANA	Fine-mode Sodium
PMFINE_CA	ACA	Fine-mode Calcium
PMFINE_MG	AMG	Fine-mode Magnesium
PMFINE_K	AK	Fine-mode Potassium
PMCOARSE_SEACAT	ASEACAT	Coarse-mode Sea Spray Cations
PMFINE_CR_VI	ACR_VI	Fine-mode Hexavalent Chromium
PMFINE_NI	ANI	Fine-mode Nickel
PMCOARSE_NI	ANI	Coarse-mode Nickel
PMFINE_AS	AAS	Fine-mode Arsenic
PMCOARSE_AS	AAS	Coarse-mode Arsenic
PMFINE_BE	ABE	Fine-mode Beryllium
PMCOARSE_BE	ABE	Coarse-mode Beryllium
PMFINE_PHG	APHG	Fine-mode Mercury
PMCOARSE_PHG	APHG	Coarse-mode Mercury
PMFINE_PB	APB	Fine-mode Lead
PMCOARSE_PB	APB	Coarse-mode Lead
PMFINE_CD	ACD	Fine-mode Cadmium
PMCOARSE_CD	ACD	Coarse-mode Cadmium
PMFINE_MN_HAPS	AMN_HAPS	Fine-mode Manganese (air toxic)
PMCOARSE_MN_HAPS	AMN_HAPS	Coarse-mode Manganese (air toxic)
PMFINE_BR	ABR	Fine-mode Bromine
PMCOARSE_BR	ABR	Coarse-mode Bromine
PMFINE_H2O	AH2O	Fine-mode Water
PMCOARSE_H2O	AH2O	Coarse-mode Water

Lightning NO In retrospective applications over the continental U.S., National Lightning Detection Network (NLDN) lightning data can be used directly to generate NO produced by lightning in CMAQ. For real-time forecasts or other applications where lightning data are not available, lightning NO is produced based on statistical relationships with the simulated convective rainfall rate (Kang et al., 2019).

There are three options for including NO from lighting. All three options require setting the CTM_LTNG_NO flag to Y in the RunScript.

```
setenv CTM_LTNG_NO Y
```

Option 1 - Offline NO – user provides a gridded lightning NO emissions file calculated with a preprocessor external to the CMAQ repository

For this option set the LTNGNO environment variable in the RunScript to the location of the gridded netCDF file of NO emissions:

```
setenv LTNGNO /home/user/path-to-file/ltngno_emiss_from_user.nc
```

Option 2 - Inline NO with NLDN Data – user uses hourly NLDN lightning strike netCDF file.

Hourly NLDN lightning strike data can be purchased. In addition to the hourly lightning strike netCDF file, this option requires a lightning parameters netCDF file. This file contains the intercloud to cloud-to-ground flash ratios, which are the scaling factors for calculating flashes using the convective precipitation rate, land-ocean masks, and the moles of NO per flash (cloud-to-ground and intercloud). The lightning parameters file for a domain over the continental US at 12km horizontal resolution (12US1) can be downloaded from the CMAS Data Warehouse. This file can be regridded to support other domains within the continental US.

For this option, set the following environment variables in the RunScript:

```
setenv LTNGNO INLINE

setenv USE_NLDN Y

setenv NLDN_STRIKES /home/user/path-to-file/nldn_hourly_ltng_strikes.nc

setenv LTNGPARMS_FILE /home/user/path-to-file/LTNG_AllParms_12US1.nc
```

Option 3 - Inline NO without NLDN Data – lightning NO is calculated within CCTM based on statistical relationships with the simulated convective rainfall rate.

This option also requires a lightning parameters netCDF file which contains the linear regression parameters for generating lightning NO. The lightning parameters file for the continental US at 12km horizontal resolution can be downloaded from the CMAS Data Warehouse. This file can be regridded to support other domains within the continental US.

For this option, set the following environment variables in the RunScript:

```
setenv LTNGNO INLINE

setenv USE_NLDN N

setenv LTNGPARMS_FILE /home/user/path-to-file/LTNG_AllParms_12US1.nc
```

6.10 Gas Phase Chemistry

6.10.1 Gas Phase Chemical Mechanisms

The CMAQ modeling system accounts for chemistry in three phases: a gas, particulate (solid or liquid), and aqueous-cloud phase. Refer to the release notes to find the gas□phase chemistry mechanisms available in each version of CMAQ. Several variations of the base gas-phase mechanisms, with and without chlorine, mercury, and toxic species chemistry, are distributed with CMAQ. The modularity of CMAQ makes it possible to create or modify the gas-phase chemical mechanism.

Gas-phase chemical mechanisms are defined in CMAQ based on Fortran source files. Located in subdirectories of the CCTM/src/MECHS directory (each corresponding to a mechanism name), these files define the source, reaction parameters, and atmospheric processes (e.g., diffusion, deposition, advection) of the various mechanism species. The species definitions for each mechanism are contained in namelist files that are read in during execution of the CMAQ programs. The CMAQ mechanism configuration is more similar to the science module configuration than to the horizontal grid or vertical layer configuration in that the mechanism is defined at build time, resulting in executables that are hard-wired to a specific gas-phase mechanism. To change chemical mechanisms between simulations, a new executable that includes the desired mechanism configuration must be compiled.

Using predefined chemical mechanisms

To select a predefined mechanism configuration in CMAQ, set the *Mechanism* variable in the Build-Script to one of the mechanism names listed in Table 6-3.

```
set Mechanism = MECHANISM_NAME
```

Refer to the README.md under CCTM/src/MECHS for detailed information reactions and on model species names for each mechanism.

Chemical mechanisms available with CMAQv5.3 can be found in Table 6-3. Atmospheric chemistry mechanisms of varying complexity are available to support diverse applications across scales and explore extensions for emerging problems and contaminants.

Table 6-3. Chemical Mechanisms Available with CMAQv5.3

Mechanism Name	Comment
cb6r3_ae7_aq	Carbon Bond 6 version r3 with aero7 treatment of SOA set up for standard cloud chemistry
cb6r3_ae7_aqkmt2	Carbon Bond 6 version r3 with aero7 treatment of SOA set up for expanded organic cloud chemistry version 2
cb6r3m_ae7_kmtbr	Carbon Bond 6 version r3 with aero7 treatment of SOA and DMS and marine halogen chemistry set up for expanded organic and halogen cloud chemistry
cb6r3_ae6_aq	Carbon Bond 6 version r3 with aero6 treatment of SOA set up for with standard cloud chemistry

Mechanism Name	Comment
cb6mp_ae6_aq	Carbon Bond 6 version r3 with air toxics and aero6 treatment of SOA set up for standard cloud chemistry
racm2_ae6_aq	Regional Atmospheric Chemistry Mechanism version 2 with aero6 treatment of SOA set up for with standard cloud chemistry
saprc07tic_ae7i_aq	State Air Pollution Research Center version 07tc with extended isoprene chemistry and aero7i treatment of SOA set up for with standard cloud chemistry
saprc07tic_ae7i_aqkmt2	State Air Pollution Research Center version 07tc with extended isoprene chemistry and aero7i treatment of SOA for expanded organic cloud chemistry version 2
saprc07tic_ae6i_aq	State Air Pollution Research Center version 07tc with extended isoprene chemistry and aero6i treatment of SOA set up for standard cloud chemistry
saprc07tic_ae6i_aqkmti	State Air Pollution Research Center version 07tc with extended isoprene chemistry and aero6i treatment of SOA for expanded organic cloud chemistry for isoprene
saprc07tc_ae6_aq	State Air Pollution Research Center version 07tc with aero6 treatment of SOA set up for with standard cloud chemistry

6.10.2 Solvers

To solve the photochemistry, the model uses one of three numerical methods or solvers. They differ by accuracy, generalization, and computational efficiency, i.e. model run times. Options include Euler Backward Iterative (EBI) solver (Hertel et al., 1993), Rosenbrock (ROS3) solver (Sandu et al., 1997), and Sparse Matrix Vectorized GEAR (SMVGEAR) solver (Jacobson and Turco, 1994). The EBI solver is default method because it is the fastest but is less accurate and must be *tailored* for each mechanism. The BuildScript defines which EBI solver to use as below.

```
set ModGas     = gas/ebi_${Mechanism}
```

If a user creates new FORTRAN modules representing the photochemical mechanism or modifies the existing modules, they must create a new EBI solver by using the create_ebi utility. Documentation on compiling and running create_ebi is available under the UTIL/create_ebi folder. The remaining two solvers, SMVGEAR and ROS3, are more accurate and less prone to convergence errors. Both methods are labeled as "generalized" because they only require the mechanism's namelist and FORTRAN modules representing the photochemical mechanism. Rosenbrock is preferred over SMVGEAR because it several times faster. To use either SMVGEAR and ROS3, the BuildScript defines ModGas as below.

```
set ModGas     = gas/smvgear
```

or

```
set ModGas     = gas/ros3
```

6.10.3 Photolysis

All the mechanisms include photolysis rates. The BuildScript has two options for calculating the rates.

```
set ModPhot    = phot/inline
```

or

```
set ModPhot    = phot/table
```

The in-line method (Binkowski et al., 2007) is the preferred option because it includes feedbacks from meteorology in addition to predicted ozone and aerosol concentrations. Three ASCII files support the in-line method. **PHOT_OPTICS** describes the optical properties of clouds, aerosols, and the earth's surface. The **OMI** file is used to determine how much light is absorbed by ozone above the model domain. Both files are included in the released version of CMAQ. Calculating photolysis rates uses one more file, the **CSQY_DATA_${Mechanism}** file, that depends on the mechanism used. It contains the cross sections and quantum yields of photolysis rates used by the mechanism. The files are provided for each mechanism in a released version of CMAQ. If a user creates a mechanism using new or additional photolysis rates, they have to create a new **CSQY_DATA_${Mechanism}** file. The inline_phot_preproc utility produces this file based on the Fortran modules describing the mechanism and data files describing the absorption cross-section and quantum yields described for each photolysis reaction. The CCTM RunScript sets values for each file's path through the environment variables OPTICS_DATA, OMI, and CSQY_DATA.

The other option uses look-up tables that contain photolysis rates under cloud free conditions based on a fixed meridional cross-section of atmospheric composition, temperature, density and aerosols. The values represent rates as a function of altitude, latitude and the hour angle of the sun on a specified Julian date. In model simulations, the method interpolates rates in the table for the date and corrects them to account for clouds described by the meteorology. Tables are dependent on the photochemical mechanism used. The jproc utility creates them based on the photochemical mechanism's FORTRAN modules. The CCTM RunScript sets the value for a table's path with the environment variable XJ_DATA.

6.10.4 Nitrous Acid (HONO)

In CMAQ, HONO is produced from emissions, gas-phase chemical reactions, and a heterogenous reaction on aerosol and ground surfaces. The contribution of emissions to HONO production is accounted for by including HONO emissions estimates from certain combustion sources. Each gas-phase chemical mechanism contains several gas-phase chemical reactions which also contributes to the HONO production. The heterogeneous production of HONO from the interaction of NO2 on aerosol surface is accounted for by including a heterogeneous reaction in the chemical mechanism. The heterogeneous production of HONO from the interaction of NO2 on ground surface is included in the air-surface exchange calculation and is controlled by the following RunScript flag:

```
setenv CTM_SFC_HONO Y
```

CMAQ uses a default setting of Y to include the production of HONO from the heterogeneous reaction on ground surface. The user can set it to N to exclude the heterogeneous production from the reaction. Note that the default setting for the inline deposition calculation (CTM_ILDEPV) flag is Y. If the flag is changed to N, then the production of HONO from the heterogeneous reaction on ground surface will not work properly. Additional description of the HONO chemistry in CMAQ can be found in Sarwar et al. (2008).

6.11 Aerosol Dynamics and Chemistry

Particulate Matter (PM) can be either primary (directly emitted) or secondary (formed in the atmosphere) and from natural or anthropogenic (man-made) sources. Secondary sources include gas-phase oxidation of SO2 to sulfate, condensation of ammonia and nitrate, and oxidation of gas-phase VOCs such as isoprene, monoterpenes, aromatics, and alkanes. Cloud processes also contribute to the formation of PM; for example, aqueous oxidation of sulfur dioxide in cloud droplets is a significant pathway for production of particulate sulfate. CCTM represents PM size using three interacting lognormal distributions, or modes. Two modes (Aitken and accumulation) are generally less than 2.5 μm in diameter while the coarse mode contains significant amounts of mass above 2.5 μm. PM2.5 and PM10, species aggregate metrics within the NAAQS, can be obtained from the model-predicted mass concentration and size distribution information.

The 6th generation CMAQ aerosol module (AERO6) was introduced in CMAQv5.0.2 and expanded the chemical speciation of PM. Eight new PM species were added to CMAQ in AERO6: Al, Ca, Fe, Si, Ti, Mg, K, and Mn. Four species that were explicitly treated in previous versions of CMAQ but were not modeled can now be treated as primary anthropogenic species: H2O, Na, Cl, and NH4. The PM emissions mass that remains after speciation into the new components is now input to the model as PMOTHER. AERO6 requires 18 PM emissions species: OC, EC, sulfate, nitrate, H2O, Na, Cl, NH4, NCOM, Al, Ca, Fe, Si, Ti, Mg, K, Mn, and Other (Reff et al., 2009). AERO6 continued to be incrementally updated in CMAQ v5.1-5.2.1 (see https://www.epa.gov/cmaq/how-cite-cmaq or release notes for when specific updates occured).

The 7th generation aerosol module (AERO7) is introduced in CMAQv5.3 with modifications and updates to the speciation and prediction of organic aerosols. For computational efficiency, the 2-product style speciation for SOA species from traditional aromatic VOC precursors (alkanes, toluene, xylenes, and benzene) has been replaced with four surrogate species with specific vapor pressures, following a VBS-style approaches used widely in models. In addition, the yield of organic aerosol from monoterpene reactions with OH and ozone has been increased, and monoterpene organic nitrates are explicitly treated as a SOA source. The treatment of alpha-pinene has also been made explicit in AERO7 in order to exclude alpha-pinene reactions with nitrate as a source of SOA. If users are employing online biogenic VOC emissions (via BEIS), then the alpha-pinene emissions will be treated correctly. If however, users are providing biogenic emissions to CMAQ from offline and only TERP is specified, we recommend scaling the alpha-pinene emissions to 30% of the total TERP emisisons and treating the remaining 70% of emitted TERP as TERP in CMAQ. This can be accomplished with the DESID

emissions interface. AERO7 also includes consideration of water uptake to the organic particle phase (ORGH2O).

Selection of AERO7 or AERO6 is accomplished through selection of the chemical mechanism in the build script as described in section 6.10 and Table 6-3. The aerosol microphysics (i.e. coagulation, condensation, new particle formation, deposition, etc.) are consistent for the two modules. The modules differ by the chemical species used to treat the PM constituents.

Both AERO7 and AERO6 mechanisms available in CMAQv5.3 are compatible with semivolatile primary organic aerosol (POA). For the nonvolatile POA configuration, mass is tracked separately in terms of its carbon (OC) and non-carbon (NCOM) content. With this approach, mass can be added to the non-carbon species to simulate the aging of POA in response to atmospheric oxidants. Simon and Bhave (2012) document the implementation of the second-order reaction between primary organic carbon and OH radicals. The semivolatile POA configuration segregates POA into several model species based on a combination of volatility and oxidation state. There are five POA species at low oxidation state representing low volatility, semivolatile and intermediate volatility compounds (LVPO1, SVPO1, SVPO2, SVPO3, IVPO1). As the gas-phase species (e.g. VLVPO1) oxidize with OH they form species with higher oxidation state (i.e. LVOO1, LVOO2, SVOO1, SVOO2, SVOO3). The multigenerational aging chemistry for the semivolatile POA configuration is derived from the approach of Donahue et al. (2012) which takes into account the functionalization and fragmentation of organic vapors upon oxidation. The semivolatile POA configuration also includes the option (on by default) of potential secondary organic aerosol from combustion sources (pcSOA). This species is emitted as a VOC (pcVOC) and forms SOA after reaction with OH. The emissions of pcVOC may be zeroed out by the user for specific sources using the DESID emissions control file; zeroing out pcVOC emissions is recommended for biomass and wood burning sources.

The aerosol module uses ISORROPIA v2.2 in the reverse mode to calculate the condensation/evaporation of volatile inorganic gases to/from the gas-phase concentrations of known coarse particle surfaces. It also uses ISORROPIA in the forward mode to calculate instantaneous thermodynamic equilibrium between the gas and fine-particle modes. The mass transfer of all semivolatile organic species is calculated assuming equilibrium absorptive partitioning, although some nonvolatile species do exist (e.g. cloud-processed organic aerosol, oligomers, nonvolatile POA (if selected)).

CMAQ can output the reduction in visual range caused by the presence of PM, perceived as haze. CCTM integrates Mie scattering (a generalized particulate light-scattering mechanism that follows from the laws of electromagnetism applied to particulate matter) over the entire range of particle sizes to obtain a single visibility value for each model grid cell at each time step. More detailed descriptions of the PM calculation techniques used in CCTM can be found in Binkowski and Shankar (1995), Binkowski and Roselle (2003), and Byun and Schere (2006).

For easier comparison of CMAQ's output PM values with measurements, time-dependent cutoff fractions may be output by the model (e.g. Jiang et al., 2006). These include quantities for describing the fraction of each mode that would be categorized as PM2.5 (i.e. PM25AT, PM25AC, and PM25CO) and PM1.0 (i.e. PM1AT, PM1AC, and PM1CO) as well as the fraction of particles from each mode that would be detected by an AMS (i.e AMSAT, AMSAC, and AMSCO). There is also a surface interaction module in the multipollutant version of CMAQ that calculates the flux of mercury to and from the surface (rather than just depositing mercury).

Further discussion on the scientific improvements to the CMAQ PM treatment is available in the release notes.

6.12 Aqueous Chemistry, Scavenging and Wet Deposition

Clouds are an important component of air quality modeling and play a key role in aqueous chemical reactions, vertical mixing of pollutants, and removal of pollutants by wet deposition. Clouds also indirectly affect pollutant concentrations by altering the solar radiation, which in turn affects photochemical pollutants (such as ozone) and the flux of biogenic emissions. The cloud module in CMAQ performs several functions related to cloud physics and chemistry. Three types of clouds are modeled in CMAQ: sub-grid convective precipitating clouds, sub-grid nonprecipitating clouds, and grid-resolved clouds. Grid-resolved clouds are provided by the meteorological model and no additional diagnosis is performed by CMAQ for those clouds. For the two types of sub-grid clouds, the cloud module in CCTM vertically redistributes pollutants, calculates in-cloud and precipitation scavenging, performs aqueous chemistry calculations, and accumulates wet deposition amounts. Aqueous chemistry and scavenging is calculated for resolved clouds as well, using the cell liquid water content and precipitation from the meteorological model.

When liquid water content (LWC), represented as the sum of cloud water, rain water, and graupel, in a cell (or column average in the case of sub-grid clouds) exceeds a critical threshold of 0.01 gm-3, a call is made to the cloud chemistry module where in-cloud scavenging and wet deposition are calculated in addition to aqueous phase chemistry. Accumulation and coarse mode aerosols are assumed to be instantaneously activated (i.e., nucleation scavenging), and Aitken mode particles (i.e., interstitial aerosol) are scavenged by the cloud droplets for the duration of cloud processing (Binkowski and Roselle, 2003). Gas phase species that participate in aqueous chemistry are taken up into the cloud water according to their Henry's Law coefficient, dissociation constants, and droplet pH. For each cloud chemistry time step, dissolved gas and aerosol species and associated ions are deposited out of the system according to a scavenging rate that is based on precipitation rate, cloud/layer thickness, and total water content (i.e., the sum of cloud water, rain water, graupel, ice, and snow). When the liquid water content does not exceed the threshold to call the cloud chemistry module (or for all species that do not participate in cloud chemistry), the wet deposition is calculated in a similar way in the "scavwdep" subroutine. Using the same expression for the washout coefficient as in the aqueous chemistry module, aerosol species are subject to wet removal assuming they are incorporated into cloud/rain water as above; while the fraction of gas phase species' concentrations subject to wet removal is a function of their effective Henry's Law coefficients at a prescribed droplet pH of 4. Essentially what is represented in CMAQ is in-cloud scavenging (or "rainout"); though arguably some effects of below-cloud scavenging (or "washout") may also be represented by including rain water in the LWC considered in calling/calculating cloud chemistry, as well as calculating aqueous chemistry and scavenging for the column (extending from the cloud top to the ground) in the case of sub-grid raining clouds. Explicit treatment of below-cloud scavenging (e.g., impaction scavenging of below-cloud aerosols by rain drops and snow) is not implemented at this time.

CMAQ's standard cloud chemistry treatment (AQCHEM) estimates sulfate production from five sulfur oxidation pathways and also includes a simple parameterization to estimate secondary organic aerosol formation from the reactions of glyoxal and methylglyoxal with the hydroxyl radical. The distribution

between gas and aqueous phases is determined by instantaneous Henry's law equilibrium, and the bi-section method is used to estimate pH (and the distribution of ionic species) assuming electroneutrality. Beginning with CMAQv5.1 a new set of options for cloud chemistry was introduced that relies on the Kinetic PreProcessor (KPP), version 2.2.3 (Damian et al., 2002) to generate a Rosenbrock integrator to solve the chemical kinetics, ionic dissociation, wet deposition, and kinetic mass transfer between the gas and aqueous phases in CMAQ clouds. These options can be collectively referred to as the AQCHEM "KMT" cloud chemistry treatments (Fahey et al., 2017).

There are several KMT chemistry options currently available in CMAQv5.3. AQCHEM-KMT treats the standard cloud chemistry mechanism and only differs from AQCHEM with the treatment of kinetic mass transfer between the phases (Schwartz, 1986) and Rosenbrock solver. AQCHEM-KMTI also includes an expanded aqueous-phase chemical mechanism that treats SOA formation from biogenic-derived epoxides (Pye et al., 2013) in cloud, in addition to the standard sulfur and alpha-dicarbonyl oxidation reactions. With CMAQv5.3 we introduce two additional cloud chemistry options: AQCHEM-KMT2 and AQCHEM-KMTBR. AQCHEM-KMT2 replaces the simple yield parameterization of SOA from glyoxal and methylglyoxal with a more mechanistic representation of the multi-step formation of oxalic acid/oxalate and other organic acids (assumed here to remain in the aerosol phase after cloud droplet evaporation) from the reactions of hydroxyl radical with glyoxal, methylglyoxal, glycolaldehyde, and acetic acid (Lim et al., 2005; Tan et al., 2009). AQCHEM-KMT2 also expands upon the reactions in AQCHEM-KMTI with additional chemistry for S, N, O-H, and C species (Leriche et al., 2013; Warneck, 1999; Lee and Schwartz, 1983). AQCHEM-KMTBR is the companion aqueous chemistry routine to the gas-phase cb6r3m_ae7_kmtbr mechanism and contains the standard 5 S(IV) oxidation reactions, SOA parameterization from glyoxal and methylglyoxal, as well as additional reactions involving Bromine species in cloud water (Sarwar et al., 2019).

The AQCHEM KMT family of cloud chemistry options can be significantly more computationally demanding than standard AQCHEM and may be thus better suited for research applications, particularly those investigating cloud/fog events or the evolution of species whose concentrations are potentially heavily influenced by cloud processing and not explicitly represented in the standard AQCHEM mechanism (e.g., oxalate – AQCHEM-KMT2). Note that when using the gas-phase mechanism with marine chemistry (CB6R3M_AE7_KMTBR), one is required to also run the companion aqueous chemistry routine, AQCHEM-KMTBR. For limited-area simulations where the primary focus is on simulating ozone or total PM2.5 concentrations, especially for longer-term averages, standard AQCHEM would likely capture the most important cloud chemistry impacts (i.e., sulfate formation from the main aqueous oxidation pathways) and is significantly more computationally efficient.

To invoke the default AQCHEM cloud chemistry option, the BuildScript under the CCTM Science Modules section should be set as follows:

```
set ModCloud  = cloud/acm_ae7
or
set ModCloud  = cloud/acm_ae6
```

For the AQCHEM-KMT cloud chemistry option, use the following option in the BuildScript:

```
set ModCloud  = cloud/acm_ae6_kmt
```

AQCHEM and AQCHEM-KMT can be used with any of the cb6r3_ae6, cb6r3_ae7, racm, or saprc07 gas phase chemistry mechanisms.

For the AQCHEM-KMTI cloud chemistry option, use the following option in the BuildScript:

```
set ModCloud  = cloud/acm_ae6i_kmti
```

AQCHEM-KMTI is meant to be used with the saprc07tic_ae6i gas phase chemistry option; i.e., in the BuildScript:

```
set Mechanism = saprc07tic_ae6i_aqkmti
```

For the AQCHEM-KMT2 cloud chemistry option, use the following option in the BuildScript:

```
set ModCloud  = cloud/acm_ae7_kmt2
```

AQCHEM-KMT2 should only be used in conjunction with the cb6r3_ae7 or saprc07tic_ae7i gas phase chemical mechanisms; i.e., in the BuildScript:

```
set Mechanism = cb6r3_ae7_aqkmt2
```

OR

```
set Mechanism = saprc07tic_ae7i_aqkmt2
```

For the AQCHEM-KMTBR cloud chemistry option, use the following option in the BuildScript:

```
set ModCloud  = cloud/acm_ae7_kmtbr
```

Note that this cloud chemistry option will be used automatically (and should be used only) when the gas phase chemistry option “cb6r3m_ae7_kmtbr” is chosen.

For toxics/Hg simulations (using the gas phase “cb6mp_ae6_aq” mechanism), one may also invoke the complementary cloud chemistry routine that includes aqueous phase chemistry for some toxic species in addition to the default chemistry:

```
set ModCloud  = cloud/acm_ae6_mp
```

6.13 Potential Vorticity Scaling

Since cross-tropopause transport of O3 can be a significant contributor to the tropospheric O3 budget, accurately characterizing the fraction of O3 in the troposphere, especially at the surface, that is of stratospheric origin is of interest in many model applications. This fraction varies spatially and seasonally in response to the tropopause height, and perhaps even more episodically, from deep intrusion events associated with weather patterns and frontal movement (e.g., Mathur et al., 2017). Potential vorticity (PV; 1 PV unit = 106 m2 K kg-1 s-1) has been shown to be a robust indicator of air mass exchange between the stratosphere and the troposphere with strong positive correlation with O3 and other trace species transported from the stratosphere to the upper troposphere (Danielsen, 1968). This correlation can be used to develop scaling factors that specify O3 in the modelled upper troposphere/lower stratosphere (UTLS) based on estimated PV. CMAQ uses a dynamical PV-scaling parameterization developed by correlating model potential vorticity fields and measured O3 (from World Ozone and Ultraviolet Radiation Data Centre) between 100-50mb over a 21-year period. This generalized parameterization, detailed in Xing et al. (2016), can dynamically represent O3 in the UTLS across the Northern Hemisphere. The implementation of the new function significantly improves CMAQ's simulation of UTLS O3 in both magnitude and seasonality compared to observations, which results in a more accurate simulation of the vertical distribution of O3 across the Northern Hemisphere (Xing et al., 2016; Mathur et al., 2017). It should be noted that to represent stratosphere-troposphere exchange of O3, appropriate vertical grid resolution near the tropopause should also be used with the PV scaling scheme.

To invoke the potential vorticity scaling of modeled O3 in the upper layers (100-50mb),

```
set potvort03
```

should be specified in the BuildScript. Also, potential vorticity fields must be available in the METCRO3D files generated by MCIP. This is enabled by setting LPV = 1 in the MCIP runscript.

6.14 References

Bash, J.O. (2010). Description and initial simulation of a dynamic bidirectional air-surface exchange model for mercury in Community Multiscale Air Quality model. J. Geophys. Res., 115, D06305, doi:10.1029/2009JD012834.

Binkowski, F.S., Arunachalam, S., Adelman, Z., & Pinto, J. (2007). Examining photolysis rates with a prototype on-line photolysis module in CMAQ. J. Appl. Meteor. and Clim., 46, 1252-1256. doi:10.1175/JAM2531.1.

Binkowski, F.S., & Roselle, S.J. (2003). Models-3 Community Multiscale Air Quality (CMAQ) model aerosol component. 1. Model description. J. Geophys. Res., 108, 4183, doi:10.1029/2001JD001409.

Binkowski, F.S., & Shankar, U. (1995). The regional particulate model: Part I. Model description and preliminary results. J. Geophys. Res., 100, 26 191–26 209.

Byun, D.W., & Ching, J.K.S. (1999). Science algorithms of the EPA models-3 Community Multiscale Air Quality (CMAQ) Modeling system. U. S. Environmental Protection Agency Rep. EPA 600/R 99/030, 727 pp.

Byun, D., & Schere, K.L. (2006). Review of the governing equations, computational algorithms, and other components of the Models-3 Community Multiscale Air Quality (CMAQ) modeling system. Appl. Mech. Rev., 59, 51–77. doi:10.1115/1.212863.

Colella, P., & Woodward, P.L. (1984). The piecewise parabolic method (PPM) for gas-dynamical simulations. J. Comput. Phys., 54, 174–201.

Cooter, E.J., Bash, J.O., Benson V., & Ran, L.M. (2012). Linking agricultural management and air-quality models for regional to national-scale nitrogen deposition assessments. Biogeosciences, 9, 4023-4035.

Damian, V., Sandu, A., Damian, M., Potra, F., & Carmichael, G.R. (2002). The kinetic preprocessor KPP – A software environment for solving chemical kinetics. Computers and Chemical Engineering, 26, 1567-1579.

Danielsen, E. F. (1968). Stratospheric-tropospheric exchange based on radioactivity, ozone and potential vorticity. J. Atmos. Sci., 25, 502–518.

Donahue, N.M., et al. (2012). A two-dimensional volatility basis set – Part 2: Diagnostics of organic-aerosol evolution. Atmospheric Chemistry and Physics, 12(2), 615-634.

Fahey, K.M., Carlton, A.G., Pye, H.O.T., Baek, J., Hutzell, W.T., Stanier, C.O., Baker, K.R., Appel, K.W., Jaoui, M., & Offenberg, J.H. (2017). A framework for expanding aqueous chemistry in the Community Multiscale Air Quality (CMAQ) model version 5.1. Geosci. Model Dev., 10, 1587-1605.

Gantt, B., Kelly, J.T., & Bash, J.O. (2015). Updating sea spray aerosol emissions in the Community Multiscale Air Quality (CMAQ) model version 5.0.2. Geosci. Model Dev., 8, 3733-3746. doi:10.5194/gmd-8-3733-201.

Giorgi, F. (1986). A particle dry-deposition parameterization scheme for use in tracer transport models. J. Geophys. Res., 91(D9), 9794-9806.

Gong, S.L. (2003). A parameterization of sea-salt aerosol source function for sub- and super-micron particles. Global Biogeochem. Cy., 17. doi: 10.1029/2003gb002079.

Hertel O., Berkowicz, R., Christensen, J., & Hov, O. (1993). Test of two numerical schemes for use in atmospheric transport-chemistry models. Atmos. Environ., 27A, 2591–2611.

Jacobson, M., & Turco, R.P. (1994) SMVGEAR: A sparse-matrix, vectorized Gear code for atmospheric models. Atmos. Environ., 28, 2991–3003.

Jaeglé, L., Quinn, P.K., Bates, T.S., Alexander, B., & Lin, J.T. (2011). Global distribution of sea salt aerosols: new constraints from in situ and remote sensing observations. Atmos. Chem. Phys., 11, 3137–3157. doi: 10.5194/acp-11-3137-2011.

Jiang, W., Smyth, S., Giroux, É., Roth, H., & Yin, D. (2006). Differences between CMAQ fine mode particle and PM2.5concentrations and their impact on model performance evaluation in the lower Fraser valley. Atmos. Environ., 40, 4973–4985.

Kang, D., Pickering, K., Allen, D., Foley, K., Wong, D., Mathur, R., & Roselle, S. (2019). Simulating lightning NOX production in CMAQv5.2: Evolution of scientific updates. Geosci. Model Dev. Discuss.,1-23. doi:10.5194/gmd-2019-33.

Kelly, J.T., Bhave, P.V., Nolte, C.G., Shankar, U., & Foley, K.M. (2010).Simulating emission and chemical evolution of coarse sea-salt particles in the Community Multiscale Air Quality (CMAQ) model. Geosci. Model Dev., 3, 257-273. doi: 10.5194/gmd-3-257-2010.

Lee, Y.N. & Schwartz, S.E. (1983). Kinetics of oxidation of aqueous sulfur(IV) by nitrogen dioxide. In H.R. Pruppacher et al. (eds.), precipitation scavenging, dry deposition, and resuspension, version 1. Elsevier.

Leriche, M., Pinty, J.P., Mari, C., & Gazen, D. (2013). A cloud chemistry module for the 3-D cloud-resolving mesoscale model Meso-NH with application to idealized cases. Geosci. Mod. Dev., 6, 1275-1298.

Lim, H., Carlton, A.G., & Turpin, B.J. (2005). Isoprene forms secondary organic aerosol through cloud processing: model simulations. Environ. Sci. Technol., 39, 4441–4446.

Massad, R.S., Nemitz, E., Sutton, M.A. (2010). Review and parameterization of bi-directional ammonia exchange between vegetation and the atmosphere. Atmos. Chem. Phys., 10, 10359-10386.

Mathur, R. & Peters, L.K. (1990). Adjustment of wind fields for application in air pollution modeling, Atmos. Environ., 24(5), 1095-1106.

Mathur, R., Xing, J., Gilliam, R., Sarwar, G., Hogrefe, C., Pleim, J., Pouliot, G., Roselle, S., Spero, T. L., Wong, D. C., and Young, J. (2017) Extending the Community Multiscale Air Quality (CMAQ) modeling system to hemispheric scales: overview of process considerations and initial applications, Atmos. Chem. Phys., 17, 12449-12474, doi: 10.5194/acp-17-12449-2017.

Murphy, B.N., Woody, M.C., Jimenez, J.L., Carlton, A.M.G., Hayes, P.L., Liu, S., Ng, N.L., Russell, L.M., Setyan, A., Xu, L., Young, J., Zaveri, R.A., Zhang, Q., & Pye, H.O.T. (2017). Semivolatile POA and parameterized total combustion SOA in CMAQv5.2: impacts on source strength and partitioning. Atmospheric Chemistry and Physics Discussions, 1-44.

Nemitz, E., Milford, C., Sutton, M.A. (2001). A two-layer canopy compensation point model for describing bi-directional biosphere-atmosphere exchange of ammonia. Q. J. Roy. Meteor. Soc.,127, 815-833.

Odman, M.T., & Russell, A.G. (2000). Mass conservative coupling of non-hydrostatic meteorological models with air quality models, in Air Pollution Modelling and Its Application XIII, edited by S.-E. Gryning and E. Batchvarova. Kluwer Academic/Plenum Publishers, New York, 651-660.

Ovadnevaite, J., Manders, A., de Leeuw, G., Ceburnis, D., Monahan, C., Partanen, A.I., Korhonen, H., & O'Dowd, C. D. (2014). A sea spray aerosol flux parameterization encapsulating wave state. Atmos. Chem. Phys., 14, 1837-1852. doi: 10.5194/acp-14-1837-2014.

Pleim, J., Venkatram, A., Yamartino, R. (1984). ADOM/TADAP Model development program: The dry deposition module. Ontario Ministry of the Environment, 4.

Pleim, J.E. (2007a). A combined local and nonlocal closure model for the atmospheric boundary layer. Part I: Model description and testing. Journal of Applied Meteorology and Climatology, 46(9), 1383-1395.

Pleim, J.E. (2007b). A combined local and nonlocal closure model for the atmospheric boundary layer. Part II: Application and evaluation in a mesoscale meteorological model. Journal of Applied Meteorology and Climatology, 46(9), 1396-1409.

Pleim, J., & Ran, L. (2011). Surface flux modeling for air quality applications. Atmosphere, 2(3), 271-302.

Pleim, J. E., Ran, L., Appel, W., Shephard, M.W., & Cady-Pereira K. (2019). New bidirectional ammonia flux model in an air quality model coupled with an agricultural model. JAMES in review.

Pye, H.O.T., Pinder, R.W., Piletic, I.R., Xie, Y., Capps, S.L., Lin, Y.H., Surratt, J.D., Zhang, Z.F., Gold, A., Luecken, D.J., Hutzell W.T., Jaoui, M., Offenberg, J.H., Kleindienst, T.E., Lewandowski, M., & Edney, E.O. (2013). Epoxide pathways improve model predictions of isoprene markers and reveal key role of acidity in aerosol formation. Environ. Sci. Technol., 47(19), 11056-11064.

Ran, L., Cooter, E., Benson, V., & He, Q. (2011). Chapter 36: Development of an agricultural fertilizer modeling system for bi-directional ammonia fluxes in the CMAQ model. In D. G. Steyn, & S. Trini Castelli (Eds.), air pollution modeling and its application XXI. Springer, 213-219.

Reff, A., Bhave, P.V., Simon, H., Pace, T.G., Pouliot, G.A., Mobley, J.D., & Houyoux, M. (2009). Emissions inventory of PM2.5 trace elements across the United States. Env. Sci. & Technol. 43, 5790-5796.

Sandu, A., Verwer, J.G., Blom, J.G., Spee, E.J., Carmichael, G.R., & Potra, F.A. (1997). Benchmarking stiff ODE solvers for atmospheric chemistry problems. II: Rosenbrock solvers. Atmos. Environ., 31, 3459–3472.

Sarwar, G., Gantt, B., Foley, K., Fahey, K., Spero, T.L., Kang, D., Mathur, R., Foroutan, H., Xing, J., Sherwen, T, & Saiz-Lopez, A. (2019). Influence of bromine and iodine chemistry on annual, seasonal, diurnal, and background ozone: CMAQ simulations over the Northern Hemisphere. Atmos. Env., Accepted.

Sarwar, G., Roselle, R., Mathur, R., Appel, W., Dennis, R. L., & Vogel, B. (2008). A Comparison of CMAQ HONO Predictions with Observations from the Northeast Oxidant and Particle Study, Atmospheric Environment, 42, 5760-5770.

Schwartz, S.E. (1986). Mass transport considerations pertinent to aqueous-phase reactions of gases in liquid water clouds. In chemistry of multiphase atmospheric systems, NATO ASI Series, G6, 415-471.

Simon, H., & Bhave, P.V. (2012). Simulating the degree of oxidation in atmospheric organic particles. Environ. Sci. Technol., 46(1), 331-339.

Skamarock, W.C., Klemp, J.B., Dudhia, J., Gill, D.O., Liu, Z., Berner, J., Wang, W., Powers, J.G., Duda, M.G., Barker, D.M., & Huang, X.Y. (2019). A description of the advanced research WRF version 4. NCAR Technical Note, NCAR/TN–556+STR.

Slinn, W.G.N. (1982). Predictions for particle deposition to vegetative canopies, Atmos. Environ., 16, 1785-1794.

Smagorinsky, J. (1963). General circulation experiments with the primitive equations. Mon. Wea. Rev., 91/3, 99-164.

Tan, Y., Perri, M.J., Seitzinger, S.P., & Turpin, B.J. (2009). Effects of precursor concentration and acidic sulfate in aqueous glyoxal-OH radical oxidation and implications for secondary organic aerosol. Env. Sci. Technol., 43, 8105–8112.

Warneck, P. (1999). The relative importance of various pathways for the oxidation of sulfur dioxide and nitrogen dioxide in sunlit continental fair weather clouds. Phys. Chem. Chem. Phys., 1, 5471-5483.

Xing, J., Mathur, R., Pleim, J., Hogrefe, C., Wang, J., Gan, C.M., Sarwar, G., Wong, D., & McKeen, S. (2016). Representing the effects of stratosphere-troposphere exchange on 3D O3 distributions in chemistry transport models using a potential vorticity based parameterization, Atmos. Chem. Phys., 16, 10865-10877, doi:10.5194/acp-16-10865-2016.

Yi, C. (2008). Momentum transfer within canopies. J. App. Meteor. Clim., 47, 262-275.

7. Model Output Files

7.1 Introduction

In this section, details on the routine CCTM output files are provided. All CMAQ programs produce model output files that adhere to the netCDF format. In addition to model data output, CMAQ can optionally produce ASCII log files that contain intermediate model execution information from the various CMAQ processes and captured with respect to processor number. If the log file option is not selected by the user and the simulation is run interactively, CMAQ will write all of the log information to the screen along with the standard error, which can be captured to a text file using basic UNIX syntax. Additional output files are created when using the Process Analysis (PA), Integrated Source Apportionment Method (ISAM) and Detailed Emissions Scaling, Isolation and Diagnostics Module (DESID) options. The files associated with these options are discussed in Chapter 9, Chapter 11, and Appendix B, respectively.

Table 7-1. CMAQ Output files

File Name1	File Type	Time-Dependence2	Spatial Dimensions3
Standard			
Output Log	ASCII	n/a	n/a
CCTM_CONC	GRDDED3	Hourly	XYZ’
CCTM_ACONC	GRDDED3	Hourly	XYZ’
CCTM_DRYDEP	GRDDED3	Hourly	XY
CCTM_WETDEP1	GRDDED3	Hourly	XY
Restart			
CCTM_CGRID	GRDDED3	1-hour	XYZ
CCTM_MEDIA	GRDDED3	Hourly	XY
CCTM_SOILOUT	GRDDED3	n/a (see detailed file description below)	XY
Diagnostic and Advanced			
CCTM_PMDIAG	GRDDED3	Hourly	XYZ’
CCTM_APMDIAG	GRDDED3	Hourly	XYZ’
CCTM_B3GTS_S	GRDDED3	Hourly	XY
CCTM_DEPV	GRDDED3	Hourly	XY
CCTM_PT3D	GRDDED3	Hourly	XYZ
CCTM_DUSTEMIS	GRDDED3	Hourly	XY
CCTM_DEPVMOS	GRDDED3	Hourly	XYW
CCTM_DEPVFST	GRDDED3	Hourly	XYW

File Name1	File Type	Time-Dependence2	Spatial Dimensions3
CCTM_DDEP_MOS	GRDDED3	Hourly	XYW
CCTM_DDEP_FST	GRDDED3	Hourly	XYW
CCTM_LTNGHRLY	GRDDED3	Hourly	XYZ
CCTM_LTNGCOL	GRDDED3	Hourly	XY
CCTM_PHOTDIAG1	GRDDED3	Hourly	XY
CCTM_PHOTDIAG2	GRDDED3	Hourly	XYZ'
CCTM_PHOTDIAG3	GRDDED3	Hourly	XYZ'
CCTM_SSEMIS	GRDDED3	Hourly	XY
CCTM_WETDEP2	GRDDED3	Hourly	XY
CCTM_VEXT	GRDDED3	Hourly	WZ

1By default, output files are named CCTM_XXX_${CTM_APPL}.nc where XXX is the file identifier and ${CTM_APPL} is a user defined string that identifies the model run.
2While "Hourly" is indicated, users may define a different time step (e.g. 30 minutes) for model output by changing the TSTEP variable in the runscript. From here onward, the term "Hourly" will be used for description purposes.
3X is the dimension along the x-axis, Y is the dimension along the y-axis, Z is the vertical dimension, Z' is the user pre-defined size of the vertical dimension controlled by the environment variables CONC_BLEV_ELEV, ACONC_BLEV_ELEV, APMDIAG_BLEV_ELEV, and NLAYS_PHOTDIAG (range from 1 to all layers) and W is a non-layer dimension, e.g. number of LU fractions, number of sites for vertical extraction.
4A special ASCII output file, FLOOR_xxx with xxx being the processor number, contains information when a simulation results in negative concentrations.

7.2 CCTM Output Files

Some output files created by the CCTM are considered standard output as these contain hourly concentration and deposition values and information to document the run. Options for these files are controlled by their corresponding environment variable in the CCTM RunScript (e.g. run_cctm.csh).

CMAQ output log Return to Table 7-1 All of the CMAQ processors generate standard output and standard error during execution. When you run the CMAQ executable interactively, diagnostic output information can be captured to a log file using a UNIX redirect command:

```
run.cctm >& tee cctm.log
```

The LOGFILE environment variable allows users to specify the name of a log file for capturing the standard output from the program. If this variable is not set, the standard output is written to the terminal and can be captured using the UNIX redirect command (">"), as shown in the example above.

CCTM_CONC: CCTM hourly instantaneous concentration file Return to Table 7-1

The 2-D or 3-D CCTM hourly concentration file (CONC) contains instantaneous gas-phase species mixing ratios (ppmV) and aerosol species concentrations (µg m-3) at the end of each model output

time step. The number and type of species contained in the CONC files depends on the chemical mechanism and aerosol model configurations that are selected when the CCTM is compiled. The Species NameLists files within the mechanism directories list the modeled species, and contain a column that specifies which species are written to the CONC files (e.g. AE_cb6r3_ae7_aq.nml). The GC_*mechname*.nml file lists the gas-phase species, the AE_*mechname*.nml file lists the aerosol species, and the NR_*mechname*.nml lists the nonreactive (inert) species. Species can be removed from the CONC file by editing the CONC column in the NameList file(s) to reduce the number of species that are written to, and thus the size of the CONC file. Users can also specify the output species list by modifying the environment variable CONC_SPCS in the RunScript which overrides the setting of the CONC column in the NameList file(s). By default, concentrations for all model layers are output to the CONC file. Users may specify the layers to output using the CONC_BLEV_ELEV environment variable in the RunScript where BLEV corresponds to the bottom layer number and ELEV corresponds to the top layer number.

CCTM_ACONC: hourly average concentration file Return to Table 7-1

The 2-D or 3-D CCTM integral average concentration file contains average model species concentrations for each model hour, as opposed to instantaneous concentrations at the end of each output time step. The species written to the ACONC file are set by the user in the CCTM RunScript using the environment variable AVG_CONC_SPCS. The model layers for which hourly average concentrations are calculated are also set in the CCTM RunScript using the environment variable ACONC_BLEV_ELEV, where BLEV corresponds to the bottom layer number and ELEV corresponds to the top layer number. An example setting for the ACONC_BLEV_ELEV variable is "1 6", which defines layers 1 through 6 as the vertical extent for which hourly average concentrations are calculated and written to the ACONC file.

CCTM_DRYDEP: hourly cumulative dry deposition file Return to Table 7-1

The 2-D CCTM dry deposition file contains cumulative hourly dry deposition fluxes (kg hectare-1) for selected model species. CCTM calculates dry deposition for all of the species listed in the dry deposition column of the Species NameLists files within the mechanism directories. The GC_*mechname*.nml file lists the gas-phase species, the AE_*mechname*.nml file lists the aerosol species, and the NR_*mechname*.nml lists the nonreactive (inert) species. Species can be removed from the dry deposition file by editing the DDEP column in the NameList file(s).

CCTM_WETDEP1: hourly cumulative wet deposition file Return to Table 7-1

The 2-D CCTM wet deposition file contains cumulative hourly wet deposition fluxes (kg hectare-1) for selected model species. CCTM calculates wet deposition for all of the species listed in the wet deposition column of the Species NameLists files within the mechanism directories. The GC_*mechname*.nml file lists the gas-phase species, the AE_*mechname*.nml file lists the aerosol species, and the NR_*mechname*.nml lists the nonreactive (inert) species. Species can be removed from the wet deposition file by editing the WDEP column in the NameList file(s).

7.3 Restart Files

There are several files created by the CCTM that are used to enable a restart of the run for any specific day. The files contain values for parameters at the end of the day which are used to initialize the values

for the start of calculations for the next day.

CCTM_CGRID: gridded concentration restart file Return to Table 7-1

The 3-D CCTM ending concentration file contains gas-phase species mixing ratios (ppmV) and aerosol species concentrations (µg m-3) at the end of each simulation period. The number and types of species contained in the output CGRID files depend on the chemical mechanism and aerosol model configurations that are selected when CCTM is compiled. This file can be used to initialize CCTM from a simulation period that the model completed. For example, if the CCTM is configuring to produce daily output files, a CGRID file will be written out at the end of each simulation day. These concentrations then become the initial conditions for the next simulation period.

CCTM_MEDIA: Bidirectional modeling media concentration file Return to Table 7-1

This 2-D CCTM file contains the soil NH4+ and pH concentrations and/or the soil, vegetation and water Hg concentrations. This file is only created when the CTM_ABFLUX environment variable or the CTM_HGBIDI variable in the RunScript is set to Y (Default is N) and is used to initialize the next day of the model simulation.

CCTM_SOILOUT Return to Table 7-1

This optional 2-D CCTM file contains hourly total rainfall information for subsequent use by the CCTM in-line biogenics module. It is written out at the end of each simulation day and is only created if the CTM_BIOGEMIS environment variable in the RunScript is set to Y (Default is N). With the exception of the first day of the simulation when the environment variable INITIAL_RUN is set to Y, the previous day's rainfall information contained in the file is used in the calculation of soil NO emissions by the CCTM in-line biogenics module. This is accomplished by setting the SOILINP environment variable in the RunScript for a given day to the CCTM_SOILOUT file created at the end of the previous day's simulation. Note that even though this file contains 24 hourly gridded rainfall fields, it has a time-independent file structure and stores these 24 values as 24 separate time-independent variables (RAINFALL01, … RAINFALL24). However, while the structure of the file is time-independent, each day's CCTM_SOILOUT file is unique due to the daily variations in meteorology. Therefore, care must be taken to ensure that the SOILINP file specified for a given day is indeed the CCTM_SOILOUT file for the previous day rather than that for a different day.

7.4 Diagnostic and Advanced CMAQ Output Files

Along with the standard output files detailed in the previous section, CCTM can be configured to output several auxiliary files for diagnostic model purposes. Each option is controlled by its corresponding environment variable in the CCTM RunScript (e.g. run_cctm.csh). For logical values, TRUE/T is equivalent to Y and FALSE/F is equivalent to N.

Note that I/O APIv3.2 supports up to MXFILE3=64 open files, each with up to MXVARS3=2048. Turning on all of the diagnostic and advanced CMAQ output files can exceed this upper limit of open files, leading to a model crash. To avoid this issue, users may use I/O API version 3.2 "large" that increases MXFILE3 to 512 and MXVARS3 to 16384. This version is available as a zip file from the following address:

https://www.cmascenter.org/ioapi/download/ioapi-3.2-large.tar.gz

Installation instructions for I/O API v5.3-large are provided in README.txt in the .tar.gz file.

FLOOR: concentration-reset diagnostics file Return to Table 7-1

This optional ASCII file contains specific gridcells/timesteps in which species with negative concentrations are reset to zero. The location and name of the file is set by the FLOOR_FILE environment variable.

CCTM_PMDIAG: instantaneous hourly aerosol diagnostics file Return to Table 7-1

This optional 2-D or 3-D CCTM diagnostic file contains instantaneous information at the end of the hour for each model hour on the geometric mean diameters, geometric standard deviations, bulk densities, 2nd moments and 3rd moments for the lognormal modes. It also includes the fraction of each mode that contributes to PM1, PM2.5, and PM10 and the AMS transmission factor for each mode. Many diagnostics relating to heterogenous chemistry are provided including the N2O5 reaction probability, the ClNO2 reaction yield, and the IEPOX uptake coefficient. Units for all variables are specified in the output file. The number of layers in this output file is determined by the setting of the CONC_BLEV_ELEV environment variable in the RunScript. This file is only created if the CTM_APMDIAG environment variable in the RunScript is set to Y (Default is N).

CCTM_APMDIAG: average hourly aerosol diagnostics file Return to Table 7-1

This optional 2-D or 3-D CCTM diagnostic file contains integral average information for each model hour on the geometric mean diameters, geometric standard deviations, bulk densities, 2nd moments and 3rd moments for the lognormal modes. It also includes the fraction of each mode that contributes to PM1, PM2.5, and PM10 and the AMS transmission factor for each mode. Many diagnostics relating to heterogenous chemistry are provided including the N2O5 reaction probability, the ClNO2 reaction yield, and the IEPOX uptake coefficient. Units for all variables are specified in the output file. The number of layers in this output file is determined by the setting of the APMDIAG_BLEV_ELEV environment variable in the RunScript. This file is only created if the CTM_APMDIAG environment variable in the RunScript is set to Y (Default is N).

CCTM_B3GTS_S: biogenic emissions diagnostic file Return to Table 7-1

This optional 2-D CCTM hourly output file contains total hourly biogenic emissions in mass units calculated in-line by the CCTM when the CTM_BIOGEMIS environment variable is set to Y. This file is only created if the B3GTS_DIAG environment variable in the RunScript is set to Y (Default is Y).

CCTM_DEPV: inline deposition diagnostics file Return to Table 7-1

This optional 2-D CCTM file contains the deposition velocity (m/s) for each chemical species calculated for the final time step for the hour. CCTM calculates the deposition velocity for all of the species listed in the deposition velocity column of the Species NameLists files files within the mechanism directories. The GC_*mechname*.nml file lists the gas-phase species, the AE_*mechname*.nml file lists the aerosol species, and the NR_*mechname*.nml lists the nonreactive (inert) species. Species can be removed from the deposition velocity file by editing the DDEP column in the NameList file(s). This file is only created if the CTM_DEPV_FILE environment variable in the RunScript is set to Y (Default is N).

CCTM_PT3D: point source emissions diagnostics file Return to Table 7-1

This optional 3-D CCTM file records the 3-D point source emissions for each layer as a linear average over the output timestep. This file is only created if the PT3DDIAG environment variable in the RunScript is set to Y (Default is N).

CCTM_DUSTEMIS: dust emissions diagnostic file Return to Table 7-1

This optional 2-D CCTM hourly output file contains dust emissions in mass units calculated in-line by the CCTM when the CTM_WB_DUST environment variable is set to Y. This file is only created if the CTM_DUSTEM_DIAG environment variable in the RunScript is set to Y (Default is N).

CCTM_DEPVMOS: land use specific deposition velocity file Return to Table 7-1

This optional 3-D CCTM file contains the deposition velocity (m s-1) for the final time step of the hour for each land use type within a grid cell. This output file is structured with the land use category being the 3rd dimension (i.e. equivalent to the layers in a concentration file). So, for model runs using the NLCD land use category system, the files will have 40 "layers". This file is only created if the DepMod environment variable in the BuildScript is set to stage (rather than m3dry) and if the CTM_MOSAIC environment variable in the RunScript is set to Y (Default is N).

CCTM_DDMOS: land use specific deposition flux file Return to Table 7-1

This optional 3-D CCTM file contains the total deposition (kg hectare-1) for the hour for each land use type within each grid cell. This output file is structured with the land use category being the 3rd dimension (i.e. equivalent to the layers in a concentration file). So, for model runs using the NLCD land use category system, the files will have 40 "layers". This file is only created if the ModDepv environment variable in the BuildScript is set to stage (rather than m3dry) and if the CTM_MOSAIC environment variable in the RunScript is set to Y (Default is N).

CCTM_DEPVFST: stomatal deposition velocity file Return to Table 7-1

This optional 3-D CCTM file contains the deposition velocity (m s-1) through the stomatal pathway for the final time step of the hour for each land use type within a grid cell. This output file is structured with the land use category being the 3rd dimension (i.e. equivalent to the layers in a concentration file). So, for model runs using the NLCD land use category system, the files will have 40 "layers". This file is only created if the ModDepv environment variable in the BuildScript is set to stage (rather than m3dry) and if the CTM_FST environment variable in the RunScript is set to Y (Default is N).

CCTM_DDFST: stomatal flux file Return to Table 7-1

This optional 3-D CCTM file contains the total deposition (kg hectare-1) through the stomatal pathway for the hour for each land use type within each grid cell. This output file is structured with the land use category being the 3rd dimension (i.e. equivalent to the layers in a concentration file). So, for model runs using the NLCD land use category system, the files will have 40 "layers". This file is only created if the ModDepv environment variable in the BuildScript is set to stage (rather than m3dry) and if the CTM_FST environment variable in the RunScript is set to Y (Default is N).

CCTM_LTNGHRLY: hourly lightning emissions file Return to Table 7-1

This optional 3-D CCTM file contains hourly lightning NO emissions (mol/s) calculated in-line by the CCTM when setting the CTM_LTNG_NO environment variable to Y. This file is only created if the CTM_LTNGDIAG_1 environment variable in the RunScript is set to Y (Default is N).

CCTM_LTNGCOL: hourly column total lightning emissions Return to Table 7-1

This optional 2-D CCTM file contains hourly column-total lightning NO emissions (mol/s) calculated in-line by the CCTM when setting the CTM_LTNG_NO environment variable to Y. This file is only created if the CTM_LTNGDIAG_2 environment variable in the RunScript is set to Y (Default is N).

CCTM_PHOTDIAG1: In-line photolysis inputs and outputs - summary file Return to Table 7-1

This optional 2-D CCTM file contains general summary information for the photolysis calculation including the surface albedo, select photolysis rates and flux values. This file is only created if the CTM_PHOTDIAG environment variable in the RunScript is set to Y (Default is N).

CCTM_PHOTDIAG2_2: In-line photolysis output – gridded photolysis rates Return to Table 7-1

This optional 3-D CCTM file contains the photolysis rates calculated in-line by the CCTM. The number of layers is set by the NLAYS_PHOTDIAG environment variable (Default is all layers). This file is only created if the CTM_PHOTDIAG environment variable in the RunScript is set to T (Default is N).

CCTM_PHOTDIAG3: In-line photolysis inputs and outputs – detailed Return to Table 7-1

This optional 3-D CCTM file contains detailed inputs and results from the photolysis rate calculation done in-line by the CCTM. The number of layers is set by the NLAYS_PHOTDIAG environment variable (Default is all layers). The number of wavelengths included in the file is set by the NWAVE_PHOTDIAG environment variable (Default is all wavelengths). This file is only created if the CTM_PHOTDIAG environment variable in the RunScript is set to T (Default is N).

CCTM_SSEMIS: Sea salt emissions diagnostic file Return to Table 7-1

This optional 2-D CCTM hourly output file contains calculated sea salt emissions (g/s). This file is only created if the CTM_SSEMDIAG environment variable in the RunScript is set to Y (Default is N).

CCTM_WETDEP2: CCTM cloud diagnostics file Return to Table 7-1

In CMAQ, wet deposition is calculated separately for resolved (grid-scale) clouds and for convective (subgrid) clouds. The WETDEP1 file contains the total wet deposition, i.e., the sum of both resolved-scale and subgrid-scale deposition. The WETDEP2 file contains only subgrid-scale deposition, plus some cloud diagnostic variables. The 2-D CCTM wet deposition file (WETDEP2) includes cumulative hourly wet deposition fluxes (kg hectare-1) for selected model species. CCTM calculates wet deposition for all of the species listed in the wet deposition column of the Species NameLists files files within the mechanism directories. The GC_*mechname*.nml file lists the gas-phase species, the AE_*mechname*.nml file lists the aerosol species, and the NR_*mechname*.nml lists the nonreactive (inert) species. Species can be removed from the wet deposition file by editing the WDEP column in the NameList file(s). This file is only created if the CLD_DIAG environment variable in the RunScript is set to Y (Default is N).

CCTM_VEXT: file of vertical profiles of concentration at selected sites Return to Table 7-1

This optional 3-D CCTM file contains vertical profiles of the concentration of multiple chemical species for latitude / longitude coordinates specified in the VERTEXT_COORD_PATH file. The species written to this output file are identical to those written to the 3D CONC file which in turn are controlled either by the setting of CONC_SPCS in the RunScript or the last column in the GC, AE, NR, and TR namelist files. There is one row for each location specified. The coordinates for each location are echoed in the file metadata in the "history" field. This file is only created if the VERTEXT environment variable in the RunScript is set to Y (Default is N).

8. Analysis Tools for CMAQ output

8.1 Introduction

Many software programs are freely available for pre- and post-processing, evaluating and visualizing CMAQ outputs. Examples of such freeware are provided in Table 8-1. Several other commercial packages, including MATLAB and IDL, also support the analysis and visualization of CMAQ inputs and outputs. Most visualization and analysis software that support netCDF file formats will work with CMAQ outputs.

Table 8-1. Software Programs for Evaluating and Visualizing CMAQ Data

Software	Description	Source
Post-processing		
CMAQ POST Tools	Programs released with CMAQ source code to prepare output data for model evaluation	https://github.com/USEPA/CMAQ
I/O API Tools	Postprocessing tools for manipulating data in the I/O API/netCDF format	https://www.cmascenter.org/ioapi
NCO	netCDF Operators: Postprocessing tools for manipulating data in the netCDF format	http://nco.sourceforge.net
Evaluation/Visualization		
AMET	Atmospheric Model Evaluation Tool for analysis and evaluation of meteorological and air quality models	https://www.epa.gov/cmaq/atmospheric-model-evaluation-tool
VERDI	Visualization Environment for Rich Data Interpretation for graphical analysis of netCDF gridded data	https://www.cmascenter.org/verdi
PseudoNetCDF	Reading, plotting, and sometimes writing capabilities for atmospheric science data formats including CMAQ files	https://github.com/barronh/pseudonetcdf/wiki
RSIG	2D and 3D visualization of satellite and modeled data	https://www.epa.gov/hesc/remote-sensing-information-gateway
NCL	NCAR Command Language for scientific data processing and visualization	http://www.ncl.ucar.edu

Software	Description	Source
IDV	Integrated Data Viewer for 3-D graphical analysis of netCDF gridded data	http://www.unidata.ucar.edu/software/idv/

This chapter briefly describes how to use some of the software tools supported by the EPA and CMAS to aggregate CMAQ output, pair aggregated CMAQ output in space and time to air quality observations, create various evaluation plots, and visualize model fields.

8.2 Aggregating and Transforming Model Species

Many CMAQ output species need to be postprocessed to allow comparisons to measurements. For example, the CMAQ aerosol module explicitly represents a number of individual PM2.5 species that need to be combined for comparisons to measured total PM2.5 mass. The *combine* utility released as part of the CMAQ POST tools can be used to accomplish this task. Since the number and definition of model output species differs between chemical mechanisms, the *combine* utility relies on a mechanism-specific "Species Definition" file that prescribes how model output variables should be combined to become comparable to different measured gas, particle and deposition species. When you download the CMAQ code for version 5.2 or later, these definition files are automatically included under the subdirectory "CCTM/src/MECHS". Within each of the listed mechanism folders, you will find files "SpecDef_MECH_NAME.txt" and "SpecDef_dep_MECH_NAME.txt" that contain a long list of species definitions and corresponding documentation. For example, to find how to calculate PM2.5 using the cb6r3_ae7_aq mechanism, open the file "SpecDef_cb6r3_ae7_aq.txt" and read the documentation on PM2.5 calculations. The species definition file will indicate which species should be included in PM2.5 (for example: sulfate, ammonium, and organic carbon) as well as factors to obtain the fraction of each CMAQ size distribution mode that corresponds to 2.5 μm (diameter) and smaller particles. Similar information is available for calculating PM10 and PM1.0. These species definition files are designed to be used with the combine utility to extract model variables that match observed quantities from specific monitoring networks. More information on the combine utility and its use can be found in this README file.

8.3 Model-Observation Pairing for Model Evaluation

Once model output has been processed using *combine*, the *sitecmp* and *sitecmp_dailyo3* utilities can be used to match air pollutant measurements with the appropriate model predicted variables. This pairing of model and observed variables is specified in the run scripts for *sitecmp* and *sitecmp_dailyo3*. In *sitecmp_dailyo3* this step is controlled by the definition of environment variables OBS_SPECIES and OZONE. See the README.md and the sample run script in the *sitecmp_dailyo3* scripts folder for more information on setting these environment variables. The run script for the *sitecmp* utility can be customized for many different types of chemical and meteorological quantities as described in the README.md for sitecmp. Sample run scripts for the AQS, CSN, IMPROVE, NADP and SEARCH networks based on the 2016 CMAQ test case are provided in the *sitecmp* scripts folder. In addition, the

README.txt file within the *sitecmp* scripts folder provides the configuration options for monitoring networks. Note that there are multiple formats for CSN and SEARCH observed data files depending on the year. The README.txt file is broken into different sections to reflect the change in species names in the observation files for these two networks. (For example, elemental carbon measurements from the CSN network are labeled as "ec_niosh" in 2009 and earlier, "ec_tor" in 2010, and "88380_val" starting in 2011.)

8.3.1 Spatial matching in sitecmp and sitecmp_dailyo3

In *sitecmp*, model values are extracted for the grid cell containing the monitor location. In *sitecmp_dailyo3* the model value of the grid cell containing the observation is provided, as well as the maximum model value of the 9 grid cells centered on the monitor location. These variables in the output file contain the character string "9cell" in the variable name.

8.3.2 Temporal matching in sitecmp and sitecmp_dailyo3

- **AQS_HOURLY, CASTNET_HOURLY, SEARCH_HOURLY, NAPS_HOURLY, AERONET**: Air quality observations are assumed to be hourly averages time stamped at the beginning of the hour with local standard time (LST). The *sitecmp* utility will use the time stamp from the observations to determine the matching model time step, accounting for the time zone of the monitor. Therefore, best practice would be for the model time step to also represent hourly average time stamped at the beginning of the hour. This can be accomplished by running the *combine* utility on the CMAQ "ACONC" output files which follow this convention. These networks also include meteorological measurements. Since meteorological observations are near instantaneous measurements (e.g. 1- or 5-minute averages), using meteorological fields from MCIP or wrfout in *combine* results in the correct matching since these fields are also instantaneous. One exception is the calculation of modeled relative humidity (RH). This variable is not available from MCIP or wrfout files but is stored in the CMAQ "APMDIAG" output file which represents hourly average values. This creates a slight inconsistency between observed and modeled values for this variable in the sitecmp output files. Note that modeled and observed precipitation for a given hour represents the hourly total rather than the hourly average.
- **AQS_DAILY_O3, CASTNET_DAILY_O3, NAPS_DAILY_O3**: *sitecmp_dailyo3* computes various daily metrics from observed and modeled hourly ozone values. The temporal matching of the hourly observed and modeled values used in these computations follows the same approach described above for AQS_HOURLY. Therefore, it is best practice to use output from CMAQ "ACONC" files for modeled ozone predictions. Details on the computation of the various daily metrics is provided in the *sitecmp_dailyo3* documentation.
- **AQS_DAILY, CSN, IMPROVE, SEARCH_DAILY**: Air quality observations are daily averages time stamped with the date in local standard time. The *sitecmp* utility will use the date from the observations to compute daily averages using 24 hourly modeled values, accounting for the time zone of the monitor. Therefore, it is best practice to use output from CMAQ "ACONC" files for modeled air quality predictions which represent hourly average concentrations.

- **CASTNET**: Air quality observations are weekly averages time stamped with beginning and end date and time of the weekly interval in local standard time. The *sitecmp* utility will use the start and end date and time from the observations to compute weekly averages using hourly modeled values, accounting for the time zone of the monitor. Therefore, it is best practice to use output from CMAQ "ACONC" files for modeled air quality predictions which represent hourly average concentrations.
- **NADP**: Air quality observations are weekly sums time stamped with beginning and end date of the weekly interval in local standard time. The *sitecmp* utility will use the start and end date from the observations to compute weekly sums using hourly modeled values, accounting for the time zone of the monitor. Observations are matched to output from CMAQ "DEP" files which represent hourly totals.
- **TOAR**: Air quality observations are daily average values of O3, MDA8 O3, O3 daytime average and O3 nighttime average. The *sitecmp* utility must be given daily average values computed from hourly values using the *hr2day* utility.

8.4 The Atmospheric Model Evaluation Tool (AMET)

The Atmospheric Model Evaluation Tool (AMET) was developed to aid in the evaluation of the meteorological and air quality models within the CMAQ modeling system (i.e. WRF, MPAS, CMAQ-CTM). AMET organizes, provides consistency and speeds-up the evaluation process for operational meteorological and air quality model simulations. The AMET software is written primarily in R, with support from several fortran programs and cshell scripts. The tool also requires the presence of a MySQL database for analysis of meteorological data and full functional analysis of air quality (CMAQ) data (analysis of CMAQ output can be done without a database present). Although it was developed specifically to aid in the evaluation of the CMAQ modeling system, the AMET software can be adapted to work with other modeling systems.

There are separate modules in AMET for evaluating meteorological and air quality model output. This separation is necessary because both the observed and predicted meteorological and air quality data are quite different, utilizing different file formats for both the observed and model data. In addition, the observed meteorological and air quality data are often obtained from networks that use different sampling protocols, which can make pairing meteorological and air quality data together difficult. One advantage of separate meteorological and air quality modules in AMET is that the modules can be installed individually, allowing a user to reduce installation time and complexity if only meteorological or air quality analysis is required.

A more detailed description of AMET can be found at https://www.epa.gov/cmaq/atmospheric-model-evaluation-tool, including a flow diagram of the AMET system and example output plots from the tool. The AMET github repository resides at https://github.com/USEPA/AMET. The repository includes the latest version of AMET, along with a complete description of the tool, a comprehensive User's Guide, an Installation Guide, and a Quick Start Guide. Finally, additional information regarding AMET (including how to download AMET-ready observation data files) can be found on the CMAS Center website: https://www.cmascenter.org/amet/.

The AMETv1.4 repository includes a new script to set up and execute multiple post-processing steps from a single file, including running *combine*, *sitecmp*, *sitecmp_dailyo3*, and "batch" AMET evalua-

tion plots. After installing AMET, users can find this script under scripts_db/aqExample/aqProject_pre_and_post.csh. Documenation for configuring this master evaluation script are provided in the AMET docs folder: AMET_aqProject_Pre_and_Post_Analysis_Script_Guide_v14b.md

8.5 Visualization Environment for Rich Data Interpretation (VERDI)

The Visualization Environment for Rich Data Interpretation (VERDI) is a visual analysis tool for evaluating and plotting multivariate gridded results from meteorological and air quality models. VERDI is written in Java, so it can be run on a variety of computer operating systems; VERDI packages are currently released for Linux, Windows, and Mac. In addition to supporting the CMAQ modeling system, VERDI also currently supports analysis and visualization of model results from the regional Weather Research and Forecasting (WRF) model, the global Model for Prediction Across Scales (MPAS), the Meteorology-Chemistry Interface Processor (MCIP), and the Comprehensive Air Quality Model with Extensions (CAMx). In addition, VERDI can read and overlay observational data at monitoring site locations to visually compare model results to observations, both spatially and temporally.

VERDI's interactive graphical user interface (GUI) allows for quick examination of model results, while the command line scripting capability in VERDI can be used for more routine analysis and plot production. Supported input data formats include I/O API, netCDF, and UAM-IV from models and ASCII text, I/O API, and netCDF for observational data sets. Supported map projections include Lambert conformal conic, Mercator, Universal Transverse Mercator, and polar stereographic. Once data are loaded into VERDI, individual selected variables can be plotted or utilized as inputs to mathematical formulas which can then be plotted. Available plot types include spatial tile, areal interpolation based on shapefiles, vertical cross section, times series, time series bar, scatter, and 3-D contour plots. Plots can then be enhanced with overlays of observations from monitoring sites, wind vectors, grid lines/cell boundaries, and additional GIS layers, such as boundaries for states, counties, HUCs (hydrologic unit codes), rivers, roads, and user-defined shapefiles. Plotting of variables can be limited to specified spatial and/or temporal ranges, with minimum/maximum values for the variable for the displayed area and time automatically shown at the bottom of each plot frame. Plots can be saved as raster images (BMP, JPEG, PNG, TIFF) of a chosen pixel size, vector images (EPS), or animated GIF "movies." Areal ESRI-compatible shapefiles and ASCII text or comma-separated-values can also be exported. Interactive analysis is aided with the ability to quickly zoom into areas of interest and to probe data values within a grid cell. To facilitate plot reproducibility, VERDI can save the session as a project file and the customization of each plot (e.g., data range, color palette, font characteristics, titles, and labels) as a plot configuration file. Plus, quick statistical analysis of the displayed data is easily accomplished by using VERDI's built-in algorithms for minimum/maximum, mean, geometric mean, median, first and third quartiles, variance, standard deviation, coefficient of variance, range, interquartile range, sum, timesteps of minimum and maximum, hours of non-compliance, maximum 8-h mean, count, fourth max, and custom percentiles.

The CMAS Center currently hosts VERDI at https://www.cmascenter.org/verdi, providing a brief description with links to download VERDI and its documentation. The main code repository for VERDI resides at https://github.com/CEMPD/VERDI where users can download the latest release, peruse the documentation, and note the latest known issues and bugs.

9. Process Analysis

9.1 Introduction

Most applications of CMAQ, as well as other 3-D grid models, output concentration fields of chemical species of interest at selected time steps that reflect the cumulative effect of all processes (emissions, chemical reaction, transport, etc.) that act on the chemical species over the time period. While concentrations are useful per se, knowing only the net result of all processes can limit the understanding of *why* the concentrations are the levels that are calculated. For some applications, the user may want to unravel this net impact and examine the quantitative impact of the individual processes, to identify those which are most important or uncertain. Process Analysis (PA) is a technique for separating out and quantifying the contributions of individual physical and chemical processes to the changes in the predicted concentrations of a pollutant. PA does *not* have to be activated in a CMAQ simulation but including PA in a simulation during runtime provides additional information that can be useful in interpreting CMAQ results. PA has two components: Integrated Process Rate (IPR) analysis, and Integrated Reaction Rate (IRR) analysis. IPR analysis quantifies the net change in species through physical processes of advection, diffusion, emissions, dry deposition, aerosol processes, and cloud processes, and the overall impact of chemical processes. IRR analysis allows the output of individual chemical reaction rates or user-specified combinations of chemical reactions and species cycling.

As a tool for identifying the relative importance of individual chemical and physical processes, PA has many applications, including:

- Quantifying major contributors to the concentration of a chemical species at a grid cell. PA can be used to split out the contributions of multiple, complex processes that control species concentrations. PA is useful for species that have both production and decay processes occurring in the same time step, including cases where the final concentration may show little change, but individual decay and production rates may be large.
- Characterizing the chemical state of a particular grid cell. PA with IRR can be used to calculate quantities such as the production of odd oxygen, the production of new radicals, the ozone production efficiency and the termination of radicals. (For example, see Tonnesen and Dennis, 2000.)
- Aiding model development. PA can help predict and evaluate the effect of modifications made to a model or process module.
- Identifying compensating or unresolved errors in the model or input data which may not be reflected in the total change in concentration. For example, if an error in the emissions input data causes the model to calculate negative concentration values in an intermediate step, this could be masked in the final predicted concentrations if compensated for by larger positive values resulting from the chemistry calculations.

PA variables are computed by saving the differential operators associated with each process or reaction, integrated over the model synchronization time step for the same variables that are used in solving

the continuity equations within the model. For processes that are solved simultaneously in the same operator, PA uses mass balance to compute the contribution of each process.

A user activates PA during CMAQ runtime and includes a PA input file to specify whether IPR, IRR or both analyses are performed, and defining what variables are required for each analysis. The IRR parameters are highly customizable and can be easily modified but must be checked carefully before running the model to ensure that they correspond to the mechanism being used. In CMAQ v5.2 and earlier versions, IRR could only be run when CMAQ was compiled with the Rosenbrock (ros3) or Sparse-Matrix-Vectorized Gear (smvgear) solvers. However, in CMAQv5.3, the code was updated to enable use of IRR with the ebi solver, and to implement enhanced output of aerosol processes. The derivation of PA and format of input files specific to CMAQ incorporation is detailed in Gipson et al., (1999), and a further description of the science behind PA is provided in Tonnesen (1994) and Jeffries and Tonnesen (1994).

9.2 Use of Process Analysis

Step 1: Activate Process Analysis and specify control files

- setenv CTM_PROCAN Y:N

Set this variable to Y to indicate that you want process analysis to be activated. The default is N. If this is set to Y, then you must also specify the following two files:

- setenv PACM_INFILE [filename]
- setenv PACM_REPORT [filename]

PACM_INFILE is the input file that specifies the desired output information (read by pa_read.F). Gipson et al., (1999), details the types of equations and operators that can be used, with a brief summary here in Table 1. PACM_REPORT is the output file that displays how CMAQ translates the variables listed in PACM_INFILE, and lists the reactions (including reactants, products and yields) that will be used in calculating the IPR and IRR values. Users should check this file on the first iteration of a new PA simulation to ensure that CMAQ is interpreting the variables as the user intended.

The user can also specify an optional subdomain for the IPR/IRR output. If these variables are not specified, the default domain is the entire CMAQ domain, however, the user may want to limit the portion of the domain where output is written because the files can get large. This is done using the variables:

- setenv PA_BCOL_ECOL “[start] [end]”
- setenv PA_BROW_EROW “[start] [end]”
- setenv PA_BLEV_ELEV “[start] [end]”

where integers [start] and [end] are the starting and ending grid columns, rows, or vertical levels. These optional variables are used to specify the modeling grid domain column range, row range, and layer range for the process analysis calculations. Set to the two digits representing the start and end columns, rows and layer numbers bounding the process analysis domain. The user must be careful that the columns, rows, and levels are not outside of the current CMAQ domain.

Step 2: Ensure that output files are being written and that the output variables have been correctly specified

A PACM_REPORT file, with the name specified in Step 1, is output for every day of simulation, along with daily IRR or IPR files, depending on whether IRR or IPR was specified. If there is a formatting error in the PACM_INPUT file, CMAQ will not run and the CMAQ log files must be checked to determine where the error occurred. The PACM_REPORT file will list the reactions that are used to interpret each of the reactions/families/cycles/operators specified by the user in the PACM_INPUT file. For complex operations (such as those including families or cycles), the user must ensure that the output conveys the appropriate quantities.

The output files are specified in the CMAQ runscript by:

- setenv CTM_IPR_1 [filename] (….similarly for CTM_IPR_2 and CTM_IPR_3)
- setenv CTM_IRR_1 [filename] (….similarly for CTM_IRR_2 and CTM_IRR_3)

IPR files are only created if IPR is turned on in the PACM_INFILE (IPR_OUTPUT specified), and IRR files are only created if IRR is specified (IRRTYPE = PARTIAL or FULL). The number of output files created (whether 1,2 or 3) depends on the number of variables specified; for example, only one will be created for the sample input file delivered as part of the CMAQ release.

Step 3: Post process output files

The output files are in the same units as the concentration files and can be post-processed using the same utilities used to post-process the CMAQ concentration files. In particular the following utilities may be helpful:

- combine (to combine multiple days in one file, to match density or layer height with IPR or IRR variables)
- m3tproc (to sum up throughput over multiple days)
- vertot (to sum up throughput over several layers, such as the PBL)
- verdi (to view spatial heterogeneity in process throughput)

9.3 Description of the PACM_INFILE

The PA input file (PACM_INFILE) is the user-tailored file that controls the parameters that are calculated and output at each time step. Depending on the specificity of the output, the file will need to be tailored to the chemical mechanism used in the simulation, because species names and reaction numbers vary among different mechanisms. For example, components of oxidized nitrogen, such as organic nitrates, are represented by species NTR1 + NTR2 + INTR in CB6, but by RNO3 in SAPRC07.

In addition, if IRR outputs are specified by label, the user must ensure that the labels are appropriate for the mechanism being used.

The user can define families of similar pollutants, specify cycles, and reaction sums that can be used in subsequent IPR and IRR equations, which can simplify the specification of quantities. DEFINE FAMILY is useful when the user wants to follow the sum of several different species, for example:

- DEFINE FAMILY NOX = NO + NO2

This will allow the user to specify operations of both NO and NO2 by using the user-specified family name NOX. Cycles are important because many species have reactions in which they decay and reform quickly. In some cases, the production and loss terms may both be large and obscure the information that is desired.

- DEFINE FAMILY PANcyc = PAN

A user-specified name, PANcyc, can be used in place of PAN in further operations, to remove the effect of rapid reactions that recycle PAN rapidly. Cycles (as well as other quantities) can also be defined with the RXNSUM statement:

- DEFINE RXNSUM CLNO3cyc = <CL28> - <CL30>

This will store the net throughput of the reaction labeled CL28 minus reaction labeled CL30 (i.e. in CMAQ-CB6, the net production of species CLNO3) in a user-specified variable named CLNO3cyc.

Table 9-1. Parameters used in PACM_INFILE

First string	Second string	Third string	Remainder of line
DEFINE FAMILY	[descriptor]	=	list of chemicals separated by +
DEFINE CYCLE	[descriptor]	=	chemical name
DEFINE RXNSUM	[descriptor]	=	list of reactions separated by +
IPR_OUTPUT	[chemical/family name]	=	physical process names and/or CHEM
IRR_OUTPUT	FULL:PARTIAL:NONE		
IRR_OUTPUT	[descriptor]	=	reaction label, combination of species reactions, etc.

9.4 Parameters for IPR

Each line for IPR output begins with IPR_OUTPUT, followed by the chemical species or the species family for which output is desired and the processes to be output. If the processes are omitted, then the default is all processes. The available processes are listed in Table 9-2. In the sample file, for example:

IPR_OUTPUT O3 = CHEM+DDEP+CLDS+AERO+TRNM;

specifies that the output includes the change in species O3 over the time step for the net sum of all chemistry processes, the net dry deposition, the net change in concentration due to clouds, aerosol processes, and total transport. If the species is a family name instead of a species name, the outputs will be calculated for the sum of each species in the family.

Table 9-2. Allowable parameters for process outputs

LPROC	process	Description
1	ZADV	vertical advection
2	HADV	horizontal advection
3	HDIF	horizontal diffusion
4	VDIF	vertical diffusion
5	EMIS	emissions contribution to concentration
6	DDEP	dry deposition of species
7	CLDS	change due to cloud processes; includes aqueous reaction and removal by clouds and rain
8	CHEM	net sum of all chemical processes for species over output step
9	COND	change in aerosol species due to condensation
10	COAG	change in aerosol species due to coagulation
11	NPF	change in aerosol species due to new particle formation
12	GROW	change in aerosol species due to aerosol growth
9+10+11+12	AERO	change due to aerosol processes
1+2	MADV	Horizontal and vertical advection
3+4	TDIF	Total diffusion of species
1+2+3+4	TRNM	Total mass-conserving transport of species

9.5 Parameters for IRR

The specification for parameters in IRR output begins with IRR_OUTPUT followed by a user-defined name for the quantity and an equation specifying how it is to be calculated. The operators used in constructing these equations are explained in more detail in Gipson et al. (1999), and a brief summary of the allowable operators is included in Table 9-3. The equation could include a reaction label or an operator for a chemical species or family. For example:

IRR_OUTPUT NewClrad = 2.0*<CL1> + <CL2> + <CL8> + <CL25>

would sum the throughput of reactions labeled CL2, CL8, and CL25 and 2 times the throughput of reaction labeled CL1. The sum for each time step and each grid would be in the variable NewCLrad defined by the user.

Table 9-3. Allowable operators for Integrated Reaction Rate outputs

Operator	Description
PROD[x] {FROM [y] {AND/OR [z]}}	sum of throughput from all reactions where species (or family) x is a product. Optional qualifiers limit to reactions were species y or z are reactants
NETP[x] {FROM [y] {AND/OR [z]}}	similar to PROD but only uses reactions where net production is greater than zero
LOSS[x] {AND/OR[y]}	All loss of species/family x; could be limited to only reactions where both x and y or x or y are lost
NETL[x] {AND/OR [y]}	similar to LOSS but only uses reactions where net loss of x {and/or y} is greater than zero
NET[x]	the net of the production and loss for all reactions in which x is a product or reactant
cyclename[POSONLY:NEGONLY]	calculates the net impact of a cycle defined earlier. Using optional qualifiers will output values only if the net is positive or negative.

9.6 Example IRR applications

IRR can be endlessly customized to examine many different processes and combinations of processes. Below are two examples of how IRR can be used.

Example 1: Examine fate of VOC reactions. In this example, we have defined a family “aVOC” to be the sum of all (mostly) anthropogenic VOCs used in CMAQ-CB6 (Luecken et al., 2019), and then quantified the loss of the family through reaction with the oxidants OH, O3 and NO3:

- DEFINE FAMILY aVOC = FORM +ALD2 +ALDX +PAR +ETHA +PRPA +MEOH +ETOH +ETH +OLE +IOLE +ACET +TOL +XYLMN +GLYD +GLY +MGLY +KET +CRON +NTR1 +OPEN +CAT1 +CRES +XOPN +NAPH + TO2 + BZO2 + XLO2 + ROR + XPRP + ROOH;
- IRR_OUTPUT aVOCwithOH = LOSS[aVOC] AND [OH];
- IRR_OUTPUT aVOCwithO3 = LOSS[aVOC] AND [NO3];
- IRR_OUTPUT aVOCwithNO3 = LOSS[aVOC] AND [O3];

We have summed the throughputs over 2 weeks (July 1-14, 2011), within the first level of the model, and compared these three output pathways at four locations throughout the U.S. Figure 9-1 shows the sum through each of these processes at 6 different grid areas, including 4 urban areas and 2 rural areas. This figure also includes the corresponding fate for the largely biogenic VOCs, although note that some VOCs, such as formaldehyde and ethanol can be both anthropogenic and biogenic.

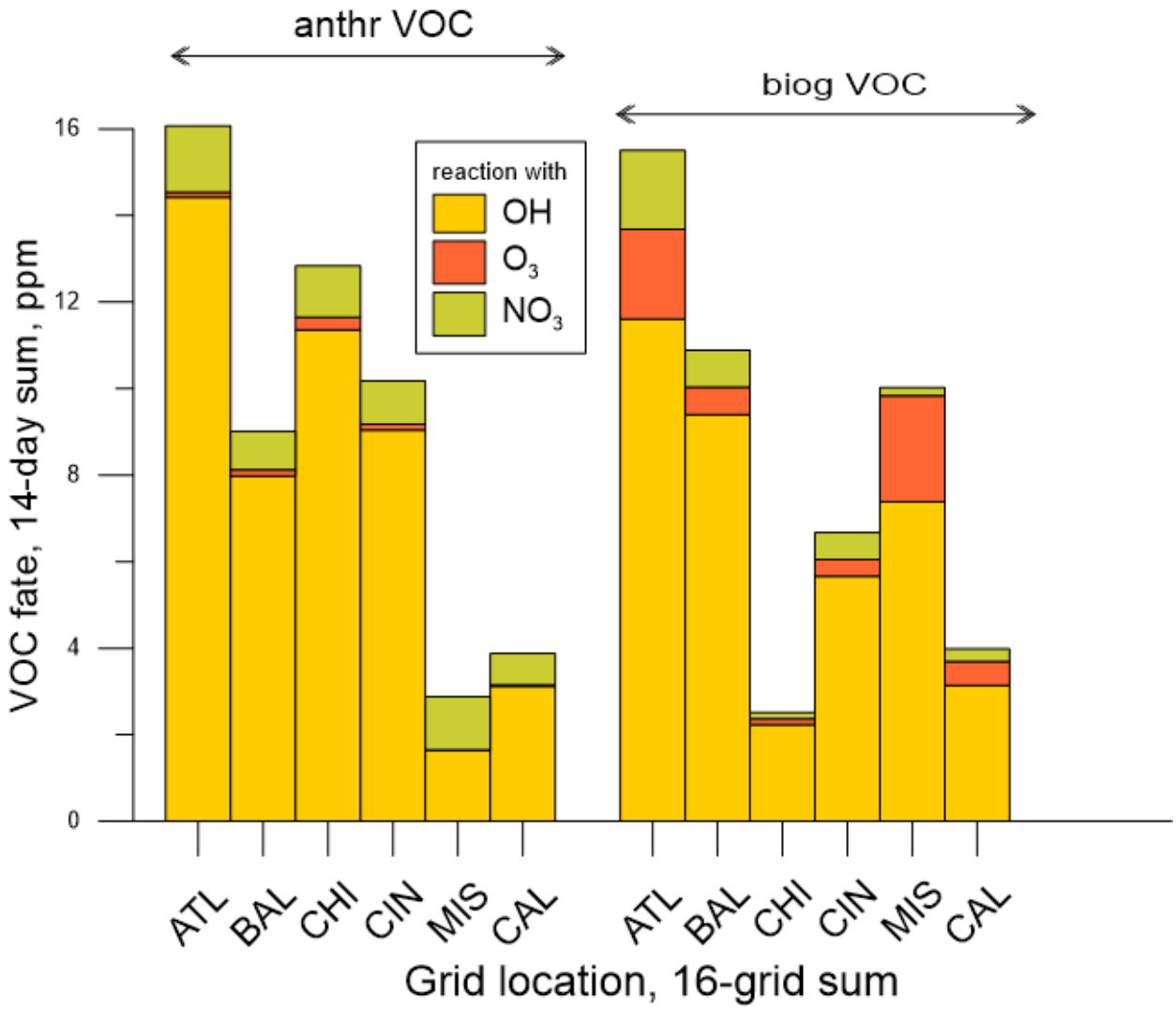


Figure 9-1. Relative contribution of oxidation pathways for VOCs using Process Analysis

Example 2: Quantify the major contributors to the production of HNO3. In this example, we develop output variables to represent the total production of HNO3 in CMAQ-CB6 and the individual reactions which contribute to this total production:

- IRR_OUTPUT HNO3prod = PROD[HNO3];
- IRR_OUTPUT HNO3fromOHNO2 = <R45>;
- IRR_OUTPUT HNO3fromhetNTR = ;
- IRR_OUTPUT HNO3fromN2O5 = + + +;
- IRR_OUTPUT HNO3fromNO3 = PROD[HNO3] FROM [NO3];
- IRR_OUTPUT HNO3frCLNO3= ;

In this case, we have summed up all throughput over the first 15 levels of the model (approximating the PBL) and over the same 14-day time period. Figure 9-2 shows the relative contribution of processes to the total HNO3 formation at three grids. In this case, at the two more urban grids, the reaction of OH+NO2 dominates the formation of HNO3 in summer, while at the rural grid cell (Missouri), the heterogeneous hydrolysis of alkyl nitrates is predominant.

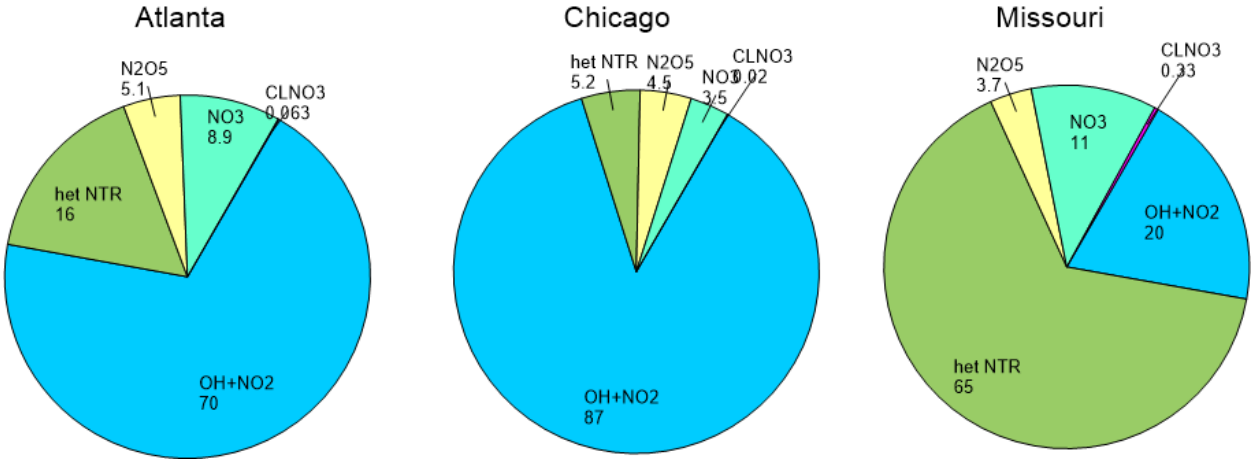


Figure 9-2. Relative contribution of HNO3 formation pathways at three grid locations

9.7 References

Gipson, G.L. (1999). Chapter 16: Process analysis. In science algorithms of the EPA models-3 Community Multiscale Air Quality (CMAQ) Modeling System. EPA/600/R-99/030.

Jeffries, H. E., & Tonnesen, S. (1994). A comparison of two photochemical reaction mechanisms using mass balance and process analysis. Atmos. Env., 28(18), 2991-3003.

Luecken, D.J., Yarwood, G., & Hutzell, W.H. (2019). Multipollutant modeling of ozone, reactive nitrogen and HAPs across the continental US with CMAQ-CB6. Atmospheric Environment, 201, 62-72.

Tonnesen, S., & Jeffries, H.E. (1994). Inhibition of odd oxygen production in the carbon bond four and generic reaction set mechanisms. Atmospheric Environment, 28(7), 1339-1349.

10. Decoupled Direct Method in Three Dimensions (CMAQ-DDM-3D)

10.1 Introduction

The Decoupled Direct Method in Three Dimensions (DDM-3D) provides CMAQ concentration and deposition sensitivity information for user specified model parameters.

In air quality modeling, sensitivities measure the response of a model output to a change in one or several predefined model parameters. In policy applications, the parameters of interest are usually emissions and the output of interest is pollutant concentrations. We may be interested in emissions from a particular geographical region, like an urban area, a group of states, or a country, and/or emissions from a particular source, such as wildfires, electricity generating units (EGUs), or light duty diesel trucks.

Emissions sensitivities can be calculated by simply running the air quality model twice – once with standard emissions inputs, and once with the emissions of interest adjusted in some way. The difference in outputs between the two runs in relation to the size of the adjustment then becomes the model sensitivity. While this process is fairly easy to implement and interpret, it quickly becomes computationally complex as the number of desired sensitivities increases. For example, calculating sensitivity to EGU emissions from 10 southeastern states in the U.S. would require 11 separate air quality model simulations.

Alternatively, model sensitivities can be calculated with CMAQ-DDM-3D. This is done by altering the existing model algorithms to allow for sensitivity propagation through every science module in CMAQ. While this process does require more computational resources than standard CMAQ, it scales favorably with the number of desired parameters.

Besides emissions, sensitivities to other model parameters can also be calculated. Currently, CMAQ-DDM-3D can be used for sensitivity to emission rates, boundary conditions, initial conditions, reaction rates, potential vorticity, or any combination of these parameters. Second order sensitivity calculations, or sensitivity of sensitivity, sometimes known as higher-order DDM-3D (HDDM-3D) are also available. Note: second order sensitivity outputs for particulate matter species are still in development and should be considered as a research option.

10.2 CMAQ-DDM-3D Releases

Current CMAQ-DDM-3D implementation is available for version 5.2 of the Community Multiscale Air Quality (CMAQ) model.

- Link to CMAQv5.2 DDM-3D source code and scripts
- Direct download to CMAQv5.2 DDM-3D source code and scripts
- Documentation for CMAQv5.2 DDM-3D

The migration of DDM-3D to the more recent CMAQ release is currently in development and will be included sometime after the base model release. This documentation will be updated at that time.

A note about I/O API installation for DDM applications

I/O APIv3.2 supports up to MXFILE3=64 open files, each with up to MXVARS3=2048. DDM applications configured to calculate sensitivity to a large number of parameters may exceed this upper limit of model variables, leading to a model crash. To avoid this issue, users may use I/O API version 3.2 "large" that increases MXFILE3 to 512 and MXVARS3 to 16384. This version is available as a zip file from the following address:

https://www.cmascenter.org/ioapi/download/ioapi-3.2-large.tar.gz

Installation instructions for I/O API v5.3-large are provided in README.txt in the .tar.gz file.

10.3 References

Cohan, D.S., & Napelenok, S.L. (2011). Air Quality Response Modeling for Decision Support. Atmosphere, 2(3), 407-425. doi: 10.3390/atmos2030407

Napelenok, S.L., Cohan, D.S., Odman, M.T., & Tonse, S. (2008). Extension and evaluation of sensitivity analysis capabilities in a photochemical model. Environmental Modelling & Software, 23(8), 994-999. doi: 10.1016/j.envsoft.2007.11.004

Napelenok, S.L., Cohan, D.S., Hu, Y.T., & Russell, A.G. (2006). Decoupled direct 3D sensitivity analysis for particulate matter (DDM-3D/PM). Atmospheric Environment, 40(32), 6112-6121. doi: 10.1016/j.atmosenv.2006.05.039

Contact

Sergey L. Napelenok, Computational Exposure Division, U.S. EPA

11. Integrated Source Apportionment Method (CMAQ-ISAM)

11.1 Introduction

The Integrated Source Apportionment Method (ISAM) calculates source attribution information for user specified ozone and particulate matter precursors within the CMAQ model. CMAQ-ISAM has been substantially updated in the CMAQv5.3 release.

The CMAQ model provides users the concentration and deposition fields of many pollutant species. These species are usually combinations of different types of primary emissions and secondary formation that have been physically and chemically transformed in the model. However, sometimes it is desirable to know specific source attribution information for the model outputs. For example, how much of the ozone in an urban area was formed due to nitrogen oxides emitted from motor vehicles in a neighboring state?

Answering this type of question often requires running an air quality model twice, once with the standard emissions scenario and once with the source of interest completely removed. The difference between the two runs is then assumed to be attributed to the removed source. While this approach is reasonably straightforward to implement, it has some drawbacks. For example, removing a large

source from the system in a highly nonlinear chemical mixture can lead to some errors. Also, calculating source attribution of many sources can be logistically and computationally prohibitive.

Alternatively, running CMAQ with ISAM enabled allows the user the ability to calculate source attribution of a large number of sources directly by the model in one simulation.

Note: While full model species list apportionment is in development, currently ISAM is limited to the following species classes in CMAQ:

```
SULFATE   - ASO4J, ASO4I, SO2, SULF, SULRXN
NITRATE   - ANO3J, ANO3I, HNO3, ANO3J, ANO3I, HNO3, NO, NO2, NO3, HONO, N2O5, PNA, PAN, PANX, N
AMMONIUM  - ANH4J, ANH4I, NH3
EC        - AECJ, AECI
OC        - ALVPO1I, ALVPO1J, ASVPO1I, ASVPO1J, ASVPO2I, ASVPO2J, ASVPO3J, AIVPO1J, VLVPO1, VSV
VOC       - 21 species in CB6R3 (see OZ_DEFN.F)
PM25_IONS - ACLI/J,ANAI/J,AMGJ,AKJ,ACAJ,AFEJ,AALJ,ASIJ,ATIJ,AMNJ,AOTHRI/J
CO        - CO
OZONE     - all NITRATE species + all VOC species
```

11.2 Build Instructions

Starting with CMAQv5.3 model release, ISAM is provided directly with the source code of the base model. To use ISAM, follow the normal build process for CMAQ described in Chapter 5 but make sure to uncomment the following line in bldit_cctm.csh:

```
set ISAM_CCTM
```

A note about I/O API installation for ISAM applications

I/O APIv3.2 supports up to MXFILE3=64 open files, each with up to MXVARS3=2048. ISAM applications configured to calculate source attribution of a large number of sources may exceed this upper limit of model variables, leading to a model crash. To avoid this issue, users may use I/O API version 3.2 “large” that increases MXFILE3 to 512 and MXVARS3 to 16384. This version is available as a zip file from the following address:

https://www.cmascenter.org/ioapi/download/ioapi-3.2-large.tar.gz

Installation instructions for I/O API v5.3-large are provided in README.txt in the .tar.gz file.

11.3 Run Instructions

To begin a CMAQ simulation with source apportionment enabled, the ISAM section of the runscript must be configured. The additional necessary environment variables are listed in Table 11-1.

Table 11-1. ISAM run script variables

Variable	Settings	Description
CTM_ISAM	Y/N	Set this to Y to enable ISAM
SA_IOLIST	path/filename	Provide the location of the ISAM control file (discussed below)
ISAM_BLEV_ELEV	" MINVALUE MAX VALUE "	LAYER range for the instantaneous ISAM output concentrations
AISAM_BLEV_ELEV	" MINVALUE MAX VALUE "	LAYER range for the average ISAM output concentrations
ISAM_NEW_START	Y/N	set Y for a new simulation and N for continuing from a previous day's outputs
ISAM_PREVDAY	path/filename	Provide the location of the previous day's ISAM restart file
SA_ACONC_1	path/filename	ISAM output for average apportioned concentrations
SA_CONC_1	path/filename	ISAM output for instanteneous apportioned concentrations
SA_DD_1	path/filename	ISAM output for apportioned dry deposition
SA_WD_1	path/filename	ISAM output for apportioned wet deposition
SA_CGRID_1	path/filename	ISAM output for a restart file to continue the simulation further in time

Additionally, ISAM can track emissions confined to geographic regions. This functionality can be enabled through CMAQ's `RegionsRegistry` set in the `EmissCtrl` namelist (Appendix B.4) and is discussed further below.

11.3.1 ISAM control file (SA_IOLIST)

The ISAM `SA_IOLIST` is a text file used to configure which tag classes, emissions streams, and source regions the model will track. An example of this file, `isam_control.txt`, is provided in $CMAQ_HOME/CCTM/scripts. The formating of this file must be kept intact, but it does allow for insertion of comment lines.

Each ISAM simulation requires the specification of the `TAG CLASSES` that the user desires to apportion. The current list includes the following choices `SULFATE, NITRATE, AMMONIUM, EC, OC, VOC, PM25_IONS, CO, OZONE`. Species associated with each of these are provided in section 11.1. One or more of these tag classes must be specified in `SA_IOLIST`. Multiple tag classes are comma delimited.

```
TAG CLASSES     |OZONE, SULFATE
```

After setting tag classes for the simulation, information for one or more tags is required. Each individual tag will track the species from the specified `TAG CLASSES` and has its own set of three options in the control file. The first option is the name:

```
TAG NAME        |EGU
```

It is recommended that the text string for the tag name be kept short (ideally three characters) in order to accommodate the longer species names from some chemical mechanisms in the ISAM output files.

The second option is the comma delimited list of regions to track with this tag. The keyword 'EVERYWHERE' is used to track domain-wide emissions. To track region-constrained emissions, variable names from the regions file specified in the `EmissCtrl` namelist are used instead of the "EVERYWHERE' keyword. The regions file requirements are identical to the optional file used to scale emissions in predetermined geographical areas (Appendix B.4).

```
REGION(S)       |EVERYWHERE
```

or

```
REGION(S)       |NC, SC, GA
```

Finally, the emissions streams labels are required as the third option in the control file. These are the labels set in the runscript for the base CMAQ simulation.

```
EMIS STREAM(S)  |PT_EGU, PT_NONEGU
```

The final line in the control file needs to be kept unchanged in order to aid the file parser in reading this file.

```
ENDLIST eof
```

In addition to the user-specified list, ISAM will alway track and output three additional default tags with every simulation (note, that at least one user-specified tag must be defined):

```
ICON - contribution from initial conditions specified for the first day of the simulation
BCON - contribution form boundary condtions throughout the simulation
OTHR - contribution from all emissions not tagged by the user through isam control file.
```

11.4 ISAM Benchmark data

The input files for the CMAQv5.3 ISAM benchmark case are the same as the benchmark inputs for the base model, described in the CMAQ Benchmark Tutorial. Output source apportionment files associated with the sample `isam_control.txt` provided in this release package are included in the benchmark outputs for the base model.

The CMAQ benchmark data can be downloaded from the CMAS Center Data Warehouse SE53BENCH Google Drive folder. The CMAQ benchmark test case is a two day simulation for July 1-2 2016 on a 100 column x 80 row x 35 layer 12-km resolution domain over the southeast U.S. Input and output files for a two week case covering July 1-14, 2016 are also available within the same Google Drive folder.

- Metadata for the CMAQ benchmark inputs: https://doi.org/10.15139/S3/IQVABD
- Metadata for the CMAQ benchmark outputs: https://doi.org/10.15139/S3/PDE4SS

11.5 References

Kwok, R.H.F, Napelenok, S.L., & Baker, K.R. (2013). Implementation and evaluation of PM2.5 source contribution analysis in a photochemical model. Atmospheric Environment, 80, 398–407 doi:10.1016/j.atmosenv.2013.08.017.

Kwok, R.H.F, Baker, K.R., Napelenok, S.L., & Tonnesen, G.S. (2015). Photochemical grid model implementation of VOC, NOx, and O3 source apportionment. Geosci. Model Dev., 8, 99-114. doi:10.5194/gmd-8-99-2015.

Contact

Sergey L. Napelenok, Computational Exposure Division, U.S. EPA

12. Sulfur Tracking Method

12.1 Introduction

CMAQv5.3 includes a runtime diagnostic model option that provides detailed information on the modeled sulfur budget. This option, referred to as the "Sulfur Tracking Method (STM)", tracks sulfate production from gas- and aqueous-phase chemical reactions, as well as contributions from emissions and initial and boundary conditions. Each tracked species is treated as other modeled species, undergoing transport (advection, diffusion, cloud-mixing) and removal by deposition (both wet and dry). Several notable features in the CMAQv5.3 release of STM include:

- The STM is now a runtime option enabled by an environment variable.
- Additional species (Table 12-2) are included to track the loss of inorganic sulfate to organosulfate for chemical mechanisms that include this loss pathway.

12.2 Usage

To activate the STM option, edit the CTM runscript and set the following environment variable to "Y" (the default is "N"):

- setenv STM_SO4TRACK Y

The STM option does not require any additional input files, and uses the initial conditions, boundary conditions, and emissions files available for with the standard, non-instrumented CCTM.

Next, run the CMAQ CTM following the instructions described in Chapter 5, section 5.7.

Note that several of the standard CMAQ output files (ACONC, CONC, CGRID, DDEP, and WDEP) will include additional species beyond the standard base model species list. A list of the additional species output by the STM option are provided in Table 12-1 and Table 12-2. These data can be post-processed using standard utilities, such as:

- combine (to combine multiple days into one file or to aggregate various tracking species)
- m3tproc (to sum/average over multiple days)
- verdi (for data visualization)

Table 12-1. Sulfur Tracking Species List

Species Group	Species Name	MW	Description
AE	ASO4AQH2O2J	96.0	Accumulation mode sulfate (ASO4J) produced by aqueous-phase hydrogen peroxide oxidation reaction: H2O2 + S(IV) -> S(VI) + H2O
AE	ASO4AQO3J	96.0	ASO4J produced by aqueous-phase ozone oxidation reaction: O3 + S(IV) -> S(VI) + O2
AE	ASO4AQFEMNJ	96.0	ASO4J produced by aqueous-phase oxygen catalyzed by Fe3+ and Mn2+ oxidation reaction: O2 + S(IV) -> S(VI)
AE	ASO4AQMHPJ	96.0	ASO4J produced by aqueous-phase methyl hydrogen peroxide oxidation reaction: MHP + S(IV) -> S(VI)
AE	ASO4AQPAAJ	96.0	ASO4J produced by aqueous-phase peroxyacetic acid oxidation reaction: PAA + S(IV) -> S(VI)
AE	ASO4GASJ	96.0	ASO4J condensation following gas-phase reaction: OH + SO2 -> SULF + HO2
AE	ASO4EMISJ	96.0	ASO4J from source emissions
AE	ASO4ICBCJ	96.0	ASO4J from boundary and initial conditions

Species Group	Species Name	MW	Description
AE	ASO4GASI	96.0	Aitken mode sulfate (ASO4I) nucleation and/or condensation following gas-phase reaction: OH + SO2 -> SULF + HO2
AE	ASO4EMISI	96.0	ASO4I from source emissions
AE	ASO4ICBCI	96.0	ASO4I from boundary and initial conditions
AE	ASO4GASK	96.0	Coarse mode sulfate (ASO4K) condensation following gas-phase reaction: OH + SO2 -> SULF + HO2
AE	ASO4EMISK	96.0	ASO4K from source emissions
AE	ASO4ICBCK	96.0	ASO4K from boundary and initial conditions
NR	SULF_ICBC	98.0	Sulfuric acid vapor (SULF) from boundary and initial conditions

Table 12-2. Additional Tracking Species Representing Loss of Inorganic Sulfate to Organosulfate (only included if using SAPRC07TIC_AE6I, SAPRC07TIC_AE7I, CB6R3_AE7, or CB6R3M_AE7 mechanisms)

Species Group	Species Name	MW	Description
AE	OSO4J	96.0	Loss of ASO4J to organosulfate
AE	OSO4AQH2O2J	96.0	Loss of ASO4AQH2O2J to organosulfate
AE	OSO4AQO3J	96.0	Loss of ASO4AQO3J to organosulfate
AE	OSO4AQFEMNJ	96.0	Loss of ASO4AQFEMNJ to organosulfate
AE	OSO4AQMHPJ	96.0	Loss of ASO4AQMHPJ to organosulfate
AE	OSO4AQPAAJ	96.0	Loss of ASO4AQPAAJ to organosulfate
AE	OSO4GASJ	96.0	Loss of ASO4GASJ to organosulfate
AE	OSO4EMISJ	96.0	Loss of ASO4EMISJ to organosulfate
AE	OSO4ICBCJ	96.0	Loss of ASO4ICBCJ to organosulfate

Contact

Shawn Roselle, Computational Exposure Division, U.S. EPA

Appendix A: Model options

A.1 Configuration Options (config_cmaq.csh)

Consistency of configuration variables is critical for building CMAQ itself, not just its libraries. Accordingly, CMAQ includes the configuration script config_cmaq.csh to help enforce consistent environment settings for CMAQ and its associated libraries. The following lists the config_cmaq.csh variables defined for the build process and suggests values to which to set those variables.

Note that for multiprocessor applications it is recommended that the Fortran MPI wrapper script mpif90 be specified for the Fortran compiler (myFC). Using this script, instead of a direct call to the Fortran compiler, will ensure that the full suite of MPI components (libraries and include files) for the compiler are included in the parallel build.

- `CMAQ_HOME` The central CMAQ installation directory. For example, if you installed the CMAQ source code in the directory `/home/user/CMAQ` set CMAQ_HOME with `export CMAQ_HOME=/home/user/CMAQ` for bash or `setenv CMAQ_HOME /home/user/CMAQ` for csh; note that this variable is M3HOME prior to CMAQv5.2
- `CMAQ_DATA` Automatically set by config_cmaq.csh; note that this variable is M3DATA prior to CMAQv5.2
- `CMAQ_LIB` Automatically set by config_cmaq.csh; note that this variable is M3LIB prior to CMAQv5.2
- `compiler` Set the Fortran compiler type that you will use to compile CMAQ; choices are intel, pgi, or gcc
- `compilerVrsn` (Optional) Set the Fortran compiler version number that you will use to compile CMAQ; if you employ this variable, it will be appended to the compiler type when naming build directories and executables
- `IOAPI_INCL_DIR` Location of the I/O API include files installed on your Linux system
- `IOAPI_LIB_DIR` Location of the I/O API library on your Linux system
- `NETCDF_LIB_DIR` Location of the netCDF C Library on your Linux system
- `NETCDF_INCL_DIR` Location of the netCDF C include files on your Linux system
- `NETCDFF_LIB_DIR` Location of the netCDF Fortran Library on your Linux system
- `NETCDFF_INCL_DIR` Location of the netCDF Fortran include files on your Linux system
- `MPI_LIB_DIR` Location of the Message Passing Interface Library on your Linux system

  - `ioapi_lib` Name of the I/O API libraryar on your system; set to "-lioapi"

- `netcdf_lib` Name of the netCDF library C on your system; set to "-lnetcdf" for versions < 4.2.0, "-lnetcdf" for version 4.2.0 and later
- `netcdff_lib` Name of the netCDF Fortran library on your system; set to "-lnetcdff" for versions 4.2.0 and later, for version before 4.2.0 this library is bundled with the C library.
- `pnetcdf_lib` Name of the parallel netCDF library on your system; set to "-lpnetcdf"
- `mpi_lib` Name of the MPI library on your system; set to "-lmpich" for MVAPICH, "-lmpi" for OpenMPI
- `myFC` Set to match the `FC` (Fortran compiler) you use to compile netCDF
- `myCC` Set to match the `CC` (C compiler) you use to compile netCDF
- `myFSTD` Standard Mode Fortran compiler optimization flags for your Linux system; suggested values for CMAQ are in the distributed script
- `myDBG` Debug Mode Fortran compiler optimization flags for your Linux system; suggested values for CMAQ are in the distributed script
- `myLINK_FLAGS` Fortran compile linker flags for your Linux system; suggested values for CMAQ are in the distributed script
- `myFFLAGS` Fixed-format Fortran compiler optimization flags for your Linux system; suggested values for CMAQ are in the distributed script
- `myCFLAGS` C compiler optimization flags for your Linux system; suggested values for CMAQ are in the distributed script
- `myFRFLAGS` Free form-format Fortran compiler optimization flags for your Linux system; suggested values for CMAQ are in the distributed script
- `extra_lib` Set to other libraries required for compiling on your Linux system; users will likely need to change this setting in the distributed script for portability to their system.
- `EXEC_ID` build tag, should be automatically set by config_cmaq.csh
- `CMAQ_REPO` This is always the location of the CMAQ repository that the user will pull from to create exectuables. If the user is building CMAQ inside the repository then it will be equal to CMAQ_HOME. If not, the user must supply an alternative folder location.

A.2 Compilation Options (bldit_cctm.csh)

The configuration options listed here are set during compilation of the CCTM executable through the build script, bldit_cctm.csh, located under the CCTM/scripts folder. When these options are invoked they create a binary executable that is fixed to the specified configuration. To change these options, you must recompile CCTM and create a new executable.

Several of the CCTM science modules have more than one option. Brief descriptions of these options are provided here.

The following options are invoked by uncommenting the line in the CCTM build script. Comment the line in the script using a "#" to turn the option off.

- `CompileBLDMAKE`
  Uncomment to use an existing BLDMAKE executable to build CCTM executable. If commented out, recompile BLDMAKE utility from the source.

- `CopySrc`
  Uncomment to copy the source code into a working build (BLD) directory. If commented, only the compiled object and executable files will be placed in the BLD directory.

- `MakeFileOnly`
  Uncomment to build a Makefile but to not compile the executable. The Makefile will be located in the BLD directory and can subsequently be used to manually compile the executable by typing 'make' in the BLD direcotry. Comment out to both create a Makefile and compile the executable when invoking the bldit_cctm.csh script.

- `ParOpt`
  Build an executable for running on multiple processors. Invoking this command requires the availability of the MPI library/INCLUDE files.

- `build_parallel_io`
  Uncomment to build CMAQ with true parallel I/O feature (requires mpi version of ioapi 3.2 and pnetcdf, refer to Appendix D.)

- `Debug_CCTM`
  Uncomment to compile the CCTM executable in Debug Mode.

- `ISAM_CCTM`
  Uncomment to compile the CCTM executable with Integrated Source Apportionment Method (ISAM). See Chapter 11 for futher information before invoking this option.

- `build_twoway`
  Uncomment to build WRF-CMAQ two way model with explicit meteorological-chemical feedbacks - to build a stand-alone CMAQ, comment this option out. This option is currently not supported. Please contact David Wong (wong.david@epa.gov) for specific instructions for building WRF-CMAQ.

- `potvortO3`
  Uncomment to build CMAQ with potential vorticity free-troposphere O3 scaling. See Chapter 6 for futher information before invoking this option.

The following configuration settings may have multiple options. Select one option in the CCTM build script.

- `ModGrid: [default: Cartesian]`
  The CCTM model grid configuration module. Currently only Cartesian coordinates are supported by CMAQ. Do not change this module setting.
  - `grid/cartesian`

- `ModAdv: [default: wrf_cons]`
  3-D Horizontal module. See Chapter 6 for further information.
  - `wrf_cons`
    use the WRF vertically integrated column mass to calculate vertical advection
  - `local_cons`
    use the layer-by-layer integrated mass-conserving scheme to calculate vertical advection

- `ModHdiff: [default: hdiff/multiscale]`
  The only option in CMAQv5 for the horizontal diffusion module is `hdiff/multiscale`, which uses a diffusion coefficient based on local wind deformation. Do not change this module setting. See Chapter 6 for further information.
  - `hdiff/multiscale`

- `ModVdiff: [default: vdiff/acm2]`
  Vertical diffusion and surface exchange module. Do not change this module setting. See Chapter 6 for further information.
  - `vdiff/acm2`
    calculate vertical diffusion using the Asymmetric Convective Model version 2 (ACM2)

- `ModDepv: [default: depv/m3dry]`
  Deposition calculation module. Users may choose between the msdry and stage options. If CMAQ output of land use specific deposition or stomatal flux is desired, then the stage option must be selected. See Chapter 6 for further information.
  - `depv/m3dry`
    CMAQ m3dry dry deposition routine. This is an updated version of the routine that has always been in CMAQ. See Chapter 6 for further information.
  - `depv/stage` CMAQ stage dry deposition routine. This option is new in version 5.3. See Chapter 6 for further information.

- `ModEmis: [default: emis/emis]`
  CMAQ inline anthropogenic and natural emissions module. Inline emissions are activated by the user via the CCTM run script. Do not change this module setting. See Chapter 6 for further information.

  - `emis/emis`

- `ModBiog: [default: biog/beis3]`
  Calculate biogenic emissions online with the BEIS3 model. Online biogenic emissions are activated in the CCTM run script. Do not change this module setting. See Chapter 6 for further information.

  - `biog/beis3`

- `ModPlmrs: [default: plrise/smoke]`
  Calculate inline plume rise for large point sources using the Briggs algorithm as it is implemented in SMOKE. Inline emissions plume rise is controlled in the CCTM run script. Do not change this module setting. See Chapter 6 for further information.

  - `plrise/smoke`

- `ModCgrds: [default: spcs/cgrid_spcs_nml]`
  CMAQ model species configuration module.

  - `spcs/cgrid_spcs_nml`
    namelist files used to configure CMAQ model species
  - `spcs/cgrid_specs_icl`
    use Fortran INCLUDE files to configure CMAQ model species

- `ModPhot: [default: phot/inline]`
  Photolysis calculation module. See Chapter 6 for further information.

  - `phot/inline`
    calculate photolysis rates inline using simulated aerosols and ozone concentrations
  - `phot/table`
    calculate clear-sky photolysis rates off-line using the CMAQ program JPROC; provide daily photolysis rate look-up tables to CCTM

- `Mechanism: [default: cb05e51_ae6_aq]`
  Chemistry mechanism for gas, aerosol, and aqueous chemistry. See the CMAQv5.3 Chemical Mechanisms Table for a listing of the mechanism choices that are available in CMAQv5.3. See Chapter 6 for further information.

- `Tracer [default trac0]`
  Specifies tracer species. Invoking inert tracer species in CMAQ requires defining the tracers using namelist files and compiling the CMAQ programs with these files. The setting for this

module corresponds to the directory name in the `$CMAQ_HOME/CCTM/src/MECHS` directory that contains the namelist files for the tracer configuration. The default setting does not use any tracers.

  - `trac[n]`

- `ModGas: [default: gas/ebi_${Mechanism}]`
  Gas-phase chemistry solver module. See Chapter 6 for further information.

  - `smvgear`
    use the SMVGEAR chemistry solver
  - `ros3`
    use gas/the Rosenbrock chemistry solver
  - `ebi`
    use the Euler Backward Iterative solver

- `ModDiag` use various diagnostic routines. Currently only the vertical extraction tool is implemented here.

- `ModAero: [default: aero7]`
  CMAQ aero/aerosol module. See Chapter 6 for further information.

  - `aero7`
    seventh-generation modal CMAQ aerosol model with extensions for sea salt emissions and thermodynamics; includes a new formulation for secondary organic aerosol yields

- `ModCloud: [default: cloud/acm_ae6]`
  CMAQ cloud module for modeling the impacts of clouds on deposition, mixing, photolysis, and aqueous chemistry. See Chapter 6 for further information.

  - `cloud/acm_ae6`
    ACM cloud processor that uses the ACM methodology to compute convective mixing with heterogeneous chemistry for AERO6
  - `cloud/acm_ae6_mp`
    ACM cloud processor that uses the ACM methodology to compute convective mixing with heterogeneous chemistry for AERO6 and air toxics; this is the multipollutant mechanism in CMAQv5
  - `cloud/acm_ae6_kmt`
    ACM cloud processor that uses the ACM methodology to compute convective mixing with heterogeneous chemistry for AERO6 and aqueous chemistry with kinetic mass transfer and Rosenbrock solver
  - `cloud/acm_ae6i_kmti`
    ACM cloud processor that uses the ACM methodology to compute convective mixing with heterogeneous chemistry for AERO6 and aqueous chemistry with kinetic mass transfer and Rosenbrock solver with an extension to simulate the aqueous phase formation of SOA in cloud droplets, see: CMAQv5.1 Aqueous Chemistry

- `ModUtil: [default: util]`
  CMAQ utility modules. Do not change this module setting.
  - `util/util`
- `ModPa: [default: procan/pa]` Process analysis is controlled in the CCTM run script. Do not change this module setting. - `procan/pa`
- `ModPvO3: [default: pv_o3]` Potential vorticity parameterization for free-troposphere exchange of ozone. This option is configured using the potvorO3 variable in the CCTM build script. Do not change this module setting. See Chapter 6 for further information.
  - `pv_o3`

A.3 Execution Options (run_cctm.csh)

The environment variables listed below are invoked during execution of the CCTM and are set in the CCTM run script, run_cctm.csh located under the CCTM/scripts folder.

- `compiler [default: intel]`
- `compilerVrsn [default: 13.1]`
- `VRSN [default: v53]`
- `PROC [default: mpi]`
  Sets if the CCTM will run in multi-processor or serial mode.
  - `mpi`
    Use MPI multi-processor configuration. The CCTM executable must have been built to support MPI, see bldit_cctm.csh compilation options above. The run script requires settings for the number of processors and other MPI configuration variables required by the Linux system.
  - `serial`
    Run the CCTM in serial, single-processor mode.
- `MECH [default: None]`
  CMAQ chemical mechanism. Must match `Mechanism` variable setting in the CCTM build script. See Chapter 6 for further information.
- `APPL [default: SE53BENCH]`
  Application name used to label output binaries and log files.
- `RUNID [default: $VRSN_compiler_APPL]`
  Run ID used to track version number, compiler, and application case name.

- `BLD`
  Directory path of the built CCTM executable

- `EXEC [default: CCTM_$APPL_$EXECID]`
  The name of the CCTM executable.

MPI Configuration

- `NPCOL_NPROW [default: 1 1]`
  The numbers of columns and rows for decomposing the modeling domain in an MPI configuration. The product of this pair of numbers must equal the total number of processors allocated to the CCTM simulation. For serial or single-processor MPI runs set to `1 1`. For example, for an 8 processor MPI simulation, set to `4 2`.

- `NPROCS [default: 1]`
  Number of processors to allocate for the CCTM simulation; equal to the product of NPCOL x NPROW. For serial or single-processor MPI runs set to `1`, otherwise set to the product of the two numbers used in NPCOL_NPROW.

Vertical extent

- `NZ [default: 35]`
  Set the number of vertical layers.

Timestep Configuration

- `NEW_START [default: TRUE]`
  Value should be true for new simulations starting from an initial condition file. To restart from a previous days simulation output, set to FALSE. For all standard runscripts, this variable is automatically set to FALSE after looping to the second day of the simulation.

- `START_DATE`
  Simulation start date in Gregorian format (YYYY-MM-DD)

- `END_DATE`
  Simulation end date in Gregorian format (YYYY-MM-DD)

- `STTIME`
  Simulation start time (HHMMSS)

- `NSTEPS [default: 240000]`
  Number of simulation time steps (HHMMSS)

- `TSTEP [default: 010000]`
  Simulation output time step interval (HHMMSS)

CCTM Configuration Options

- `GRID_NAME [default: Blank]`
  Name of the grid definition contained in the GRIDDESC file that specifies the horizontal grid for the current application of the model.

- `GRIDDESC [default: Path to GRIDDESC file]`
  Grid description file for setting the horizontal grid definition.

- `CTM_APPL [default: ${RUNID}_${YYYYMMDD}]`
  CCTM log and output file naming extension.

- `CONC_SPCS [if commented out, all species]`
  Model species to be written to the CCTM_CONC file.
- `CONC_BLEV_ELEV [if commented out, all layers]`
  Vertical model layer range for the CCTM_CONC file concentrations; this variable sets the lower and upper layers over which to output the CCTM_CONC file. In the example script, BLEV and ELEV are both set to 1, so concentrations will only be written for the first layer.
- `AVG_CONC_SPCS [if commented out, output all species]`
  Model species for calculating integral average concentrations for each output time step. Options can be any of the standard output species that are written to the CCTM_CONC file. The species in this list will be written to the CCTM_ACONC output file.
- `ACONC_BLEV_ELEV [default: if commented out, all layers]`
  Vertical model layer range for integral average concentrations; this variable sets the lower and upper layers over which to calculate integral average concentrations. For example, setting this variable to "1 5" will produce integral average concentrations for model layers 1 through 5.
- `AVG_FILE_END_TIME [default: N]`
  Change the time stamp of the ACONC file output time step from the default of the beginning of the hour to the end of the hour.
  - `Y`: Set the time stamp to the end of each hour.
  - `N`: Set the time stamp to the beginning of the hour.

- `EXECUTION_ID [default: Blank]`
  The name of the CCTM executable; automatically set by the script.

Synchronization Time Step and Tolerance Options

- `CTM_MAXSYNC [default: 720]`
  Maximum synchronization time step in seconds

- `CTM_MINSYNC [default: 60]`
  Minimum synchronization time step in seconds

- `SIGMA_SYNC_TOP [default: .70]`
  Top sigma level thru. which sync step determined

- `ADV_HDIV_LIM [default: .9]`
  Maximum horizontal divergence limit for advection time step adjustment

- `CTM_ADV_CFL [default: .75]`
  Maximum Courant–Friedrichs–Lewy (CFL) condition

- `RB_ATOL [default: 1.0E-07]`
  If using Rosenbrock (ros3) photochemistry solver, the absolute tolerance for converging to solution
- `RB_RTOL [default: 1.0E-03]`
  If using Rosenbrock (ros3) photochemistry solver, relative tolerance for converging to solution
- `GEAR_ATOL [default: 1.0E-09]`
  If using Gear (smvgear) photochemistry solver, the absolute tolerance for converging to solution
- `GEAR_RTOL [default: 1.0E-03]`
  If using Gear (smvgear) photochemistry solver, relative tolerance for converging to solution

Science Options

- `CTM_OCEAN_CHEM [default: True]`
  Use Online Sea Spray Aerosol emissions and Halogen ozone chemistry. See Chapter 6 for further information.

- `CTM_WB_DUST [default: False]`
  Setting to calculate online windblown dust emissions in CCTM. Setting this variable to Y also enables the option to provide additional gridded landuse input files beyond the land use information contained in the MCIP files. Whether or not additional landuse information is provide and, if yes, whether that additional landuse information is provided in one or two files is controlled by the environment variable CTM_WBDUST_BELD. See Chapter 6 for further information.
- `CTM_WBDUST_BELD [default: UNKNOWN]`
  Landuse database for identifying dust source regions; ignore if CTM_WB_DUST = FALSE
  - BELD3: Use BELD3 landuse data for windblown dust calculations. The user needs to specify the DUST_LU_1 and DUST_LU_2 files described in Chapter 4. These files typically are available for North American domains only.
  - BELD4: Use BELD4 landuse data for windblown dust calculations. The user needs to specify the E2C_LU file described in Chapter 4. This file typically is available for North American domains only.
  - UNKNOWN: Use landuse information provided by MCIP for windblown dust calculations

- `CTM_LTNG_NO [default: Y]`
  Y/N setting to activate lightning NO emissions. Setting this variable to Y requires additional variables to define the configuration of the lightning NO emissions calculation. See the settings for `LTNGNO`, `LTNGPARAMS`, `NLDN_STRIKES`, and `LTNGDIAG` below. See Chapter 6 for further information.
- `CTM_WVEL [default: Y]`
  Y/N setting to output the CCTM-calculated vertical velocities to the CONC file.
- `KZMIN [default: Y]`
  If KZMIN is set to Y, CCTM will read the urban land use fraction variable (PURB) from the GRID_CRO_2D meteorology file and use this information to determine the minimum eddy diffusivity in each grid cell. In CMAQv5, grid cells that are predominantly urban use a KZMIN value of 1.0 m2/s and non-urban cells use a value of 0.01 m2/s. If this variable is set to N, the PURB variable will not be used and a uniform KZMIN value of 1.0 m2/s will be used throughout the modeling domain.
- `CTM_MOSAIC [default N]`
  Y/N setting to ouput land use specific deposition velocities and fluxes. This option is only available when using the STAGE deposition module. See Chapter 6 for further information.
- `CTM_FST [default: N]`
  Y/N setting to output land-use specific stomatal flux. This option is only available when using the STAGE deposition module and when CTM_MOSAIC is set to Y. See Chapter 6 for further information.
- `PX_VERSION` Y/N setting to indicate whether the Pleim-Xiu land-surface model was used for the input meteorology. If this setting is set to Y the input meteorology data must include soil moisture (SOILM), soil temperature (SOILT), and soil type (ISLTYP) variables for use in the calculation of soil NO emissions. Additionally, the soil properties from PX will be used in the dust model and in the STAGE deposition module for calculating the soil compensation point for ammonia bidirectional exchange. See Chapter 6 for further information.
- `CLM_VERSION` Y/N setting to indicate whether the Community Land Model (CLM) land-surface model was used in generating the input meteorology. If this setting is set to Y the input meteorology data must include soil moisture (SOILM), soil temperature (SOILT), and soil type (ISLTYP) variables for use in the calculation of soil NO emissions. Additionally, the soil properties from CLM will be used in the dust model and in the STAGE deposition module for calculating the soil compensation point for ammonia bidirectional exchange. See Chapter 6 for further information.
- `NOAH_VERSION` Y/N setting to indicate whether the Noah land-surface model was used in generating the input meteorology. If this setting is set to Y the input meteorology data must include soil moisture (SOILM), soil temperature (SOILT), and soil type (ISLTYP) variables for use in the calculation of soil NO emissions. Additionally, the soil properties from Noah will be used in the dust model and in the STAGE deposition module for calculating the soil compensation point for ammonia bidirectional exchange. See Chapter 6 for further information.
- `CTM_ABFLUX [default: Y]`
  Y/N setting to activate fertilizer ammonia bidirectional flux for in-line emissions and deposition velocities. Setting this variable to Y requires four additional input files that include gridded fractional crop distributions (E2C_LU), soil properties (E2C_SOIL), fertilizer conditions (E2C_CHEM), and an agricultural soil initial conditions file (INIT_MEDC_1). Activation of this setting will produce additional variables in the output dry deposition file.

- `CTM_BIDI_FERT_NH3` Y/N setting to indicate whether fertilizer NH3 should be subtracted from the emissions and handled instead by the NH3 bidirectional flux model. Note that the bidirectional flux model must also be invoked by setting CTM_ABFLUX to Y.

- `CTM_HGBIDI [default: N]` Y/N setting to activate mercury bidirectional flux for in-line emissions and deposition velocities. Activation of this setting will produce additional variables in the output dry deposition file.
- `CTM_SFC_HONO [default: Y]` Y/N setting to include surface HONO interactions. See Chapter 6 for further information.
- `CTM_GRAV_SETL [default Y]`
  Y/N setting to activate gravitational sedimentation for aerosols.
- `CTM_BIOGEMIS [default: Y]`
  Y/N setting to calculate biogenic emissions. If this option is activated, several additional variables must be set (see the online biogenic emissions configuration settings). See Chapter 6 for further information.
- `OPTICS_MIE_CALC [default: N]`
  In the inline option for photolysis rates, solve Mie Theory to calculate the optical properties of the aerosol modes based on uniformly mixed spheres.
- `CORE_SHELL_OPTICS [default: N]`
  In the inline option for photolysis rates, solve Mie Theory to calculate the optical properties of the aerosol modes based on spheres with an elemental carbon core.

Process analysis options

- `CTM_PROCAN [default: N]`
  Activate process analysis in the CCTM. Set this to Y and use $CMAQ_DATA/pacp/pacp.inp to configure the integrated process rate and integrated reaction rate settings for the CCTM. Additional process analysis output files will be created when this setting is activated.

- `PA_BCOL_ECOL [default: 0]`
  Modeling grid domain column range for the process analysis calculations. Set to the two digits representing the beginning and ending column number bounding the process analysis domain.

- `PA_BROW_EROW [default: 0]`
  Modeling grid domain row range for the process analysis calculations. Set to the two digits representing the beginning and ending row number bounding the process analysis domain.

- `PA_BLEV_ELEV [default: 0]`
  Modeling grid domain layer range for the process analysis calculations. Set to the two digits representing the bottom and top layer numbers bounding the process analysis domain.

- `PACM_INFILE`
  Input file that specifies the desired output information (read by pa_read.F). See Table 1 in Chapter 9 for details on the types of equations and operators that can be used in this file. A sample file is

included in each of the mechanism folders under the CCTM/src/MECHS directory. For example, the file pa_cb6r3_ae7_aq.ctl in CCTM/src/MECHS/cb6r3_ae7_aq provides a template of IRR and IPR commands.

- `PACM_REPORT`
  The output file that displays how CMAQ translates the variables listed in `PACM_INFILE`, and lists the reactions (including reactants, products and yields) that will be used in calculating the IPR and IRR values.

I/O Controls

- `IOAPI_LOG_WRITE [default:False]`
  Set to T to turn on excess WRITE3 logging by the I/O API.

- `FL_ERR_STOP [default: False]`
  Set to T to configure the program to exit if inconsistent headers are found in the input files.

- `PROMPTFLAG [default: False]`
  Turn on I/O-API PROMPTFILE interactive mode. Set to T to require interactive prompts for different I/O API operations.

- `IOAPI_OFFSET_64 [default: True]`
  I/O API setting for large time step records. If your output time step is going to produce data that are >2GB per time step, then this needs to be set to YES.

Aerosol Diagnostics Controls

- `CTM_PMDIAG [default: False]`
  Output aerosol diagnostics and properties file. These data are required for post-processing the fraction of aerosol mass is in various size ranges (e.g. PM2.5, PM10, etc.). The file also includes physical parameters describing the aerosol size distribution like the following: dry diameter, wet diameter, standard deviation, wet second moment, dry second moment, wet thrid moment, dry third moment, and density.

- `CTM_APMDIAG [default: False]`
  Output hourly average aerosol diagnostics and properties file. These data are required for post-processing the fraction of aerosol mass is in various size ranges (e.g. PM2.5, PM10, etc.). The file also includes physical parameters describing the aerosol size distribution like the following: dry diameter, wet diameter, standard deviation, wet second moment, dry second moment, wet thrid moment, dry third moment, and density.

- `APMDIAG_BLEV_ELEV [default: 0]`
  Modeling grid domain layer range for the hourly average aerosol diagnostics and properties file. Set to the two digits representing the bottom and top layer numbers to bound the output domain.

Comment out this variable or set it to 0 to output all layers. Set the value to "1 1" to output just the surface layer.

Diagnostic Output Flags

- `CTM_CKSUM [default: True]`
  Write science processes summaries to the standard output. Impacts run speed and log file output size.
- `CLD_DIAG [default: False]`
  Output an hourly wet deposition diagnostic file (CTM_WET_DEP_2) that includes convective wet deposition estimates.
- `CTM_PHOTDIAG [default: False]`
  Output files for viewing the photolysis rates used in the model simulation and what meterological and other factors determined the rates. The inline and table options produce three files (`CTM_RJ_1`, `CTM_RJ_2` and `CTM_RJ_3`) and one file (`CTM_RJ_2`), respectively. `CTM_RJ_1` is a two dimensional file that contains key photolysis rates and radiative parameters. `CTM_RJ_2` contains the photolysis rates used over the model domain. `CTM_RJ_3` contains data used to calculate the photolysis rates.
  – `NLAYS_PHOTDIAG [default: 1]` : Number of layers in `CTM_RJ_2` and `CTM_RJ_3` files. Permitted values equal 1 to number of layers in model domain. Only the inline option uses this runtime option.
  – `NWAVE_PHOTDIAG [default:294 303 310 316 333 381 607]` : In `CTM_RJ_3`, the wavelengths of diagnostic data written. The user can use or subset the default values.

- `CTM_SSEMDIAG [default: False]`
  Output the calculated sea salt emissions to a diagnostic netCDF output file (CTM_SSEMIS_1).
- `CTM_DUSTEM_DIAG [default: False]`
  Output the online dust emissions to a diagnostic netCDF output file (CTM_DUST_EMIS_1). The diagnostic file includes not only the total dust emissions, but also dust emissions by land use category and dust model parameters, such as gridded erodible land use fractions.
- `CTM_DEPV_FILE [default: False]`
  Output an hourly diagnostic file (CTM_DEPV_DIAG) for the inline deposition velocity calculations.
- `LTNGDIAG [default: False]`
  Output a lightning NO emissions diagnostics file.

Inline emissions configuration

- `STK_GRPS_###`
  Directory path and file name of the stack groups file for sector ###, where ### = 001, 002,…,N_EMIS_PT. Each ### refers to one of the inline plume rise point-source sectors. See Chapter 6 for further information.

- `STK_EMIS_###`

Directory path and file name of the point emissions file for sector ###, where ### = 01, 02,…,N_EMIS_PT. Each ### refers to the one of the plume rise point-source sectors. See Chapter 6 for further information.

- `STK_EMIS_DIAG_###`
  Logical for turning on/off diagnostic output for point emissions file for sector ###, where ### = 01, 02,…,N_EMIS_PT. Each ### refers to the one of the plume rise point-source sectors. These data reflect the emission rates after scaling rules have been applied by DESID, the emissions control interface. Values for STK_EMIS_DIAG_### include FALSE, TRUE, 2D, 2DSUM, and 3D. The TRUE and 2D options are synonymous and will output just the surface layer of emissions. The 2DSUM option outputs a 2D file with values calculated from summing the entire column of emissions in each horizontal grid cell. The 3D option outputs a full 3D file. All options provide output across all output time steps during the simulation day. See Chapter 6 for further information.

- `LAYP_STDATE [HHMMSS]`
  Start date for calculating elevated-point-source emissions.

- `LAYP_STTIME [HHMMSS]`
  Start time for calculating elevated-point-source emissions.

- `LAYP_NSTEPS [HHHHHH]`
  Number of time steps for calculating elevated-point-source emissions.

Lightning NOx configuration

- `LTNGNO [default: "InLine"]`
  Setting to define whether the lightning emissions calculation will be inline or off-line. This variable can be set to a gridded netCDF file of lightning NO emissions to use emissions calculated with a preprocessor outside of CCTM. Setting this variable to "inline" activates the inline emissions calculation in CCTM and requires the LTNGPARMS_FILE variable (see below) to provide parameters for generating inline lightning NO emissions. See Chapter 6 for further information.

- `USE_NLDN [default: False]`
  Use hourly NLDN strikes file to compute inline lightning NO emissions. Activating this setting requires the NLDN_STRIKES input file. If USE_NLDN is set to N and LTNGNO set to "InLine", lightning NO emissions will be generated using parameters provided in the LTNGPARMS_FILE.
  Lightning parameters netCDF file, which contains the linear regression parameters for generating lightning NO using the parameterization scheme when LTNGNO set to "InLine" and USE_NLDN set to N. In addition, it also contains the intercloud to cloud-to-ground flash ratios, scaling factors for calculating flashes using the convective precipitation rate, land-ocean masks, and the moles of NO per flash (cloud-to-ground and intercloud) which are used by both lightning production schemes (NLDN and parameterization). Ingore if LTINGNO set to an external input file. See Chapter 6 for further information.

- `CTM_LTNGDIAG_1`
  Lightning diagnostics output 3D netCDF file; ignore if `LTNGDIAG = N`

- `CTM_LTNGDIAG_2`
  Lightning diagnostics output 2D netCDF file (column total lightning NO emissions); ignore if `LTNGDIAG = N`

Online biogenic emissions configuration

- `GSPRO [default: Build Directory]`
  Directory path and file name for input ASCII speciation profiles. See Chapter 6 for further information.

- `B3GRD [default: None]`
  Grid-normalized biogenic emissions input netCDF file. See Chapter 6 for further information.

- `BIOSW_YN [default: Y]`
  Use the frost dates switch file to determine whether to use winter or summer biogenic emissions. See Chapter 6 for further information.

- `BIOSEASON [default: False]`
  File name for the frost dates switch input netCDF file. See Chapter 6 for further information.

- `SUMMER_YN [default: False]`
  Toggle for summer season normalized biogenic emissions. This variable is ignored if BIOSW_YN is set to Y. Comment out or set to Y to select summer season biogenic emissions factors; set to N to turn off. See Chapter 6 for further information.

- `PX_VERSION [default: True]`
  Setting to indicate whether the Pleim-Xiu land-surface model was used for the input meteorology. If this setting is set to Y the input meteorology data must include soil moisture (SOILM), soil temperature (SOILT), and soil type (ISLTYP) variables for use in the calculation of soil NO emissions.

- `SOILINP [default: [Out Directory/CCTM_SOILOUT_$RUNID_$YESTERDSY]`
  Directory path and file name of biogenic NO soil emissions file. If NEW_START is set to N or F, the soil NO emissions file from the previous day's simulation will be a required input file. See Chapter 6 for further information.

- `B3GTS_DIAG [default: False]`
  Write the online biogenic emissions (mass units) to a diagnostic netCDF output file (B3GTS_S).

- `B3GTS_S [default: [Output Directory]/CCTM_B3GTS_$CTM_APPL.nc`
  Diagnostic output netCDF file of biogenic emissions. This variable is ignored if B3GTS_DIAG is set to N.

Windblown dust emissions configuration

- `DUST_LU_1 [default: Path to BELD3 Data]`
  Input BELD "A" landuse netCDF file gridded to the modeling domain. Used if `CTM_WBDUST_BELD` is set to BELD3.
- `DUST_LU_2 [default: Path to BELD3 Data]`
  Input BELD "TOT" landuse netCDF file gridded to the modeling domain. Used if `CTM_WBDUST_BELD` is set to BELD3.
- `E2U_LU [default: path to BELD4 data file]`
  Input BELD4 landuse netCDF file gridded to the modeling domain. Used if `CTM_WBDUST_BELD` is set to BELD4.

Appendix B: Emissions Input and Control

Jump to DESID Tutorial for step by step instructions on performing some basic manipulation of emission streams.

Jump to Emissions overview in Chapter 6 of this User's Guide.

B.1 Emissions Control with the Detailed Emissions Scaling, Isolation and Diagnostics Module (DESID)

In addition to the options available in the RunScript, CMAQ now reads a dedicated namelist in order to apply comprehensive rules for reading and scaling emissions. The namelist, called the **Emission Control Namelist** is named "EmissCtrl.nml" by default and a separate version exists for every mechanism because these namelists are preloaded with likely rules linking emissions of important CMAQ primary species to their typical surrogate names as output by SMOKE. By default, this namelist is stored in each chemical mechanism folder (e.g. MECHS/cb6r3_ae7_aq) and is copied into the user's build directory when bldit_cctm.csh is executed. If the user modifies the name or location of this namelist, then the following command in the RunScript should be updated as well:

```
setenv EMISSCTRL_NML ${BLD}/EmissCtrl.nml
```

The Detailed Emissions Speciation, Isolation and Diagnostics (DESID) module included with CMAQv5.3 provides comprehensive customization and transparency of emissions manipulation to the user. The customization of emissions is accomplished via the Emission Control Namelist, which contains four sections of variables that modify the behavior of the emissions module. These include ***General Specs***, ***Emission Scaling Rules***, ***Size Distributions***, and ***Regions Registry***

B.2 General Specs

These variables modify or constrain the effects of other sections of the namelist. The "Guard_XXX" options allow the user to protect specific streams from being modified by scaling rules (explained in section B.3) with the "ALL" keyword in the stream field. For example, the "Guard_BiogenicVOC" option instructs the model not to scale biogenic VOC emissions from the online BEIS module, even if a rule indicates that "ALL" streams are to be scaled. The other "Guard_XXX" options achieve the same effect for other online emissions sources like wind-blown dust, sea spray, marine gas, and lightning NO.

B.3 Emission Scaling Rules

With the rules present in this section, the user is able to exert sophisticated, precise control over the scaling applied to emissions from specific streams, in specific geographic areas, and/or for specific compounds. The set of rules used by CMAQ to interpret emissions shall be provided in one array called EM_NML. It is necessary that every field (i.e. column) be populated for every rule. The fields are given and defined here and in the comment section of the Emission Control Namelist:

```
! Region      | Stream Label  |Emission | CMAQ-        |Phase/|Scale |Basis |Op
!  Label      |               |Surrogate| Species      |Mode  |Factor|      |
```

- 'Region Label' - Apply scaling for specific regions of the domain. Set this field to "EVERYWHERE" to apply the rule to the entire domain.
- 'Stream Label' - Short Name from Run Script (e.g. the value of GR_EMIS_01_LAB or STK_EMIS_01_LAB). There are a few reserved names that apply to online emissions streams. These are:
  - BIOG - Biogenic VOC emissions
  - MGEM - Marine Gas Emissions
  - LTNG - Lightning NO Emissions
  - WBDUST - Wind-Blown Dust Emissions
  - SeaSpray - Sea Spray Aerosol Emissions

  Set this field to 'ALL' to apply the rule to all emission streams.

- 'Emission Surrogate' - The character string identifying the surrogate on the emission file or in the online calculation that the CMAQ species should be mapped to. Usually this name is the same as the CMAQ species for convenience. For aerosols, it's usually slightly different (e.g ANO3 vs. PNO3). Set this field to 'ALL' to apply the rule to all emission surrogates.

- 'CMAQ-Species' - Internal Species Name. Set this field to 'ALL' to apply the rule to all CMAQ internal species.
- 'Phase/Mode' - If the CMAQ-Species is a Gas, this field should equal 'Gas'. If the CMAQ-Species is an aerosol, this field should indicate one of the possible emission aerosol modes. Every stream by default is given a 'COARSE' and 'FINE' mode. The user may refer to these or

define others above and refer to them as well. This level of specificity is needed so that aerosol number and surface area are calculated correctly, and so that any unit conversions between gases and aerosols can be handled correctly.

- 'Scale Factor' - Adjustment factor to be applied to the mapping
- 'Basis' - Specifies whether the scaling option should directly apply, or if the operation should conserve moles or mass when performing scaling operations. CMAQ has a lookup table of molecular weights for known emission surrogate species and can use these to translate molar and mass emission rates from the input file to the CMAQ species. CMAQ determines the units of the emission surrogate species by reading the file header (i.e. it is important the units are accurate. Options for input are:
  - 'MASS' - Conserve Mass. For example, if emissions of an aerosol are to be scaled to emissions of a gas surrogate, it is common to want to conserve mass.
  - 'MOLE' - Conserve Moles. For example, if emissions of a gas-phase species are to be scaled to another gas, it is sometimes desired to conserve moles since gas emissions are provided on a mole basis.
  - 'UNIT' - Ignore molecular weight conversions and apply emission rate directly regardless of units.
- 'Operation' - Specifies the kind of rule to be carried out. Options are:
  - 'a' - add the rule to existing instructions. This operation should be used for new entries, too.
  - 'm' - find existing scaling instructions matching this rule's features (ie. species, streams, etc) and multiply them by the factor in this particular rule.
  - 'o' - find existing scaling instructions matching this rule and overwrite them.

B.3.1 Default Rules

The Emission Control Namelists provided with the CMAQ repo have default rules included that correspond to each chemical mechanism. Here is an example default rule that links NO in CMAQ to NO from every emission stream in every model grid cell with a scale factor of 1.0.

```
! Region      | Stream Label  |Emission | CMAQ-        |Phase/|Scale |Basis |Op
!  Label      |               |Surrogate| Species      |Mode  |Factor|      |
'EVERYWHERE'  , 'All'         ,'NO'     ,'NO'          ,'GAS' ,1.0  ,'UNIT','a',
```

Many rules are needed here in order to properly link every emitted pollutant to a CMAQ species. Rules are needed for gas- and aerosol-phase species. Additional rules also exist for online aerosol modules like wind-blown dust and sea spray because the names of aerosol surrogates from these modules are different than those typically used for SMOKE output. For example, fine-mode aerosol sulfate is commonly called PSO4 in SMOKE, but is PMFINE_SO4 from dust and sea spray.

B.3.2 Modifying Default rules

The user can modify any default rule to change the scale factor applied. Alternatively, the user can add new rules after the default rules to customize the emissions. Typical modifications may include multiplying the emissions of a particular species from a particular stream by a factor of 2, zeroing out emissions of all species from a particular stream, etc. Please see the tutorial on Prescribing Emissions with DESID for specific examples of modifications and the syntax used to invoke them.

B.3.2.1 Supporting the Volatility Basis Set

The *Volatility Basis Set* for treating the semivolatile partitioning of primary organic emissions is an example of a model feature that is well-supported by DESID. The approach involves distributing the emissions of total primary organic aerosol (carbon and noncarbon mass, or POC and PNCOM) among a series of aerosol and gas species of varying volatility.

If the user would like to invoke the nonvolatile partitioning assumption, it can be accomplished by directing all POC and PNCOM emissions to the POC and PNCOM species in CMAQ.

```
! --> Nonvolatile POA
'EVERYWHERE', 'ALL'         ,'POC'    ,'APOC'        ,'FINE',1.    ,'MASS','a',
'EVERYWHERE', 'ALL'         ,'PNCOM'  ,'APNCOM'      ,'FINE',1.    ,'MASS','a',
```

If the user would like to apply the default volatility distribution to the POA emissions, it can be accomplished with the following default rules.

```
! --> Semivolatile POA
'EVERYWHERE', 'ALL'         ,'POC'    ,'VLVPO1'      ,'GAS' ,0.    ,'MASS','a',
'EVERYWHERE', 'ALL'         ,'PNCOM'  ,'VLVPO1'      ,'GAS' ,0.    ,'MASS','a',
'EVERYWHERE', 'ALL'         ,'POC'    ,'VSVPO1'      ,'GAS' ,0.045 ,'MASS','a',
'EVERYWHERE', 'ALL'         ,'PNCOM'  ,'VSVPO1'      ,'GAS' ,0.045 ,'MASS','a',
'EVERYWHERE', 'ALL'         ,'POC'    ,'VSVPO2'      ,'GAS' ,0.14  ,'MASS','a',
'EVERYWHERE', 'ALL'         ,'PNCOM'  ,'VSVPO2'      ,'GAS' ,0.14  ,'MASS','a',
'EVERYWHERE', 'ALL'         ,'POC'    ,'VSVPO3'      ,'GAS' ,0.18  ,'MASS','a',
'EVERYWHERE', 'ALL'         ,'PNCOM'  ,'VSVPO3'      ,'GAS' ,0.18  ,'MASS','a',
'EVERYWHERE', 'ALL'         ,'POC'    ,'VIVPO1'      ,'GAS' ,0.50  ,'MASS','a',
'EVERYWHERE', 'ALL'         ,'PNCOM'  ,'VIVPO1'      ,'GAS' ,0.50  ,'MASS','a',
'EVERYWHERE', 'ALL'         ,'POC'    ,'ALVPO1'      ,'FINE',0.09  ,'MASS','a',
'EVERYWHERE', 'ALL'         ,'PNCOM'  ,'ALVPO1'      ,'FINE',0.09  ,'MASS','a',
'EVERYWHERE', 'ALL'         ,'POC'    ,'ASVPO1'      ,'FINE',0.045 ,'MASS','a',
'EVERYWHERE', 'ALL'         ,'PNCOM'  ,'ASVPO1'      ,'FINE',0.045 ,'MASS','a',
'EVERYWHERE', 'ALL'         ,'POC'    ,'ASVPO2'      ,'FINE',0.    ,'MASS','a',
'EVERYWHERE', 'ALL'         ,'PNCOM'  ,'ASVPO2'      ,'FINE',0.    ,'MASS','a',
'EVERYWHERE', 'ALL'         ,'POC'    ,'ASVPO3'      ,'FINE',0.    ,'MASS','a',
'EVERYWHERE', 'ALL'         ,'PNCOM'  ,'ASVPO3'      ,'FINE',0.    ,'MASS','a',
'EVERYWHERE', 'ALL'         ,'POC'    ,'AIVPO1'      ,'FINE',0.    ,'MASS','a',
'EVERYWHERE', 'ALL'         ,'PNCOM'  ,'AIVPO1'      ,'FINE',0.    ,'MASS','a',
```

Notice that for each species (e.g. ALVPO1) a rule is needed to link the species to the emissions of POC and another rule is needed to add PNCOM. This is because both carbon and noncarbon mass are part of the emissions of every semivolatile species. To change the volatility distribution for all streams, the user may modify the scaling factors in the default rules above. To introduce specialized volatility distributions for specific stream (e.g. residential wood burning, forest fires, diesel vehicles, etc), rules may be added which explicitly identify a stream in the "Stream Label" field.

B.4 Applying Masks for Spatial Dependence

Gridded masks are used to apply rules to specific areas of the domain. For example, the following rule:

```
! Region      | Stream Label  |Emission | CMAQ-        |Phase/|Scale |Basis |Op
!  Label      |               |Surrogate| Species      |Mode  |Factor|      |
'KENTUCKY'    , 'All'         ,'All'    ,'All'         ,'All' ,1.50 ,'UNIT','m',
```

will scale emissions of all species from all streams by +50% but only in grid cells in the state of Kentucky. One or more I/O API formatted input files containing geographic region definitions are required to take advantage of this option. Such files should contain a separate variable for each spatial region of interest. Each variable is a gridded field of real numbers from 0.0 to 1.0, with 0.0 outside of the region of interest and 1.0 completely inside the region. Region border grid cells should have the geographic fraction attributed to the region (for example, a grid cell that 35% in Kentucky and 65% in Tennessee would have have the number 0.35 for the variable representing the Kentucky mask.

These mask files are read by CMAQ through environmental variables, which are identified in the RunScript. For example:

```
setenv US_STATES /home/${CMAQ_HOME}/CCTM/scripts/us_states.nc
```

If variables from multiple mask files are used, each of these mask files needs to be defined in the RunScript.

The *RegionsRegistry* section of the Emission Control Namelist maps each "Region Label" to specific variables on specific files. Here is the *RegionsRegistry* section in the default namelist:

```
&RegionsRegistry
 RGN_NML  =
 !          | Region Label   | File_Label  | Variable on File
 !<Default>    'EVERYWHERE'  ,'N/A'        ,'N/A',
 !<Example>    'WATER'       ,'CMAQ_MASKS' ,'OPEN',
 !<Example>    'ALL'         ,'CMAQ_MASKS' ,'ALL',
/
```

As indicated, the Region Label "EVERYWHERE" is active by default and returns a mask that operates uniformly across the entire domain. The "File_Label" field identifies the environment variable in the RunScript that stores the location and name of the file containing the mask. The user may modify this to any name they wish as long as it is consistent with the variable name on the RunScript. The "Variable on File" field identifies the variable on the input file that stores the gridded field to be used for this region. Examples are provided for two cases.

In the first case, a region with label "WATER" is defined and referenced to the variable "OPEN" (which is short for *open water*) in the file 'CMAQ_MASKS' which needs to be defined in the RunScript. Using this "WATER" region will apply a scaling rule only for open water grid cells and fractionally along coastlines. The second example provides a shortcut for files with many variables that are all desired (e.g. states of the Unites States). Rather than listing out all variables on the file and explicitly linking them to "Region Labels", the user can invoke the "ALL" keyword and all variables will be read and stored with "Region Labels" that equal the names of the variables on the file.

B.5 Aerosol Size Distributions

The treatment of aerosol size distributions in CMAQv5.3 has been updated to be more consistent with the way particle sizes and modes are treated by the National Emission Inventory and in emissions processing tools like SMOKE, MOVES, SPECIATE, and Speciation Tool. Specifically, in these tools, aerosol emissions are typically parameterized into two main modes, Fine and Coarse. Although the size distribution parameters (i.e. total number, diameter, standard deviation, etc.) for these modes will vary among emission sources, previous versions of CMAQ assumed that all primary fine particles had the same size distribution upon emission. Coarse-mode particles were assumed to exhibit a larger diameter but were also uniform across all sources (excluding wind-blown dust and sea spray).

In CMAQv5.3, users link particle emission surrogates to CMAQ particle species via the Emission-ScalingRules section of the Emission Control Namelist. Examples of default mapping rules can be found in any of the Emission Control Namelists in the CMAQ repository. The three lines below assign emissions for all streams for particulate-phase sulfate, ammonium, and nitrate.

```
! Region      | Stream Label  |Emission | CMAQ-        |Phase/|Scale |Basis |Op
!  Label      |               |Surrogate| Species      |Mode  |Factor|      |
'EVERYWHERE'  , 'ALL'         ,'PSO4'   ,'ASO4'        ,'FINE',1.0   ,'UNIT','a',
'EVERYWHERE'  , 'ALL'         ,'PNH4'   ,'ANH4'        ,'FINE',1.0   ,'UNIT','a',
'EVERYWHERE'  , 'ALL'         ,'PNO3'   ,'ANO3'        ,'FINE',1.0   ,'UNIT','a',
```

The CMAQ-Species field should be populated with bulk chemical names (e.g. ASO4, AEC, AK, ACA, etc). In other words, the 'i','j', or 'k' which usually designates the mode of the aerosol species name should be omitted. A list of the valid aerosol bulknames exists in the source file "AERO_DATA.F" in the array named "aerolist". The user should also identify the aerosol mode to be populated using the "Phase/Mode" field. In the example above, all of the rules identify the "FINE" mode as the destination mode. CMAQ uses this value to look up the size distribution parameters (diameter and standard deviation) to apply for this particular emission.

Aerosol mode keywords from the EmissionScalingRules section are linked to reference mode labels in the SizeDistributions section of the Emission Control Namelist. These assignments can be made for all streams at once, as demonstrated by the first two default entries initializing the 'FINE' and 'COARSE' modes, or they can be made on a stream-by-stream basis as shown below for Wind-Blown Dust and Sea Spray aerosol.

```
&SizeDistributions
 SD_NML    =
 !         | Stream Label   | Mode Keyword | Ref. Mode
 !<Default>  'ALL'          ,'FINE'        ,'FINE_REF',
 !<Default>  'ALL'          ,'COARSE'      ,'COARSE_REF',
             'WBDUST'       ,'FINE'        ,'FINE_WBDUST',
             'WBDUST'       ,'COARSE'      ,'COARSE_WBDUST',
             'SEASPRAY'     ,'FINE'        ,'FINE_SEASPRAY',
             'SEASPRAY'     ,'COARSE'      ,'COARSE_SEASPRAY',
 !<Example>  'AIRCRAFT'     ,'FINE'        ,'AIR_FINE',   !To use these examples, you
 !<Example>  'AIRCRAFT'     ,'COARSE'      ,'AIR_COARSE', ! must add entries for AIR_FINE
                                                         ! and AIR_COARSE to the data structur
                                                         ! em_aero_ref in AERO_DATA.
```

The 'Ref. Mode Labels' are used to lookup size distribution parameters in AERO_DATA.F. The following reference modes are defined in this file:

```
TYPE em_aero
    Character( 20 ) :: name
    Real            :: split( n_mode )  ! dimensionless
    Real            :: dgvem( n_mode )  ! meters
    Real            :: sgem ( n_mode )  ! dimensionless
END TYPE em_aero
INTEGER, PARAMETER  :: n_em_aero_ref = 9

TYPE( em_aero ), Parameter :: em_aero_ref( n_em_aero_ref ) = (/

!              ----Name----     -----Split-----    ---Geo. Mean Diameter---   ---Stnd Dev.---
& em_aero('FINE_REF       ',(/0.1,0.9,0.0/),(/0.06E-6,0.28E-6 ,6.0E-6 /),(/1.7,1.7,2.2/)), ! De
& em_aero('ACC_REF        ',(/0.0,1.0,0.0/),(/0.06E-6,0.28E-6 ,6.0E-6 /),(/1.7,1.7,2.2/)), ! Ju
& em_aero('COARSE_REF     ',(/0.0,0.0,1.0/),(/0.06E-6,0.28E-6 ,6.0E-6 /),(/1.7,1.7,2.2/)), ! Ju
& em_aero('UNITY_REF      ',(/1.0,1.0,1.0/),(/0.06E-6,0.28E-6 ,6.0E-6 /),(/1.7,1.7,2.2/)), ! Us
& em_aero('ZERO_REF       ',(/0.0,0.0,0.0/),(/0.06E-6,0.28E-6 ,6.0E-6 /),(/1.7,1.7,2.2/)), ! Ze
& em_aero('FINE_WBDUST    ',(/0.0,1.0,0.0/),(/0.06E-6,1.391E-6,5.26E-6/),(/1.7,2.0,2.0/)), ! De
& em_aero('COARSE_WBDUST  ',(/0.0,0.0,1.0/),(/0.06E-6,1.391E-6,5.26E-6/),(/1.7,2.0,2.0/)), ! De
& em_aero('FINE_SEASPRAY  ',(/0.0,1.0,0.0/),(/0.06E-6,1.391E-6,5.26E-6/),(/1.7,2.0,2.0/)), ! Fi
& em_aero('COARSE_SEASPRAY',(/0.0,0.0,1.0/),(/0.06E-6,1.391E-6,5.26E-6/),(/1.7,2.0,2.0/))  ! Co
                                                                                           !  T
```

```
                                                                                ! a
                                                                                ! a
& /)
```

Users can add as many new size distributions as they want, as long as they increment the variable n_em_aero_ref to always equal the number of size distributions in the lookup array (em_aero_ref).

CMAQ will use the size distribution reference value linked to each emissions scaling rule via the phase/mode keyword to calculate the fraction of each aerosol primary emission that should go into the 'i', 'j', and 'k' modes in the internal aerosol module. At first, it may seem that the linking step between phase/mode keywords in the EmissionsScalingRules section, the corresponding mode keywords in the SizeDistributions section, and the reference mode labels is unnecesary, but it serves an important function. As stated earlier, it is common that modes of similar size from a variety of sources will be referred to by common names like 'FINE' and 'COARSE', even though the size distribution parameters may differ considerably. With the linking step provided in the SizeDistributions section, parameters for several streams can be specified individually, but all be labeled 'FINE' and applied with one rule in the EmissionsScalingRules section.

In the example above, fine mode Wind-Blown Dust are linked to 'FINE_WBDUST', sea spray aerosols are linked to 'FINE_SEASPRAY' and all other sources are linked to 'FINE_REF'. Thus, different size distributions will be calculated for each of these streams. However, if the user wants to scale the mass of all fine mode aerosol by a factor of 2, the following emission rule is valid:

```
! Region      | Stream Label  |Emission | CMAQ-        |Phase/|Scale |Basis |Op
!  Label      |               |Surrogate| Species      |Mode  |Factor|      |
'EVERYWHERE'  , 'ALL'         ,'ALL'    ,'ALL'         ,'FINE',1.0   ,'UNIT','m',
```

B.6 Additional DESID Features

B.6.1 Summary Output to Processor-Specific Logfiles

Diagnostic output is an important feature of the new emissions module, DESID. Because the impact of emissions is so critical for CMAQ predictions and because the features available for scaling emissions are now quite complex, a comprehensive text-based output has been added to the CMAQ logfiles to enhance transparency.

The logfiles now provide several lists of information to help protect users from mistakes like inconsistent naming between emissions and CMAQ speciation. First, CMAQ reports for each stream the number and names of all the surrogate species that were not used. Second, it prints the names of surrogates that the user told it to look for but that it could not find on any of the emission streams. If the environment variable:

```
setenv CTM_EMISCHK Y        #> Abort CMAQ if missing surrogates from emissions Input files
```

is set to 'Y' or 'True', then the model will abort if it cannot find any individual surrogate. If the variable is set to 'N' or 'False' then CMAQ will print a warning and proceed.

Finally, CMAQ loops through streams and outputs the size distribution modes available for each stream and the full list of every emission instructions applied to each stream. These are ordered by CMAQ species (with 'i', 'j', and 'k' modes listed separately) and surrogate species name so that a full understanding of the scaling rules applied to each CMAQ species' emissions can be grasped quickly. Columns are printed for the applicable region(s) of the grid, the phase/mode applied, the input scale factor, the scaling basis, the operation, and the final scale factor applied taking into account any molecular weight conversions, if needed, and size distribution fractions.

B.6.2 Diagnostic Gridded Output Files

Many complex scaling procedures are now possible with DESID. Users are advised to confirm that the emissions are scaled the way they have intended. One tool to help this step is the Gridded Diagnostic Output. This is enabled on a stream-by-stream basis in the CMAQ RunScript with the following options:

```
# Gridded Emissions Diagnostic files
  setenv GR_EMIS_DIAG_001 TRUE
  setenv GR_EMIS_DIAG_002 2D

# Stack emissions diagnostic files
  setenv STK_EMIS_DIAG_001 2DSUM
  setenv STK_EMIS_DIAG_002 2DSUM
  setenv STK_EMIS_DIAG_003 FALSE
  setenv STK_EMIS_DIAG_004 2DSUM
  setenv STK_EMIS_DIAG_005 2DSUM
```

The lines above set the behavior of the gridded diagnostic output for gridded and inline emission streams. The values available for each stream are 'TRUE', 'FALSE', '2D', '2DSUM', and '3D'. The '2D' option prints just the surface layer of emissions for a particular stream. The '3D' option prints all layers populated by that stream. The '2DSUM' option prints one 2D field, but it equals the column of sum of emissions throughout the gridded model domain. The 'TRUE' option equates to '2D'. The user can also set the diagnostic behavior of online streams using the following variables:

```
setenv BIOG_EMIS_DIAG TRUE
setenv MG_EMIS_DIAG TRUE
setenv LTNG_EMIS_DIAG TRUE
setenv DUST_EMIS_DIAG TRUE
setenv SEASPRAY_EMIS_DIAG TRUE
```

The gridded diagnostic output files that are created are named systematically with the format "CCTM_EMDIAG_[XXX]*[CTM_APPL]*[DATE].nc" where XXX is the emissions stream label,

CTM_APPL is the application name defined in the CCTM runscript, and DATE is the date of the simulation. To change the default value for the diagnostic output of all emission streams, modify the "EMIS_DIAG" variable:

```
setenv EMIS_DIAG TRUE
```

This variable sets the default behavior for all streams. If the variables for any specific streams are provided in the RunScript, they will override this default value.

The emission rates printed to the diagnostic files reflect all the scaling rules applied and are written just before the emissions are added to the CMAQ transport module. Because the model interpolates in time, it is very likely that the rates written to the diagnostic file will not correspond in time to the rates from the input files. In most cases, the rates will be one-half time step before the top of the hour, the time point of the emission inputs. For this reason, it is not entirely helpful for users to compare the scaled emissions directly to the rates on the input files. However, comparing them qualitatively can be helpful.

Appendix C: Processing Spatial Data for CMAQ Inputs

C.1 Geospatial Data

Air quality modeling requires many spatial data to generate anthropogenic, biogenic, fire, sea salt, dust, and NH3 emissions. In addition, land surface characteristics such as land cover types with vegetation leaf area index (LAI) and fraction, albedo, and soil types are required in modeling the exchanges of heat, moisture, and momentum between the land and atmosphere and dry deposition of trace chemicals (e.g. O3 and NH3). It is important to use a consistent coordinate system for all the geospatial data used in emission, meteorology, and air quality modeling. Most of the geospatial data required for the Sparse Matrix Operator Kernel Emissions (SMOKE)/Weather Research and Forecasting (WRF)/CMAQ modeling can be generated using the Spatial Allocator (SA) that includes three components developed for specific applications: Vector, Raster, and Surrogate Tools.

In using the spatial data, it is important to know the datum, which is a spheroidal (either spherical or ellipsoidal) surface that represents the surface of the earth, and the projection, which is a mathematic transformation that converts a location on the datum to the location on a flat plane. The following sections briefly describe the appropriate datum and projections to use with the CMAQ system and the methods for generating the needed spatial data in the correct form.

C.2 Geodetic datum

A geodetic datum is a coordinate system used to define a location on the Earth. There are many datums used in spatial datasets depending on what geographic regions they are and how the Earth's surface is approximated as a spheroid. Most of U.S. geospatial data are defined in North American Datum of 1983 (NAD83) and the global data sets are often defined in World Geodetic System 1984 (WGS84).

WRF datasets are in WGS84. All latitude-longitude geographic data sets used in a CMAQ simulation, such as emissions shapefiles, land use or biogenic data files, and the ocean file, should be in WGS84 so that they are spatially aligned with the WRF files. For simulations over North America, NAD83 is only slightly different from the WGS84 datum. As a result, NAD83 can be used for North America domains without introducing spatial misalignment issues in the model datasets.

C.3 Spatial Data Projection

CMAQ can use any of the four map projections defined for WRF. The four map projection coordinate systems are regular latitude-longitude geographic, Lambert conformal conic, Mercator, and Polar stereographic. It is important to know that in projecting spatial data that is in WGS84 to the CMAQ projection or projecting CMAQ data to another map projection, users SHOULD NOT do any datum transformation. This is consistent with the WRF preprocessing system (WPS). Datum transformation will result in geographic location shifting.

The CMAQ domain projection is defined through the PROJ coordinate transformation software library using a spherical surface with an earth radius of 6370000 m to match the WRF domain projection definition. Once an input dataset is in WGS84 the following examples can be used to define the projection transformation needed to match the WRF data:

Lambert Conformal Conic: "+proj=lcc +a=6370000.0 +b=6370000.0 +lat_1=33 +lat_2=45 +lat_0=40 +lon_0=-97"

Polar stereographic: "+proj=stere +a=6370000.0 +b=6370000.0 +lat_ts=33 +lat_0=90 +lon_0=-97 +k_0=1.0"

Mercator: "+proj=merc +a=6370000.0 +b=6370000.0 +lat_ts=33 +lon_0=0"

Geographic: "+proj=latlong +a=6370000.0 +b=6370000.0"

C.4 Spatial Data Generation

Emission spatial allocation surrogates are required for generating anthropogenic emissions by SMOKE to spatially allocate county-based emission inventories to model grid cells. Emission surrogates can be based on population, roads, airports, railroads, and land use spatial data sets. The SA Vector and Surrogate Tools can be used to generate all needed emission surrogates for SMOKE.

- SA Vector and Surrogate Tools

Biogenic emissions require land use input including different tree species. There are two ways to compute the required input for the domain covering the continental U.S. (CONUS).

1. The original method—re-grid Biogenic Emissions Landuse Database, Version 3 (BELD3) using a SA Vector allocation tool. The BELD3 data is generated from the early 1990s AVHRR land cover data and FIA tree species at the county level.

2. The second method—use the SA Raster BELD4 land cover generation tool to generate model domain land use data with tree species. Then, a provided utility is used to convert the generated land cover data into an I/O API format for CMAQ input. The limitation for this tool is that the early 1990s county-level FIA tree species table is still used in allocating FIA tree species (this is also the case for the 1st approach).

- SA Vector and Surrogate Tools
- SA Raster BELD4 land cover generation tool

Fire emissions require fire location, burned areas, and detailed fuel load information. Fire locations are available via satellite detections from the Hazard Mapping System (HMS) or ground level reports from the National Fire and Aviation Management web application. Burn Area estimates can be obtained from GIS based sources such as the Geospatial Multi-Agency Coordination (GeoMac) website or the U.S. National Historical Fire Perimeters Data Basin Dataset. Fuel loading is estimated using a geospatial dataset such as the US Forest Service Fuel Characteristic Classification System (FCCS). All these information sources can be used to estimate fire emissions. An example of a tool that can be used to generate fire emissions is the BlueSky modeling framework. BlueSky modularly links a variety of independent models of fire information, fuel loading, fire consumption, fire emissions, and smoke dispersion. Using these tools and estimating fire emissions can be quite complex so datasets of fire emissions have been created for the community. Examples of such datasets are the Fire INventory from the National Center for Atmospheric Research (FINN) or the Global Fire Emissions Database (GFED).

- Hazard Mapping System Fire and Smoke Product
- National Fire and Aviation Management Web Application
- Geospatial Multi-Agency Coordination website
- U.S. National Historical Fire Perimeters Data Basin Dataset
- US Forest Service Fuel Characteristic Classification System
- US Forest Service BlueSky Modeling Framework

- Fire INventory from the National Center for Atmospheric Research
- Global Fire Emissions Database

Sea spray emissions require open ocean and surf zone (50m) buffer fractions for the modeling grid cells in an I/O API file. For most of North American domain, a SA Vector allocation tool can be used to generate the surf zone and open ocean file from a polygon shapefile with land, surf zone buffer, and open ocean in SA data directory. For areas outside U.S., users have to generate a surf zone polygon shapefile with has the same attribute as the file in the SA to use the tool. See the CMAQ Tutorial on creating an ocean file for step by step instructions on creating this CMAQ input file. Chapter 6 has additional information on sea spray module in CMAQ.

NH3 emissions from agricultural lands can be estimated using the CMAQ bi-directional NH3 model. The input for the CMAQ bi-directional NH3 model is generated by the Fertilizer Emission Scenario Tool for CMAQ (FEST-C) system. FEST-C contains three main components: Java interface, Environmental Policy Integrated Climate (EPIC) model, and SA Raster Tools. The interface guides users

through generating required land user and crop data and EPIC input files and simulating EPIC, and extracting EPIC output for CMAQ. The generated BELD4 land use data by FEST-C needs to be converted into an I/O API format using a utility program in FEST-C for CMAQ input. Note that the BELD4 data used for FEST-C is generated by the 2nd approach described above in Biogenic emission generation approaches.

- Fertilizer Emission Scenario Tool for CMAQ (FEST-C)

Land use and land cover data for surface flux modeling in meteorology and air quality can be generated using WPS or the SA Raster Tools. It is important to use consistent land use data in both meteorology and air quality modeling. For the U.S., WPS contains re-gridded 9-arc second (around 250 m resolution) 2011 NLCD land cover, imperviousness, and canopy data while 2011 MODIS land cover is used for areas outside the U.S. In addition, users can use the land use re-gridding tool in the SA Raster Tools system to generate land cover data for any domain directly using NLCD (at 30 m resolution) or/and MODIS land cover data (at 1 km or 500 m resolution). Users can use a provided R utility in SA to update their geogrid land cover data using the more accurate land cover data generating using SA.

Appendix D: Parallel Implementation of CMAQ

D.1 General Structure of data

There are a few approaches to parallelize an application, such as data-parallelism. Data-parallelism is a paradigm which decomposes data into “equal” sections and distributes them among allocated processors. Each processor works on the portion it owns. CMAQ parallel implementation is based on this methodology.

The CMAQ model operates on a 4D space (ncols, nrows, nlays, nspcs) and only the spatial domain is decomposed. When NPROCS processors are used to run CMAQ, NPCOL number of processors are assigned to the column dimension and NPROCS/NPCOL processors are assigned to the row dimension (NPROCS = NPCOL x NPROW). In the case that the column dimension is not divisible by NPCOL, the remainder is distributed equally to NPCOL processors. The same approach is applied to the row dimension. For example (illustrated in Figure D-1), given a 100 by 75 (column x row) data grid and six processors with three processors along the column dimension and two processors along the row dimension, the following subdomain sizes (NCOLS x NROWS) are assigned to each processor: 34 x 38 (processor 0), 33 x 38 (processors 1 and 2), 34 x 37 (processor 3), and 33 x 37 (processors 4 and 5).

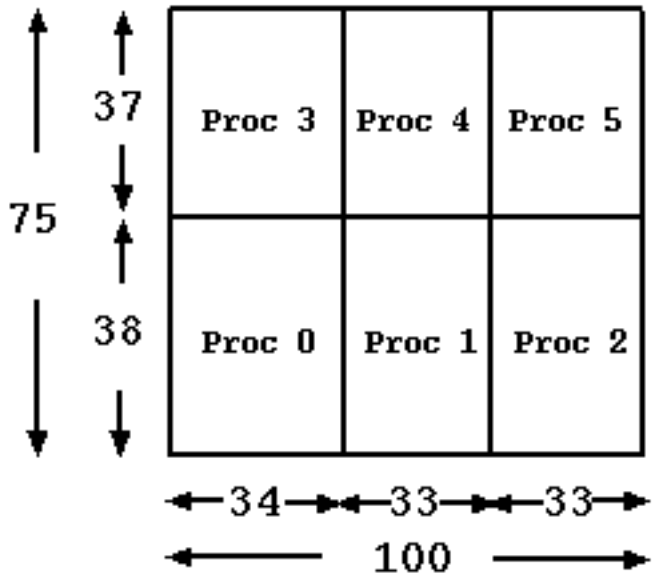


Figure D-1. Domain decomposition illustration

D.2 Interprocessor Communication

In some science processes such as advection, a processor requires data from neighboring processors (interprocessor communication) when the model runs on a distributed memory system. An interprocessor communication library, STENEX, was developed to provide a simple and robust interface to handle various kinds of near neighbor communication. Near neighbor is defined as processors which are adjacent to a given processor (blue block) in the eight major geographical directions: N, NE, E, SE. S, SW, W, and NW (Fig. D-2).

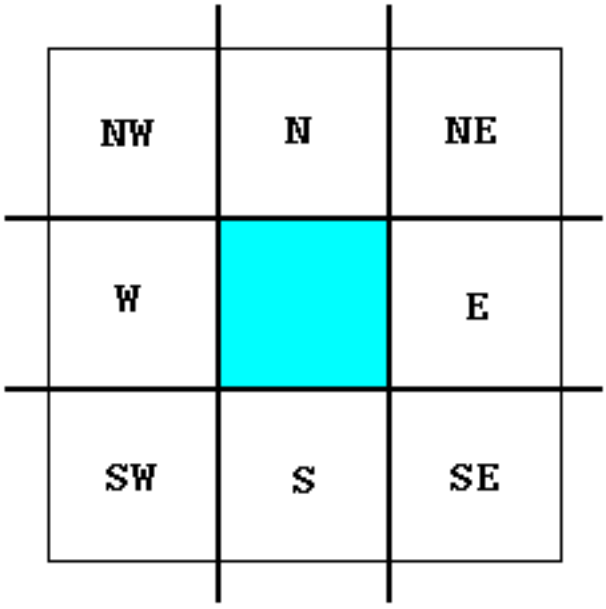


Figure D-2. A depiction of near neighbor processors

As an illustration of interprocessor data access (Fig. D-3), consider the following piece of code being executed on Processor 2 with a 2x2, 4-processor domain decomposition. It is clear that the calculation at the grid cell denoted by "X" requires data denoted by red dots which reside in near neighbor processors 0 and 3.

DIMENSION DATA(NCOLS, NROWS)

DO J = 1, NROWS DO I = 1, NCOLS DATA(I,J) = A(I+2,J) * A(I, J-1) END DO END DO

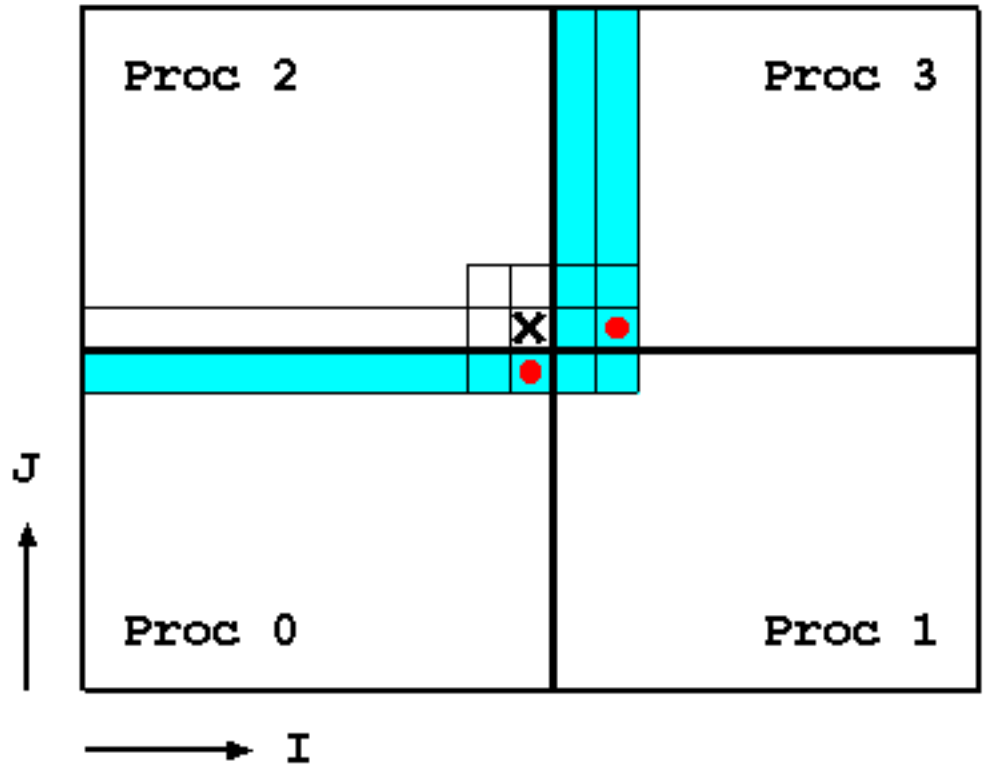


Figure D-3. An example to show interprocessor data access is needed

To facilitate interprocessor communication as shown in the example above, "ghost" regions are used (extra space in the data structure), i.e. DIMENSION DATA (NCOLS+2, NROWS+1). The thickness of the ghost region depends of the amount of overlap that is required by the algorithm.

The Stencil Exchange (SE) Library is designed in Fortran 90 using Object Oriented-base technology to handle various types of communication with the objective of hiding the management of the low-level data movement. SE addresses four types of communication and a brief description of each type is provided below.

- interior to ghost region, which is indicated in light blue in Figure D-4. This particular type of communication is being used in various places such as HADV and HDIFF.

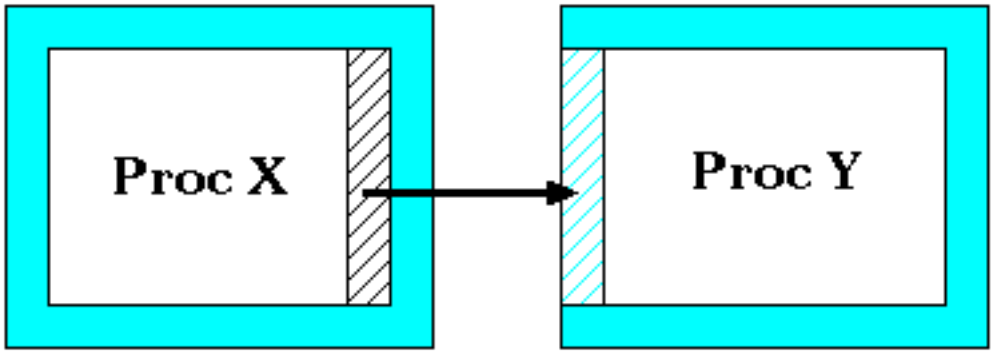


Figure D-4. Interior to ghost region communication

- sub-section data redistribution (Fig. D-5). This particular type of communication is being used in Process Analysis.

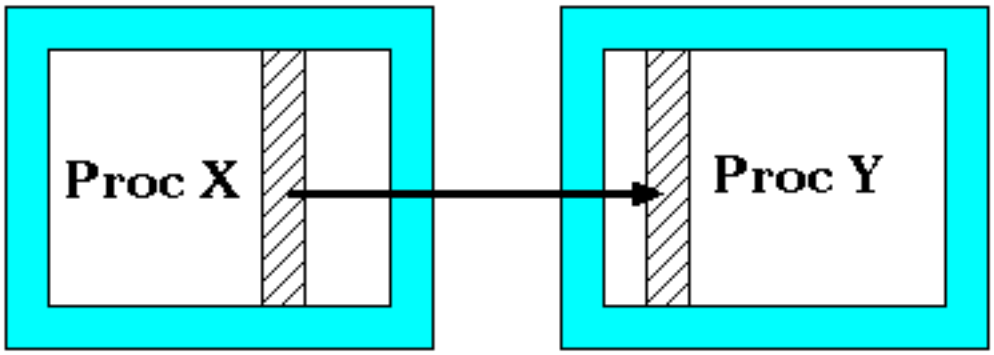


Figure D-5. Sub-section data redistribution communication

D.3 Parallel I/O

All I/O operations in CMAQ are handled by the I/O API library. Furthermore, the I/O API library was designed for serial code. As a result, CMAQ cannot utilize any I/O functions contained in the in I/O API library (such as READ3 and WRITE3) directly in any parallel computing platform.

CMAQv4.7.1 and later releases include a directory called 'PARIO' which was developed to bridge this gap. PARIO contains a smaller set of functions which are counterparts to equivalent functions in IOAPI but capable to run in parallel. The following I/O API routines have PARIO equivalents: READ3, INTERP3, WRITE3, CHECK3, OPEN3, CLOSE3, DESC3, M3ERR, M3EXIT, M3WARN. Each file name in the PARIO library has a "P" prefix to distinguish it from its counterpart in the I/O API library, e.g. POPEN3 and PINTERP3. Substitution with the PARIO subroutines is done at compilation through CPP flags. Note that the subroutine argument lists in any PARIO routine are identical to those in the I/O API counterpart routine.

On the output side, all processors are required to send their portion of data to processor 0, which will stitch together each sub-part and then output it to the file (Fig. D-8). This is considered a "pseudo" parallel I/O approach and this approach is being using in PARIO.

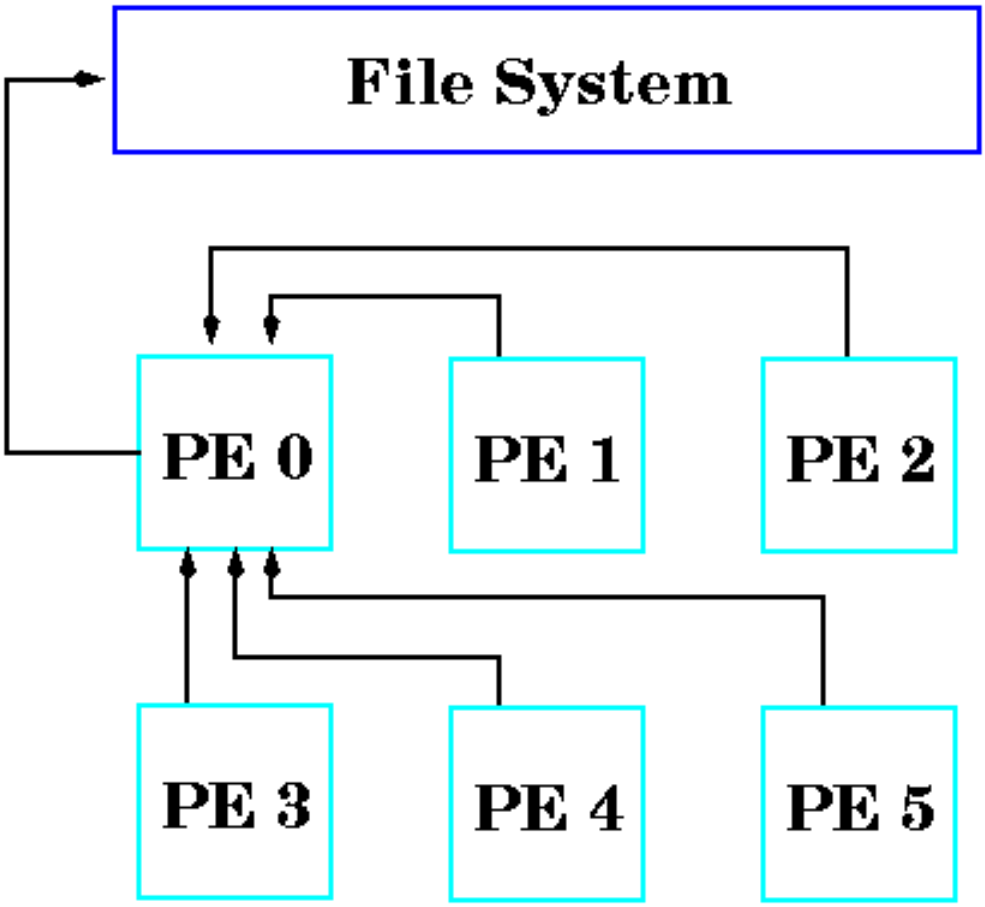


Figure D-8. Combine all sub-domain data from each processor in an I/O processor

In CMAQv5.2 and later versions, we have developed a true parallel I/O approach, referred to as PIO (Wong et. al.). PIO allows each processor to write their portion to the output file simultaneously (Fig. D-9).

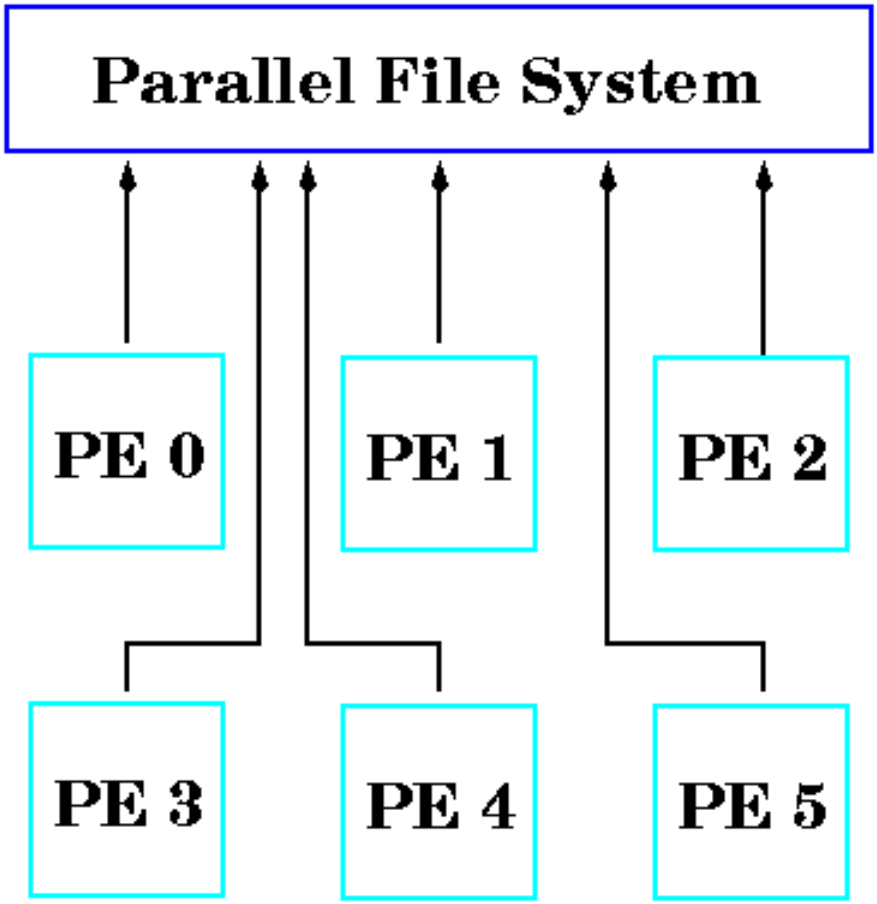


Figure D-9. True parallel I/O approach

To invoke this feature users have to re-build CMAQ after building additional libraries not used with CMAQ traditionally as well as retaining the traditional libraries with the exception of the non-"mpi" IOAPI library downloaded in Chapter 3. The additional libraries required by invoking this option include the PnetCDF library and the "mpi" version of the IOAPI library.

PnetCDF library

The PnetCDF library is the parallel I/O implementation to complement the classic netCDF library. The PnetCDF library is available for download at https://parallel-netcdf.github.io/ users should find and follow the instructions for proper installation given on the website. Users should install a stand alone PnetCDF library using MPI Fortran 90 and C compilers. After successful installation, check the environment PATH & LD_LIBRARY_PATH to ensure that the paths have been updated to include the path of the PnetCDF libraries and bin. Note that users may have to set these paths manually if not set, and that these paths must be loaded every time a new shell is started. Note: users should not re-build their netCDF library at this point, within CMAQ the classic netCDF library and PnetCDF library interact as two stand alone libraries.

IOAPI library

The I/O API library provides an interface between the netCDF libraries and CMAQ to handle input and output (I/O) calls throughout the CMAQ code. The latest version of the IOAPI library (version 3.2) is available for download at https://www.cmascenter.org/ioapi/documentation/all_versions/html/AVAIL.html#v32.

The general steps for installation of IOAPI libraries on a Linux system (with C-shell and GNU compilers) are below. These instructions are an example and we recommend using the latest release available at the time of your CMAQ installation.

This approach also requires installation of "mpi" I/O API libraries as shown below (note these steps should be followed after completing the steps in Chapter 3 section 3.2.3):

```
setenv BIN Linux2_x86_64gfortmpi
```

Edit the file in the ioapi folder called Makeinclude.Linux2_x86_64gfortmpi to comment out all openMP options as CMAQ does not support openMP. Note: If users are using the ifort compiler you also need to remove -Bstatic flag within the ioapi/Makeinclude.Linux2_x86_64ifortmpi file as well.

```
OMPFLAGS = # -fopenmp
OMPLIBS = # -fopenmp
```

In the top level IOAPI_3.2 directory run:

```
make configure
make
```

After building the reqiured libraries, users must build CCTM. Before compilation of CCTM, users must turn on this feature by uncommenting the following line in bldit_cctm.csh at the model build step and link with I/O API 3.2.

```
#set MakefileOnly                    #> uncomment to build a Makefile, but do not compile;
#set build_parallel_io               #> uncomment to build with parallel I/O (pnetcdf);
```

After building the BLD directory (where the Makefile lives), change to this directory and edit the Makefile to include PNETCDF and the correct I/O API BIN before compiling the code. An example of these edits are shown below:

```
LIB = /home/CMAQ_PIO/CMAQ_libs
include_path = -I /home/CMAQ_PIO/CMAQ_libs/ioapi_3.2/Linux2_x86_64ifortmpi \
               -I /home/CMAQ_PIO/CMAQ_libs/ioapi_3.2/ioapi/fixed_src \
               -I $(LIB)/mpi/include -I.

 IOAPI  = -L/home/CMAQ_PIO/CMAQ_libs/ioapi_3.2/Linux2_x86_64ifortmpi -lioapi
 NETCDF = -L$(LIB)/netcdf/lib -lnetcdf -lnetcdff
 PNETCDF = -L$(LIB)/pnetcdf/lib -lpnetcdf
 LIBRARIES = $(IOAPI) $(NETCDF) $(PNETCDF)
```

Lastly, users must also edit the CCTM run script by inserting MPI: in front of the output file path as shown below:

setenv CTM_CONC_1 "MPI:$OUTDIR/CCTM_CONC_${CTM_APPL}.nc -v" #> On-Hour Concentrations

For further directions on installation of PIO please contact David Wong at wong.david-c@epa.gov

D.4 Reference:

Wong, D.C., Yang, C.E., Fu, J.S., Wong, K., & Gao, Y. (2015). An approach to enhance pnetCDF performance in environmental modeling applications. Geosci. Model Dev., 8, 1033-1046.

Appendix E: Configuring the Weather Research and Forecasting Model (WRF) for Use with Air Quality Models

E.1 WRF version 4+

- WRF4.0 has updates to the ACM2 PBL model to account for the new default hybrid coordinate system. Our internal model runs suggest that the hybrid option (hybrid_opt =2) improves the model in areas where topographical variations are more extreme like the Rocky Mountains. As such, it is suggested, but not a requirement, to use this option in WRF that became the default in WRF4.0.

- Several important updates were made to the Pleim-Xiu LSM in WRF4.1. Soil hydraulics are calculated using analytical equations (Noilhan and Mahfouf, 1996) rather than lookup tables. And, users can opt to use time-varying vegetation fraction from a wrflowinp input file in WRF that can be based off MODIS satellite data instead of the old weighting method that is based on lookup tables and landuse fraction. This satellite data option is activated using the physics namelist option "pxlsm_modis_veg = 1".

E.2 WRF version 3.7

- **Section from WRFv3.7 Technical Documentation related to air quality modeling:** This 8 page pdf provides description and procedures for using the Pleim-Xiu LSM, ACM2 PBL and Pleim Surface Layer Scheme in WRF including best practices and namelist options.

E.3 WRF with lightning assimilation

- **WRF with Lightning Assimilation User's Guide:** This 3 page pdf describes how to run WRF with the lightning assimilation technique described in Heath et al. (2016). The assimilation method uses gridded lightning data to trigger and suppress sub-grid deep convection in Kain-Fritsch. The gridded lightning data (variable name is 'LNT') is read in through auxinput8. The lightning data is grouped into 30-min intervals and treated as simple zeros (no lightning) or ones (lightning) for the assimilation method. All of the necessary code modifications and data are described in the document.

- **WRF with Lightning Assimilation Code:** This .zip file (ltgda_wrf_16feb2017.zip; 220K) contains the registry and FORTRAN files with the updates needed to run WRF with lightning assimilation, as well as a generic Python script to grid lightning data to your WRF domain.

E.4 Reference:

Heath, N. K., J. E. Pleim, R. C. Gilliam, & D. Kang (2016). A simple lightning assimilation technique for improving retrospective WRF simulations, J. Adv. Model. Earth Syst., 8, 1806 – 1824, http://dx.doi.org/10.1002/2016MS000735.

Noilhan, J., & Mahfouf, J. F. (1996). The ISBA land surface parameterization scheme. Global and planetary Change, 13(1-4), 145-159.