Models-3/Community Multiscale Air Quality Model (CMAQ)
User's Guide to Alternative Particulate Matter and Cloud Modules

Prepared by

Betty Pun, Yang Zhang, Shiang-Yuh Wu,
Krish Vijayaraghavan and Christian Seigneur
Atmospheric & Environmental Research, Inc.
2682 Bishop Drive, Suite 120
San Ramon, CA 94583

Prepared for

Dr. Naresh Kumar
EPRI
3412 Hillview Avenue
Palo Alto, CA 94304

Document Number CP083-03-1b
December 2003

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

The U.S. Environmental Protection Agency (EPA) has developed and released the Models-3/Community Multiscale Air Quality Model (CMAQ) (Byun and Ching, 1999). Models-3/CMAQ is a three-dimensional grid-based air quality model that can be applied to simulate ozone (O_3) and other photochemical oxidants, particulate matter (PM), and the deposition of pollutants such as acids (i.e., sulfate, nitrate), nitrogen species, and toxic air pollutants. Under contract with EPRI and the California Air Resources Board (ARB), Atmospheric & Environmental Research, Inc. (AER), in collaboration with Professor John Seinfeld, California Institute of Technology (Caltech), has incorporated two new enhanced PM modules and an enhanced cloud module (Pun et al., 2001; Zhang et al., 2002, Zhang et al., 2004) into version 4.2.1 of Models-3/CMAQ, released in July 2001. Models-3/CMAQ is supported by EPA and the user is referred to EPA documentation to install and set up Models-3/CMAQ. This user's guide focuses on the selection and application of the new PM and cloud modules incorporated into Models-3/CMAQ.

The enhanced PM modules are two versions of the Model for Aerosol Dynamics, Reaction, Ionization, and Dissolution (MADRID), thereafter referred to as MADRID 1 and MADRID 2. Both modules use a sectional representation of the particle size distribution. The formation of new particles is treated by calculating the relative rates of condensation and new particle formation given a minimum particle diameter threshold for the particle size distribution. Coagulation is neglected. Gas-to-particle conversion is treated for both inorganic and organic species. ISORROPIA (Nenes et al, 1999) is used to simulate the thermodynamic equilibrium of inorganic species. For organic species, MADRID 1 uses an empirical approach based on the results of smog chamber experiments conducted at Caltech, whereas MADRID 2 uses a mechanistic approach with ten surrogate organic compounds. MADRID 1 is coupled to the Carbon-Bond Mechanism Version IV (CBM-IV), the RADM2, and the RADM2_CI4 (RADM2 chemistry with the 4-product isoprene chemistry) for gas-phase chemistry whereas MADRID 2 is coupled to the Caltech Atmospheric Chemistry Mechanism (CACM) (Griffin et al., 2002). The transfer of chemical species from the gas phase to the particles can be simulated using either a full equilibrium assumption or hybrid algorithms that

approximate the kinetic process but are more computationally efficient. Growth of particles by condensation (and shrinkage by volatilization) is simulated with the moving-center method of Jacobson (Jacobson and Turco, 1995; Jacobson, 1997), which minimizes numerical diffusion and allows both particle mass and number concentrations to be tracked accurately. The PM number concentrations are explicitly treated by accounting for their changes due to various atmospheric processes (e.g., emission, diffusion, advection, deposition, and new particle formation). Table 1-1 summarizes the various components of MADRID.

For cloud chemistry, MADRID functions with either the RADM module or the CMU cloud chemistry. The RADM module was modified from the original CMAQ release to be compatible sectional PM, while the CMU module was implemented within CMAQ-MADRID. The CMU cloud chemistry module was developed by Professor Spyros Pandis and co-workers at Carnegie-Mellon University (CMU) (Strader et al., 1998). The CMU module provides a detailed representation of gas/liquid equilibria, aqueous equilibria and aqueous reactions among 69 species. It is coupled to both versions of MADRID.

The formulations of MADRID and of the CMU cloud chemistry module are described in detail in an article in press in the Journal of Geophysical Research, Atmospheres (Zhang et al., 2004) and the EPRI report titled "CMAQ-MADRID Technical Documentation" by Zhang et al. (2002). A copy of the EPRI report is provided under the "doc" directory in the current CMAQ-MADRID distribution.

Table 1-1. Processes simulated in MADRID. (Transport processes are those of CMAQ and are not listed here.)

Process	Module (Options are indicated with numbers)	Comments
Gas-phase chemistry	1. CBM-IV 2. RADM2 3. RADM2-CI4 4. CACM	CBM-IV, RADM2, and RADM2-CI4 modified to account for additional VOC for SOA formation and heterogeneous reactions. CBM-IV and RADM2-CI4 are solved with both the QSSA and the MEBI gas-phase chemistry solvers, and RADM2 and CACM are solved with the QSSA gas-phase chemistry solver.
Gas-particle thermodynamic equilibrium for inorganic species	ISORROPIA version 1.6 (sulfate, nitrate, ammonium, sodium, chloride, water)	ISORROPIA 1.6 is used in the July 2002 version of CMAQ.
Gas-particle equilibrium for organic species	1. Empirical partition coefficients for absorption into an organic phase of 4 SOA from anthropogenic precursors and 34 SOA from biogenic precursors (MADRID 1) 2. Gas/particle thermodynamic equilibrium with 10 surrogate compounds by aqueous dissolution or absorption into an organic phase (MADRID 2)	Partition coefficients from Odum et al. (1997) and Griffin et al. (1999); compatible with CBM-IV, RADM2, and RADM2-CI4. See Pun et al. (2002) for module description; compatible with CACM.
Particle size distribution	Sectional with at least 2 size sections 1. 2-section representation (fine and coarse particles) 2. Multi-section representation	The Stokes diameter is used to define the size section boundaries; note that the $PM_{2.5}$ and PM_{10} definitions are based on the aerodynamic diameter.
Coagulation	None	Coagulation is negligible compared to other processes under most conditions
Nucleation	1. New particle formation theory of McMurry and Friedlander (1979) 2. None	Look-up table is the default option since the full model may be too computationally demanding for 3-D simulations. Nucleation can be neglected under conditions with high PM concentrations (e.g., polluted urban environment).
Condensational growth/shrinkage by volatilization	Diffusion-limited condensation/volatilization using the moving-center algorithm	Condensational growth algorithms do not apply to the two-section option Moving-center algorithm tracks both mass and number concentrations

Table 1-1. Processes simulated in MADRID (continued). (Transport processes are those of CMAQ and are not listed here.)

Process	Module (Options are indicated with numbers)	Comments
Gas/particle mass transfer for inorganic species	1. Hybrid algorithm – CIT 2. Hybrid algorithm – CMU 3. Full equilibrium algorithm	The CIT hybrid approach is the default approach. The CMU hybrid approach simulates explicit mass transfer for particles with diameter > 2.15 μm. It is computationally more expensive and can be numerically unstable under some conditions.
Gas/particle mass transfer for organic species	1. Hybrid algorithm – CIT 2. Hybrid algorithm – CMU 3. Full equilibrium algorithm	SOA formation occurs on fine particles when full equilibrium is selected; it occurs on fine particles with the CMU hybrid algorithm.
Cloud chemistry	1. None 2. RADM 3. CMU	
Heterogeneous chemistry	1. None 2. Four reactions with PM and one reaction with droplets	Jacob (2000). Three sets of reaction probabilities (i.e., base, lower bound, and upper bound) recommended by Jacob (2000) may be selected. The base values are the default values.
Dry deposition	Integrated flux approach	Venkatram and Pleim (1999)
Wet deposition	In-cloud (rainout) and below-cloud (washout) scavenging of gases and particles	Effective Henry's law constants used for gases

2. USING MODELS-3 WITH ENHANCED PM AND CLOUD MODULES

2.1 Installation of New Modules and Compilation of Models-3/CMAQ

Models-3/CMAQ provides modularity based on the operator splitting approach and the source codes are arranged in the repository based on a directory structure that follows the science processes: (1) aero (PM thermodynamics and kinetics), (2) aero_depv (PM deposition), (3) chem (gas-phase chemistry and solver), (4) ping (plume in grid), (5) cloud (cloud dynamics and chemistry), (6) hadv (horizontal advection), (7) vadv (vertical advection), (8) hdiff (horizontal diffusion), and (9) vdiff (vertical diffusion). The routines for initialization and photolytic rate calculation are included in "init" and "phot", respectively. There are also other directories for utility routines, driver routines, and codes for process analysis.

At present, CMAQ-MADRID is implemented within the July 2002 version of Models-3/CMAQ. All source codes, including a complete set of the original CMAQ routines from the July 2002 distribution, are provided in the accompanying tape. The new CMAQ-MADRID source code files are incorporated into the Models-3/CMAQ repository under the various existing process directories, with one exception. A new directory called "common_MADRID" has been created to store an ordinary differential equation solver that is used in the simulations with particles or clouds, or both. If the user is running a different version of Models-3/CMAQ, we recommend setting up a new repository for CMAQ-MADRID in parallel with the CMAQ repository. If the user is using the July 2002 version of CMAQ, the source code directory provided in the tape can replace the /*/models/CCTM/src/ directory on the local computer's Models-3 repository during installation. A comparison with the original code will show that several new subdirectories have been added, as listed in Table 2-1. While the pa_MADRID module is functional in MADRID simulations, processes analysis results have not been analyzed for MADRID.

Several MADRID-relevant subroutines have also been added in the CMAQ's existing subdirectories such as "aero_noop" and "cloud_noop" to handle the simulations

Table 2-1. Incorporation of new modules into the Models-3 repository.

Parent Directory	New Subdirectory	Comments
aero/	aero_MADRID1_qssa or aero_MADRID1_mebi	The subroutine "aerorate.F" depends on the gas-phase chemistry solvers. Include files are mechanism- and size-dependent
	aero_MADRID2_qssa	The subroutine "aerorate.F" works for the qssa solver. Currently operational for 2 size sections
cloud/	cloud_radm_aero_MADRID1_qssa or cloud_radm_aero_MADRID1_mebi	RADM cloud chemistry for use with aero_MADRID1_qssa or aero_MADRID1_mebi. The subroutine "cldrate.F" depends on the gas-phase chemistry solvers
	cloud_CMU_aero_MADRID1_qssa or cloud_CMU_aero_MADRID1_mebi	CMU cloud module that works with aero_MADRID1_qssa or aero_MADRID1_mebi. The subroutine "cldrate.F" depends on the gas-phase chemistry solvers
	cloud_CMU_aero_MADRID2_qssa	CMU cloud module that works with aero_MADRID2_qssa. The subroutine "cldrate.F" depends on the gas-phase chemistry solvers
chem/	qssa_MADRID1	QSSA solver for CBM-IV, RADM 2, and RADM2-CI4 gas-phase chemistry with heterogeneous reactions
	mebi_cb4_MADRID1	MEBI solver for CBM-IV gas-phase chemistry with heterogeneous reactions. MEBI solver is hard-coded for CBM-IV mechanism
	mebi_radam2_cis4_MADRID1	MEBI solver for RADM2-CI4 gas-phase chemistry with heterogeneous reactions. MEBI solver is hard-coded for RADM2-CI4 mechanism
	qssa_MADRID2	QSSA solver for CACM gas-phase chemistry with heterogeneous reactions
phot/	phot_aqCMU	used with CMU cloud modules
init/	init_MADRID	used for all sectional simulations
vdiff/	eddy_MADRID	used for all sectional simulations
aero_depv/	aero_depv_MADRID	used for all sectional simulations
procan/	pa_MADRID	used for all sectional simulations
driver/	ctm_MADRID1	Special computing structure for driver.F and sciproc.F. Used with aero_MADRID1_qssa or aero_MADRID1_mebi
	ctm_MADRID2	Special computing structure for sciproc.F. sciproc.F modified to call PM module only once per hour. Recommended with aero_MADRID2_qssa

without particles or clouds. Two subroutines (aerorate_noop.F and movebin_noop.F) are added in "aero_noop" to turn off particle growth and heterogeneous reactions on the surface of particles when particles are not simulated. Similarly, one subroutine (cldrate_noop.F) is added in "cloud_noop" to turn off heterogeneous reactions in cloud droplets. These "aero_noop" and "cloud_noop" modules work with both the original modal CMAQ and CMAQ-MADRID.

The new modules in CMAQ-MADRID comply with Models-3's coding standard. Therefore, these modules should be easily transferable to future versions of Models-3/CMAQ, unless there are significant changes in Models-3/CMAQ's data structures and subroutine arguments. In that case, modifications may be necessary based on the design changes in Models-3/CMAQ.

The file "modules" in CCTM/CVSROOT/ directory is modified to reflect the addition of new modules and/or new classes in the Models-3/CMAQ repository. Since "common_MADRID" is a new class of science directory, it should be added in the repository and the new ODE solver module "ae_aq_solver_MADRID" should be added under the new CLASS called "common_MADRID". This is done by adding the following lines in the file "modules":

```
   common_MADRID_C  -s CLASS_common_MADRID  src/common_MADRID
ae_aq_solver_MADRID -s MODULE_common_MADRID src/common_MADRID/ae_aq_solver_MADRID
```

Other new "MODULE" entries are added under each "CLASS." A simple example is shown below:

```
vdiff           -s CLASS_vdiff       src/vdiff
eddy_MADRID     -s MODULE_vdiff      src/vdiff/eddy_MADRID
```

A new "modules" file is provided with the codes.

Include files specific to the processes are required to build the executable code. The subdirectories of include files are located in the Models-3/CMAQ include/release/ directory. Each combination of gas-phase chemistry, aerosol chemistry and dynamics, aqueous-phase chemistry modules, and particle size resolution requires an appropriate directory of include files. As a result of the incorporation of one new and three modified gas-phase mechanisms (CACM, CBM-IV, RADM2, and RADM2-CI4) with two new

aerosol modules (MADRID 1 and MADRID 2) and one new and one modified cloud modules (CMU and RADM), there are 25 new directories of include files for 25 mechanism/cloud module/PM module/particle size combinations in the Models-3/CMAQ include/release/ directory. Each of the directories of include files is labeled by the gas-phase chemistry, aqueous-phase chemistry, aerosol module, followed by the number of sections for the application. The appropriate include directory to be used with specific applications will be discussed in greater detail in Section 2.2.

In each include file directory, there are two files to describe the gas-phase mechanism and the heterogeneous reactions (i.e., RXCM.EXT and RXDT.EXT). The file called "mech.def" in each directory is the input file for the Models-3 chemistry reader used to generate RXCM.EXT and RXDT.EXT, and provides a description of all the gas-phase and heterogeneous reactions, including those responsible for the formation of SOA. There are four module-specific include files (i.e., PARAMETER.EXT, ARRAYMAP.EXT, AQ_PARAMS.EXT and HETERPAR.EXT) to specify constants, variables, arrays, species mapping, and number of size sections that are used for specific mechanism/particle size combinations. The file "PARAMETER.EXT" is needed for all MADRID simulations regardless of the mechanism/particle size combination. The file "ARRAYMAP.EXT" is needed for simulations with particles (i.e. gas+aerosol and gas+aerosol+cloud). The file "AQ_PARAMS.EXT" is needed for simulations with clouds (i.e., gas+cloud and gas+aerosol+cloud). The file "HETERPAR.EXT" is needed for simulations with particles, or clouds, or both. Options within CMAQ-MADRID are selected using flags in PARAMETER.EXT and HETERPAR.EXT as follows:

PARAMETER.EXT:

- IAERORATE (1 to turn on heterogeneous reactions on aerosol surfaces, 0 to turn off. 0 is always used in simulations without aerosols.)
- ICLDRATE (1 to turn on cloud heterogeneous reactions, 0 to turn off. 0 is always used in simulations without clouds.)
- IAPART (1 to select CIT hybrid algorithm for gas/particle mass transfer; 2 to select full equilibrium approach, 3 to select CMU hybrid algorithm.)
- INUCL (1 to turn on nucleation; 0 to turn off.)

HETERPAR.EXT:

- NGAMMA (1 to select the base values of reaction probability for heterogeneous reactions on the surface of particles; 2 to select the lower bound values, 3 to select the upper bound values.)

Where applicable, a default value of 1 is currently selected for the above flags in CMAQ-MADRID.

In addition, the include directory includes files for gas-phase species (GC_*.EXT), aerosol species (AE_*.EXT), and non-reactive species (NR_*.EXT) to specify species names, process participation, output, and correspondence between species in different phases. A list of files for gas-phase species is provided in Table 2-2. Corresponding files for aerosol, non-reactive, and tracer species have similar structures and functions. For example, the list of species for each PM module can be found in AE_SPC.EXT. Note that only one file is used for the correspondence of the aerosol species: AE_A2AQ.EXT.

The include files GC_*.EXT, AE_*.EXT and NR_*.EXT are specific to each application, because (1) MADRID 1, MADRID 2, and the original Models-3 modal PM module treat different organic aerosols; (2) MADRID treats the additional inorganic species of HCl and, in MADRID 2 only, a nonreactive organic compound BUTAN; and (3) the CMU and RADM aqueous-phase chemistry modules treat different sets of aqueous-phase equilibria and kinetic reactions among different gaseous and aqueous-phase species. In MADRID, the mapping of gas-phase and non-reactive species to aerosol species are defined in ARRAYMAP.EXT. As a result, the include files G2AE.EXT and N2AE.EXT are not used in MADRID simulations.

2.2 Compilation of CMAQ-MADRID

Models-3/CMAQ can be compiled from the graphical user interface (GUI) or using scripts. The new modules can be selected in the same manner as the default

Table 2-2. Gas-phase species extension files (GC_*.EXT).

Filename (* =)	Description	Comments
SPC	Species name and general characteristics	Corresponds to the gas-phase mechanism
ADV	Species that undergo advection	
DEPV	Species that undergo dry deposition and deposition velocity data	
DIFF	Species that undergo diffusion	
EMIS	Species with emission and emission factor data	
ICBC	Parameters for initial and boundary conditions	
SCAV	Species that are scavenged by rain (wet deposition)	needed for simulations with clouds
CONC	species with concentration output	
DDEP	species with dry deposition output	
WDEP	Species with wet deposition output	needed for simulations with clouds
G2AE	Correspondence between gas-phase and aerosol species	G2AE.EXT not used with CMAQ-MADRID
G2AQ	Correspondence between gas-phase and aqueous species	needed for simulations with clouds

options provided by EPA using a script for compilation. In the code-building script, the paths of the include directories are specified, along with the selected modules.

MADRID is a Fortran code, with the exception of the SOA module in aero_MADRID 2, which is composed of both Fortran and C routines. During compilation, all (Fortran and C) source code files are copied from the repository into a separate source code directory corresponding to the particular application. The script used for compilation automatically selects the appropriate compiler for each piece of code. The machine codes generated by the compiler, contained in objective files, are linked with the appropriate libraries to build the executable in the source code directory for the application. Include files are not copied into the directory where the code is built, and they remain in the include directory of the repository.

Two sample scripts are shown below for the examples to be discussed in Section 4. In the script shown in Appendix A, aero_MADRID1_mebi is selected for PM with 8 size sections and cloud_CMU_aero_MADRID1_mebi is selected for cloud processes. In the script shown in Appendix B, the aero_MADRID2_qssa module (2 sections) is selected in a model that does not contain cloud processes.

Figure 2-1 illustrates the associations among different gas-phase mechanisms, PM modules, and cloud modules. The user should not combine codes from the original modal version of CMAQ and the sectional CMAQ-MADRID in one application.

The thermodynamics of inorganic species and aerosol dynamics are treated similarly in the two MADRID modules. The organic condensable compounds are specifically produced in different mechanisms. There are, therefore, compatibility requirements with the two SOA modules that treat different organic condensable species. The Odum/Griffin SOA module in aero_MADRID1_qssa (or aero_MADRID1_mebi) is used with modified versions of CBM-IV, RADM2, and RADM2-CI4. On the other hand, the AEC module in aero_MADRID2_qssa is compatible with CACM. Therefore, the new PM modules are designed to be used in combination with specific gas-phase chemical mechanisms.

A modified version of the original Models-3/CMAQ cloud module (RADM) was developed to function with MADRID 1. At present, the RADM cloud module cannot be applied with MADRID 2. The CMU cloud module can interface with either MADRID 1

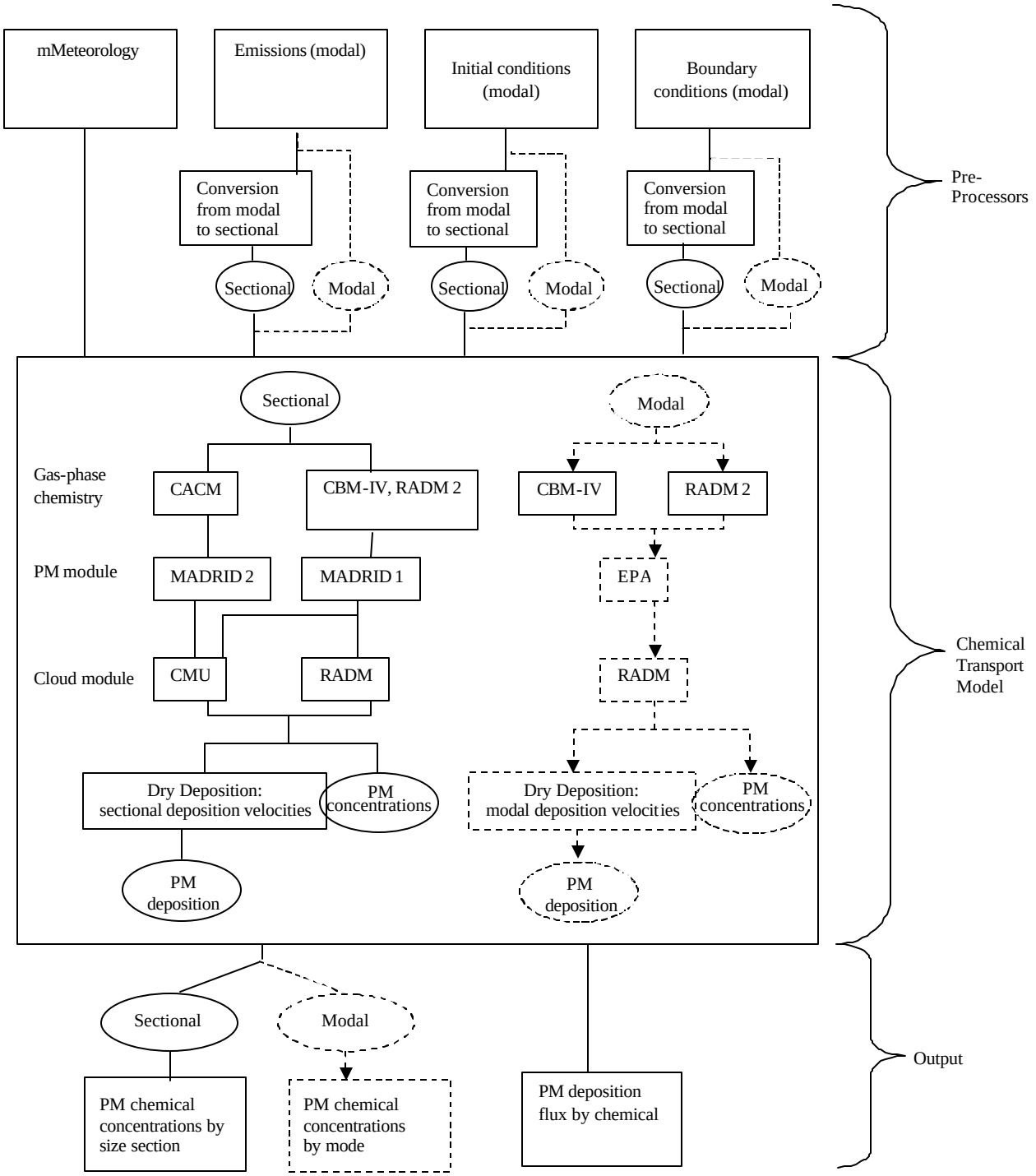

Figure 2-1. Flow chart indicating the associations of various gas-phase mechanisms, PM modules, and cloud modules. RADM2-CI4 is not represented explicitly in this figure, although it can be applied with both sectional and modal models.

or MADRID 2, and their corresponding gas-phase mechanisms (i.e., CBM-IV, RADM2 or RADM2_CI4 for MADRID 1 and CACM for MADRID 2).

The current release of MADRID supports a total of 16 combinations of gas-phase mechanisms, aerosol modules, and cloud modules. For CBM-IV and RADM2-CI4, these combinations can work with two different gas-phase chemistry solvers (i.e., QSSA and MEBI), two different horizontal diffusion modules (i.e., unif and multiscale), resulting in a total of 16 application configurations for each mechanism family (i.e., 4 mechanism combinations x 2 gas-phase chemistry solvers x 2 horizontal diffusion modules). For RADM2 and CACM, these combinations can work with two different horizontal diffusion modules (i.e., unif and multiscale) but only one gas-phase chemistry solver (i.e., QSSA), resulting in a total of 8 application configurations for each mechanism family (i.e., 4 mechanism combinations x 1 gas-phase chemistry solver x 2 horizontal diffusion modules). In addition, two size resolutions (i.e., 2 and 8 size sections) can be selected for applications with aerosols for mechanism families of CBM-IV, RADM2 and RADM2-CI4. Note that since QSSA solves gas-phase chemistry by blocks, whereas MEBI solves it by individual grid cells, different versions of aerorate.F and cldrate.F are needed to compute the heterogeneous reaction rate constants within the gas-phase chemistry modules with QSSA or MEBI. Correspondingly, there are two MADRID1 aerosol modules (i.e., aero_MADRID1_qssa and aero_MADRID1_mebi), two CMU cloud modules (i.e., cloud_CMU_aero_MADRID1_qssa and cloud_CMU_aero_MADRID1_mebi) and two RADM cloud modules (i.e., cloud_RADM_aero_MADRID1_qssa and cloud_RADM_aero_MADRID1_mebi). These module pairs differ only in whether the block or cell version of "aerorate.F" and/or "cldrate.F" is used.

Table 2-3 lists 4 selected application combinations of modules for each mechanism family. When developing a new application, the user is encouraged to compare the selected modules in the build script against those listed in Table 2-3. For each combination of gas, PM, and aqueous modules, the appropriate include directory (also listed in Table 2-3) needs to be specified in the build script. Example build scripts (LINUX platform) are provided in the tape.

Table 2-3a. Configurations of modules and mechanism available in CMAQ-MADRID: CBM-IV, MADRID 1 and/or CMU.

Configuration*	CBM-IV + MADRID 1 (no cloud)	CBM-IV + CMU (no aerosol)	CBM-IV + CMU + MADRID 1	CBM-IV + RADM + MADRID 1
Mechanism/ Include directory	cb4_aeMADRID1_2sec cb4_aeMADRID1_8sec	cb4_aqCMU	cb4_aqCMU_aeMADRID1_2sec cb4_aqCMU_aeMADRID1_8sec	cb4_aqRADM_aeMADRID1_2sec cb4_aqRADM_aeMADRID1_8sec
Module				
ModDriver	ctm_MADRID1	ctm_MADRID1	ctm_MADRID1	ctm_MADRID1
ModPar	par_noop	par_noop	par_noop	par_noop
ModInit	init_MADRID	init_MADRID	init_MADRID	init_MADRID
ModAdjc	denrate	denrate	denrate	denrate
ModCpl	gencoor	gencoor	gencoor	gencoor
ModHadv	hbot_MADRID	hbot_MADRID	hbot_MADRID	hbot_MADRID
ModVadv	vbot	vbot	vbot	vbot
ModHdiff	Multiscale$^{\xi}$	multiscale$^{\xi}$	multiscale$^{\xi}$	multiscale$^{\xi}$
ModVdiff	eddy_MADRID	eddy_MADRID	eddy_MADRID	eddy_MADRID
ModPhot	phot	phot_aqCMU	phot_aqCMU	phot
ModPing	ping_noop	ping_noop	ping_noop	ping_noop
ModChem	mebi_cb4_MADRID1$^{\phi}$	mebi_cb4_MADRID1$^{\phi}$	mebi_cb4_MADRID1$^{\phi}$	mebi_cb4_MADRID1$^{\phi}$
ModAero	aero_MADRID1_mebi$^{\phi}$	aero_noop	aero_MADRID1_mebi$^{\phi}$	aero_MADRID1_mebi$^{\phi}$
ModAdepv	aero_depv_MADRID	aero_depv_noop	aero_depv_MADRID	aero_depv_MADRID
ModCloud	cloud_noop	cloud_CMU_aero_MADRID1_mebi$^{\phi}$	cloud_CMU_aero_MADRID1_mebi$^{\phi}$	cloud_RADM_aero_MADRID1_mebi$^{\phi}$
ModPa	pa_MADRID	pa_MADRID	pa_MADRID	pa_MADRID
ModUtil	util	util	util	util
Modaeaqsolver	ae_aq_solver_MADRID	ae_aq_solver_MADRID	ae_aq_solver_MADRID	ae_aq_solver_MADRID

* The combination of gas-phase mechanism, cloud module, and aerosol module. CBM-IV, MADRID 1, CMU and RADM denote the CBM-IV gas-phase chemical mechanism, the MADRID 1 aerosol module, the CMU aqueous-phase chemical mechanism and the RADM aqueous-phase chemical mechanism, respectively. The CBM-IV gas-phase mechanism includes heterogeneous reactions on surface of particles and droplets; these reactions are not simulated when aqueous-phase chemistry and/or aerosol processes are not selected. They can also be neglected by the user. Two flags (IAERORATE and ICLDRATE) are included in PARAMETER.EXT to provide options to turn on/off these heterogeneous reactions.

$^{\xi}$ "Multiscale" can be replaced by "unif".

$^{\phi}$ When QSSA is used as a gas-phase chemistry solver, the modules "mebi_cb4_MADRID1", "aero_MADRID1_mebi", and "cloud_CMU_aero_MADRID1_mebi" (or "cloud_RADM_aero_MADRID1_mebi") should be replaced by "qssa_MADRID1", "aero_MADRID1_qssa" and "cloud_CMU_aero_MADRID1_qssa" (or "cloud_RADM_aero_MADRID1_qssa"), respectively.

Table 2-3b. Configurations of modules and mechanism available in CMAQ-MADRID: RADM2, MADRID 1 and/or CMU.

Configuration*	RADM2 + MADRID 1 (no cloud)	RADM2 + CMU (no aerosol)	RADM2 + CMU + MADRID 1	RADM2 + RADM + MADRID 1
Mechanism/ Include directory	radm2_aeMADRID1_2sec radm2_aeMADRID1_8sec	radm2_aqCMU	radm2_aqCMU_aeMADRID1_2sec radm2_aqCMU_aeMADRID1_8sec	radm2_aqRADM_aeMADRID1_2sec radm2_aqRADM_aeMADRID1_8sec
Module				
ModDriver	ctm_MADRID1	ctm_MADRID1	ctm_MADRID1	ctm_MADRID1
ModPar	par_noop	par_noop	par_noop	par_noop
ModInit	init_MADRID	init_MADRID	init_MADRID	init_MADRID
ModAdjc	denrate	denrate	denrate	denrate
ModCpl	gencoor	gencoor	gencoor	gencoor
ModHadv	hbot_MADRID	hbot_MADRID	hbot_MADRID	hbot_MADRID
ModVadv	vbot	vbot	vbot	vbot
ModHdiff	multiscale$^{\xi}$	multiscale$^{\xi}$	multiscale$^{\xi}$	multiscale$^{\xi}$
ModVdiff	eddy_MADRID	eddy_MADRID	eddy_MADRID	eddy_MADRID
ModPhot	phot	phot_aqCMU	phot_aqCMU	phot
ModPing	ping_noop	ping_noop	ping_noop	ping_noop
ModChem	qssa_MADRID1	qssa_MADRID1	qssa_MADRID1	qssa_MADRID1
ModAero	aero_MADRID1_qssa	aero_noop	aero_MADRID1_qssa	aero_MADRID1_qssa
ModAdepv	aero_depv_MADRID	aero_depv_noop	aero_depv_MADRID	aero_depv_MADRID
ModCloud	cloud_noop	cloud_CMU_aero_MADRID1_qssa	cloud_CMU_aero_MADRID1_qssa	cloud_RADM_aero_MADRID1_qssa
ModPa	pa_MADRID	pa_MADRID	pa_MADRID	pa_MADRID
ModUtil	util	util	util	util
Modaeaqsolver	ae_aq_solver_MADRID	ae_aq_solver_MADRID	ae_aq_solver_MADRID	ae_aq_solver_MADRID

* The combination of gas-phase mechanism, cloud module, and aerosol module. RADM2, MADRID 1, CMU and RADM denote the RADM2 gas-phase chemical mechanism, the MADRID 1 aerosol module, the CMU aqueous-phase chemical mechanism and the RADM aqueous-phase chemical mechanism, respectively. The RADM2 gas-phase mechanism includes heterogeneous reactions on surface of particles and droplets; these reactions are not simulated when aqueous-phase chemistry and/or aerosol processes are not selected. They can also be neglected by the user. Two flags (IAERORATE and ICLDRATE) are included in PARAMETER.EXT to provide options to turn on/off these heterogeneous reactions.

$^{\xi}$ "Multiscale" can be replaced by "unif"

Table 2-3c. Configurations of modules and mechanism available in CMAQ-MADRID: RADM2_CI4, MADRID 1 and/or CMU.

Configuration*	RADM2_CI4 + MADRID 1 (no cloud)	RADM2_CI4 + CMU (no aerosol)	RADM2_CI4 + CMU + MADRID 1	RADM2_CI4 + RADM + MADRID 1
Mechanism/ Include directory	radm2_ci4_aeMADRID1_2sec radm2_ci4_aeMADRID1_8sec	radm2_ ci4_aqCMU	radm2_ci4_aqCMU_aeMADRID1_2sec radm2_ci4_aqCMU_aeMADRID1_8sec	radm2_ci4_aqRADM_aeMADRID1_2sec radm2_ci4_aqRADM_aeMADRID1_8sec
Module				
ModDriver	ctm_MADRID1	ctm_MADRID1	ctm_MADRID1	ctm_MADRID1
ModPar	par_noop	par_noop	par_noop	par_noop
ModInit	init_MADRID	init_MADRID	init_MADRID	init_MADRID
ModAdjc	denrate	denrate	Denrate	denrate
ModCpl	gencoor	gencoor	Gencoor	gencoor
ModHadv	hbot_MADRID	hbot_MADRID	hbot_MADRID	hbot_MADRID
ModVadv	vbot	vbot	Vbot	vbot
ModHdiff	multiscale$^{\xi}$	multiscale$^{\xi}$	multiscale$^{\xi}$	multiscale$^{\xi}$
ModVdiff	eddy_MADRID	eddy_MADRID	eddy_MADRID	eddy_MADRID
ModPhot	phot	phot_aqCMU	phot_aqCMU	phot
ModPing	ping_noop	ping_noop	ping_noop	ping_noop
ModChem	mebi_radm2_cis4_MADRID1$^{\phi}$	mebi_radm2_cis4_MADRID1$^{\phi}$	mebi_radm2_cis4_MADRID1$^{\phi}$	mebi_radm2_cis4_MADRID1$^{\phi}$
ModAero	aero_MADRID1_mebi$^{\phi}$	aero_noop	aero_MADRID1_mebi$^{\phi}$	aero_MADRID1_mebi$^{\phi}$
ModAdepv	aero_depv_MADRID	aero_depv_noop	aero_depv_MADRID	aero_depv_MADRID
ModCloud	cloud_noop	cloud_CMU_aero_MADRID1_mebi$^{\phi}$	cloud_CMU_aero_MADRID1_mebi$^{\phi}$	cloud_RADM_aero_MADRID1_mebi$^{\phi}$
ModPa	pa_MADRID	pa_MADRID	pa_MADRID	pa_MADRID
ModUtil	util	util	Util	util
Modaeaqsolver	ae_aq_solver_MADRID	ae_aq_solver_MADRID	ae_aq_solver_MADRID	ae_aq_solver_MADRID

* The combination of gas-phase mechanism, cloud module, and aerosol module. RADM2_CI4, MADRID 1, CMU and RADM denote the RADM2_CI4 gas-phase chemical mechanism, the MADRID 1 aerosol module, the CMU aqueous-phase chemical mechanism and the RADM aqueous-phase chemical mechanism, respectively. The RADM2_CI4 gas-phase mechanism includes heterogeneous reactions on surface of particles and droplets; these reactions are not simulated when aqueous-phase chemistry and/or aerosol processes are not selected. They can also be neglected by the user. Two flags (IAERORATE and ICLDRATE) are included in PARAMETER.EXT to provide options to turn on/off these heterogeneous reactions.

$^{\xi}$ "Multiscale" can be replaced by "unif".

$^{\phi}$ When QSSA is used as a gas-phase chemistry solver, the modules "mebi_radm2_cis4_MADRID1", "aero_MADRID1_mebi", and "cloud_CMU_aero_MADRID1_mebi" (or "cloud_RADM_aero_MADRID1_mebi") should be replaced by "qssa_MADRID1", "aero_MADRID1_qssa" and "cloud_CMU_aero_MADRID1_qssa" (or "cloud_RADM_aero_MADRID1_qssa"), respectively.

Table 2-3d. Configurations of modules and mechanism available in CMAQ-MADRID: CACM.

***Configuration** *	CACM (gas-phase only)	CACM + MADRID 2 (no cloud)	CACM + CMU (no aerosol)	CACM + CMU + MADRID 2
Mechanism/ Include directory	CACM	CACM_aeMADRID2_2sec	CACM_aqCMU	CACM_aqCMU_aeMADRID2_2sec
Module				
ModDriver	ctm_MADRID2	ctm_MADRID2	ctm_MADRID2	ctm_MADRID2
ModPar	par_noop	par_noop	par_noop	par_noop
ModInit	init_MADRID	init_MADRID	init_MADRID	init_MADRID
ModAdjc	denrate	denrate	denrate	Denrate
ModCpl	gencoor	gencoor	gencoor	Gencoor
ModHadv	hbot_MADRID	hbot_MADRID	hbot_MADRID	hbot_MADRID
ModVadv	vbot	vbot	vbot	Vbot
ModHdiff	multiscale[ξ]	multiscale[ξ]	multiscale[ξ]	multiscale[ξ]
ModVdiff	eddy_MADRID	eddy_MADRID	eddy_MADRID	eddy_MADRID
ModPhot	phot	phot	phot_aqCMU	phot_aqCMU
ModPing	ping_noop	ping_noop	ping_noop	ping_noop
ModChem	qssa_MADRID2	qssa_MADRID2	qssa_MADRID2	qssa_MADRID2
ModAero	aero_noop	aero_MADRID2_qssa	aero_noop	aero_MADRID2_qssa
ModAdepv	aero_depv_noop	aero_depv_MADRID	aero_depv_noop	aero_depv_MADRID
ModCloud	cloud_noop	cloud_noop	cloud_CMU_aero_MADRID2_qssa	cloud_CMU_aero_MADRID2_qssa
ModPa	pa_MADRID	pa_MADRID	pa_MADRID	pa_MADRID
ModUtil	util	util	util	Util
Modaeaqsolver	--	ae_aq_solver_MADRID	ae_aq_solver_MADRID	ae_aq_solver_MADRID

* The combination of gas-phase mechanism, cloud module, and aerosol module. CACM, MADRID 2 and CMU denote the CACM gas-phase chemical mechanism, the MADRID 2 aerosol module and the CMU aqueous-phase chemical mechanism, respectively. The CACM gas-phase mechanism includes heterogeneous reactions on surface of particles and droplets; these reactions are not simulated when aqueous-phase chemistry and/or aerosol processes are not selected. They can also be neglected by the user. Two flags (IAERORATE and ICLDRATE) are included in PARAMETER.EXT to provide options to turn on/off these heterogeneous reactions.

[ξ] "Multiscale" can be replaced by "unif"

2.3 Input Files and Preprocessors

The Models-3/CMAQ inputs that need to be modified as a result of incorporating the new PM and cloud modules include emissions and initial/boundary conditions. PM inputs need to be commensurate with the sectional formulation. Since the August 2000 version, the modal formulation of Models-3/CMAQ has accepted two formats for PM emissions. If the emissions of $PM_{2.5}$ and PM_{10} are supplied, PM speciation is performed internally based on some default profile for inorganic PM, elemental carbon (EC) and organic compounds. There are different profiles available within CMAQ. Recent versions of the emission processor SMOKE also produce speciated PM output for 6 species, including sulfate (PSO4), nitrate (PNO3), EC (PEC), organic aerosols (POA), other fine PM (PMFINE), and coarse PM (PMCORS). The MADRID PM modules require speciated and size-specific PM emission inputs. As shown in Figure 2-1, a preprocessor, mode2sec_emis (available upon request) can be used to generate such inputs based on the $PM_{2.5}$ and PM_{10} emissions. Based on the speciation profiles selected in CMAQ, the mass of individual species is assigned to Aitken, fine, or coarse modes. The mass of the species in each section is then calculated by numerical integration. The user can also specify a PM speciation profile that is specific to the application by modifying the fraction of each species in the pre-processor code and recompiling after the change is made. Speciated PM emissions from SMOKE can be assigned to sections using the same pre-processor. PSO4 (which does not include sea salt sulfate), PNO3, PEC, POA, and PMFINE are assigned to the fine sections and PMCORS is assigned to the "other" species in the coarse sections. The number of particles emitted into each particle size section contributes to changes in the particle number concentrations and are, therefore, included in the aerosol emission species table (i.e., AE_EMIS.EXT), although the number of particles is not generated by the emission preprocessor and not included in the emission input file. The number of particles emitted is internally calculated based on the total emitted PM mass in a size section in the subroutine called "AERO_EMIS_MADRID.F", assuming an initial particle size distribution for the emitted particles.

Initial and boundary conditions are also needed to drive the new aerosol modules with sectional PM representation. The original modal PM module of Models-3/CMAQ requires speciated IC/BC inputs for the Aitken, accumulation, and coarse modes. Two processors are available (upon request) to prepare sectional inputs for the initial conditions (three-dimensional) and boundary conditions (three-dimensional and time-varying), respectively, from the modal inputs. In these preprocessors, mode2sec_ic and mode2sec_bc, the mass of PM species in each mode is redistributed among the corresponding sections of the particle size distribution (see Figure 2-1). Note that the initial conditions and boundary conditions for the number of particles for each size section are not generated by the IC/BC preprocessors. They are internally calculated in subroutines "initscen.F" and "rdbcon.F" , respectively.

2.4 Output Files

Models-3/CMAQ allows the user to specify the output species in extension files for gas-phase and particulate-phase concentrations (e.g., GC_CONC.EXT, AE_CONC.EXT, NR_CONC.EXT) as well as the dry and wet deposition fluxes of particulate and soluble gaseous compounds (e.g., GC_DDEP.EXT, AE_DDEP.EXT, GC_WDEP.EXT, AE_WDEP.EXT). The output files are in NetCDF format in three dimensions for concentrations and in two dimensions for wet and dry deposition fluxes.

3. SUBROUTINES

3.1 PM Modules

The subroutines contained in the sectional PM module are listed in Table 3-1 for MADRID 1. Key functions performed in the subroutines are listed, together with the names of the calling program(s) and the subroutine(s) used. These subroutines are housed in the PM (aero) repository of Models-3/CMAQ under the aero_MADRID1_qssa (or aero_MADRID1_mebi) subdirectory. Note that subroutines used in ISORROPIA are collected in aeroiso.F, isocom.F, isofwd.F, and isorev.F. Subroutines used in the CMU hybrid approach are contained in block.F, diff.F, diffund.F, dvode.F, eqpart.F, equaer.F, errprt.F, hybrid.F, madm.F, negchk.F, and step.F. These subroutines are not listed in further detail. Include files that are specific to the PM module are also listed in Table 3-1. These files (except for HETERPAR.EXT, ARRAYMAP.EXT and PARAMETER.EXT) are located in the same repository directory as the source code (akin to the original modal version of Models-3/CMAQ) and are copied into the working directory with the PM source codes before Models-3/CMAQ is compiled. Mechanism- and/or size-dependent files (e.g., ARRAYMAP.EXT, PARAMETER.EXT, HETERPAR.EXT) are located in the include directories listed in Table 2-3.

Except for the SOA module, the same PM subroutines are used for both versions of MADRID. However, some subroutines may contain module-specific lines of codes due to the differences in the representation of SOA. MADRID 2 uses the AER/EPRI/Caltech (AEC) module for SOA formation. Subroutines used in the AEC SOA library are presented in Table 3-2. These programs replace the CALTECH_SOA, CALTECH_SOA_REV, and SOASUB modules of Table 3-1. Note that decorated names that end with an underscore are used in C subroutines that are called by Fortran subroutines (e.g., aec_soa_ in C; aec_soa in the Fortran calling program) or that call Fortran subroutines (e.g., unifac_ in the calling C program; unifac in the Fortran subroutine). Note also that the *variable notation indicates that a pointer or an array is being passed as an argument in a C subroutine. The variable[] notation is also used for an array argument in C. Arrays are not explicitly noted in subroutine arguments in Fortran.

Table 3-1. Subroutines contained in MADRID 1.

Subroutine (Arguments; Return values)	Function(s)	Called By	Routine(s) Called
AERO SUBST_GRID_ID (CGRID, DPCTR, JDATE, JTIME, TSTEP)	Interfaces between the PM module and modules for other processes in Models-3/CMAQ	SCIPROC	NEXTIME, BOXAER
AEROCMU (GAS, CAERO, TEMPK, RH, PRES, Q, NG)	Interfaces between the PM module and the module for the CMU hybrid approach for gas/particle mass transfer calculation	AERODRIV	STEP, EQPART, HEQDYN
AERODRIV (TEMPE, AHE, TEMPE1, AHE1, PRES, USTAR)	Main aerosol driver	BOXAER	AEROCMU, AEROEQ, MASSFIX, NUCLRATE, WVPRE
RDEMIS_AE_MADRID (MDATE, MTIME, EMISLYRS, NSPC_EMIS, RJACM, EMIS)	Reads aerosol emissions and calculate aerosol number emissions	VDIFF	SUBHFILE M3EXIT M3WARN
AEROEQ (GAS, CAERO, TEMPK, RH, USTAR)	Interfaces between the PM module and the modules for the CIT hybrid and full equilibrium approaches for gas/particle mass transfer calculation	AERODRIV	AEROISO, CALTECH_SOA, AEROPARAM
aeroiso.F; isocom.F; isorev.F; isofwd.F	ISORROPIA routines	AEROEQ, EQPART, EQUAER	(various, not listed)
AEROPARAM (TEMPE, IAERO, USTAR, KN, SC, ST, VSETT)	Calculates the Knudsen, Schmidt, and Stokes numbers as a function of the input temperature and particle size.	AEROEQ,	QGAUS, FUNDIFF
AERORATE (CNBLK_AE, BLKLEN, BTEMP, BPRESS, RTDAT_AE, BLKDP)	Calculates the first-order heterogeneous loss rates of gases on particles with a specified size distribution	CALCKS	--

Table 3-1. Subroutines contained in MADRID 1 (continued).

Subroutine (Arguments; Return Values)	Function(s)	Called by	Routine(s) Called
BISECTNUCL (A, B, E, N, XM)	Performs iterations to solve g(x) = 0 based on the Bisection method	NUCLRATE	G
block.F; diff.F; diffund.F; dvode.F, eqpart.F; equaer.F; errprt.F; hybrid.F; madm.F; negchk.F; step.F	Routines for the CMU hybrid approach	AEROCMU	(various, not listed)
BOXAER (DT, CONCS, TEMPEK, TEMPEK1, PRES, QV, QV1, DENS, USTAR)	Interfaces between Models-3/CMAQ and Aerosol routines	AERO_DRIVER	AERODRIV, INITDSECF
CALTECH_SOA (WORG, GASORG, PARTORG , CURTEMP)	Calculates total absorbing organic matter for use in Odum/Griffin SOA partition. Calls SOASUB. Checks SOA solution	AEROEQ, EQPART	SOASUB
CALTECH_SOA_REV (K, Q, CURPOC, CURTEMP)	Calculates partial pressures of organics at the surface of the particles.	STEP	--
FUNDIFF (DP)	Defines external function	QGAUS	--
G (X)	Defines external function G used in BISECTNUCL	BISECTNUCL	--
INITDSECF	Initializes the particle size structure for the CMU hybrid approach	BOXAER	--
MASSFIX (CAERO0, CAERO, AERGAS0, AERGAS)	Ensures mass conservation in the CMU hybrid approach	AERODRIV	--
MOVEBIN SUBST_GRID_ID (CGRID, DPCTR)	Interfaces between the sciproc.F and the growth modules for the PM	SCIPROC	MOVECENTER
MOVECENTER (CAERO)	The moving-center scheme for PM growth	MOVEBIN	--

Table 3-1. Subroutines contained in MADRID 1 (continued).

Subroutine (Arguments; Return Values)	Function(s)	Called by	Routine(s) Called
NUCLRATE (GMASS, GNUM, AERGAS, TEMPK, RH)	Calculates new particle formation rates based on McMurry and Friedlander's method	AERODRIV	BISECTNUCL
QGAUS (FUNC, A, B, SS)	Program for Gaussian quadrature integration solver	AEROPARAM	--
SIZEINIT (DPCTR)	Initializes the particle size distributions that are based on the CIT or GATOR size structures	INITSCEN	--
SOASUB (MSUM, WORG, PARTORG, GASORG , CURTEMP)	Calculates equilibrium partition of individual SOA species in Odum/Griffin SOA formulation.	CALTECH_SOA	--
Include Files	**Content**	**Used by**	
ARRAYMAP.EXT	Mapping information between main and working arrays	various subroutines	
DYNAMIC.EXT, EQUAER.EXT	Common variables used in the routines for the CMU hybrid approach	various subroutines	
HETERPAR.EXT	Assigns species indices for gas-phase species that partition into heterogeneous reactions in cloud droplet and on aerosol surface	AERORATE, CLDRATE	
ISRPIA.EXT	Common variables used in ISORROPIA	ISORROPIA subroutines	
MW.EXT	Molecular weight data	various subroutines	
PARAMETER.EXT	Defines variables, arrays, and constants used in the PM module	various subroutines	
PARTITION.EXT	SOA partition coefficients based on experimental data of Odum/Griffin	CALTECH_SOA	

Table 3-2. Subroutines and include files used in the AEC SOA library.

Subroutine (Arguments; Return Values)	Function(s)	Called by	Routine(s) Called
aec_soa_ (called as aec_soa) (*tempk, *rh, *worg, *gasorg, *partorg, *lwc, *protonconc, *Organion, *DeltaLWC, *tboaflag)	Main SOA program.	AEROEQ EQPART	amain, bmain
amain (*aero, *aeros; deltaLWC)	Determines phases of particles. Specify initial guesses for PM composition. Calls Newton solver to solve aqueous PM composition. Calculates dry particle composition. Calls ZSR to calculate organic water content	aec_soa	amainabs, saturation, newt_double, TypeA, ZSR
amainabs (aabaero[])	Main program for absorption partitioning Type A compounds with no existing water	amain	newt_double, TypeAabs
bmain (gas[], aero[],)	Calculates total condensables. Makes initial guesses for particulate-phase concentrations for use in newt.	aec_soa	newt, TypeB
bpartfuncapprox (n, aero)	Calculates approximated equilibrium for hydrophobic SOA by assuming fixed absorbing medium amount and composition	TypeB	--
bpartitionfunc (n, aerof[], ff[ ])	Calculates the deviation from equilibrium for the absorptive SOA given set of concentrations and partition constants	fmin1 (a subroutine used by newt1)	--
bunidriver (XPASS[], GAMMA[], n)	Sets up unifac parameters for hydrophobic SOA and calls unifac subroutine	TypeB	unifac_
kpart (n, ac[], mwom, KB[])	Calculates partition coefficients for hydrophobic compounds	Type B	--
lwcorgfunc (n, dLWC[], f[])	Calculates the deviation of water activity from RH	fmin1_double (a subroutine used by newt1_double)	unidriver

Table 3-2. Subroutines and include files used in the AEC SOA library (continued).

Subroutine (Arguments; Return Values)	Function(s)	Called by	Routine(s) Called
newt newt1 newt_double newt1_double	Solves simultaneous equations using the Newton/Line Search method from Numerical Recipes (Press et al., 1997)	bmain Typeb amain ZSR	(numerous, not listed)
partfunc2 (n, aero[],gamma[], f[])	Calculates deviation from equilibrium for hydrophilic solutes	TypeA	--
partiaabs (n, aerod[], ff[])	Calculates deviation from equilibrium for hydrophilic solutes in absorption calculation	fmin1_double (which is called by TypeAabs)	--
partiabsapprox (n, aerod[]);	Calculates approximate partition of Type A compounds when LWC = 0 by assuming fixed absorbing mediumamount and composition	Typeaabs	
saturation (tot, vp, *pmconc, *gasconc)	Calculates saturation partition	amain	--
TypeA (n, aero[],f[])	Calculates Henry's law activity coefficient based on Raoult's law activity coefficients from unifac routines. Calls partfunc2 to calculate deviation from equilibrium. Calculates negative ion (anion) concentration	fmin_double (a subroutine used by newt_double) or amain	unidriver, partfunc2
TypeAabs	Given initial SOA concentrations, calls subroutines to calculate activity coefficients, partition coefficients, and equilibrium SOA concentrations. Calculates difference between input and equilibrium SOA concentrations.	fmin_double (a subroutine used by newt_double) or by amainabs	newt1_double, unidriver, partiaabs, Kpart
TypeB (n, aero[], f[])	Given initial SOA concentrations, calls subroutines to calculate activity coefficients, partition coefficients, and equilibrium SOA concentrations. Calculates difference between input and equilibrium SOA concentrations.	fmin (a subroutine used by newt) or bmain	bpartfuncapprox, bpartitionfunc. bunidriver, Kpart, newt1

Table 3-2. Subroutines and include files used in the AEC SOA library (continued).

Subroutine (Arguments; Return Values)	Function(s)	Called by	Routine(s) Called
unidriver (XPASS[], GAMMA[], n)	Sets up unifac parameters for hydrophilic SOA and calls unifac subroutine	TypeA	unifac_
unifac (called using unifac_) (NMOL, NFUNC, NU, X, A, RG, QG, Z, TEMP, GAMA)	Calculates activity coefficients	bunidriver unidriver (called as unifac)	--
ZSR (*aerog; dLWC[1])	Calculates the amount of water associated with hydrophilic SOA, either by an iterative procedure based on unifac or by ZSR	amain	newt1_double, lwcorgfunc
Include Files	**Content**	**Used by**	
binsolu.h	binary solution data for aqueous particles	ZSR	
bunifacparam.h	unifac parameters for hydrophobic SOA	bunidriver	
glo.h	global variables used in AEC_SOA routines	all subroutines in AEC_SOA routines	
glodef.h	global parameters used in AEC_SOA routines	all subroutines in AEC_SOA routines	
nr.h; nr_double.h; nrutil.h	function definitions used in numerical recipes	newt, newt1, newt_double, newt1_double and other numerical recipe subroutines	
unifacparam.h	unifac parameters for hydrophilic SOA	unidriver	

The globally convergent Newton/line search method is used to solve simultaneous equations in the AEC SOA library. The subroutines newt, newt1, newt_double, and newt1_double are copies (with minor changes) of the Newton method routine used in various points in the SOA code. Routines used by newt, newt1, newt_double, and newt1_double can be found in Press et al. (1997) and are not listed in detail here.

3.2 Cloud Modules

CMAQ-MADRID provides options to use two cloud modules: either cloud_RADM (original module modified for interfacing with sectional aerosol modules) or cloud_CMU. These modules simulate the physical and chemical processes occurring in clouds including aerosol scavenging by cloud droplets, aqueous-phase chemistry, and removal by wet deposition. The two cloud modules have the same treatments of cloud dynamics and aerosol scavenging by cloud droplets. They differ primarily in the aqueous-phase chemical mechanism used. A modified RADM aqueous-phase chemical mechanism (Chang et al., 1987; Walcek and Taylor, 1986), which includes updated reaction rate constants, is used in the cloud_RADM module. The cloud_CMU module uses the CMU aqueous-phase chemical mechanism (Strader et al., 1998).

The subroutines contained in the cloud_RADM modules are listed in Table 3-3, along with brief descriptions of their functions. These subroutines are used in the simulations with gas-phase and heterogeneous chemistry and with CAMQ-MADRID 1 with various size sections (i.e., all cloud_RADM_* directories). The names of the subroutine(s) by which each subroutine is called and those that it calls (if any) are also given in the table. The listing of the subroutines is provided in alphabetical order. The cloud_RADM module requires only one include file: AQ_PARAMS.EXT, which is required for a specific gas-phase mechanism (e.g., CBM-IV, RADM2, or RADM2-CI4) and a specific size resolution (e.g., 2 or multiple size sections). Those mechanism- and size-dependent AQ_PARAMS.EXT files are located in the include directories listed in Table 2-3.

Table 3-3. The subroutines contained in the cloud_RADM modules used with CMAQ-MADRID 1.

Subroutine (Arguments)	Function	Called by	Routine(s) called
AQCHEM (JDATE, JTIME, TEMP, PRES_PA, TAUCLD, PRCRATE, WCAVG, WTAVG, AIRM, ALFA0, ALFA3, GAS, AEROSOL, GASWDEP, AERWDEP, HPWDEP)	Computes concentration changes in cloud due to aqueous-phase chemistry, scavenging, and wet deposition.	AQMAP	
AQMAP (JDATE, JTIME, WTBAR, WCBAR, TBARC, PBARC, CTHK1, AIRM, PRATE1, TAUCLD, POLC, CEND, REMOV, REMOVAC, ALFA 0, ALFA3)	Provides an interface processor between the cloud dynamics module(s) and the aqueous-phase chemistry module.	RESCLD RADMCLD	CGRID_MAP AQCHEM
CLDPROC SUBST_GRID_ID (CGRID, TOT_WCBAR, JDATE, JTIME, TSTEP)	Provides an interface processor between the cloud module and the modules for other processes.	SCIPROC	CGRID_MAP RESCLD RADMCLD NEXTIME
CLDRATE (BLKLWC, BLKLEN, BTEMP, BPRESS, RTDAT_CLD)	Calculates the first-order heterogeneous loss rates of gases on cloud droplets. This is the version used with the QSSA gas-phase chemistry solver	CALCKS	--
CLDRATE (TOT_LWC, BTEMP, BPRESS, RTDAT_CLD)	Calculates the first-order heterogeneous loss rates of gases on cloud droplets. This is the version used with the MEBI gas-phase chemistry solver	CALCKS	--
GETALPHA (MASSI, LWC, T, P, RHOAIR, ALFA0, ALFA2, ALFA3)	Calculates the in-cloud scavenging coefficients for aerosol number and mass.	SCAVWDEP	
HLCONST (NAME, TEMP)	Sets the Henry's law constant of species at a given temperature.	SCAVWDEP	
INDEXN (NAME1, N, NAME2, INDICES)	Searches for all occurrences of NAME1 in list NAME2.	SCAVWDEP AQMAP	
RESCLD SUBST_GRID_ID (CGRID, JDATE, JTIME, TSTEP, N_SPC_WDEP, WDEP_MAP, DEP, RES_WCBAR)	Calculates cloud characteristics for grid-resolved clouds and uses them to generate cumulative and net timestep deposition, cloud top, cloud bottom, and pressure at the lifting level.	CLDPROC	CGRID_MAP NEXTIME SCAVWDEP AQ_MAP

Table 3-3. The subroutines contained in the cloud_RADM modules used with CMAQ-MADRID 1 (continued).

Subroutine (Arguments)	Function	Called by	Routine(s) called
RADMCLD SUBST_GRID_ID (CGRID, JDATE, JTIME, TSTEP, ICLDTYPE, N_SPC_WDEP, WDEP_MAP, DEP, CONV_WCBAR)	Calculates cloud characteristics for subgrid convective precipitating clouds and subgrid non-precipitating clouds, and uses them to generate cumulative and net timestep deposition, cloud top, cloud bottom, and pressure at the lifting level.	CLDPROC	CGRID_MAP NEXTIME SCAVWDEP AQ_MAP
SCAVWDEP (JDATE, JTIME, WTBAR, WCBAR, TBARC, PBARC, CTHK1, AIRM, PRATE1, TAUCLD, POLC, CEND, REMOV, REMOVAC, ALFA 0, ALFA2, ALFA3)	Computes in-cloud scavenging and wet removal.	RESCLD RADMCLD	CGRID_MAP GETALPHA
Include Files	**Content**	**Used by**	
AQ_PARAMS.EXT	Pointers for arrays GAS and AEROSOL used in the CBM-IV and RADM gas-phase chemistry module and parameters used in aerosol activation parameterization	AQMAP, AQCHEM	

The cloud_CMU subroutines used with CMAQ-MADRID are listed in Table 3-4. Due to different chemical species in the two MADRID modules, several of the subroutines and include files listed in Table 3-4 differ slightly between the cloud_CMU_aero_MADRID1_qssa (or cloud_CMU_aero_MADRID1_mebi) and cloud_CMU_aero_MADRID2_qssa directories. There is no difference, however, in the subroutine arguments and functions. Two NETLIB subroutines are used in the cloud_CMU modules: the Variable-coefficient Ordinary Differential Equation solver package (DVODE) in double precision and the Powell hybrid method package (HYBRD). DVODE is used to solve the ordinary differential equations (ODEs) for the aqueous-phase reaction system, and HYBRD is used to obtain the steady-state concentrations of the aqueous-phase steady-state species. Note that the subroutines called by DVODE and HYBRD are given in Table 3-4, but their functions are not provided (except for the subroutines FEX and STATE, which provide the functions for DVODE and HYBRD, respectively).

All the cloud include files except for AQ_PARAMS.EXT are located in the same source code repository directory as the source code and are copied into the working directory with cloud module source codes before CMAQ is compiled. A specific "AQ_PARAMS.EXT" is required for a specific gas-phase mechanism (e.g., CBM-IV or RADM2, RADM2-CI4, or CACM) and a specific size resolution (e.g., 2 or multiple size sections). Those mechanism- and size-dependent AQ_PARAMS.EXT files are located in the include directories listed in Table 2-3.

Table 3-4. The subroutines contained in the cloud_CMU modules used with CMAQ-MADRID.

Subroutine (Arguments)	Function	Called by	Routine(s) Called
ADDIT (RR, ARYTM, RP, RL)	Sums up the rates of the aqueous-phase kinetic reactions to give the rates for the main aqueous-phase species.	AQRATES	
AQCHEM (JDATE, JTIME, TEMP, PRES_PA, TAUCLD, PRCRATE, WCAVG, WTAVG, AIRM, ALFA0, ALFA3, GAS, AEROSOL, GASWDEP, AERWDEP, HPWDEP, PHRATENO2)	Maps species concentrations for the call to the Strader et al. aqueous-phase chemistry module; Computes concentration changes in cloud due to aqueous-phase chemistry, scavenging, and wet deposition.	AQMAP	AQ_STRADER
AQINTEGR (Y, STIME, STOUT)	Prepares the necessary variables for the call to the stiff ODE integrator (DVODE).	AQ_STRADER	DVODE AQRATES
AQMAP (JDATE, JTIME, WTBAR, WCBAR, TBARC, PBARC, CTHK1, AIRM, PRATE1, TAUCLD, POLC, CEND, REMOV, REMOVAC, ALFA0, ALFA2, ALFA3, PHRATENO2)	Provides an interface processor between the cloud dynamics module(s) and the aqueous-phase chemistry module.	RESCLD RADMCLD	CGRID_MAP AQCHEM
AQRATES (YAQ, AQPROD, AQDEST, YAQPRIME)	Calculates the rates of change of the species concentrations at a time step for the aqueous-phase species.	FEX	FULLEQUIL VALUES STEADY HYBRD REACT ADDIT MASS DIFFER
AQ_STRADER (TBEG,DELTAT,GAS,CCO2,AEROSOL,HSO5,HMSA,LWC, PRATE,TEMP,P,IRAD,PHOTOFAC,NSECT, DAER, FDIST)	Initializes the aqueous-phase chemistry calculation with the Strader et al. aqueous-phase chemical mechanism.	AQCHEM	DROPINIT CONSTANTS AQINTEGR
BADDIT (RR, R)	Calculates the reaction rates for the aqueous-phase steady-state species.	STATE	
BMASS (WL, RADIUS, TEMP, FGL, FLG, GFGL, GFLG)	Calculates the mass fluxes for the aqueous-phase steady-state species.	STATE	

Table 3-4. The subroutines contained in the cloud_CMU modules used with CMAQ-MADRID (continued).

Subroutine (Arguments)	Function	Called by	Routine(s) Called
CLDPROC SUBST_GRID_ID (CGRID, TOT_WCBAR, JDATE, JTIME, TSTEP)	Provides an interface processor between the cloud module and the modules for other processes.	SCIPROC	CGRID_MAP RESCLD RADMCLD NEXTIME
CLDRATE (BLKLWC, BLKLEN, BTEMP, BPRESS, RTDAT_CLD)	Calculates the first-order heterogeneous loss rates of gases on cloud droplets. This is the version used with the QSSA gas-phase chemistry solver.	CALCKS	--
CLDRATE (TOT_LWC, BTEMP, BPRESS, RTDAT_CLD)	Calculates the first-order heterogeneous loss rates of gases on cloud droplets. This is the version used with the MEBI gas-phase chemistry solver.	CALCKS	--
CONSTANTS (TEMP)	Calculates the equilibrium constants, the Henry's law constants, and the reaction rate constants.	AQ_STRADER	
DIFFER (RP, RL, FGL, FLG, GFGL, GFLG, DP, DL)	Calculates the differentials for the gas- and aqueous-phase species.	AQRATES	
DROPINIT	Sets molecular weights for aqueous-phase species.	AQ_STRADER	
DVODE* (F, NEQ, Y, T, TOUT, ITOL, RTOL, ATOL, ITASK, ISTATE, IOPT, RWORK, LRW, IWORK, LIW, JAC, MF, RPAR, IPAR)	Solves the system of the ODEs for the aqueous-phase reaction system.	AQINTEGR	DCOPY F DEWSET DVHIN DSCAL DVINDY XERRWD DVSTEP DVJUST DVSET VNLS DAXPY DVJAC DVSOL JAC DGEFA DACOPY DGBFA DGESL DGBSL

Table 3-4. The subroutines contained in the cloud_CMU modules used with CMAQ-MADRID (continued).

Subroutine (Arguments)	Function	Called by	Routine(s) called
ELECTRO (X, F, CON, SPRES, DCMET, AKEQ, AKHEN, WV, TEMP)	Calculates the electron balance for the electroneutrality equation.	FULLEQUIL	
FEX (NEQA, T, Y, F)	Provides the function for the aqueous-phase chemistry integrator.	DVODE	AQRATES
FULLEQUIL (ASPRES, AWV, ATEMP, AXSOL)	Solves the electroneutrality equation using the bisection method.	AQRATES	ELECTRO
GETALPHA (MASSI, LWC, T, P, RHOAIR, ALFA0, ALFA2, ALFA3)	Calculates the in-cloud scavenging coefficients for aerosol number and mass.	SCAVWDEP	
HLCONST (NAME, TEMP, PBARC)	Sets the Henry's law constants of species at a given temperature.	SCAVWDEP	
HYBRD (STATE, N, X, FVEC, XTOL, MAXFEV, ML, MU, EPSFCN, DIAG, MODE, FACTOR, NPRINT, INFO, NFEV, FJAC, LDFJAC, R, LR, QTF, WA1, WA2, WA3, WA4)	Computes the steady-state concentrations for the aqueous-phase steady-state species using the Powell hybrid method.	AQRATES	STATE FDJAC1 QRFAC QFORM DOGLEG R1UPDT R1MPYQ
INDEXN (NAME1, N, NAME2, INDICES)	Searches for all occurrences of NAME1 in list NAME2.	SCAVWDEP AQMAP	
MASS (WL, RADIUS, TEMP, FGL, FLG, GFGL, GFLG)	Calculates the mass fluxes for the mass balances.	AQRATES	
RESCLD (CGRID, JDATE, JTIME, TSTEP, N_SPC_WDEP, WDEP_MAP, DEP, RES_WCBAR)	Calculates cloud characteristics for grid-resolved clouds and uses them to generate cumulative and net timestep deposition, cloud top, cloud bottom, and pressure at the lifting level.	CLDPROC	CGRID_MAP NEXTIME SCAVWDEP AQ_MAP
RADMCLD (CGRID, JDATE, JTIME, TSTEP, ICLDTYPE, N_SPC_WDEP, WDEP_MAP, DEP, CONV_WCBAR)	Calculates cloud characteristics for subgrid convective precipitating clouds and subgrid non-precipitating clouds, and uses them to generate cumulative and net timestep deposition, cloud top, cloud bottom, and pressure at the lifting level.	CLDPROC	CGRID_MAP NEXTIME SCAVWDEP AQ_MAP
REACT (RR, ARYTM)	Calculates the rates of the aqueous-phase reactions.	AQRATES STATE	

Table 3-4. The subroutines contained in the cloud_CMU modules used with CMAQ-MADRID (continued).

Subroutine (Arguments)	Function	Called by	Routine(s) Called
SCAVWDEP (JDATE, JTIME, WTBAR, WCBAR, TBARC, PBARC, CTHK1, AIRM, PRATE1, TAUCLD, POLC, CEND, REMOV, REMOVAC, ALFA0, ALFA2, ALFA3)	Computes in-cloud scavenging and wet removal.	RESCLD RADMCLD	CGRID_MAP GETALPHA
STATE (N, X, F, PAR)	Provides the vector of the functions for the aqueous-phase steady-state species.	HYBRD	VALUES REACT BADDIT BMASS
STEADY (RADIUS, TEMP)	Calculates the steady-state species concentrations.	AQRATES	
VALUES (X)	Calculates the concentrations of the aqueous-phase species.	AQRATES STATE	
Include Files	**Content**	**Used by**	
AERPAR.EXT	Aerosol parameters used in the CMU aqueous-phase chemistry module	DROPPAR.EXT	
AER_SECTIONS.EXT	Aerosol size information used in the calculation of aerosol activation by cloud droplets	AQMAD, AQCHEM, AQ_STRADER	
AQCONSTS.EXT	Defines constants used for the CMU aqueous-phase chemistry module	AQCHEM, AQ_STRADER, AQRATES, BMASS, ELECTRO, MASS, STEADY	
AQPAR.EXT	Species indices and molecular weight used for aqueous-phase chemistry module	ADDIT, BADDIT, DIFFER, ELECTRO, HLCONST, DROPCOM.EXT	
AQUEOUS.EXT	Defines fraction of crustal material, flag for chlorine chemistry, and error tolerance for numerical integration	AQCHEM, AQINTEGR, AQ_STRADER, AQRATES, REACT	
AQ_PARAMS.EXT	Pointers for arrays GAS and AEROSOL used in the CMU aqueous-phase chemistry module and parameters used in aerosol activation parameterization	AQCHEM, AQMAP, AQ_STRADER, AQRATES	

Table 3-4. The subroutines contained in the cloud_CMU modules used with CMAQ-MADRID (continued).

Include Files	Content	Used by
DROPCOM.EXT	Variables and arguments shared by the CMU aqueous-phase chemistry subroutines	AQCHEM, AQ_STRADER, BMASS, CONSTANTS, DROPINIT, FULLEQUIL, MASS, REACT, STATE, STEADY, VALUES
DROPPAR.EXET	Aqueous-phase parameters and variables	AQCHEM, AQINTEGR, AQ_STRADER, AQRATES, FEX, REACT
HYBRD.EXT	Input parameters for subroutine HYBRD	AQRATES
PHOTNO2.EXT	NO_2 photolysis rate used in the CMU aqueous-phase chemistry module	PHOT, RADMCLD, RESCLD

* subroutines of DVODE are located in the ae_aq_solver_MADRID subdirectory under the common_MADRID class of modules

4. TEST CASES

Two example simulations are provided to demonstrate the application of the new aerosol and cloud modules described in this document. In the tape accompanying this user's guide, emissions, meteorology, and result files are provided for these two simulations. Both simulations are for the August 1987 SCAQS episode and more details can be found in the final report titled "CMAQ-MADRID Technical Documentation" (Zhang et al., 2002).

The first test case demonstrates the new CMU cloud module in an application with MADRID 1 using 8 particle size sections. Note that new particle formation and the growth/formation of particles due to cloud droplets are simulated in this example. The gas-phase mechanism, CBM-IV is solved using the MEBI chemistry solver, which has been modified for MADRID 1. The script for building Models-3 with the specifications used in this first test case is shown in Appendix A. The results of this example are somewhat different from those reported in Zhang et al. (2002), since Zhang et al. used the observed mixing heights instead of the simulated one from MM5 used here.

The second test case is an aerosol-only simulation with MADRID 2, using two size sections to represent aerosols. As shown in the script in Appendix B, cloud modules and heterogeneous reactions in cloud droplets are turned off by selecting cloud_noop. New particle formation is simulated.

5. REFERENCES

Byun, D.W., and J.K.S. Ching, 1999. Science algorithms of the EPA Models-3 Community Multiscale Air Quality (CMAQ) Modeling System. EPA/600/R-99/030. U.S. Environmental Protection Agency, Research Triangle Park, NC 27711.

Chang, J.S., R.A. Brost, I.S.A. Isaksen, S. Madronich, P. Middleton, W.R. Stockwell, and C.J. Walcek, 1987. A three-dimensional Eulerian acid deposition model: Physical concepts and formulation, *J. Geophys. Res*., **92**, 14681-14700.

Griffin, R.J., D.R. Cocker III, R.C. Flagan and J.H. Seinfeld, 1999. Organic aerosol formation from the oxidation of biogenic hydrocarbons, *J. Geophys. Res.*, **104**, 3555-3567.

Griffin, R.J., D. Dabdub, J.H. Seinfeld, 2002. Secondary organic aerosols: I. Atmospheric chemical mechanism for production of molecular constituents, *J. Geophys. Res.*, in press.

Jacob, D., 2000. Heterogeneous chemistry and tropospheric ozone, *Atmos. Environ.*, **34**, 2132-2159.

Jacobson, M.Z., 1997a. Development and application of a new air pollution modeling system – II. Aerosol module structure and design, *Atmos. Environ.*, **31**, 131-144.

Jacobson, M.Z. and R.P. Turco, 1995. Simulating condensational growth, evaporation, and coagulation of aerosols using a combined moving and stationary size grid, *Aerosol Sci. Technol.*, **22**, 29-92.

McMurry, P.H. and S.K. Friedlander, 1979. New particle formation in the presence of an aerosol, *Atmos. Environ.*, **13**, 1635-1651.

Nenes, A., C. Pilinis and S.N. Pandis, 1999. Continued development and testing of a new thermodynamic aerosol module for urban and regional air quality models, *Atmos. Environ.*, **33**, 1553-1560.

Odum, J.R., T.P.W. Jungkamp, R.J. Griffin, H.J.L. Forstner, R.C. Flagan and J.H. Seinfeld, 1997. Aromatics, reformulated gasoline, and atmospheric organic aerosol formation, *Environ. Sci. Technol.*, **31**, 1890-1897.

Press, W.H., S.A. Teulolsky, W.T. Vetterling, B.P. Flannery, 1997. Numerical Recipes in C, Second Edition, 3.5" Diskette for Windows3.1, 95, or NT. Cambridge University press, Cambridge, UK.

Pun, B.K., Y. Zhang, K. Vijayaraghavan, S.Y. Wu, C. Seigneur, J.H. Seinfeld, 2001. Development of new aerosol modules and incorporation into Models-3/CMAQ, AER Draft report to EPRI, Palo Alto, CA.

Pun, B.K., R.J. Griffin, C.Seigneur, J.H. Seinfeld, 2002. Secondary organic aerosol 2. Thermodynamic model for gas/particle partitioning of molecular constituents, *J. Geophys. Res.*, **107**, *D17*, 4333, doi:10.1029/2001J000542.

Strader, R., C. Gurciullo, S. Pandis, N. Kumar, F.W. Lurmann, 1998. Development of gas-phase chemistry secondary organic aerosol, and aqueous-phase chemistry modules for PM modeling, STI Final report to Coordinating Research Council, Atlanta, GA.

Venkatram, A. and J. Pleim, 1999. The electrical analogy does not apply to modeling dry deposition of particles, *Atmos. Environ.*, **33**, 3075-3076.

Walcek, C.J. and G.R. Taylor, 1986. A theoretical method for computing vertical distributions of acidity and sulfate production within cumulus clouds, *J. Atmos. Sci.*, **43**, 339-355.

Zhang, Y., B. Pun, K. Vijayaraghavan S.-Y. Wu, and C. Seigneur, 2002. CMAQ-MADRID Technical Documentation, AER Report to EPRI, Document Number CP083-02-03a, Palo Alto, CA.

Zhang, Y., B.K. Pun, K. Vijayaraghavan, S.-Y. Wu, C. Seigneur, S.N. Pandis, M.Z. Jacobson, A. Nenes, J.H. Seinfeld, 2004. Development and Application of the Model of Aerosol Dynamics, Reaction, Ionization and Dissolution (MADRID), *J. Geophys. Res.*, in press.

APPENDIX A.

Script used to compile Models-3/CMAQ with aero_MADRID1 and cloud_CMU_aero_MADRID1.

```
#! /bin/csh -f

#> RCS file, release, date & time of last delta, author, state, [and locker]
#> $Header$

#> what(1) key, module and SID; SCCS file; date and time of last delta:
#> %W% %P% %G% %U%

setenv BASE /usr/home/models3
setenv M3MODEL $BASE/july2002/models    # code archive
setenv M3TOOLS $BASE/july2002/tools      # libraries

 if ( ! -e $M3MODEL || ! -e $M3TOOLS ) then
    echo "   $M3MODEL or $M3TOOLS directory not found"
    exit 1
    endif
 echo "   Model archive path: $M3MODEL"
 echo "           Tools path: $M3TOOLS"

 set BLD_OS = `uname –s`
 if ($BLD_OS != 'Linux') then
    echo "   $BLD_OS -> wrong makit script for host! "
    exit 1
    endif

 set echo

#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#
#> user choices: cvs archives
 set Project = $M3MODEL/CCTM
 set GlobInc = $M3MODEL/include/release

#> user choices: base directory
#set Base = /your_dir/you/working_dir
 set Base = $cwd

 set APPL  = cb4_aqCMU_aeMADRID1_8sec_mebi_multi
 set CFG   = cfg.$APPL
 set MODEL = CCTM_$APPL

#> user choices: m3bld command
#set Opt = compile_all  # force compile, even if object files are current
#set Opt = clean_up     # remove all source files upon successful completion
#set Opt = no_compile   # do everything except compile
#set Opt = no_link      # do everything except link
#set Opt = one_step     # compile and link in one step
#set Opt = parse_only   # checks config file syntax
#set Opt = show_only    # show requested commands but doesn't execute them
 set Opt = verbose      # show requested commands as they are executed
#set MakeOpt            # builds a Makefile instead of the model

#> user choices:  single or multiple processors
#set ParOpt          # multiple PE's

#> user choices: various modules
```

```
 set Revision = release       # release = latest CVS revision
#set Revision = '"STA2_1"'    # monocode (>="REL1_4")
#set Revision = '"REL1_3"'    # last release before monocode
#> NOTE: m3bld will try to compile with existing code; it will not retrieve
#>       new (different release) code. So if your "BLD" directory contains
#>       code from a release different than the one you have specified above.
#>       m3bld will tell you, but will merrily compile the original code.
#>       The workaround is to remove your "BLD" directory and start fresh.

 set ModDriver = ( module ctm_MADRID1        $Revision; )
#set ModDriver = ( module ctm_MADRID2        $Revision; )

 if ( $?ParOpt ) then
    set ModPar = ( module par                $Revision; )
    else
    set ModPar = ( module par_noop           $Revision; )
    endif

 set ModInit   = ( module init_MADRID        $Revision; )

#set ModAdjc   = ( module adjcon_noop        $Revision; )
 set ModAdjc   = ( module denrate            $Revision; )

 set ModCpl    = ( module gencoor            $Revision; )

#set ModHadv   = ( module hadv_noop          $Revision; )
 set ModHadv   = ( module hbot_MADRID        $Revision; )
#set ModHadv   = ( module hppm               $Revision; )

#set ModVadv   = ( module vadv_noop          $Revision; )
 set ModVadv   = ( module vbot               $Revision; )
#set ModVadv   = ( module vppm               $Revision; )

#set ModHdiff  = ( module hdiff_noop         $Revision; )
#set ModHdiff  = ( module unif               $Revision; )
 set ModHdiff  = ( module multiscale         $Revision; )

#set ModVdiff  = ( module vdiff_noop         $Revision; )
 set ModVdiff  = ( module eddy_MADRID        $Revision; )

#set ModPhot   = ( module phot_noop          $Revision; )
#set ModPhot   = ( module phot               $Revision; )
 set ModPhot   = ( module phot_aqCMU         $Revision; )

 set ModPing   = ( module ping_noop          $Revision; )

#set ModChem   = ( module chem_noop          $Revision; )
#set ModChem   = ( module qssa_MADRID1       $Revision; )
 set ModChem   = ( module mebi_cb4_MADRID1   $Revision; )
#set ModChem   = ( module mebi_radm2_cis4_MADRID1  $Revision; )
#set ModChem   = ( module qssa_MADRID2       $Revision; )

#set ModAero   = ( module aero_noop          $Revision; )
#set ModAero   = ( module aero_MADRID1_qssa       $Revision; )
 set ModAero   = ( module aero_MADRID1_mebi       $Revision; )
#set ModAero   = ( module aero_MADRID2_qssa       $Revision; )
#set ModAero   = ( module aero_MADRID2_mebi       $Revision; )

#set ModAdepv  = ( module aero_depv_noop     $Revision; )
 set ModAdepv  = ( module aero_depv_MADRID   $Revision; )

#set ModCloud  = ( module cloud_noop         $Revision; )
#set ModCloud  = ( module cloud_CMU_aero_MADRID1_qssa  $Revision; )
```

```
 set ModCloud  = ( module cloud_CMU_aero_MADRID1_mebi  $Revision; )
#set ModCloud  = ( module cloud_CMU_aero_MADRID2_qssa  $Revision; )
#set ModCloud  = ( module cloud_CMU_aero_MADRID2_mebi  $Revision; )
#set ModCloud  = ( module cloud_RADM_aero_MADRID1_qssa  $Revision; )
#set ModCloud  = ( module cloud_RADM_aero_MADRID1_mebi  $Revision; )

 set ModPa     = ( module pa_MADRID          $Revision; )

 set ModUtil   = ( module util               $Revision; )

 set Modaeaqsolver = ( module ae_aq_solver_MADRID    $Revision; )

#set ModMisc   = ( misc; \
#                  $Base/file1.F \
#                  $Base/file2.F \
#                  $Base/file3.F )

#> user choices: emissions processing in chem or vdiff (default) ...
#set Cemis

#> user choices: vertical layers and mechanism

#set Reso      = 05
#set Grid      = 63X28
 set Layers    = 15
#set Domain    = d${Reso}_${Layers}
 set Mechanism = cb4_aqCMU_aeMADRID1_8sec
 set Tracer    = trac0               # default: no tracer species
#set Decomp    = 1X1

#> user choices: set process analysis linkages
 set PABase    = $GlobInc
 set PAOpt     = pa_noop
#set PABase    = /project/cmaq/yoj/saprc
#set PAOpt     =

#> other user choices set below are:
#>    name of the "BLD" directory
#>    compiler/link flags
#>    library paths
#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#

 set Bld = $Base/BLD_${APPL}
#unset echo
 if ( ! -e "$Bld" ) then
    mkdir $Bld
    else
    if ( ! -d "$Bld" ) then
       echo "   *** target exists, but not a directory ***"
       exit 1
       endif
    endif
#set echo
 cd $Bld

#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#

 set MODLOC = ${M3TOOLS}/stenex/${BLD_OS}
 set seL = sef90_noop
# Note: If sef90_noop is used instead of se_noop, the values of STX1, STX2
# and STX3 have to be changed in this script.
 if ( $?ParOpt ) then
    set Mpich = /usr/local/mpich-1.2.4
```

```
   set seL = sef90
   set LIB2 = "-L${M3TOOLS}/pario -lpario"
   set LIB3 = "-L${M3TOOLS}/dynmem -ldynmem"
   set LIB4 = "-L${Mpich}/lib -lmpich"
   set Str1 = (// Parallel / Include message passing definitions)
   set Str2 = (include SUBST_MPICH ${Mpich}/include/mpif.h;)
   else
   set seL = sef90_noop
   set LIB2 =
   set LIB3 =
   set LIB4 =
   set Str1 =
   set Str2 =
   endif

set LIB1 = "-L${MODLOC} -l${seL}"
set LIB5 = "-L${M3TOOLS}/m3io/${BLD_OS} -lm3io"
set LIB6 = "-L${BASE}/aug2000/tools/netCDF/${BLD_OS} -lnetcdf"
set LIBS = "$LIB1 $LIB2 $LIB3 $LIB4 $LIB5 $LIB6"

set FC = /usr/pgi/linux86/bin/pgf90
set FP = /usr/pgi/linux86/bin/pgf90

set F_FLAGS    = "-Mfixed -Mextend -O2 -module ${MODLOC} -I."
set CPP_FLAGS  = "-Mfixed -Mextend"
set C_FLAGS    = " "
set LINK_FLAGS = ""

set Blder = $M3TOOLS/build/$BLD_OS/m3bl

set ICL_PAR   = $GlobInc
set ICL_CONST = $GlobInc
set ICL_FILES = $GlobInc
set ICL_EMCTL = $GlobInc
set ICL_IOAPI = ${M3TOOLS}/m3io/Linux
set ICL_GRID  = $BASE/july2002/models/include/release/d05_15
set ICL_MECH  = $GlobInc/$Mechanism
set ICL_TRAC  = $GlobInc/$Tracer
set ICL_PA    = $PABase/$PAOpt

if ( $?Cemis ) then
   set CV = -Demis_chem
   else
   set CV =
   endif

if ( $?ParOpt ) then   # split to avoid line > 256 char
   set PAR = ( -Dparallel\
               -DINTERPB=PINTERPB\
               -DM3ERR=PM3ERR\
               -DM3EXIT=PM3EXIT\
               -DM3WARN=PM3WARN\
               -DSHUT3=PSHUT3\
               -DWRITE3=PWRITE3 )

   set Popt = SE
   else
   echo "   Not Parallel; set Serial (no-op) flags"
   set PAR =
   set Popt = NOOP
   endif

set STX1 = ( -DSUBST_MODULES=${Popt}_MODULES\
```

```
              -DSUBST_BARRIER=${Popt}_BARRIER )
 set STX2 = ( -DSUBST_GLOBAL_MAX=${Popt}_GLOBAL_MAX\
              -DSUBST_GLOBAL_MIN=${Popt}_GLOBAL_MIN\
              -DSUBST_GLOBAL_MIN_DATA=${Popt}_GLOBAL_MIN_DATA\
              -DSUBST_GLOBAL_TO_LOCAL_COORD=${Popt}_GLOBAL_TO_LOCAL_COORD\
              -DSUBST_GLOBAL_SUM=${Popt}_GLOBAL_SUM\
              -DSUBST_GLOBAL_LOGICAL=${Popt}_GLOBAL_LOGICAL\
              -DSUBST_LOOP_INDEX=${Popt}_LOOP_INDEX\
              -DSUBST_SUBGRID_INDEX=${Popt}_SUBGRID_INDEX )
 set STX3 = ( -DSUBST_HI_LO_BND_PE=${Popt}_HI_LO_BND_PE\
              -DSUBST_SUM_CHK=${Popt}_SUM_CHK\
              -DSUBST_INIT_ARRAY=${Popt}_INIT_ARRAY\
              -DSUBST_COMM=${Popt}_COMM\
              -DSUBST_MY_REGION=${Popt}_MY_REGION\
              -DSUBST_SLICE=${Popt}_SLICE\
              -DSUBST_GATHER=${Popt}_GATHER\
              -DSUBST_DATA_COPY=${Popt}_DATA_COPY\
              -DSUBST_IN_SYN=${Popt}_IN_SYN )

 setenv CVSROOT $Project

#> make the config file

 set Cfile = ${CFG}.bld
 set quote = '"'

 echo                                                              > $Cfile
 echo "model       $MODEL;"                                       >> $Cfile
 echo                                                             >> $Cfile
 echo "FPP         $FP;"                                          >> $Cfile
 echo                                                             >> $Cfile
#set text = "$quote$CPP_FLAGS $CV $PAR $STX1 $STX2 $STX3$quote;"

 set add = "-DINTERPB=INTERP3"
 set text = "$quote$CPP_FLAGS $CV $add $PAR $STX1 $STX2 $STX3$quote;"

 echo "cpp_flags   $text"                                         >> $Cfile
 echo                                                             >> $Cfile
 echo "f_compiler  $FC;"                                          >> $Cfile
 echo                                                             >> $Cfile
 echo "f_flags     $quote$F_FLAGS$quote;"                         >> $Cfile
 echo                                                             >> $Cfile
 echo "link_flags  $quote$LINK_FLAGS$quote;"                      >> $Cfile
 echo                                                             >> $Cfile
 echo "libraries   $quote$LIBS$quote;"                            >> $Cfile
 echo                                                             >> $Cfile
 echo "global      $Opt;"                                         >> $Cfile
 echo                                                             >> $Cfile

 set text="// layers, mechanism and tracer:"
 echo "$text ${Layers}, ${Mechanism}, ${Tracer}"                  >> $Cfile
 echo "// project archive: ${Project}"                            >> $Cfile
 echo                                                             >> $Cfile

 echo "include SUBST_PE_COMM    $ICL_PAR/PE_COMM.EXT;"            >> $Cfile
 echo "include SUBST_CONST      $ICL_CONST/CONST.EXT;"            >> $Cfile
 echo "include SUBST_FILES_ID   $ICL_FILES/FILES_CTM.EXT;"        >> $Cfile
 echo "include SUBST_EMPR_VD    $ICL_EMCTL/EMISPRM.vdif.EXT;"     >> $Cfile
 echo "include SUBST_EMPR_CH    $ICL_EMCTL/EMISPRM.chem.EXT;"     >> $Cfile
 echo "include SUBST_IOPARMS    $ICL_IOAPI/PARMS3.EXT;"           >> $Cfile
 echo "include SUBST_IOFDESC    $ICL_IOAPI/FDESC3.EXT;"           >> $Cfile
 echo "include SUBST_IODECL     $ICL_IOAPI/IODECL3.EXT;"          >> $Cfile
 echo "include SUBST_COORD_ID   $ICL_GRID/COORD_63X28.EXT;"       >> $Cfile
```

```
echo "include SUBST_VGRD_ID    $ICL_GRID/VGRD_15.EXT;"            >>$Cfile
echo "include SUBST_HGRD_ID    $ICL_GRID/HGRD_63X28_1X1.EXT;"     >>$Cfile

echo "include SUBST_RXCMMN     $ICL_MECH/RXCM.EXT;"               >> $Cfile
echo "include SUBST_RXDATA     $ICL_MECH/RXDT.EXT;"               >> $Cfile
echo "include SUBST_ARRAYMAP   $ICL_MECH/ARRAYMAP.EXT;"           >> $Cfile
echo "include SUBST_PARAMETER  $ICL_MECH/PARAMETER.EXT;"          >> $Cfile
echo "include SUBST_AQ_PARAMS  $ICL_MECH/AQ_PARAMS.EXT;"          >> $Cfile
echo "include SUBST_HETERPAR   $ICL_MECH/HETERPAR.EXT;"           >> $Cfile
echo "include SUBST_GC_SPC     $ICL_MECH/GC_SPC.EXT;"             >> $Cfile
echo "include SUBST_GC_EMIS    $ICL_MECH/GC_EMIS.EXT;"            >> $Cfile
echo "include SUBST_GC_ICBC    $ICL_MECH/GC_ICBC.EXT;"            >> $Cfile
echo "include SUBST_GC_DIFF    $ICL_MECH/GC_DIFF.EXT;"            >> $Cfile
echo "include SUBST_GC_DDEP    $ICL_MECH/GC_DDEP.EXT;"            >> $Cfile
echo "include SUBST_GC_DEPV    $ICL_MECH/GC_DEPV.EXT;"            >> $Cfile
echo "include SUBST_GC_ADV     $ICL_MECH/GC_ADV.EXT;"             >> $Cfile
echo "include SUBST_GC_CONC    $ICL_MECH/GC_CONC.EXT;"            >> $Cfile
echo "include SUBST_GC_G2AE    $ICL_MECH/GC_G2AE.EXT;"            >> $Cfile
echo "include SUBST_GC_G2AQ    $ICL_MECH/GC_G2AQ.EXT;"            >> $Cfile
echo "include SUBST_GC_SCAV    $ICL_MECH/GC_SCAV.EXT;"            >> $Cfile
echo "include SUBST_GC_WDEP    $ICL_MECH/GC_WDEP.EXT;"            >> $Cfile
echo "include SUBST_AE_SPC     $ICL_MECH/AE_SPC.EXT;"             >> $Cfile
echo "include SUBST_AE_EMIS    $ICL_MECH/AE_EMIS.EXT;"            >> $Cfile
echo "include SUBST_AE_ICBC    $ICL_MECH/AE_ICBC.EXT;"            >> $Cfile
echo "include SUBST_AE_DIFF    $ICL_MECH/AE_DIFF.EXT;"            >> $Cfile
echo "include SUBST_AE_DDEP    $ICL_MECH/AE_DDEP.EXT;"            >> $Cfile
echo "include SUBST_AE_DEPV    $ICL_MECH/AE_DEPV.EXT;"            >> $Cfile
echo "include SUBST_AE_ADV     $ICL_MECH/AE_ADV.EXT;"             >> $Cfile
echo "include SUBST_AE_CONC    $ICL_MECH/AE_CONC.EXT;"            >> $Cfile
echo "include SUBST_AE_A2AQ    $ICL_MECH/AE_A2AQ.EXT;"            >> $Cfile
echo "include SUBST_AE_SCAV    $ICL_MECH/AE_SCAV.EXT;"            >> $Cfile
echo "include SUBST_AE_WDEP    $ICL_MECH/AE_WDEP.EXT;"            >> $Cfile
echo "include SUBST_NR_SPC     $ICL_MECH/NR_SPC.EXT;"             >> $Cfile
echo "include SUBST_NR_EMIS    $ICL_MECH/NR_EMIS.EXT;"            >> $Cfile
echo "include SUBST_NR_ICBC    $ICL_MECH/NR_ICBC.EXT;"            >> $Cfile
echo "include SUBST_NR_DIFF    $ICL_MECH/NR_DIFF.EXT;"            >> $Cfile
echo "include SUBST_NR_DDEP    $ICL_MECH/NR_DDEP.EXT;"            >> $Cfile
echo "include SUBST_NR_DEPV    $ICL_MECH/NR_DEPV.EXT;"            >> $Cfile
echo "include SUBST_NR_ADV     $ICL_MECH/NR_ADV.EXT;"             >> $Cfile
echo "include SUBST_NR_N2AE    $ICL_MECH/NR_N2AE.EXT;"            >> $Cfile
echo "include SUBST_NR_N2AQ    $ICL_MECH/NR_N2AQ.EXT;"            >> $Cfile
echo "include SUBST_NR_SCAV    $ICL_MECH/NR_SCAV.EXT;"            >> $Cfile
echo "include SUBST_NR_WDEP    $ICL_MECH/NR_WDEP.EXT;"            >> $Cfile
echo "include SUBST_TR_SPC     $ICL_TRAC/TR_SPC.EXT;"             >> $Cfile
echo "include SUBST_TR_EMIS    $ICL_TRAC/TR_EMIS.EXT;"            >> $Cfile
echo "include SUBST_TR_ICBC    $ICL_TRAC/TR_ICBC.EXT;"            >> $Cfile
echo "include SUBST_TR_DIFF    $ICL_TRAC/TR_DIFF.EXT;"            >> $Cfile
echo "include SUBST_TR_DDEP    $ICL_TRAC/TR_DDEP.EXT;"            >> $Cfile
echo "include SUBST_TR_DEPV    $ICL_TRAC/TR_DEPV.EXT;"            >> $Cfile
echo "include SUBST_TR_ADV     $ICL_TRAC/TR_ADV.EXT;"             >> $Cfile
echo "include SUBST_TR_T2AQ    $ICL_TRAC/TR_T2AQ.EXT;"            >> $Cfile
echo "include SUBST_TR_SCAV    $ICL_TRAC/TR_SCAV.EXT;"            >> $Cfile
echo "include SUBST_TR_WDEP    $ICL_TRAC/TR_WDEP.EXT;"            >> $Cfile
echo                                                              >> $Cfile

set text = "// Process Analysis / Integrated Reaction Rates processing"
echo $text                                                        >> $Cfile
echo "include SUBST_PACTL_ID    $ICL_PA/PA_CTL.EXT;"              >> $Cfile
echo "include SUBST_PACMN_ID    $ICL_PA/PA_CMN.EXT;"              >> $Cfile
echo "include SUBST_PADAT_ID    $ICL_PA/PA_DAT.EXT;"              >> $Cfile
echo                                                              >> $Cfile

echo "$Str1"                                                      >> $Cfile
```

```
echo "$Str2"                                                    >> $Cfile
echo                                                            >> $Cfile

echo "$ModDriver"                                               >> $Cfile
echo                                                            >> $Cfile

echo "$ModPar"                                                  >> $Cfile
echo                                                            >> $Cfile

echo "$ModInit"                                                 >> $Cfile
echo                                                            >> $Cfile

set text = "denrate and adjcon_noop"
echo "// options are" $text                                    >> $Cfile
echo "$ModAdjc"                                                 >> $Cfile
echo                                                            >> $Cfile

echo "$ModCpl"                                                  >> $Cfile
echo                                                            >> $Cfile

set text = "hbot, hppm and hadv_noop"
echo "// options are" $text                                    >> $Cfile
echo "$ModHadv"                                                 >> $Cfile
echo                                                            >> $Cfile

set text = "vbot, vppm and vadv_noop"
echo "// options are" $text                                    >> $Cfile
echo "$ModVadv"                                                 >> $Cfile
echo                                                            >> $Cfile

set text = "unif, multi_scale and hdiff_noop"
echo "// options are" $text                                    >> $Cfile
echo "$ModHdiff"                                                >> $Cfile
echo                                                            >> $Cfile

set text = "eddy and vdiff_noop"
echo "// options are" $text                                    >> $Cfile
echo "$ModVdiff"                                                >> $Cfile
echo                                                            >> $Cfile

set text = "phot and phot_noop"
echo "// options are" $text                                    >> $Cfile
echo "$ModPhot"                                                 >> $Cfile
echo                                                            >> $Cfile

set text = "ping_qssa, ping_smvgear, ping_mebi_cb4_1 and ping_noop"
echo "// options are" $text                                    >> $Cfile
echo "$ModPing"                                                 >> $Cfile
echo                                                            >> $Cfile

set text = "qssa, smvgear, mebi_cb4_1 and chem_noop"
echo "// options are" $text                                    >> $Cfile
echo "$ModChem"                                                 >> $Cfile
echo                                                            >> $Cfile

set text = "aero1, aero2 and aero_noop"
echo "// options are" $text                                    >> $Cfile
echo "$ModAero"                                                 >> $Cfile
echo                                                            >> $Cfile

set text = "aero_depv1, aero_depv2 and aero_depv_noop"
echo "// options are" $text                                    >> $Cfile
echo "$ModAdepv"                                                >> $Cfile
```

```
 echo                                                                  >> $Cfile

 set text = "cloud_radm and cloud_noop"
 echo "// options are" $text                                          >> $Cfile
 echo "$ModCloud"                                                    >> $Cfile
 echo                                                                  >> $Cfile

 set text = "pa and pa_noop, which requires the"
 echo "// options are" $text "replacement of the three"              >> $Cfile
 set text = "// global include files with their pa_noop counterparts"
 echo $text                                                            >> $Cfile
 echo "$ModPa"                                                       >> $Cfile
 echo                                                                  >> $Cfile

 echo "$ModUtil"                                                     >> $Cfile
 echo                                                                  >> $Cfile

 echo "$Modaeaqsolver"                                               >> $Cfile
 echo                                                                  >> $Cfile

 if ( $?ModMisc ) then
    echo "$ModMisc"                                                  >> $Cfile
    echo                                                            >> $Cfile
    endif

#> make the makefile or the model executable

 if ( $?MakeOpt ) then
    $Blder -make $Cfile   # $Cfile = ${CFG}.bld
    else
    set NoMake
    $Blder $Cfile
    endif
 if ( $status != 0 ) then
    echo "   *** failure in $Blder ***"
    exit 1
    endif
 if ( -e "$Base/${CFG}" ) then
    echo "   >>> previous ${CFG} exists, re-naming to ${CFG}.old <<<"
    unalias mv
#   mv $Base/${CFG} $Base/${CFG}.old
    endif
 cp ${CFG}.bld $Base/${CFG}
 if ( ( $Opt != no_compile ) && \
      ( $Opt != no_link    ) && \
      ( $Opt != parse_only ) && \
      ( $Opt != show_only  ) && \
        $?NoMake ) then
#   mv $MODEL $Base
    endif

 exit
```

APPENDIX B.

Script used to compile Models-3/CMAQ with aero_MADRID2.

```
#! /bin/csh -f

#> RCS file, release, date & time of last delta, author, state, [and locker]
#> $Header$

#> what(1) key, module and SID; SCCS file; date and time of last delta:
#> %W% %P% %G% %U%

setenv BASE /usr/home/models3
setenv M3MODEL $BASE/july2002/models    # code archive
setenv M3TOOLS $BASE/july2002/tools      # libraries

 if ( ! -e $M3MODEL || ! -e $M3TOOLS ) then
    echo "   $M3MODEL or $M3TOOLS directory not found"
    exit 1
    endif
 echo "   Model archive path: $M3MODEL"
 echo "           Tools path: $M3TOOLS"

 set BLD_OS = `uname -s`
 if ($BLD_OS != 'Linux') then
    echo "   $BLD_OS -> wrong makit script for host!"
    exit 1
    endif

 set echo

#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#
#> user choices: cvs archives
 set Project = $M3MODEL/CCTM
 set GlobInc = $M3MODEL/include/release

#> user choices: base directory
#set Base = /your_dir/you/working_dir
 set Base = $cwd

 set APPL  = CACM_aeMADRID2_2sec
 set CFG   = cfg.$APPL
 set MODEL = CCTM_$APPL

#> user choices: m3bld command
#set Opt = compile_all  # force compile, even if object files are current
#set Opt = clean_up     # remove all source files upon successful completion
#set Opt = no_compile   # do everything except compile
#set Opt = no_link      # do everything except link
#set Opt = one_step     # compile and link in one step
#set Opt = parse_only   # checks config file syntax
#set Opt = show_only    # show requested commands but doesn't execute them
 set Opt = verbose      # show requested commands as they are executed
 set MakeOpt            # builds a Makefile instead of the model

#> user choices:  single or multiple processors
#set ParOpt          # multiple PE's

#> user choices: various modules

 set Revision = release       # release = latest CVS revision
```

```
#set Revision = '"STA2_1"'    # monocode (>="REL1_4")
#set Revision = '"REL1_3"'    # last release before monocode
#> NOTE: m3bld will try to compile with existing code; it will not retrieve
#>       new (different release) code. So if your "BLD" directory contains
#>       code from a release different than the one you have specified above,
#>       m3bld will tell you, but will merrily compile the original code.
#>       The workaround is to remove your "BLD" directory and start fresh.

#set ModDriver = ( module ctm_MADRID1        $Revision; )
 set ModDriver = ( module ctm_MADRID2        $Revision; )

 if ( $?ParOpt ) then
    set ModPar = ( module par                $Revision; )
    else
    set ModPar = ( module par_noop           $Revision; )
    endif

 set ModInit   = ( module init_MADRID        $Revision; )

#set ModAdjc   = ( module adjcon_noop        $Revision; )
 set ModAdjc   = ( module denrate            $Revision; )

 set ModCpl    = ( module gencoor            $Revision; )

#set ModHadv   = ( module hadv_noop          $Revision; )
 set ModHadv   = ( module hbot_MADRID        $Revision; )
#set ModHadv   = ( module hppm               $Revision; )

#set ModVadv   = ( module vadv_noop          $Revision; )
 set ModVadv   = ( module vbot               $Revision; )
#set ModVadv   = ( module vppm               $Revision; )

#set ModHdiff  = ( module hdiff_noop         $Revision; )
 set ModHdiff  = ( module unif               $Revision; )
#set ModHdiff  = ( module multiscale         $Revision; )

#set ModVdiff  = ( module vdiff_noop         $Revision; )
 set ModVdiff  = ( module eddy_MADRID        $Revision; )

#set ModPhot   = ( module phot_noop          $Revision; )
 set ModPhot   = ( module phot               $Revision; )
#set ModPhot   = ( module phot_aqCMU         $Revision; )

 set ModPing   = ( module ping_noop          $Revision; )

#set ModChem   = ( module chem_noop          $Revision; )
#set ModChem   = ( module qssa_MADRID1       $Revision; )
#set ModChem   = ( module mebi_cb4_MADRID1   $Revision; )
#set ModChem   = ( module mebi_radm2_cis4_MADRID1  $Revision; )
 set ModChem   = ( module qssa_MADRID2       $Revision; )

#set ModAero   = ( module aero_noop          $Revision; )
#set ModAero   = ( module aero_MADRID1_qssa       $Revision; )
#set ModAero   = ( module aero_MADRID1_mebi       $Revision; )
 set ModAero   = ( module aero_MADRID2_qssa       $Revision; )
#set ModAero   = ( module aero_MADRID2_mebi       $Revision; )

#set ModAdepv  = ( module aero_depv_noop     $Revision; )
 set ModAdepv  = ( module aero_depv_MADRID   $Revision; )

 set ModCloud  = ( module cloud_noop         $Revision; )
#set ModCloud  = ( module cloud_CMU_aero_MADRID1_qssa  $Revision; )
#set ModCloud  = ( module cloud_CMU_aero_MADRID1_mebi  $Revision; )
```

```
#set ModCloud  = ( module cloud_CMU_aero_MADRID2_qssa  $Revision; )
#set ModCloud  = ( module cloud_CMU_aero_MADRID2_mebi  $Revision; )
#set ModCloud  = ( module cloud_RADM_aero_MADRID1_qssa  $Revision; )
#set ModCloud  = ( module cloud_RADM_aero_MADRID1_mebi  $Revision; )

 set ModPa     = ( module pa_MADRID          $Revision; )

 set ModUtil   = ( module util               $Revision; )

 set Modaeaqsolver = ( module ae_aq_solver_MADRID    $Revision; )

#set ModMisc   = ( misc; \
#                  $Base/file1.F \
#                  $Base/file2.F \
#                  $Base/file3.F )

#> user choices: emissions processing in chem or vdiff (default) ...
#set Cemis

#> user choices: vertical layers and mechanism

#set Reso      = 05
#set Grid      = 63X28
 set Layers    = 15
#set Domain    = d${Reso}_${Layers}
 set Mechanism = CACM_aeMADRID2_2sec
 set Tracer    = trac0               # default: no tracer species
#set Decomp    = 1X1

#> user choices: set process analysis linkages
 set PABase    = $GlobInc
 set PAOpt     = pa_noop
#set PABase    = /project/cmaq/yoj/saprc
#set PAOpt     =

#> other user choices set below are:
#>    name of the "BLD" directory
#>    compiler/link flags
#>    library paths
#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#

 set Bld = $Base/BLD_${APPL}
#unset echo
 if ( ! -e "$Bld" ) then
    mkdir $Bld
    else
    if ( ! -d "$Bld" ) then
       echo "   *** target exists, but not a directory ***"
       exit 1
       endif
    endif
#set echo
 cd $Bld

#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#:#

 set MODLOC = ${M3TOOLS}/stenex/${BLD_OS}
#set MODLOC = $BASE/aug2000/scripts/stenex/Linux2
#set seL = se_noop
 set seL = sef90_noop
# Note: If sef90_noop is used instead of se_noop, the values of STX1, STX2
# and STX3 have to be changed in this script.
 if ( $?ParOpt ) then
```

```
    set Mpich = /usr/local/mpich-1.2.4
    set seL = sef90
    set LIB2 = "-L${M3TOOLS}/pario -lpario"
    set LIB3 = "-L${M3TOOLS}/dynmem -ldynmem"
    set LIB4 = "-L${Mpich}/lib -lmpich"
    set Str1 = (// Parallel / Include message passing definitions)
    set Str2 = (include SUBST_MPICH ${Mpich}/include/mpif.h;)
    else
    set seL = sef90_noop
    set LIB2 =
    set LIB3 =
    set LIB4 =
    set Str1 =
    set Str2 =
    endif

 set LIB1 = "-L${MODLOC} -l${seL}"
 set LIB5 = "-L${M3TOOLS}/m3io/${BLD_OS} -lm3io"
 set LIB6 = "-L${BASE}/aug2000/tools/netCDF/${BLD_OS} -lnetcdf"
 set LIBS = "$LIB1 $LIB2 $LIB3 $LIB4 $LIB5 $LIB6"

 set FC = /usr/pgi/linux86/bin/pgf90
 set FP = /usr/pgi/linux86/bin/pgf90

 set F_FLAGS    = "-Mfixed -Mextend -fast -module ${MODLOC} -I."
 set CPP_FLAGS  = "-Mfixed -Mextend"
 set C_FLAGS    = "-v -fast -I."
 set LINK_FLAGS = ""

 set Blder = $M3TOOLS/build/$BLD_OS/m3bld

 set ICL_PAR   = $GlobInc
 set ICL_CONST = $GlobInc
 set ICL_FILES = $GlobInc
 set ICL_EMCTL = $GlobInc
 set ICL_IOAPI = ${M3TOOLS}/m3io/Linux
#set ICL_VCRD  = $GlobInc/vcoord
 set ICL_GRID  = $BASE/july2002/models/include/release/d05_15
 set ICL_MECH  = $GlobInc/$Mechanism
 set ICL_TRAC  = $GlobInc/$Tracer
 set ICL_PA    = $PABase/$PAOpt

 if ( $?Cemis ) then

    set CV = -Demis_chem
    else
    set CV =
    endif

 if ( $?ParOpt ) then   # split to avoid line > 256 char
    set PAR = ( -Dparallel\
                -DINTERPB=PINTERPB\
                -DM3ERR=PM3ERR\
                -DM3EXIT=PM3EXIT\
                -DM3WARN=PM3WARN\
                -DSHUT3=PSHUT3\
                -DWRITE3=PWRITE3 )

    set Popt = SE
    else
    echo "   Not Parallel; set Serial (no-op) flags"
    set PAR =
    set Popt = NOOP
```

```
   endif

 set STX1 = ( -DSUBST_MODULES=${Popt}_MODULES\
              -DSUBST_BARRIER=${Popt}_BARRIER )
 set STX2 = ( -DSUBST_GLOBAL_MAX=${Popt}_GLOBAL_MAX\
              -DSUBST_GLOBAL_MIN=${Popt}_GLOBAL_MIN\
              -DSUBST_GLOBAL_MIN_DATA=${Popt}_GLOBAL_MIN_DATA\
              -DSUBST_GLOBAL_TO_LOCAL_COORD=${Popt}_GLOBAL_TO_LOCAL_COORD\
              -DSUBST_GLOBAL_SUM=${Popt}_GLOBAL_SUM\
              -DSUBST_GLOBAL_LOGICAL=${Popt}_GLOBAL_LOGICAL\
              -DSUBST_LOOP_INDEX=${Popt}_LOOP_INDEX\
              -DSUBST_SUBGRID_INDEX=${Popt}_SUBGRID_INDEX )
 set STX3 = ( -DSUBST_HI_LO_BND_PE=${Popt}_HI_LO_BND_PE\
              -DSUBST_SUM_CHK=${Popt}_SUM_CHK\
              -DSUBST_INIT_ARRAY=${Popt}_INIT_ARRAY\
              -DSUBST_COMM=${Popt}_COMM\
              -DSUBST_MY_REGION=${Popt}_MY_REGION\
              -DSUBST_SLICE=${Popt}_SLICE\
              -DSUBST_GATHER=${Popt}_GATHER\
              -DSUBST_DATA_COPY=${Popt}_DATA_COPY\
              -DSUBST_IN_SYN=${Popt}_IN_SYN )

 setenv CVSROOT $Project

#> make the config file

 set Cfile = ${CFG}.bld
 set quote = '"'

 echo                                                              > $Cfile
 echo "model       $MODEL;"                                       >> $Cfile
 echo                                                             >> $Cfile
 echo "FPP         $FP;"                                          >> $Cfile
 echo                                                             >> $Cfile
#set text = "$quote$CPP_FLAGS $CV $PAR $STX1 $STX2 $STX3$quote;"

 set add = "-DINTERPB=INTERP3"
 set text = "$quote$CPP_FLAGS $CV $add $PAR $STX1 $STX2 $STX3$quote;"

 echo "cpp_flags   $text"                                         >> $Cfile
 echo                                                             >> $Cfile
 echo "f_compiler  $FC;"                                          >> $Cfile
 echo                                                             >> $Cfile
 echo "f_flags     $quote$F_FLAGS$quote;"                         >> $Cfile
 echo                                                             >> $Cfile
 echo "link_flags  $quote$LINK_FLAGS$quote;"                      >> $Cfile
 echo                                                             >> $Cfile
 echo "libraries   $quote$LIBS$quote;"                            >> $Cfile
 echo                                                             >> $Cfile
 echo "global      $Opt;"                                         >> $Cfile
 echo                                                             >> $Cfile

 set text="// layers, mechanism and tracer:"
 echo "$text ${Layers}, ${Mechanism}, ${Tracer}"                  >> $Cfile
 echo "// project archive: ${Project}"                           >> $Cfile
 echo                                                             >> $Cfile

 echo "include SUBST_PE_COMM    $ICL_PAR/PE_COMM.EXT;"           >> $Cfile
 echo "include SUBST_CONST      $ICL_CONST/CONST.EXT;"           >> $Cfile
 echo "include SUBST_FILES_ID   $ICL_FILES/FILES_CTM.EXT;"       >> $Cfile
 echo "include SUBST_EMPR_VD    $ICL_EMCTL/EMISPRM.vdif.EXT;"    >> $Cfile
 echo "include SUBST_EMPR_CH    $ICL_EMCTL/EMISPRM.chem.EXT;"    >> $Cfile
 echo "include SUBST_IOPARMS    $ICL_IOAPI/PARMS3.EXT;"          >> $Cfile
```

```
 echo "include SUBST_IOFDESC    $ICL_IOAPI/FDESC3.EXT;"            >> $Cfile
 echo "include SUBST_IODECL     $ICL_IOAPI/IODECL3.EXT;"           >> $Cfile
#echo "include SUBST_COORD_ID   $ICL_VCRD/COORD_${Layers}L.EXT;"   >> $Cfile
#echo "include SUBST_VGRD_ID    $ICL_VCRD/VGRD_${Layers}.EXT;"     >> $Cfile
 echo "include SUBST_COORD_ID   $ICL_GRID/COORD_63X28.EXT;"        >> $Cfile
 echo "include SUBST_VGRD_ID    $ICL_GRID/VGRD_15.EXT;"            >>$Cfile
 echo "include SUBST_HGRD_ID    $ICL_GRID/HGRD_63X28_1X1.EXT;"     >>$Cfile

 echo "include SUBST_RXCMMN     $ICL_MECH/RXCM.EXT;"               >> $Cfile
 echo "include SUBST_RXDATA     $ICL_MECH/RXDT.EXT;"               >> $Cfile
 echo "include SUBST_PARAMETER  $ICL_MECH/PARAMETER.EXT;"          >> $Cfile
 echo "include SUBST_ARRAYMAP   $ICL_MECH/ARRAYMAP.EXT;"          >> $Cfile
 echo "include SUBST_HETERPAR   $ICL_MECH/HETERPAR.EXT;"           >> $Cfile
 echo "include SUBST_GC_SPC     $ICL_MECH/GC_SPC.EXT;"             >> $Cfile
 echo "include SUBST_GC_EMIS    $ICL_MECH/GC_EMIS.EXT;"            >> $Cfile
 echo "include SUBST_GC_ICBC    $ICL_MECH/GC_ICBC.EXT;"            >> $Cfile
 echo "include SUBST_GC_DIFF    $ICL_MECH/GC_DIFF.EXT;"            >> $Cfile
 echo "include SUBST_GC_DDEP    $ICL_MECH/GC_DDEP.EXT;"            >> $Cfile
 echo "include SUBST_GC_DEPV    $ICL_MECH/GC_DEPV.EXT;"            >> $Cfile
 echo "include SUBST_GC_ADV     $ICL_MECH/GC_ADV.EXT;"             >> $Cfile
 echo "include SUBST_GC_CONC    $ICL_MECH/GC_CONC.EXT;"            >> $Cfile
 echo "include SUBST_GC_G2AE    $ICL_MECH/GC_G2AE.EXT;"            >> $Cfile
 echo "include SUBST_GC_G2AQ    $ICL_MECH/GC_G2AQ.EXT;"            >> $Cfile
 echo "include SUBST_GC_SCAV    $ICL_MECH/GC_SCAV.EXT;"            >> $Cfile
 echo "include SUBST_GC_WDEP    $ICL_MECH/GC_WDEP.EXT;"            >> $Cfile
 echo "include SUBST_AE_SPC     $ICL_MECH/AE_SPC.EXT;"             >> $Cfile
 echo "include SUBST_AE_EMIS    $ICL_MECH/AE_EMIS.EXT;"            >> $Cfile
 echo "include SUBST_AE_ICBC    $ICL_MECH/AE_ICBC.EXT;"            >> $Cfile
 echo "include SUBST_AE_DIFF    $ICL_MECH/AE_DIFF.EXT;"            >> $Cfile
 echo "include SUBST_AE_DDEP    $ICL_MECH/AE_DDEP.EXT;"            >> $Cfile
 echo "include SUBST_AE_DEPV    $ICL_MECH/AE_DEPV.EXT;"            >> $Cfile
 echo "include SUBST_AE_ADV     $ICL_MECH/AE_ADV.EXT;"             >> $Cfile
 echo "include SUBST_AE_CONC    $ICL_MECH/AE_CONC.EXT;"            >> $Cfile
 echo "include SUBST_AE_A2AQ    $ICL_MECH/AE_A2AQ.EXT;"            >> $Cfile
 echo "include SUBST_AE_SCAV    $ICL_MECH/AE_SCAV.EXT;"            >> $Cfile
 echo "include SUBST_AE_WDEP    $ICL_MECH/AE_WDEP.EXT;"            >> $Cfile
 echo "include SUBST_NR_SPC     $ICL_MECH/NR_SPC.EXT;"             >> $Cfile
 echo "include SUBST_NR_EMIS    $ICL_MECH/NR_EMIS.EXT;"            >> $Cfile
 echo "include SUBST_NR_ICBC    $ICL_MECH/NR_ICBC.EXT;"            >> $Cfile
 echo "include SUBST_NR_DIFF    $ICL_MECH/NR_DIFF.EXT;"            >> $Cfile

 echo "include SUBST_NR_DDEP    $ICL_MECH/NR_DDEP.EXT;"            >> $Cfile
 echo "include SUBST_NR_DEPV    $ICL_MECH/NR_DEPV.EXT;"            >> $Cfile
 echo "include SUBST_NR_ADV     $ICL_MECH/NR_ADV.EXT;"             >> $Cfile
 echo "include SUBST_NR_N2AE    $ICL_MECH/NR_N2AE.EXT;"            >> $Cfile
 echo "include SUBST_NR_N2AQ    $ICL_MECH/NR_N2AQ.EXT;"            >> $Cfile
 echo "include SUBST_NR_SCAV    $ICL_MECH/NR_SCAV.EXT;"            >> $Cfile
 echo "include SUBST_NR_WDEP    $ICL_MECH/NR_WDEP.EXT;"            >> $Cfile
 echo "include SUBST_TR_SPC     $ICL_TRAC/TR_SPC.EXT;"             >> $Cfile
 echo "include SUBST_TR_EMIS    $ICL_TRAC/TR_EMIS.EXT;"            >> $Cfile
 echo "include SUBST_TR_ICBC    $ICL_TRAC/TR_ICBC.EXT;"            >> $Cfile
 echo "include SUBST_TR_DIFF    $ICL_TRAC/TR_DIFF.EXT;"            >> $Cfile
 echo "include SUBST_TR_DDEP    $ICL_TRAC/TR_DDEP.EXT;"            >> $Cfile
 echo "include SUBST_TR_DEPV    $ICL_TRAC/TR_DEPV.EXT;"            >> $Cfile
 echo "include SUBST_TR_ADV     $ICL_TRAC/TR_ADV.EXT;"             >> $Cfile
 echo "include SUBST_TR_T2AQ    $ICL_TRAC/TR_T2AQ.EXT;"            >> $Cfile
 echo "include SUBST_TR_SCAV    $ICL_TRAC/TR_SCAV.EXT;"            >> $Cfile
 echo "include SUBST_TR_WDEP    $ICL_TRAC/TR_WDEP.EXT;"            >> $Cfile
 echo                                                             >> $Cfile

 set text = "// Process Analysis / Integrated Reaction Rates processing"
 echo $text                                                       >> $Cfile
#echo "include SUBST_PACTL_ID    $ICL_PA/PA_CTL_no_irr.EXT;"      >> $Cfile
```

```
echo "include SUBST_PACTL_ID    $ICL_PA/PA_CTL.EXT;"              >> $Cfile
echo "include SUBST_PACMN_ID    $ICL_PA/PA_CMN.EXT;"              >> $Cfile

echo "include SUBST_PADAT_ID    $ICL_PA/PA_DAT.EXT;"              >> $Cfile
echo                                                              >> $Cfile

echo "$Str1"                                                      >> $Cfile
echo "$Str2"                                                      >> $Cfile
echo                                                              >> $Cfile

echo "$ModDriver"                                                 >> $Cfile
echo                                                              >> $Cfile

echo "$ModPar"                                                    >> $Cfile
echo                                                              >> $Cfile

echo "$ModInit"                                                   >> $Cfile
echo                                                              >> $Cfile

set text = "denrate and adjcon_noop"
echo "// options are" $text                                       >> $Cfile
echo "$ModAdjc"                                                   >> $Cfile
echo                                                              >> $Cfile

echo "$ModCpl"                                                   >> $Cfile
echo                                                              >> $Cfile

set text = "hbot, hppm and hadv_noop"
echo "// options are" $text                                       >> $Cfile
echo "$ModHadv"                                                   >> $Cfile
echo                                                              >> $Cfile

set text = "vbot, vppm and vadv_noop"
echo "// options are" $text                                       >> $Cfile
echo "$ModVadv"                                                   >> $Cfile
echo                                                              >> $Cfile

set text = "unif, multi_scale and hdiff_noop"
echo "// options are" $text                                       >> $Cfile
echo "$ModHdiff"                                                  >> $Cfile
echo                                                              >> $Cfile

set text = "eddy and vdiff_noop"
echo "// options are" $text                                       >> $Cfile
echo "$ModVdiff"                                                  >> $Cfile
echo                                                              >> $Cfile

set text = "phot and phot_noop"
echo "// options are" $text                                       >> $Cfile
echo "$ModPhot"                                                   >> $Cfile
echo                                                              >> $Cfile

set text = "ping_qssa, ping_smvgear, ping_mebi_cb4_1 and ping_noop"
echo "// options are" $text                                       >> $Cfile
echo "$ModPing"                                                   >> $Cfile
echo                                                              >> $Cfile

set text = "qssa, smvgear, mebi_cb4_1 and chem_noop"
echo "// options are" $text                                       >> $Cfile
echo "$ModChem"                                                   >> $Cfile
echo                                                              >> $Cfile

set text = "aero1, aero2 and aero_noop"
```

```
 echo "// options are" $text                                          >> $Cfile
 echo "$ModAero"                                                     >> $Cfile
 echo                                                                >> $Cfile

 set text = "aero_depv1, aero_depv2 and aero_depv_noop"
 echo "// options are" $text                                          >> $Cfile
 echo "$ModAdepv"                                                    >> $Cfile
 echo                                                                >> $Cfile

 set text = "cloud_radm and cloud_noop"
 echo "// options are" $text                                          >> $Cfile
 echo "$ModCloud"                                                    >> $Cfile
 echo                                                                >> $Cfile

 set text = "pa and pa_noop, which requires the"
 echo "// options are" $text "replacement of the three"              >> $Cfile
 set text = "// global include files with their pa_noop counterparts"
 echo $text                                                          >> $Cfile
 echo "$ModPa"                                                       >> $Cfile
 echo                                                                >> $Cfile

 echo "$ModUtil"                                                     >> $Cfile
 echo                                                                >> $Cfile

 echo "$Modaeaqsolver"                                              >> $Cfile
 echo                                                               >> $Cfile

 if ( $?ModMisc ) then
    echo "$ModMisc"                                                 >> $Cfile
    echo                                                            >> $Cfile
    endif

#> make the makefile or the model executable

 if ( $?MakeOpt ) then
    $Blder -make $Cfile   # $Cfile = ${CFG}.bld
    else
    set NoMake
    $Blder $Cfile
    endif
 if ( $status != 0 ) then
    echo "   *** failure in $Blder ***"
    exit 1
    endif
 if ( -e "$Base/${CFG}" ) then
    echo "   >>> previous ${CFG} exists, re-naming to ${CFG}.old <<<"
    unalias mv
#   mv $Base/${CFG} $Base/${CFG}.old
    endif
 cp ${CFG}.bld $Base/${CFG}
 if ( ( $Opt != no_compile ) && \
      ( $Opt != no_link    ) && \
      ( $Opt != parse_only ) && \
      ( $Opt != show_only  ) && \
        $?NoMake ) then
#   mv $MODEL $Base
    endif

 exit
```