DESIGN AND DEVELOPMENT OF MAQM, AN AIR QUALITY MODEL ARCHITETCTURE WITH PROCESS-LEVEL MODULARITY

Weimin Jiang*, Helmut Roth, Qiangliang Li, Steven C. Smyth, Fuquan Yang
Institute for Chemical Process and Environmental Technology, National Research Council of Canada, Ottawa, Ontario, Canada

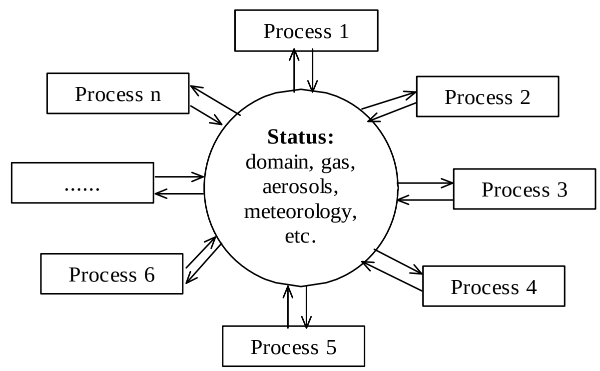
1. INTRODUCTION

Atmospheric processes affecting air quality are very complicated, and scientific understanding of these processes has been evolving rapidly. To successfully simulate concentration levels and changes of atmospheric pollutants, it is highly desirable to have air quality models with full and flexible modularity at the science process level. Such a modular air quality model would significantly enhance model transparency, facilitate scientific understanding, promote science algorithm development, and enable effective error checking and corrections. It would open the door for broad participation of the air quality modelling community in the model development and improvement processes, which is essential to ensure complete and robust science in the models.

Recently, we have completed the design and development of a software architecture that can be used to build modular air quality models. The architecture, named MAQM (Modular Air Quality Model), treats different processes as individual blocks. These blocks can be easily added, removed, modified, and/or reorganized, either individually or in combination. To demonstrate the MAQM concept and feasibility, a 3-D air quality model has also been developed and tested for a North American domain. The model is named MAQM/CMAQ to indicate that it is based on the MAQM architecture, and its science algorithms are mostly adapted from the Community Multiscale Air Quality (CMAQ) model developed by the United States Environmental Protection Agency (US EPA) (EPA 1999; Binkowski and Roselle, 2003).

2. CONCEPTUAL MAQM ARCHITECTURE

MAQM tries to mimic the natural environment as closely and as directly as possible within the constraints of operator-splitting. The following figure shows a simplified conceptual architecture of MAQM, which is built on the structural components and design principles discussed below.



1. *Status and process:*

In the context of MAQM, a status (or natural status) is a numerical description of the modelled environmental situation in the modelling domain at a particular time and modelling stage. It includes quantities that change with time and location, such as concentrations of gaseous species, chemical and size-related quantities of particles, and meteorological conditions. It is also extended to include other relevant quantities and properties that do not change with time and/or location, such as model grid definition, grid cell dimensions, terrain height, molecular weights, etc.

A process is a software representation of an event or a combination of events that brings about changes in certain quantities of the natural status. Emissions, gas phase chemistry, nucleation, and advection are all examples of MAQM processes. In MAQM, the concept of a process is also extended to include a set of computer code that extracts and outputs certain status information as required.

The main objective of a process is to modify and/or output certain status quantities, based on the status information available at the beginning of

**Corresponding author:* Weimin Jiang, National Research Council of Canada, Room 233, M2,1200 Montreal Road, Ottawa, Ontario, Canada K1A 0R6; e-mail: Weimin.jiang@nrc-cnrc.gc.ca

a time step through which a process progresses. It is also possible that a process may output certain other local quantities as required.

Note that the conceptual MAQM architecture does not restrict how the status will be modified by individual processes and where the new input data will come from. For example, modification of certain gaseous species concentrations by the emission process could be caused by reading in emissions from a file generated off-line, or by generating emissions directly on-line.

2. *Global status data repository: uniqueness of shared status information*

All status quantities in the modelling domain at a particular time that are used by multiple processes, e.g., concentrations, molecular weights, temperature, grid information, etc, are uniquely defined and represented in an on-line data repository. The data repository has a global scope and can be accessed by all relevant processes.

3. *Self-contained process modules*

A process module contains all the executable statements, data, and parameters required by itself, except the global status information available in the data repository mentioned above.

A process module is free to obtain any other input information through its own channels, such as specific input data files. However, any such information must not cause any conflicts with the unique status information given in the data repository. A process module is also allowed to output any quantities to its own output files with unique file names. All input or output files used by a process module are considered as integral parts of the module by MAQM.

4. *Structural separation of process modules*

In MAQM, each process module is an independent entity. Different process modules are treated as individual blocks, and are structurally separated. There are no direct relationships among different process modules. All data exchanges among different process modules, if any, are carried out indirectly through the global data repository. For example, a gas chemistry module needs temperature data, which are generated or obtained by a meteorology module. Instead of passing the temperature data from the meteorology module to the gas chemistry module directly, the meteorology module modifies the temperature field in the data repository, and the gas chemistry module obtains the temperature data from the data repository. As soon as the temperature field in the data repository is modified by the meteorological module, all process modules can access the same temperature data independently without repetitively calling the meteorological module and without direct data exchange with the meteorological module.

Note that a process is defined in Point 1 above as a software representation of an event or a combination of events that brings about status changes. That is to say that process modules can be organized in a flexible way. A process module can be as simple or as complicated as required. For those processes that are fundamentally un-separable, they can be combined into a single module, which is structurally separated from other modules. However, for the sake of maintaining the overall modular paradigm, the creation of these multi-process modules should be avoided if possible.

5. *Structural separation and linkages of status and processes*

As mentioned above, shared status quantities are stored in a uniquely defined global data repository and are structurally separated from the process code. However, the status is also closely linked to the processes through accessing the status quantities by the processes. Same status quantities can be accessed by different processes, and a process can access multiple and selective status quantities in the data repository. This is analogous to a modern computer hardware architecture, under which different devices and components carry out various functions by selectively accessing multiple data items stored in memories through bus wires.

A good implementation of the MAQM architecture would maintain relative independence of the status data repository and the process code, and minimize the need to change process code when status data are changed, and vice versa.

6. *The MAQM driver and its interface with the process modules*

In MAQM, all operations related to the status quantities are carried out by process modules. The MAQM driver coordinates the actions of individual process modules by calling them in sequence according to a common user-specified time step length. The MAQM driver is kept at the simplest possible form in structure.

The interface between the MAQM driver and the process modules is also kept simple. The MAQM driver passes only the time stamp and step length information to individual process modules. There is no information passed back from the process modules to the MAQM driver. Results of the process module operations are reflected in the change of status quantities in the status data repository and/or in the output files generated by the process modules.

If needed, each process module can determine and use its own internal time step length(s). However, only the status data after the time step specified by the MAQM driver are reported back to the status data repository.

7. *Minimized usage of hardcode and repetitive code*

Whenever possible, MAQM avoids using hardcoded variable names, indices, and pointers. As a consequence, usage of repetitive code is minimized.

3. IMPLEMENTATION OF THE CONCEPTUAL MAQM ARCHITECTURE IN FORTRAN

The global status data repository can be effectively realized through a set of dedicated FORTRAN module program units, i.e., the status data are represented by a set of FORTRAN data modules.

The process modules can be implemented either through FORTRAN module program units, or through sets of FORTRAN subroutines and functions. The top level subroutine in a process module is called by the MAQM driver. It accepts the time stamp and the model step length from the driver. Subroutines at any levels of a process module can access the status data in the data modules via the USE statements. The data can be selectively accessed by including the ONLY option in the USE statements.

4. THE MAQM/CMAQ MODEL: DEMONSTRATION OF THE MAQM CONCEPT AND FEASIBILITY

MAQM/CMAQ is a model that we developed to demonstrate the MAQM concept and feasibility, although it has gone beyond our original expectation and become a fully functioning model. A preliminary evaluation for a continental North American domain over the month of July 2002 showed that in comparison with CMAQ version 4.5, MAQM/CMAQ generated almost identical spatial and temporal O_3 distribution patterns with noticeably better performance in O_3 biases and errors. The MAQM/CMAQ aerosol performance also appeared to be better.

The following sections focus on the original objective of the MAQM/CMAQ development by briefly presenting the structure, composition, functionality, and capacities of MAQM/CMAQ only. Detailed evaluation of MAQM/CMAQ will be presented elsewhere.

4.1 Overview of the MAQM/CMAQ model

MAQM/CMAQ is built on the conceptual MAQM architecture and programmed in FORTRAN. The science algorithms implemented in the model are mostly adapted from CMAQ version 4.5, along with some in-house additions and revisions. When adapting algorithms from CMAQ, we tried to use the original CMAQ code whenever possible. However, we also made substantial changes and/or recoding to correct mistakes, simplify the code, and facilitate the integration of the CMAQ code into MAQM/CMAQ.

The saprc99_ae4_aq mechanism available in the CMAQ release is used as the basis of the gas chemical mechanism in MAQM/CMAQ. It was made compatible with our revised implementation of Schell's secondary organic aerosol (SOA) algorithm (Schell et al., 2001) by adding 10 semi-volatile organic compounds (SVOCs) in appropriate reactions and removing the 6 pseudo species representing reacted amounts of organic precursors. The revised mechanism file, named saprc99_ae4_aq_SVOC.def, was reformatted by reorganizing its kinetic information and adding new information regarding some properties of the chemical species, such as molecular weights. It was then processed through a new chemical mechanism processor, PROCCHEM, developed in-house to automatically generate a list of reactive chemical species and their molecular weights and several source code files. The species list is included in the gas_data module and the source code files are integrated into the gas_chem process module to be discussed later.

For aerosol modelling, the CMAQ approach of representing particles as three superimposed log normal size distributions, or modes, is adapted. However, the approach is structurally generalized so that it could be changed to use any number of modes. In addition, the code is generalized so that each mode contains the same 20 aerosol species. These species could be adjusted when needed. The 20 species contain all the aerosol species used by CMAQ, except that the 2 CMAQ SOA species are replaced by the 10 SVOC species partitioned in aerosols. Whenever feasible, hardcoded variable names, indices, and repetitive code specific to a given species and size mode are eliminated.

To facilitate the comparison of model results with size-resolved measurement data, MAQM/CMAQ generates and outputs not only prognostic modal aerosol quantities, but also diagnostic sectional aerosol quantities, which include total and speciated aerosol concentrations at specific size cutoffs. The size cutoffs can be easily set up by the users. At the moment, they are set up at default values of 0.01, 0.1, 1, 2.5, 5.0, and 10 μm.

The current version of MAQM/CMAQ contains 20 process modules and 5 global data modules. The data modules are shared and accessed by all the process modules.

4.2 The MAQM/CMAQ data modules

Four data modules, grid_data, gas_data, aero_data, and met_data, are used to store relevant information of the model grid, gaseous species, aerosol quantities, and meteorological variables, respectively. Another data module, const_data, contains commonly used mathematical, chemical, physical, and meteorological constants, as well as minimum concentrations of gas and aerosol quantities, which represent nominal zeros of theses quantities in MAQM/CMAQ.

4.3 The MAQM/CMAQ process modules

The 20 process modules currently implemented in MAQM/CMAQ are:

- **grid**:
  definition and set up of quantities related to model grid and model domain.
- **init_met**:
  meteorological fields initialization.
- **init_gas**:
  gas species concentration initialization.
- **init_aero**:
  aerosol quantity initialization.
- **BConc_gas**:
  gas species boundary condition.
- **BConc_aero**:
  aerosol quantity boundary condition.
- **emission**:
  gas and particle emissions.
- **gas_chem**:
  gas phase chemistry.
- **nucleation**:
  particle formation through nucleation.
- **condensation**:
  H_2SO_4 condensation on particles.
- **inor_aero**:
  inorganic gas/aerosol partitioning and equilibrium.
- **SOA**:
  secondary organic aerosol formation/evaporation through gas/aerosol partitioning.
- **coagulation**:
  particle coagulation.
- **advection**:
  gas and particle advection.
- **diffusion**:
  gas and particle diffusion.
- **cloud**:
  cloud processes affecting gas and aerosols, including wet deposition.
- **ddep_gas**:
  dry deposition of gaseous species.
- **ddep_part**:
  dry deposition of particles.
- **meteorology**:
  meteorological variable set-up and/or calculation.
- **output**:
  generation and output of required gas and aerosol quantities to output files.

4.4 The MAQM/CMAQ utilities

MAQM/CMAQ uses CMAQ pre-processors MCIP and JPROC. In addition, it uses three pre-processors that we developed in-house for chemical mechanism processing and initial and boundary condition preparation.

MAQM/CMAQ also contains several utility subroutines that can be called by any other program units within MAQM/CMAQ for maintenance, housekeeping, and quality control purposes.

4.5 The MAQM/CMAQ input & output files

As in CMAQ, the current version of MAQM/CMAQ reads in MCIP-generated meteorological input files and SMOKE-generated emission files directly. This ensures the compatibility of MAQM/CMAQ and CMAQ in terms of using the same meteorological and emission input files. MAQM/CMAQ reads in initial (IC) and boundary (BC) conditions from a set of ASCII files, which allow more flexible set-up of initial and boundary concentration values in different grid cells. The standard CMAQ IC and BC profiles can be converted to the MAQM/CMAQ IC and BC ASCII files by using the utility programs that we developed in-house. The CMAQ IC and BC pre-processors are not used by MAQM/CMAQ.

For each day of the model run, MAQM/CMAQ generates the following three output files in I/O API format. Users can specify to use either the conventional or SI unit for the output files. In the conventional units, gas and aerosol mass concentrations are given in ppm and $\mu g/m^3$, respectively. In the SI unit, the two quantities are $mole/m^3$ and kg/m^3, respectively.

- GasConc_yyyyddd: hourly gas concentrations on the day ‘yyyyddd’.
- AeroConc_yyyyddd: hourly modal aerosol quantities on the day ‘yyyyddd’, which include concentrations of total and speciated mass, number, and surface area of all particle modes, as well as geometric mean diameters and geometric standard deviations of all the particle number distributions.
- PMx_yyyyddd: hourly particle concentrations within specific size ranges on the day ‘yyyyddd’, which include total and speciated mass, number, and surface area concentrations. As mentioned in Secton 4.1, the cutoff sizes are 0.01, 0.1, 1, 2.5, 5, and 10 µm by default. The cut-off values can be easily changed by users.

5. POSSIBLE FUTURE WORK

The simple and flexible modular architecture of MAQM has laid the foundation for numerous possibilities for future work on model development, evaluation, science implementation, improvement, and applications. It opens doors for broad and effective participation in model development by the modelling community. Here, we briefly discuss some possible future work that could be done either by ourselves and/or by the community.

5.1 Develop new models or restructure current models using the MAQM architecture

The MAQM architecture, including its design principles, could be readily adapted for developing new models or restructuring current models.

The implementation of the architecture in MAQM/CMAQ could also be used as a prototype in future model development. For the modal approach of aerosol modelling, the number of modes and aerosol species in MAQM/CMAQ could be changed, along with adjustments of the current algorithms or the addition of new code. For the sectional approach, the current matrix of modes vs. species in the MAQM/CMAQ aerosol quantity definition could be readily converted to the definition of size ranges vs. species. Program code can then be developed in a similar fashion depending on the scientific formulations used.

5.2 Investigate and improve current science in individual process modules

Due to the relative structural independence of process modules, science implementation details and their embedded assumptions in any particular module in MAQM/CMAQ or other models based on the MAQM architecture can be effectively recovered, rigorously investigated and tested, and further improved. Broad participation by the modelling community in the investigation and improvement of embedded science details is the only way to ensure the correctness and robustness of the science algorithms implemented in complicated air quality models.

5.3 Add new or alternative process modules

New and/or alternative process modules can be added to MAQM/CMAQ or other models using the MAQM architecture with minimal interference from other process modules.

5.4 Revise current meteorology and/or emission modules to use meteorology and emissions input files from alternative sources

The revision is to accept input files with different file formats, other than the I/O API currently used, and to change the status data in the data modules using the input data available in alternative input files generated by other meteorological models or emissions processors.

5.5 Implement on-line meteorology and/or emissions modules

Instead of using off-line meteorology and emission input files, on-line meteorology and/or emissions modules can be developed and implemented in MAQM/CMAQ or other models using the MAQM architecture. These modules will be structurally independent of other process modules and their implementation will be relatively straightforward. By conducting meteorological modelling and emissions modelling/processing on-line, two-way feedbacks between air quality and meteorology and between meteorology and emissions can be achieved.

5.6 Implement time-varying boundary conditions

Subject to the availability of data or algorithms, the boundary condition modules in MAQM-based models can be easily revised to update gas and aerosol concentrations in boundary cells according to time-varying boundary conditions.

5.7 Process analysis

Under MAQM, process modules can be easily added, removed, and regrouped. Process analysis can be conducted by adding, removing, and/or regrouping certain process modules, outputting quantities of interest, and comparing the results with different groupings of process modules. Alternatively, depending on the needs of specific project(s), specific process analysis module(s) can be developed and inserted into desired locations of the process module stream in a MAQM-based model in a flexible way.

6. ACKENOWLEDGEMENT

Most science algorithms implemented in the current version of MAQM/CMAQ were adapted from the CMAQ model. Early ideas of air quality model modularity and the spirit of community modelling have been built into the development of the CMAQ system, which inspired our thinking in designing and developing the MAQM architecture and the MAQM/CMAQ model presented in this paper. In addition, many organizations have contributed to the CMAQ code that was adopted and/or adapted by MAQM/CMAQ. The contributions by the CMAQ development team at the US EPA and all the other organizations are very much appreciated. We also thank the Community Modelling and Analysis (CMAS) Center at the University of North Carolina at Chapel Hill for their work in distributing and supporting CMAQ and other associated modelling tools that are used in this project.

7. REFERENCES

Binkowski, F.S., Roselle, S.J., 2003. Models–3 Community Multiscale Air Quality (CMAQ) model aerosol component, 1, Model description, *J. Geophys. Res.,* **108**(D6), 4183, doi:10.1029/2001JD001409.

EPA, 1999. Science algorithms of the EPA Models-3 Community Multiscale Air Quality (CMAQ) modeling system, edited by D.W. Byun and J.K.S. Ching, EPA/600/R-99/030, Office of Research and Development, United States Environmental Protection Agency, Washington. DC, USA, March 1999.

Schell, B., Ackermann, I.J., Hass, H., Binkowski, F.S., Ebel, A., 2001. Modeling the formation of secondary organic aerosol within a comprehensive air quality model system, *J. Geophys. Res., 106*(D22), 28,275–28,293.