7.5

MCPL: An I/O API Output Module for MM5

Carlie J. Coats, Jr., and John N. McHenry*
MCNC Environmental Modeling Center, RTP, NC
e-mail: coats@emc.mcnc.org
Web address: http://www.emc.mcnc.org/projects/ppar/mcpl.html
Voice (919) 248-9241 Fax (919) 248-9245

1. INTRODUCTION

`MCPL` is a drop-in MM5 module designed to fit into MM5 with minimal effort, to be callable at a variety of time scales from the advection-step frequency on up. It provides functionality generalizing the present MM5 subroutine `OUTTAP`, but directly using the Models-3 I/O API [Coats, et al., 1993; Coats 1998] for its output. `MCPL` is designed for easy extensibility. Because `MCPL` uses the I/O API to provide both file storage coupled-model output, it allows MM5 to couple with other concurrently-executing models. `MCPL` is extremely flexible and configurable—which variables to output, which format to output them in, and which windows into which nests to output them for—are chosen at MM5 launch time by setting environment variable flags. The current version provides the options for writing out any combination of variables from the following:

- 2-D and 3D native MM5 variables
- 2D and 3D variables for SMOKE, MAQSIP, and CMAQ air quality modeling
- 2D variables for the TOPMODEL-Based Land Surface-Atmosphere Transfer Scheme (TOPLATS) hydrology model
- various other diagnostic variables (e.g., wind divergence and vorticity)
- Pressure-surface-interpolated 3-D variables, and pressure-surface-interpolation coeficients

`MCPL` was developed under the "Practical Parallel Computing" EPA STAR-96 cooperative agreement and has been used extensively for numerical air quality forecasting, regulatory applications, and coupling MM5 with other environmental models.

`MCPL` and its options are described in detail on web page *http://envpro.ncsc.org/projects/ppar/mcpl.html* .

* *Corresponding author address:* Carlie J. Coats, Jr., MCNC Environmental Modeling Center, P.O. Box 13261 RTP, NC 27709

2. MCPL FUNCTIONALITY IN MM5

`MCPL` is designed for extremely easy insertion into the MM5 source code. MM5's top-level file, `mm5.F`, is modified by adding 13 lines of code (including comments) in a block following the model initialization section. An additional five lines of code are modified in the code for subroutine `SOLVE`. For air quality modeling, there is an additional include file `addmdl3.incl` containing common block `/ADDMDL3/`, together with modifications to the relevant MM5 physics subroutines to save cloud and ground-slab characteristics.

MCPL is principally controlled by environment variables and by ASCII tables found in files under standard logical names, according to Models-3 conventions. It is called through two entry points:

- `MCPL_GRID` has no arguments, must be called first, and sets up the grid geometry for subsequent calls.
- `MCPL_OUT( NEST )` writes the output files for each time step of each nest. It is called immediately after initializing each nest in MM5, so as to write out the initial conditions to the files for that nest, as well as at the end of `SOLVE`, to write out files at the appropriate output time steps.

Because other models with which MM5 may be coupled (particularly air quality models) must be able to run on windows within MM5 domains, `MCPL` is designed to perform this function. It also computes derived variables (e.g., relative humidity) required by these other models. `MCPL` uses a variety of simple worker routines to extract the requested variables from MM5 common blocks; where appropriate, to compute, decouple, and/or perform units conversions; and to re-order the result into standard I/O API column-row-layer subscript order.

MCPL's output files are completely customizable by the user at MM5 program-launch time, in terms of nests and variables being output.

3. MCPL AND MODEL COUPLING

For coupling MM5 with other concurrently-running (parallel) environmental models, MCPL uses the PVM-based c**oupling-mode of the I/O API.** In this mode, `WRITE3` operations put data to "virtual files" implemented as PVM mailboxes, and `READ3` operations read data from those same mailboxes. These operations are still selective direct access by file-name, variable-name, date and time. PVM's blocking property (that `READ3` requests for data not yet written put the requester to sleep until the data becomes available) manages synchronization between the models. Senders and receivers do not need to know about each other (needing only the same `setenv`'s to identify the mailboxes to each other), nor do they need to read and write the data in the same order. One sender may well have multiple receivers, as in the coupled modeling projects described below. In setting up a coupled modeling system this way, there are two principal pitfalls the modeler needs to worry about:

- deadlocks in which two processes are both put to sleep because each is expecting data from the other; and
- one-way coupling of concurrent models, where one model might "race ahead" of the other: the modeler needs to introduce artificial feedbacks so that the processes stay synchronized.

We are currently using `MCPL` in MM5 for a number of on-going projects, including:

- Real-time numerical air quality prediction (RT-NAQP) using MM5, SMOKE, and MAQSIP (see McHenry presentation, this conference);
- Coupling MM5 with the TOPLATS very high resolution land-surface hydrology model

ACKNOWLEDGEMENTS

The authors graciously thank Dr. Nelson Seaman and colleagues at the Pennsylvania State University for help testing and validating `MCPL`. Although the research described in this article has been funded in part by the United States EPA through grants R825210, R825211, and R825388 to MCNC, and grant R825254 to the Pennsylvania State University, it has not been subjected to the Agency's required peer and policy review and therefore does not necessarily reflect the views of the Agency and no official endorsement should be inferred.

6.0 REFERENCES

Coats, C.J., Jr., 1996. High Performance Algorithms in the Sparse Matrix Operator Kernel Emissions (SMOKE) Modeling System. Ninth AMS Joint Conference on Applications of Air Pollution Meteorology with A&WMA, 584-588.

Coats, C.J., Jr., 1997-2002. The EDSS/Models-3 I/O API. URL *http://www.emc.mcnc.org/products/ioapi/*

Coats, C.J., Jr., and M.Houyoux, 1998. The Sparse-Matrix Operator Kernel Emissions (SMOKE) Modeling System. URL *http://www.emc.mcnc.org/products/smoke/*

Coats, C.J., Jr., A.F.Hanna, D.Hwang, and D.W.Byun, 1993. Model Engineering Concepts for Air Quality Models in an Integrated Environmental Modeling System. Trans. AWMA Specialty Conf. on Regional Photochemical Measurement and Modeling Studies.

Fine, S.S., and J.Ambrosiano, 1996. The Environmental Decision Support System: Overview and Air Quality Application. AMS Symp. Env. App.

Geist, Al, A.Beguelin, J. Dongarra, W.Jiang, R.Manchek, V.Sunderam, 1994. PVM: Parallel Virtual Machine, A Users' Guide and Tutorial for Networked Parallel Computing. MIT Press.

McHenry, J.N., C.D.Peters-Lidard, C.J. Coats Jr., S.Fine, and A.Trayanov, 1998. DASHMM. Data-Assimilating Hydrological Meteorological Model. URL http:// *www.emc.mcnc.org /projects/dashmm*

Peters-Lidard, C.D., M.S.Zion, and E.F.Wood, 1997. A soil-vegetation-atmosphere transfer scheme for modeling spatially variable water and energy balance processes. J Geophys Res., 102(D4), 4303-4324.

PVM: Parallel Virtual Machine, 1998. URL *http://www.epm.ornl.gov:80/pvm*

UNIDATA PROGRAM CENTER, 1998. NetCDF. URL *http://www.unidata.ucar.edu/packages/netcdf /index.htm*