# AUTOMATIC CODING OF CHEMISTRY SOLVERS IN WRF-CHEM USING KPP

M. Salzmann* and M. G. Lawrence

Max-Planck-Institute for Chemistry, Department of Atmospheric Chemistry, Mainz, Germany

## 1. Introduction

Coupled state of the art meteorology/chemistry models such as WRF-Chem (Grell et al. 2005) typically include hundreds of reactions and dozens of chemical species. Solving the corresponding huge systems of ordinary differential equations requires highly efficient numerical integrators. In the case of hard-coded manually "tuned" solvers, even minor changes to the chemical mechanism, such as updating the mechanism by additional equations, often require recasting the equation system and, consequently, major revisions of the code. This procedure is both extremely time consuming and error prone. In recent years, automatic code generation has become an appreciated and widely used tool to overcome these problems. The Kinetic PreProcessor (KPP, Damian et al. 2002; Sandu et al. 2003; Sandu and Sander 2006) is a computer program which reads chemical equations and reaction rates from an ASCII input file provided by the user (see Example File 1) and writes the program code necessary to perform the numerical integration. Efficiency is obtained by automatically reordering the equations in order to exploit the sparsity of the Jacobian. In a still experimental setup, KPP Version 2.1 has been adapted for WRF-Chem to produce Fortran 90 modules which can be used in WRF-Chem without further modifications. Furthermore, a preprocessor for WRF-Chem has been developed which automatically generates the interface routines between the KPP-generated modules and WRF-Chem, based on entries from the WRF-Chem registry and on KPP input files. This WRF-Chem/KPP coupler can be executed automatically during build time and considerably reduces the effort necessary to add chemical compounds and/or reactions to existing mechanisms as well as the effort necessary to add new mechanisms using KPP in WRF-Chem. In the following the WRF-Chem/KPP coupler is described "as is". Since the coupler is under consideration for the WRF-Chem repository some details of the design are still likely to change in response to suggestions by the WRF developers and the user community.

*Corresponding author address: Marc Salzmann, Max-Planck-Institute for Chemistry, Department of Atmospheric Chemistry, PO Box 3060, 55020 Mainz, Germany. Email: salzmann@mpch-mainz.mpg.de.

## 2. The WRF-Chem/KPP coupler

At present two mechanisms have been implemented together with the WRF-Chem/KPP coupler in a test setup (described in Section 3); these are RACM-MIM (Geiger et al. 2003), and a simple CO-$CH_4$-$HO_x$-$NO_x$ "background" chemistry. The KPP input files for the mechanisms are expected to reside in subdirectories of a directory named KPP. The KPP directory is expected to be located in the chem directory. These two mechanisms could serve as examples for adding further mechanisms with KPP in the future. Adding a new mechanism involves the modification/provision of a number of ASCII files. The basic steps are:

- editing the Registry.EM_CHEM to
  - add additional species to the chem array.
  - add a package (=a mechanism) with a name ending on "_kpp".
- providing input files (.eqn, .spc, .kpp) for KPP in a sub-directory of chem/KPP named after the package.
- optionally providing a file (wrfkpp.equiv) for mapping variable names in WRF to variable names in KPP (e.g. HO to OH).

If the environment variable WRF_CHEM is set to one, the WRF compile script copies the file Registry.EM_CHEM to the file Registry and invokes the WRF registry mechanism. The WRF registry mechanism parses the registry and writes the part of the WRF code which is responsible for memory allocation, IO, etc. The WRF-Chem/KPP coupler, which can also be invoked from the compile script, automatically compiles and runs a modified version of KPP and generates the code for the interface between WRF-Chem and the KPP generated code (see schematic in Figure 1). The wrfkpp.equiv file can be used in order to avoid renaming of species in existing KPP input files, which are already defined under a different name in the WRF-Chem registry. Note that for implementing new mechanisms, it is often necessary to calculate additional photolysis rates, include new emission datasets, specify initial and boundary conditions, calculate additional dry deposition rates, specify Henry's law coefficients for soluble trace gases and carry them through the wet deposition routines, etc., in

```
#EQUATIONS { racm-mim  }
{001} NO2+hv=O3P+NO  : j(Pj_no2)  ;
{002} O3+hv=O1D{+O2} : j(Pj_o31d) ;
...
{242} MACP+HO2=MAHP : ARR2( 1.82e-13 , -1300.0, TEMP ) ;
{243} MACP+MACP=HACE+MGLY+0.5 HCHO+0.5 CO+HO2  : 2.00e-12 ;
{244} MACP+NO2=MPAN : TROE( 9.70e-29 , 5.6 , 9.30e-12 , 1.5 , TEMP, C_M) ;
...
```

Example File 1: Excerpt from the KPP equation (.eqn) file for the RACM-MIM (Geiger et al. 2003) mechanism.

addition to the steps described above. Furthermore, it is important to note that the programs Yacc, flex, and sed are required for compiling KPP as described in the KPP documentation. The KPP documentation is available at http://people.cs.vt.edu/ asandu/Software/Kpp/docsforkpp.htm.

The coupler is just like KPP and the WRF registry mechanism written in C and is in part based on the registry mechanism. It automatically performs the following tasks:

- Gather information from the WRF-Chem registry:
  - chem array species names (from package lines)
  - names of radicals from radical arrays
  - photolysis rates
- Read KPP species files (from chem/KPP/*/*.spc)
  - read wrfkpp.equiv files
- Perform consistency checks and issue warnings if necessary
- If necessary, compile modified KPP
- Run KPP for files in chem/KPP/* directories (write Makefile(s)) and copy the KPP generated modules to the chem directory
- Modify the Makefile in the chem directory to include KPP generated modules and driver
- Write a mechanism driver for KPP mechanisms (called from chem/mechanism_driver.F)
- Write a solver interface module for each KPP mechanism implemented with KPP

After the compilation, the KPP generated mechanisms are invoked by setting the chemistry option in the WRF namelist to the value which was specified in the package specification in the registry. The WRF-Chem/KPP coupler was designed to require only minimal changes to the existing WRF-Chem code (Figure 2). Thus far, only Rosenbrock type solvers have been adapted for the use in WRF-Chem, but adapting additional solvers from the solvers provided with KPP is relatively straight forward and does not require changes to the KPP code or the WRF-Chem/KPP coupler. A number of KPP features, such as the use of adjoint methods, are currently not supported. The modifications to KPP have been implemented as an additional option which can be turned on by adding the line
`#WRFCONFORM`
to the KPP input file with suffix `.kpp`. Only relatively moderate changes to the KPP source code have been applied in order to adapt KPP for the use with WRF-Chem.

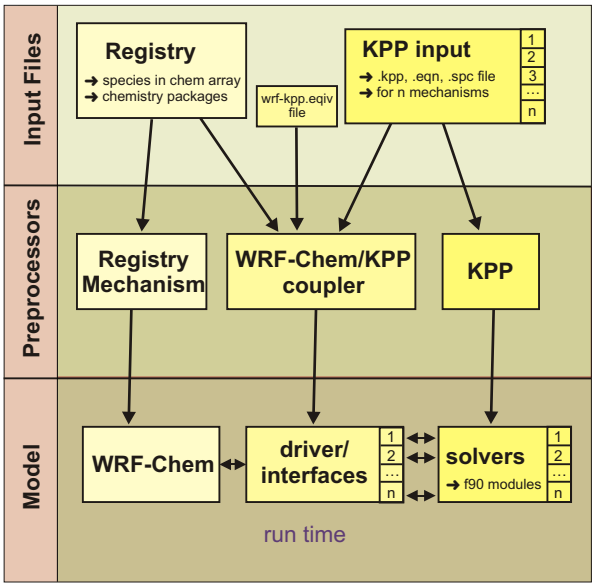


Figure 1: Schematic: Input files (ASCII), "Preprocessors" run during build time (written in C), and WRF-Chem with coupled KPP solvers (written in Fortran 90.)

## 3. Performance

KPP generated solvers have already successfully been used in a number of 3-D models (e.g. von Kuhlmann et al. 2003; Sander et al. 2005). In order to obtain a first

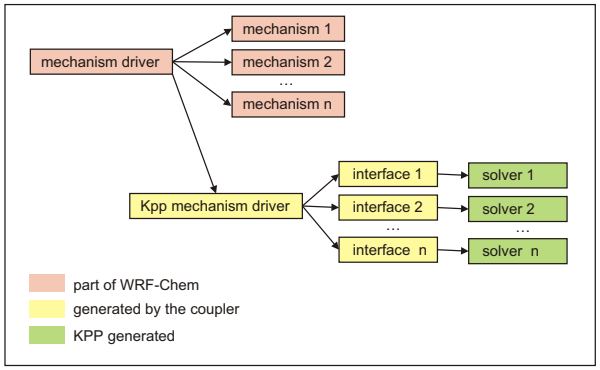


Figure 2: Call tree: The mechanism driver from WRF-Chem calls a separate mechanism driver for the mechanisms implemented with KPP. This setup requires only one call to be added in WRF-Chem and allows switching between mechanisms implemented with KPP and the other mechanisms in WRF-Chem.

Table 1: Wall-clock time $t_w$ for test runs (emissions, deep convection parameterization, aerosols, and dry deposition turned off) performed on a gigabit Ethernet Linux cluster with 3.06GHz dual-Xeon nodes. $\alpha$ is the ratio of run time $t_r$ to wall-clock time consumed $t_w$.

| | $t_w$ | $\alpha=t_r/t_w$ |
|---|---|---|
| 36 h run on 24 processors | | |
| RACM-MIM[1] | 6 h 28 min | 5.57 |
| RADM2[2] | 2 h 19 min | 15.89 |
| 2 h run on 12 processors | | |
| RACM-MIM[1] | 38 min 24 s | 3.13 |
| RADM2[2] | 13 min 1 s | 9.22 |

[1] KPP generated Rosenbrock solver (ros3) with 245 reactions and 80 predicted species.

[2] quasi steady state approximation method with 157 reactions, 38 predicted, and 22 diagnosed species as implemented in WRF-Chem.

impression of the performance of the KPP-generated code in WRF-Chem, runs using an initial test setup in which emissions, deep convection, and dry deposition have been switched off were conducted for the KPP generated and automatically coupled RACM-MIM mechanism. Some of the initial and boundary values for RACM-MIM were set to preliminary values in these test runs. Appropriate initial and boundary values for RACM-MIM may soon become available within the framework of efforts which are currently underway at the University of Chile. These efforts aim at implementing new boundary and initial values from global models into WRF-Chem (Rainer Schmitz, pers. comm., 2006). In order to avoid redundant work, we decided to use preliminary initial and boundary conditions for testing. The timestep in the test runs was 90s and the gridsize 134 x 110 x 35 points at 27 km horizontal resolution as in Grell et al. (2005). Table 1 shows the elapsed "wall-clock" time for the RACM-MIM runs. Furthermore, the wall-clock times needed for otherwise identical test runs using the hard-coded RADM2 mechanism implemented in WRF-Chem are shown. Since the number of predicted species is much higher in RACM-MIM, these numbers should not be compared directly, but can only serve to give a rough impression.

## 4. Availability and Future Perspectives

The next next step is to evaluate which changes may still be necessary in order for the WRF-Chem/KPP coupler to become part of the WRF-Chem repository. Furthermore, a more detailed documentation of the coupler is under construction. The coupler may prove particularly useful for sensitivity studies using different chemistry mechanisms or using additional reactions in existing KPP implemented mechanisms. In the future, additional preprocessors, e.g. for handling the calculation of Henry's law coefficients for soluble trace gases could be added to the framework of the WRF/Chem-KPP coupler and the coupler could be extended to support additional KPP features such as the usage of additional solvers. Coupling KPP to WRF-Chem is only one step towards generalizing WRF-Chem. For other processes (e.g. dry deposition), generalization is still an open task.

## Acknowledgments

We thank Rainer Schmitz, University of Chile, for valuable comments and for making the equation file for RACM-MIM available. Rolf Sander and Rolf von Kuhlmann are gratefully acknowledged for willingly sharing their expertise on KPP.

## References

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

Geiger, H., I. Barnes, T. Benter, and M. Spitteler, 2003: The tropospheric degradation of isoprene: An updated module for the Regional Atmospheric Chemistry Mechanism. *Atmos. Environ.*, **37**, 1503–1519.

Grell, G. A., S. E. Peckham, S. A. McKeen, J. Wilczak, and B. Eder, 2005: Fully coupled "online" chemistry within the WRF model. *Atmos. Environ.*, **39**, 6957–6975.

Sander, R., A. Kerkweg, P. Jöckel, and J. Lelieveld, 2005: Technical note: The new comprehensive atmospheric

chemistry module MECCA. *Atmos. Chem. Phys.*, **5**, 445–450.

Sandu, A., D. Daescu, and G. R. Carmichael, 2003: Direct and adjoint sensitivity analysis of chemical kinetic systems with KPP: I – Theory and software tools. *Atmos. Environ.*, **37**, 5083–5096.

Sandu, A. and R. Sander, 2006: Technical note: Simulating chemical systems in fortran90 and matlab with the kinetic preprocessor KPP-2.1. *Atmos. Chem. Phys.*, **6**, 187–195.

von Kuhlmann, R., M. G. Lawrence, P. J. Crutzen, and P. J. Rasch, 2003: A model for studies of tropospheric ozone and nonmethane hydrocarbons: Model description and ozone results. *J. Geophys. Res.*, **108**, doi: 10.1029/2002JD002893.