WRF-Chem/KPP Coupler (WKC)

Version 1.0.0 User's and Developer's Guide

Marc Salzmann
salzmann@mpch-mainz.mpg.de

September 27, 2006

Contents

1 Preface

WKC is an acronym for the **W**RF-Chem/**K**PP **C**oupler. For an introduction to WKC see the extended abstract to Presentation (6.4) by Salzmann and Lawrence, 2006 WRF-User Workshop (`chem/KPP/documentation/abstr_wkc.pdf`). Documentation for **KPP** (**K**inetic **P**re**P**rocessor) is e.g. provided at Adrian Sandu's homepage:

http://people.cs.vt.edu/ asandu/Software/Kpp/.

References for KPP are among others Damian *et al.* (2002); Sandu *et al.* (2003); Sandu and Sander (2006) (please give credit when using KPP generated code). KPP and WKC are distributed under the GNU General Public License (GPL). **At this stage, Version 1.0.0 of WKC and its documentation are still a β releases and are recommended for friendly users/developers only.** Constructive comments and suggestions regarding the coupler and/or this documentation are welcome. Only a limited number of all KPP features are available for use with WKC, but more features may be added in the future.

2 Requirements for running KPP

KPP requires `flex`, `yacc`, and `sed` to be installed. The path to the `flex` library (either `libfl.a` or `libfl.sh`) is specified by the environment variable `FLEX_LIB_DIR`. The default path is `/usr/lib`. If `libfl.a` (or `libfl.sh`) is not located in `/usr/lib` on your system `FLEX_LIB_DIR` should be set prior to compiling WRF-Chem. The C compiler is set by `configure_kpp` based on the settings in `configure.wrf`.

3 Compiling and Running KPP together with WKC

WKC and KPP are compiled and executed automatically when WRF-Chem is compiled. WKC copies the KPP generated code to the `chem` directory and automatically modifies the `Makefile` in the `chem` directory, so that the KPP generated code is compiled into WRF-Chem. The KPP and WKC generated modules in the `chem` directory contain the string `_kpp_` in their file names. Running the `clean` script removes these modules.

4 Using mechanisms already implemented with WKC

KPP files for mechanisms which have already been implemented with WKC are located in subdirectories of `chem/KPP/mechanisms`. The corresponding packages are declared in the

file `Registry/Registry.EM_CHEM` and contain the suffix "`_kpp`" in their name. In order to use one of these mechanisms inside WRF-Chem, set the `chem_opt` in the `namelist.input` to the corresponding value (find out the number by browsing the `Registry.EM_CHEM`).
The following mechanisms are currently available (thanks to Rainer Schmitz, University of Chile, who made the KPP input files for these mechanisms available):

- RACM, Stockwell *et al.* (1997).
- RACM-MIM, Geiger *et al.* (2003).

These WKC implemented mechanisms have `chem_opt` $>$ `100`. See Section 8 on how to add additional mechanisms.

5 Layout of WKC

WKC reads KPP species input files with suffix `.spc` and the file `Registry/Registry.EM_CHEM` and automatically generates the Fortran 90 interface routines between WRF-Chem and the KPP generated code (see Fig. 1). It is in parts based on the WRF registry mechanism. The WKC related files are located in the `chem/KPP` directory. This directory contains:

- a subdirectory `mechanisms` which holds directories with KPP input files for different mechanisms.
- a compile and a clean script for WKC (which are executed from the WRF-Chem compile script).
- a version of KPP v2.1 in the `kpp` subdirectory. This version of KPP was adapted to produce code which can directly be used with WRF-Chem (using the `#WRF_Conform` option in the .kpp file).
- the source code of WKC in the `util/wkc` subdirectory.
- `module_wkpp_constants.F` which allows to specify input to kpp such as RTOL and ATOL (likely to be extended in the future).
- a subdirectory `inc` containing files which are included during compile time (using "`#include`" statements). The files in `chem/KPP/inc`are not removed by the WKC clean script. Their purpose is to allow user modifications to WKC generated code.

At the heart of WKC is the routine `gen_kpp.c`, which is located in the `util/wkc` directory.

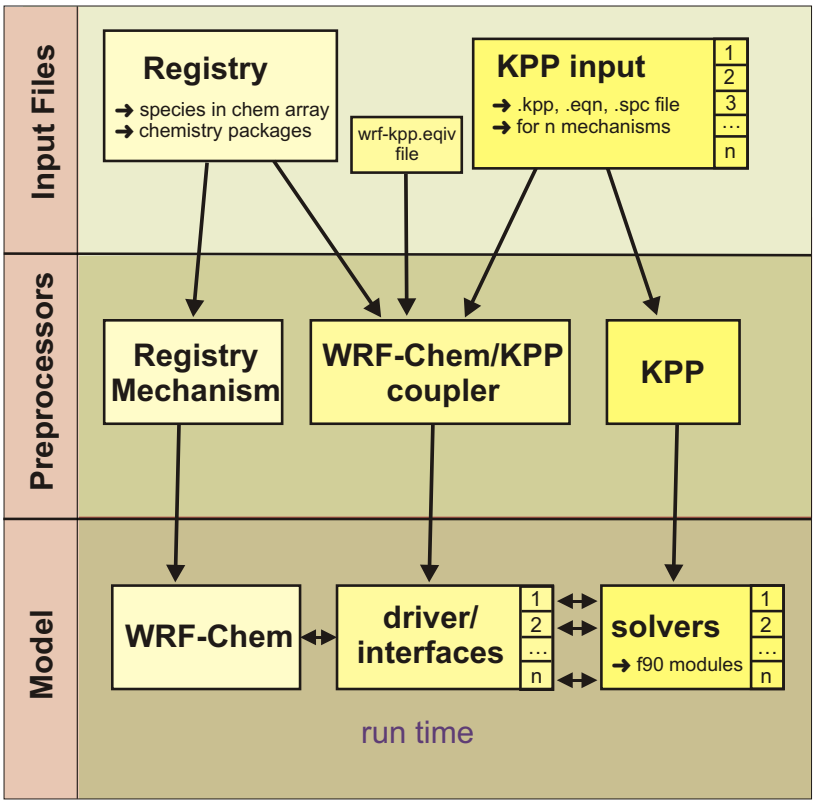

Figure 1: Schematic: Input files (ASCII), "Preprocessors" run during build time (written in C), and WRF-Chem with coupled KPP solvers (written in Fortran 90.)

6 Code produced by WKC, User Modifications

The code produced by WKC is called from the chem_driver (see schematic call tree in Fig. 2). Since parts of the code are generated automatically, manual changes will be lost when re-compiling WRF-Chem (as indicated by a warning in the header of the corresponding files). There are, however, a number of "`#INCLUDE`" preprocessor statements in the WKC generated code. The files included (in the `.f` files) are located in the chem/KPP/inc directory. These files are not removed by the clean script and can be used to inline user supplied code. In case this should not be enough, there are two ways to edit automatically generated files permanently: The files can either be renamed in such a way that they won't be removed by the `clean_kpp` script, or the C code which generated the files (either KPP or WKC) can be

edited. The latter is generally the better solution. However, the method of using include files in the `chem/KPP/inc` directory is strongly recommended.

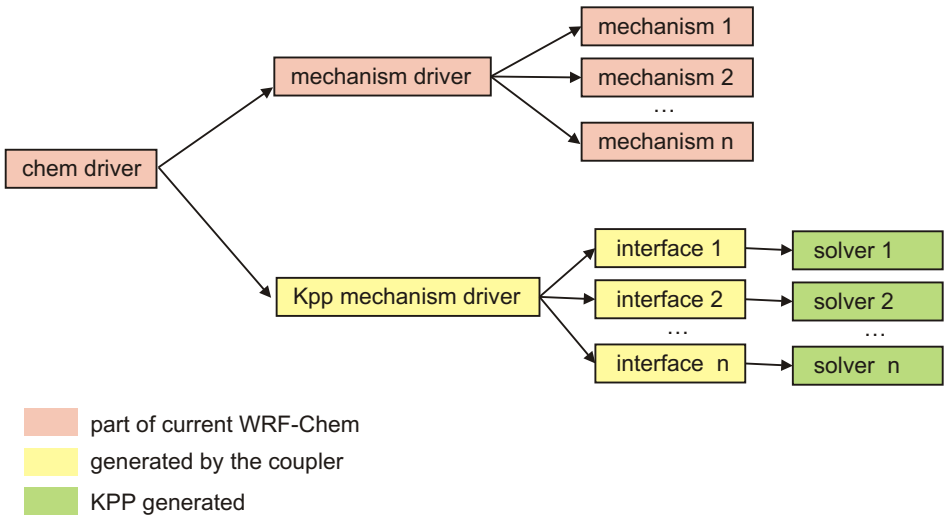

Figure 2: Call tree: The mechanism driver from WRF-Chem calls a separate mechanism driver for the mechanisms implemented with KPP. This setup requires only one call to be added in WRF-Chem and allows switching between mechanisms implemented with KPP and the other mechanisms in WRF-Chem.

7 Available integrators

References for the chosen integrator can e.g. after compiling WRF-Chem with KPP be found in the `chem` directory in `module_kpp_my_mechanism_Integrator.f90`, where ‘‘`my_mechanism`’’ refers to the mechanism chosen in the WRF-Chem namelist. Currently, only Rosenbrock type integrators are available for the use with WKC. See Section 9 on how to add additional integrators.

8 Adding additional mechanisms with WKC

The following basic steps are necessary in order to add an additional mechanism:

- edit the `Registry.EM_CHEM` to
  - add additional species to the `chem` array (only if necessary!).
  - add a package (=a mechanism) with a name ending on "`_kpp`", e.g. `my_mechanism_kpp`.
- provide input files `my_mechanism.eqn`, `my_mechanism.spc`, `my_mechanism.kpp` for KPP in a sub-directory of `chem/KPP/mechanisms` named after the package (i.e. `my_mechanism`, not `my_mechanism_kpp`).
- optionally provide a file (`my_mechanism_wrfkpp.equiv`) for mapping variable names in WRF to variable names in KPP (e.g. HO to OH).

For examples, see mechanisms which have already been implemented. When copying one of the directories in `chem/KPP/mechanisms` to another directory it is necessary to change the name of `#Model` in the `.kpp` file and the names of the `.eqn` and the `.spc` file in the `.def` file. When introducing a "new" `.kpp` file set the `#INTEGRATOR` to an integrator contained in the directory `chem/KPP/kpp/kpp-2.1/int/WRF_conform`, e.g.

```
#INTEGRATOR WRF_conform/rosenbrock
```

and add

```
#WRFCONFORM
```

to the `.kpp` file. Note, that not all KPP options are supported by WKC. Furthermore, WKC is currently not able to handle comments in the `.spc` file!

8.1 Adding a package to the Registry

Is pretty straight forward.

8.2 Adapting KPP equation files and using the wrf_kpp_equiv file for mapping species names

Adapting a KPP equation file for the use with WRF involves renaming a few variables in the equation file:

	KPP `.eqn` file	unit in `.eqn` file	Registry
Photolysis rate (e.g.)	j(Pj_no2))	s^{-1}	ph_no2
temperature	TEMP	K	t_phy
third body conc.	C_M	(molec moist air)/cm^3	calculated from rho
water vapor conc.	C_H2O	molec/cm^3	calc. from QVAPOR

```
#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} MSACP+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.

Photolysis rates, temperatures, third body concentrations, and water vapor concentrations are passed down from the WRF-Chem/KPP interface routines. Photolysis rates are stored pointwise in a 1-D array and addressed by pointers defined in the automatically generated interface routine. For example, the NO_2 photolysis rate `ph_no2` in the Registry.EM_CHEM becomes `j(Pj_no2)` in the KPP equation file (see example in Example File 1). Additional variables (e.g. user calculated N_2O_5 hydrolysis rates) can be passed down by modifying `.inc` files in the `chem/KPP/inc` directory.

9 Adapting additional KPP integrators for WKC

Currently, only Rosenbrock type solvers have been adapted for the use with the WKC. Adapting additional integrators which come with KPP is, however, rather straight forward (but can nevertheless be time consuming). The integrator files which come with KPP are located in the directory `chem/KPP/kpp/kpp2.1/int`. Integrators which have been adapted for WRF-Chem are located in a subdirectory of `chem/KPP/kpp/kpp2.1/int` named `WRF_Conform`. In order to adapt an additional solver for WRF-Chem:

- Copy the `.f90` and the `.def` file to the `WRF_Conform` directory.
- Add `KPP_ROOT` as a prefix to the names of the subroutines in all subroutine and end subroutine statements.
- Change the arguments in the `SUBROUTINE (KPP_ROOT_)INTEGRATE` statements to match the calling routine (for an example, see the existing integrator routines).

- Remove all the `USE` statements in which non-constant data is used. Instead pass down the data in the subroutine statements. This can be time consuming.

Depending on the solver, additional steps may be necessary.

10 Installing WKC into an existing WRF-Chem version

If you received this documentation together with WRF-Chem, WKC is already installed, and you can skip this section. Otherwise:

- Untar the `chem/KPP` directory into your WRF-Chem.
- Update your Registry (see `Registry/Registry.EM_CHEM` for an example).
- Add the following lines to the beginning of the WRF compile script (after where configure.wrf is written):

```
setenv WRF_KPP 1
if ( ! $?WRF_KPP )   setenv WRF_KPP  0
if ( $WRF_KPP == 1 ) then
chem/KPP/compile_wkc
endif
```

- Add the following lines to the end of the WRF clean script:

```
if ( -e chem/KPP )then
  ( cd chem/KPP; clean_kpp )
endif
```

- Add a call to `chem_driver.F`:

```
  CALL kpp_mechanism_driver(   chem,                          &
    grid%id,dtstepc,config_flags,                            &
    p_phy,t_phy,rho,moist,                                   &
    vdrog3, ldrog,                                           &
!
#include <call_to_kpp_mech_drive.inc>
```

```
!
      ids,ide, jds,jde, kds,kde,                            &
      ims,ime, jms,jme, kms,kme,                            &
      grid%i_start(ij), min(grid%i_end(ij),ide-1),      &
      grid%j_start(ij), min(grid%j_end(ij),jde-1),      &
      k_start    , min(k_end,kde-ksubt)                     )
```

11 Final Remarks

If you find it necessary to make changes to the coupler, or you have implemented a new mechanism, I would appreciate your feedback (salzmann@mpch-mainz.mpg.de).

Bibliography

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

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

Sandu, A., and Sander, R., 2006: Technical note: Simulating chemical systems in Fortran90 and Matlab with the Kinetic PreProcessor KPP-2.1. 6, 187–195.

Sandu, A., Daescu, D., and Carmichael, G. R., 2003: Direct and Adjoint Sensitivity Analysis of Chemical Kinetic Systems with KPP: I – Theory and Software Tools. 37, 5083–5096.

Stockwell, W. R., Kirchner, F., Kuhn, M., and Seefeld, S., 1997: A new mechanism for regional atmospheric chemistry modeling. 102, 25847–25879.