tutorial presentation

Using KPP generated chemistry solvers in WRF-Chem

Marc Salzmann

Max Planck Institute for Chemistry

Mainz, Germany

salzmann@mpch-mainz.mpg.de

overview

- introduction to KPP
- WRF-Chem/KPP coupler
  - “guided tour”
  - for cookbook style recipes see documentation

www.mpch-mainz.mpg.de/~salzmann/my_home/sub/wkc.html

KPP

KPP

article | discussion | edit this page | history

Your continued donations keep Wikipedia running!

KPP

From Wikipedia, the free encyclopedia

KPP may refer to:

- The Communist Party of Poland
- Kernel Patch Protection, a security feature of Microsoft Windows

This disambiguation page lists articles associated with the same title. If an internal link led you here, you may wish to change the link to point directly to the intended article.

Category: Disambiguation

KPP

- **KPP: Kinetic PreProcessor**
  - **reads ASCII input and generates code for chemistry integration**

- **e.g.:**
  - **Damian et al., 2002**
  - **Sandu et al., 2003**
  - **Sandu and Sander, 2006**

- **http://people.cs.vt.edu/~asandu/Software/Kpp/**

KPP example

$$O_2 \xrightarrow{h\nu} 2O \quad \text{(R1)}$$

$$O + O_2 \rightarrow O_3 \quad \text{(R2)}$$

$$O_3 \xrightarrow{h\nu} O + O_2 \quad \text{(R3)}$$

$$O + O_3 \rightarrow 2O_2 \quad \text{(R4)}$$

$$O_3 \xrightarrow{h\nu} O(^1D) + O_2 \quad \text{(R5)}$$

$$O(^1D) + M \rightarrow O + M \quad \text{(R6)}$$

$$O(^1D) + O_3 \rightarrow 2O_2 \quad \text{(R7)}$$

$$NO + O_3 \rightarrow NO_2 + O_2 \quad \text{(R8)}$$

$$NO_2 + O \rightarrow NO + O_2 \quad \text{(R9)}$$

$$NO_2 \xrightarrow{h\nu} NO + O \quad \text{(R10)}$$

Small "stratospheric" mechanism,
example from Sandu, 2003; Sandu and Sander, 2006

KPP example

File #1: ***species (.spc file)***

```
#INCLUDE atoms
#DEFVAR
  O    = O;
  O1D  = O;
  O3   = O + O + O;
  NO   = N + O;
  NO2  = N + O + O;

 #DEFFIX
   M    = ignore;
   O2   = O + O;
```

KPP example

File #2: ***equation (.eqn file)***

```
#EQUATIONS { Stratospheric Mechanism }
<R1>  O2  + hv = 2O        : 2.6e-10*SUN;
<R2>  O   + O2 = O3        : 8.0e-17;
<R3>  O3  + hv = O   + O2  : 6.1e-04*SUN;
<R4>  O   + O3 = 2O2       : 1.5e-15;
<R5>  O3  + hv = O1D + O2  : 1.0e-03*SUN;
<R6>  O1D + M  = O   + M   : 7.1e-11;
<R7>  O1D + O3 = 2O2       : 1.2e-10;
<R8>  NO  + O3 = NO2 + O2  : 6.0e-15;
<R9>  NO2 + O  = NO  + O2  : 1.0e-11;
<R10> NO2 + hv = NO  + O   : 1.2e-02*SUN;
```

KPP example

File #3: *.kpp file*

```
#MODEL       small_strato
#LANGUAGE    Fortran90
#DOUBLE      ON
#INTEGRATOR  rosenbrock
#DRIVER      general
#JACOBIAN    SPARSE_LU_ROW
#HESSIAN     ON
#STOICMAT    ON
```

KPP example

File #4: *.def file (optional)*

```
#include atoms_red
#include ./racm_mim.spc
#include ./racm_mim.eqn

#INLINE F90_RATES
REAL(KIND=dp) FUNCTION k46( TEMP, C_M )
    REAL(KIND=dp), INTENT(IN) :: temp, c_m
    REAL(KIND=dp) :: k0, k2, k3

   k0=7.2E-15_dp * EXP(785._dp/TEMP)
   k2=4.1E-16_dp * EXP(1440._dp/TEMP)
   k3=1.9E-33_dp * EXP(725._dp/TEMP)

   k46=k0+k3/(1+k3/k2)

END FUNCTION k46
#ENDINLINE
```

in addition: User defined functions in $KPP_HOME/util/UserRateLaws.f90

OR: $KPP_HOME/util/**WRF_Conform**/UserRateLaws.f90

KPP example

- **Output from KPP: “box model” with a driver**
  - **in Fortran 90**
  - **alternatively: Matlab**

For details: e.g. Sandu and Sander, 2006

KPP

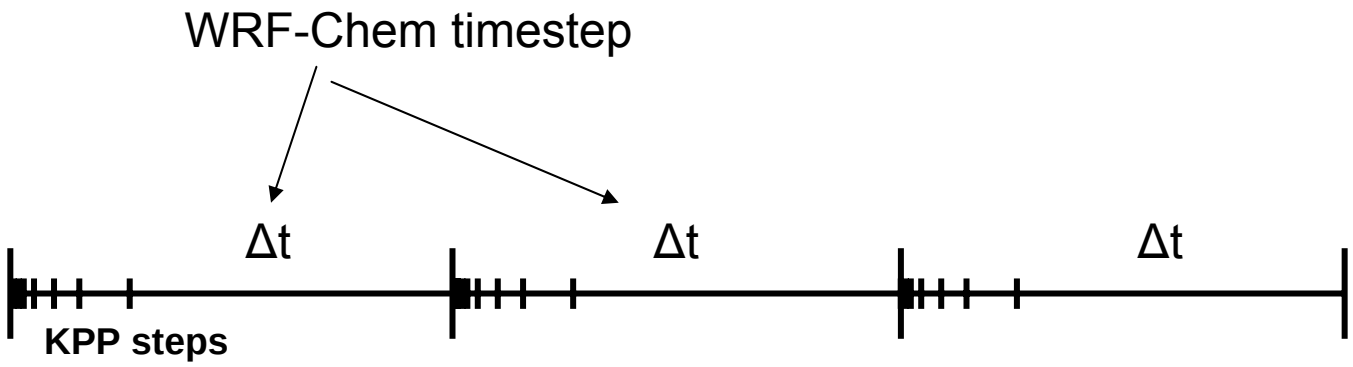
- widely used
  - box model studies
  - chemistry integration in 3-D models (for example in GEOS-Chem, E5/M1, WRF-Chem)
  - adjoints in 3-D models (for example Sandu et al., 2005, Henze et al., 2007)
- freely available
- distributed together with a few examples

KPP benchmarks

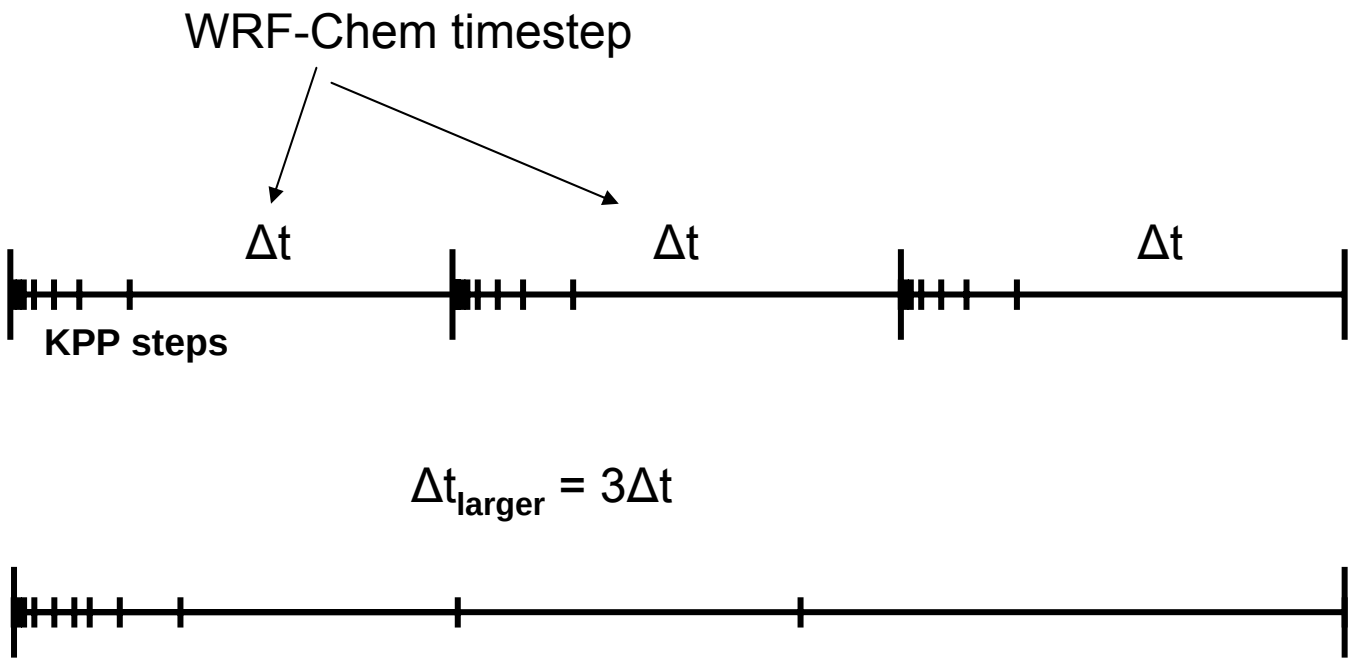
Integrator	stratospheric mechanism CPU time [s]	MECCA CPU time [s]
rodas3	0.42	38.71
kpp_lsode	0.32	39.79
ros3	0.38	41.33
rodas4	0.46	49.92
ros4	0.43	51.09
kpp_seulex	0.50	55.31
kpp_sdirk	0.86	63.24
ros2	0.39	69.43
kpp_radau5	0.49	103.33
ros2_manual	0.08	—

from: Sandu and Sander, 2006

ros3



ros3



WRF-Chem/KPP coupler

WRF-Chem/KPP coupler (WKC)

- writes .f90 interfaces between WRF-Chem and KPP generated code
- “non-destructive layout”, i.e. minimum modifications to WRF-Chem, almost everything in one directory
- script to automatically compile/run KPP during WRF compile time

WRF-Chem/KPP coupler (WKC)

- a few modifications to KPP (switch)
  - not all features available
  - currently only Rosenbrock type solvers
- written in C and in parts based on the Registry mechanism

WRF-Chem Registry

Registry/registry.chem (included in Registry.EM_CHEM)

Excerpt: chemistry array

```
# Chem Scalars
state   real    -    ikjftb   chem  1  -     -    -
state   real   so2   ikjftb   chem  1  -   irhusdf=(bdy_interp:dt)  "so2"   "SO2"   "ppmv"
state   real   sulf  ikjftb   chem  1  -   irhusdf=(bdy_interp:dt)  "sulf"  "SULF"  "ppmv"
state   real   no2   ikjftb   chem  1  -   irhusdf=(bdy_interp:dt)  "no2"   "NO2"   "ppmv"
state   real   no    ikjftb   chem  1  -   irhusdf=(bdy_interp:dt)  "no"    "NO"    "ppmv"
state   real   o3    ikjftb   chem  1  -   irhusdf=(bdy_interp:dt)  "o3"    "O3"    "ppmv"
state   real   hno3  ikjftb   chem  1  -   irhusdf=(bdy_interp:dt)  "hno3"  "HNO3"  "ppmv"
state   real   h2o2  ikjftb   chem  1  -   irhusdf=(bdy_interp:dt)  "h2o2"  "H2O2"  "ppmv"
...
```

+ arrays for radicals

+ arrays for photolysis rates

WRF-Chem Registry

+ package declaration for each mechanism

```
# CHEMISTRY PACKAGE DEFINITIONS
#
package   prescribe_aerosol     chem_opt==0          -              -
package   radm2          chem_opt==1                  -              chem:so2,sulf,
no2,no,o3,hno3,h2o2,ald,hcho,op1,op2,paa,ora1,ora2,nh3,n2o5,no3,pan,hc3,hc5,hc8,e
th,co,ol2,olt,oli,tol,xyl,aco3,tpan,hono,hno4,ket,gly,mgly,dcb,onit,csl,iso,ho,ho
2
package   radm2sorg      chem_opt==2                  -              chem:so2,sulf,
no2,no,o3,hno3,h2o2,ald,hcho,op1,op2,paa,ora1,ora2,nh3,n2o5,no3,pan,hc3,hc5,hc8,e
th,co,ol2,olt,oli,tol,xyl,aco3,tpan,hono,hno4,ket,gly,mgly,dcb,onit,csl,iso,ho,ho
2,so4aj,so4ai,nh4aj,nh4ai,no3aj,no3ai,orgaro1j,orgaro1i,orgaro2j,orgaro2i,orgalk1
j,orgalk1i,orgole1j,orgole1i,orgba1j,orgba1i,orgba2j,orgba2i,orgba3j,orgba3i,orgb
a4j,orgba4i,orgpaj,orgpai,ecj,eci,p25j,p25i,antha,seas,soila,nu0,ac0,corn
package   racm           chem_opt==3                  -              chem:so2,sulf,
no2,no,o3,hno3,h2o2,ald,hcho,op1,op2,paa,ora1,ora2,nh3,n2o5,no3,pan,hc3,hc5,hc8,e
th,co,ete,olt,oli,tol,xyl,aco3,tpan,hono,hno4,ket,gly,mgly,dcb,onit,csl,iso,co2,c
h4,udd,hket,api,lim,dien,macr,ho,ho2
package   racmsorg       chem_opt==4                  -              chem:so2,sulf,
no2,no,o3,hno3,h2o2,ald,hcho,op1,op2,paa,ora1,ora2,nh3,n2o5,no3,pan,hc3,hc5,hc8,e
th,co,ete,olt,oli,tol,xyl,aco3,tpan,hono,hno4,ket,gly,mgly,dcb,onit,csl,iso,co2,c
h4,udd,hket,api,lim,dien,macr,ho,ho2,so4aj,so4ai,nh4aj,nh4ai,no3aj,no3ai,orgaro1j
,orgaro1i,orgaro2j,orgaro2i,orgalk1j,orgalk1i,orgole1j,orgole1i,orgba1j,orgba1i,o
rgba2j,orgba2i,orgba3j,orgba3i,orgba4j,orgba4i,orgpaj,orgpai,ecj,eci,p25j,p25i,an
tha,seas,soila,nu0,ac0,corn
package   cbmz           chem_opt==5                  -              chem:so2,sulf,
no2,no,o3,hno3,h2o2,ald,hcho,op1,op2,paa,ora1,ora2,nh3,n2o5,no3,pan,hc3,hc5,hc8,e
th,co,ol2,olt,oli,tol,xyl,aco3,tpan,hono,hno4,ket,gly,mgly,dcb,onit,csl,iso,ho,ho
2,hcl,ch3o2,ethp,ch3oh,c2h5oh,par,to2,cro,open,op3,c2o3,ro2,ano2,nap,xo2,xpar,iso
prd,isopp,isopn,isopo2,dms,msa,dmso,dmso2,ch3so2h,ch3sch2oo,ch3so2,ch3so3,ch3so2o
o,ch3so2ch2oo,mtf
...
```

WRF-Chem Registry

+ package declaration for **KPP** mechanisms

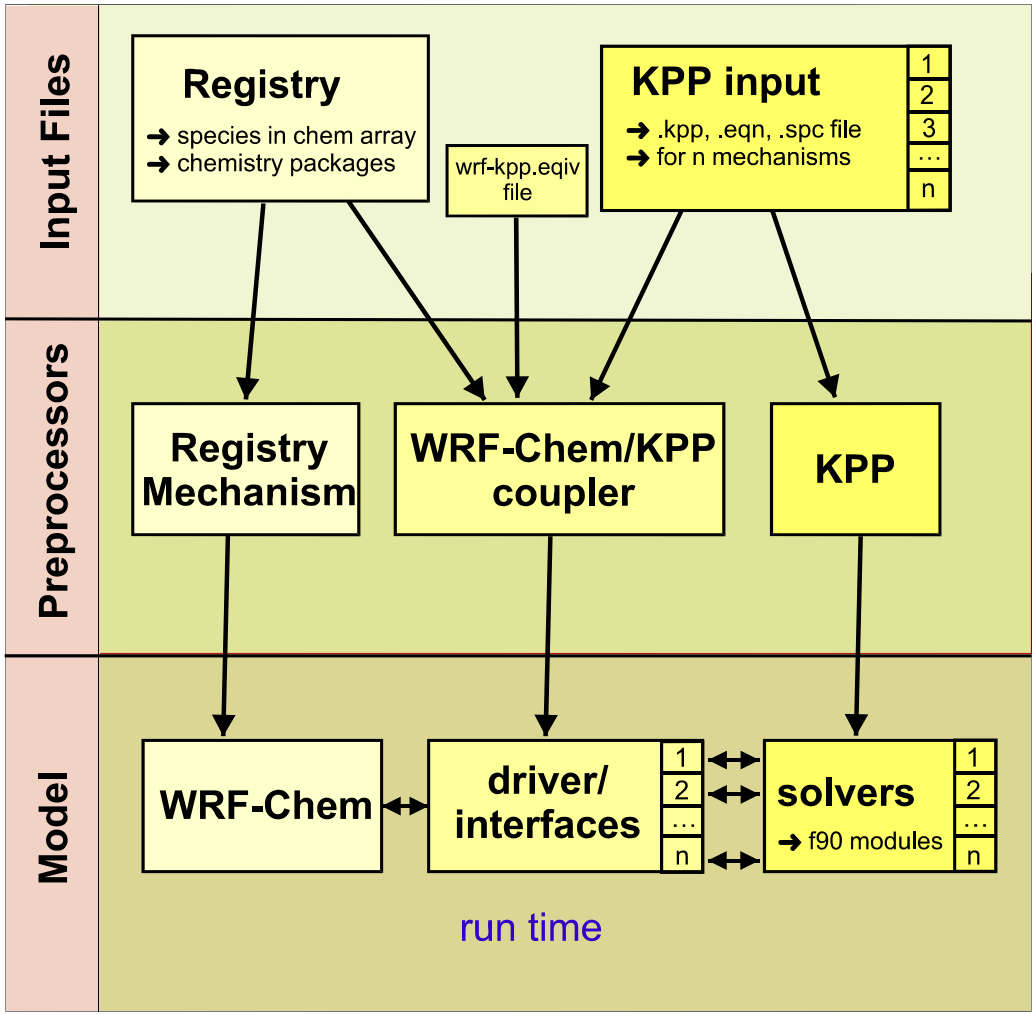
```
...
#cms++
package   radm2_kpp        chem_opt==101                    -              chem:so2,s
ulf,no2,no,o3,hno3,h2o2,ald,hcho,op1,op2,paa,ora1,ora2,nh3,n2o5,no3,pan,hc3,hc5,h
c8,eth,co,ol2,olt,oli,tol,xyl,aco3,tpan,hono,hno4,ket,gly,mgly,dcb,onit,csl,iso,c
o2,ch4,ho,ho2

package   racm_mim_kpp     chem_opt==102                    -              chem:so2,
sulf,no2,no,o3,hno3,h2o2,ald,hcho,op1,op2,paa,ora1,ora2,n2o5,no3,pan,hc3,hc5,hc8,
eth,co,ete,olt,oli,tol,xyl,aco3,tpan,hono,hno4,ket,gly,mgly,dcb,onit,csl,iso,co2,
ch4,udd,hket,api,lim,dien,macr,hace,ishp,ison,mahp,mpan,nald,ho,ho2

package   racm_kpp     chem_opt==103                    -              chem:so2,sulf,n
o2,no,o3,hno3,h2o2,ald,hcho,op1,op2,paa,ora1,ora2,nh3,n2o5,no3,pan,hc3,hc5,hc8,et
h,co,ete,olt,oli,tol,xyl,aco3,tpan,hono,hno4,ket,gly,mgly,dcb,onit,csl,iso,co2,ch
4,udd,hket,api,lim,dien,macr,ho,ho2

package   racmsorg_kpp     chem_opt==104                    -              chem:so2,
sulf,no2,no,o3,hno3,h2o2,ald,hcho,op1,op2,paa,ora1,ora2,nh3,n2o5,no3,pan,hc3,hc5,
hc8,eth,co,ete,olt,oli,tol,xyl,aco3,tpan,hono,hno4,ket,gly,mgly,dcb,onit,csl,iso,
co2,ch4,udd,hket,api,lim,dien,macr,ho,ho2,so4aj,so4ai,nh4aj,nh4ai,no3aj,no3ai,org
aro1j,orgaro1i,orgaro2j,orgaro2i,orgalk1j,orgalk1i,orgole1j,orgole1i,orgba1j,orgb
a1i,orgba2j,orgba2i,orgba3j,orgba3i,orgba4j,orgba4i,orgpaj,orgpai,ecj,eci,p25j,p2
5i,antha,seas,soila,nu0,ac0,corn
#cms--
```

WRF-Chem/KPP


Input Files
Registry
➜ species in chem array
➜ chemistry packages
wrf-kpp.eqiv file
KPP input
➜ .kpp, .eqn, .spc file
➜ for n mechanisms
1
2
3
…
n
Preprocessors
Registry Mechanism
WRF-Chem/KPP coupler
KPP
Model
WRF-Chem
driver/ interfaces
1
2
…
n
solvers
➜ f90 modules
1
2
…
n
run time

wrf-kpp.equiv file

```
! use this file for species that have different
! names in WRF and KPP
!
! currently case sensitive !
!
! left column      right column
! name in WRF      name in KPP
CH3OOH             MeOOH
ho                 OH
```

KPP

- **Advantages**
  - **significantly less time consuming than manual coding**
  - **less error prone**
  - **numerically efficient**

    - **results in greater flexibility**
      - **updating mechanisms by additional equations**
      - **sensitivity studies**

Example: modified KPP file

```
#MODEL racm_mim
#LANGUAGE Fortran90
#DOUBLE ON
#INTEGRATOR WRF_conform/rosenbrock
#DRIVER general
#JACOBIAN SPARSE_LU_ROW
#HESSIAN OFF
#STOICMAT OFF
#WRFCONFORM
```

Example: modified KPP equation file

```
#EQUATIONS {racm-mim}
 {001:J01} NO2+hv=O3P+NO    : j(Pj_no2) ;
 {002:J02} O3+hv=O1D{+O2}    : j(Pj_o31d) ;
 {003:J03} O3+hv=O3P{+O2}    : j(Pj_o33p) ;
 {004:J04} HONO+hv=HO+NO   : j(Pj_hno2) ;
 {005:J05} HNO3+hv=HO+NO2   : j(Pj_hno3) ;
 {006:J06} HNO4+hv=0.65 HO2+0.65 NO2+0.35 HO+0.35 NO3 : j(Pj_hno4) ;
 {007:J07} NO3+hv=NO{+O2}    : j(Pj_no3o2) ;
 {008:J08} NO3+hv=NO2+O3P   : j(Pj_no3o) ;
 {009:J09} H2O2+hv=HO+HO   : j(Pj_h2o2) ;
...
 {236:213} XO2+NO=NO2     : 4.00e-12 ;
 {237:214} XO2+NO3=NO2     : 1.20e-12 ;
 {238:215} ISOP+ISOP=2. MACR+HCHO+HO2   : 2.00e-12 ;
 {239:216} ISHP+HO=MACR+HO : 1.00e-10 ;
 {240:217} ISON+HO=HACE+NALD : 1.30e-11 ;
 {241:218} MACP+NO=NO2+0.25 HACE+0.25 CO+0.25 ACO3+0.5 MGLY+0.75 HCHO+0.75 HO2:
     ARR2( 2.54e-12, -360.0, TEMP ) ;
 {242:219} MACP+HO2=MAHP : ARR2( 1.82e-13 , -1300.0, TEMP ) ;
 {243:220} MACP+MACP=HACE+MGLY+0.5 HCHO+0.5 CO+HO2 : 2.00e-12 ;
 {244:221} MACP+NO2=MPAN : TROE( 9.70e-29 , 5.6 , 9.30e-12 , 1.5 , TEMP, C_M) ;
 {245:222} MPAN=MACP+NO2 : TROEE(1.11e28,14000.0,9.70e-29,5.6,9.30e-12,1.5,TEMP,C_M);
```

pre-defined variables in WRF-Chem/KPP coupler:

	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

(from WKC Users' and Developers' Guide)

chem/KPP directory

http://www.school-clip-art.com/school_bus_clipart.shtml

chem/KPP directory

```
> cd chem/KPP
> ls

clean_kpp
compile_wkc
configure_kpp
documentation
inc
kpp
mechanisms
module_wkppc_constants.F
util
```

mechanisms directory

```
> cd mechanisms
> ls

racm
racm_mim
racmsorg
radm2
```

HERE you can add your own mechanism

racm_mim directory

```
> cd racm_mim
> ls

racm_mim.def
racm_mim.eqn
racm_mim.kpp
racm_mim.spc
racm_mim_wrfkpp.equiv
```

chem/KPP directory

```
> cd ../..
> ls

clean_kpp
compile_wkc
configure_kpp
documentation
inc
kpp
mechanisms
module_wkppc_constants.F
util
```

kpp/kpp-2.1 directory

```
> cd kpp/kpp-2.1
> ls

…
compile_kpp
doc
examples
int
models
src.org
util
drv
gpl
src
test
…
```

kpp/kpp-2.1 directory

```
> cd kpp/kpp-2.1
> ls

…
compile_kpp
doc
examples
int
models
src.org
util
drv
gpl
src
test
…
```

```
WRF_conform/
        rosenbrock.f90
```

chem/KPP directory

```
> cd ../..
> ls

clean_kpp
compile_wkc
configure_kpp
documentation
inc
kpp
mechanisms
module_wkppc_constants.F
util
```

util directory

```
> cd util
> ls

Makefile
wkc
write_decomp (NEW!)
```

util directory

```
> cd util
> ls

Makefile
wkc
write_decomp (NEW!)
```

`gen_kpp.c`:

- use info from Registry and from KPP input files
- write interface

util directory

```
> cd util
> ls

Makefile
wkc
write_decomp
```

direct addressing in KPP LU decomposition (similar to a code from Edwin Spee, CWI, Amsterdam)

running **WKC**

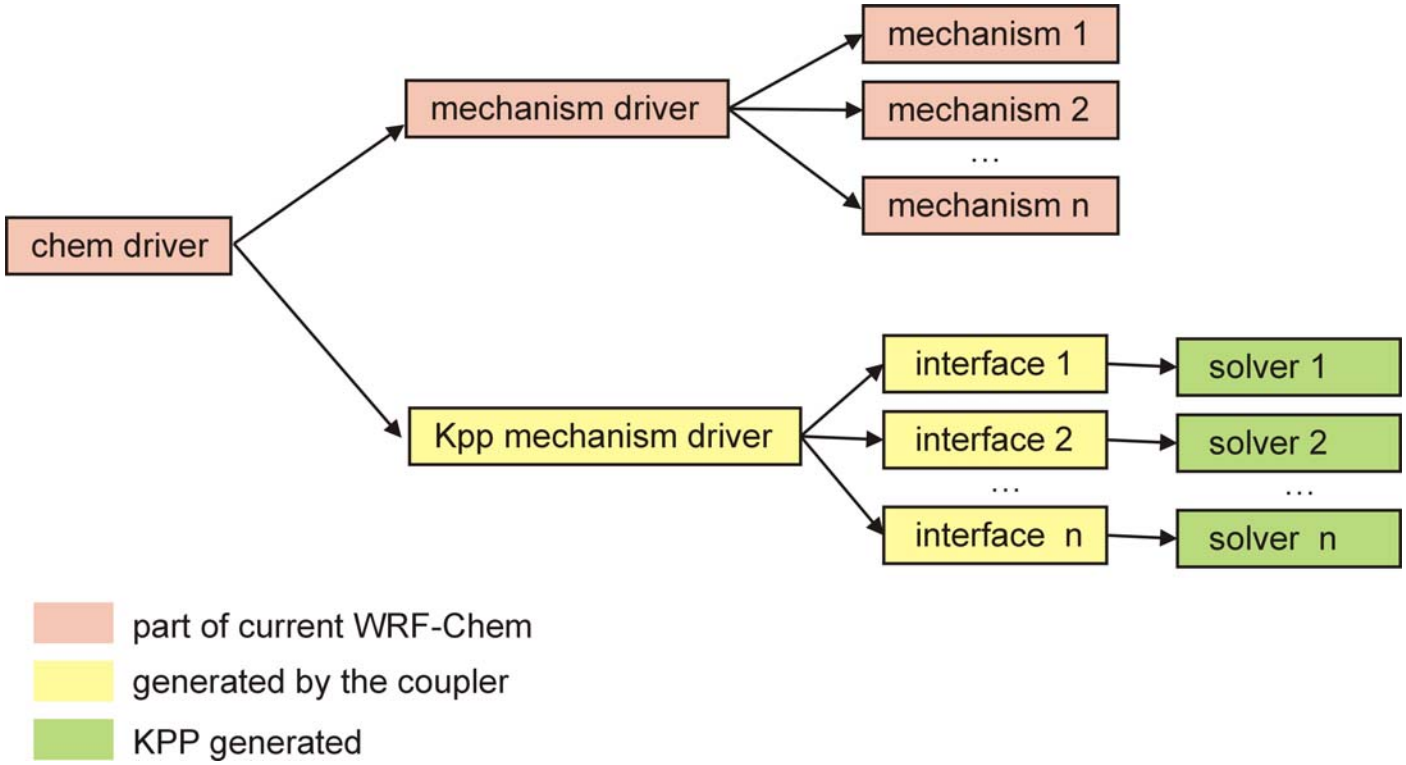
running the WRF-Chem/KPP coupler

- excerpt from WRF compile script:

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

=> set environment variable: **setenv WRF_KPP 1**

call tree for the generated code



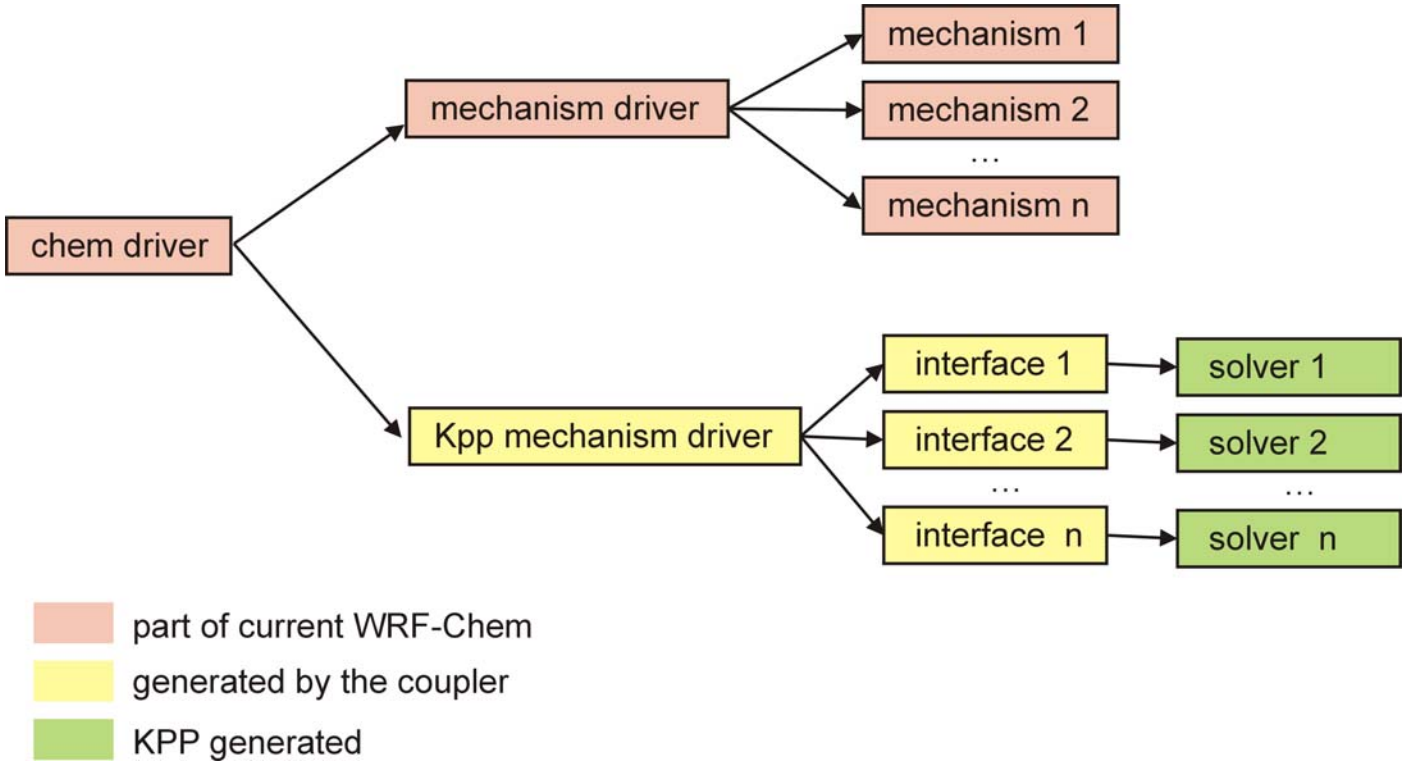
call from chem_driver.F

```
#ifdef WRF_KPP
 #ifdef WRF_KPP
    CALL wrf_debug(15,'calling kpp_mechanism_driver')

 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,                                     &
    its,ite,jts,jte,kts,kte)

 #endif
```

call tree for the generated code



output in chem directory

after compile!

```
> cd chem; ls *kpp_racm_mim*

module_kpp_racm_mim_Integr.F
module_kpp_racm_mim_Jacobian.F
module_kpp_racm_mim_JacobianSP.F
module_kpp_racm_mim_Parameters.F
module_kpp_racm_mim_Precision.F
module_kpp_racm_mim_Update_Rconst.F
module_kpp_racm_mim_interface.F
```

solver

interface

in addition:

- modified **`Makefile`**
- link to **`module_wkppc_constants.F`**
- **`kpp_mechanism_driver.F`**

customization

customization

```
#ifdef WRF_KPP
   CALL wrf_debug(15,'calling kpp_mechanism_driver')

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,                                     &
   its,ite,jts,jte,kts,kte)

#endif
```

fixed

automatically generated

customization

```
 SUBROUTINE kpp_mechanism_driver(   &
!
#include <fixed_args_kpp_interf.inc>
!
              addt, addx, addc, etep, oltp, &
              olip, cslp, limp, hc5p, hc8p, &
              tolp, xylp, apip, isop, hc3p, &
              ethp, o3p, tco3, mo2, o1d, &
              olnn, olnd, rpho, xo2, ketp, &
              xno2, ol2p, oln, macp,  &
              ph_o31d, ph_o33p, ph_no2, ph_no3o2, ph_no3o, &
              ph_hno2, ph_hno3, ph_hno4, ph_h2o2, ph_ch2or, &
              ph_ch2om, ph_ch3cho, ph_ch3coch3, ph_ch3coc2h5, ph_hcocho, &
              ph_ch3cocho, ph_hcochest, ph_ch3o2h, ph_ch3coo2h, ph_ch3ono2, &
              ph_hcochob, ph_macr, ph_n2o5, ph_o2,  &
               ids,ide, jds,jde, kds,kde,          &
               ims,ime, jms,jme, kms,kme,          &
               its,ite, jts,jte, kts,kte          )
```

fixed

automatically generated

fixed

customization

- if possible, use .inc files in inc directory:

```
> ls chem/KPP

clean_kpp
compile_wkc
configure_kpp
documentation
inc
kpp
mechanisms
module_wkppc_constants.F
util
```

customization

```
> cd inc
> ls *fixed*

 fixed_args_kpp_interf.inc
 fixed_decl_kpp_interf.inc
```

customization

```
> less fixed_args_kpp_interf.inc
```

```
chem, id, dtstepc,config_flags, &
p_phy,t_phy,rho_phy,moist,      &
vdrog3, ldrog,                  &
```

customization

```
> cd inc
> ls *racm_mim*

extra_args_to_update_rconst_racm_mim.inc
extra_args_update_rconst_racm_mim.inc
extra_decls_update_rconst_racm_mim.inc
kpp_mechd_a_racm_mim.inc
kpp_mechd_b_racm_mim.inc
kpp_mechd_e_racm_mim.inc
kpp_mechd_ia_racm_mim.inc
kpp_mechd_ib_racm_mim.inc
kpp_mechd_ibu_racm_mim.inc
kpp_mechd_l_racm_mim.inc
kpp_mechd_u_racm_mim.inc
```

installation

installation

- distributed together with WRF-Chem
- few prerequisites for compiling KPP

KPP prerequisites

- needed for installation
  - flex
  - yacc

WKC documentation

- **currently:**

www.mpch-mainz.mpg.de/~salzmann/my_home/sub/wkc.html

… will also be in WRF-Chem Users' guide

- **questions/suggestions:**

wrfchemhelp.gsd@noaa.gov

or: salzmann@mpch-mainz.mpg.de

testing

- mainly performed at NOAA ESRL (Georg Grell, Steven Peckham)
- also myself

known issues

- **major bug found in last release**
  - **see WG11 website, known issues WRF-Chem**
    http://ruc.fsl.noaa.gov/wrf/WG11/known-prob.htm

- **efficiency**
  - larger initial time increment for ros3
  - indirect addressing in KPP LU decomposition

  => WRF-Chem/KPP almost twice as fast

  => next release

future perspectives

- **improvements to WKC**
  - more options
  - easier to use
  - additional adjustments?
- **contributed mechanisms?**
- **your feedback is highly welcome**

future perspectives

- WRF-Chem/KPP coupler can be extended to handle aqueous chemistry
- KPP can also be used to solve linear first order ODEs describing mass transfer (e.g. Schwartz, 1986)

future perspectives

- more generalized
    - emsissons
    - photolysis rates
    - initial and boundary conditions
    - dry deposition rates
    - wet depostion
    - …..

Thank you!

acknowledgements

- Mark Lawrence
- Rolf von Kuhlmann, Rolf Sander
- Georg Grell, Steven Peckham