Gas-phase Chemistry with Emphasis on KPP

Si-Wan Kim

NOAA/ESRL & CIRES, U. of Colorado

Based on

Marc Salzmann's presentations and discussions with him

Outline

1. Chemical mechanisms in WRF-Chem model
2. KPP
3. WRF-Chem KPP Coupler (WKC)
4. How to add or updates mechanisms in WRF-Chem model
5. Example
6. Suggestions

1. Chemical mechanisms in WRF-Chem model

Chemical mechanisms in WRF-Chem model

/WRFV3/Registry/registry.chem

```
# Chemistry package definitions

package radm2              chem_opt==1
package radm2sorg          chem_opt==2
package racm               chem_opt==3
package racmsorg           chem_opt==4
package cbmz               chem_opt==5
                           :
package chem_tracer        chem_opt==13
package radm2_kpp          chem_opt==101
package radm_mim_kpp       chem_opt==102
```

Chemical mechanisms in WRF-Chem model

Chemistry package definitions

package racm_kpp	chem_opt==103
package racmpm_kpp	chem_opt==104
package racmsorg_kpp	chem_opt==105
package radm2sorg_kpp	chem_opt==106
package cbm4_kpp	chem_opt==110
package nmhc9_kpp	chem_opt==200
package gocart_simple	chem_opt==300
package gocartracm_kpp	chem_opt==301

2. KPP: Kinetic PreProcessor

What is KPP?

- Kinetic PreProcessor

: reads chemical reactions and rate constants from ASCII input files and automatically generates code for chemistry integration.

- Advantages

: less time consuming than manual coding

less error prone

numerically efficient

flexibility in updating mechanism with additional equations

- Reference

: Damian et al., *Computers and Chemical Engineering*, 2002

Sandu et al., *Atmos. Environ*., 2003

Sandu and Sander, *Atmos. Chem. Phys*., 2006

Input files of KPP

- .spc file

  Definition of chemical species as variable or fixed value.

- .eqn file

  Writing chemical reactions in kpp format

- .kpp file

  Model description, computer language, precision, integrator (e.g. Rosenbrock solver) etc.

- .def file

  User defined function (also check $KPP_HOME/util/UserRateLaws.f90 or $KPP_HOME/util/WRF_Conform/UserRateLaws.f90)

3. WRF-Chem/KPP Coupler (WKC)

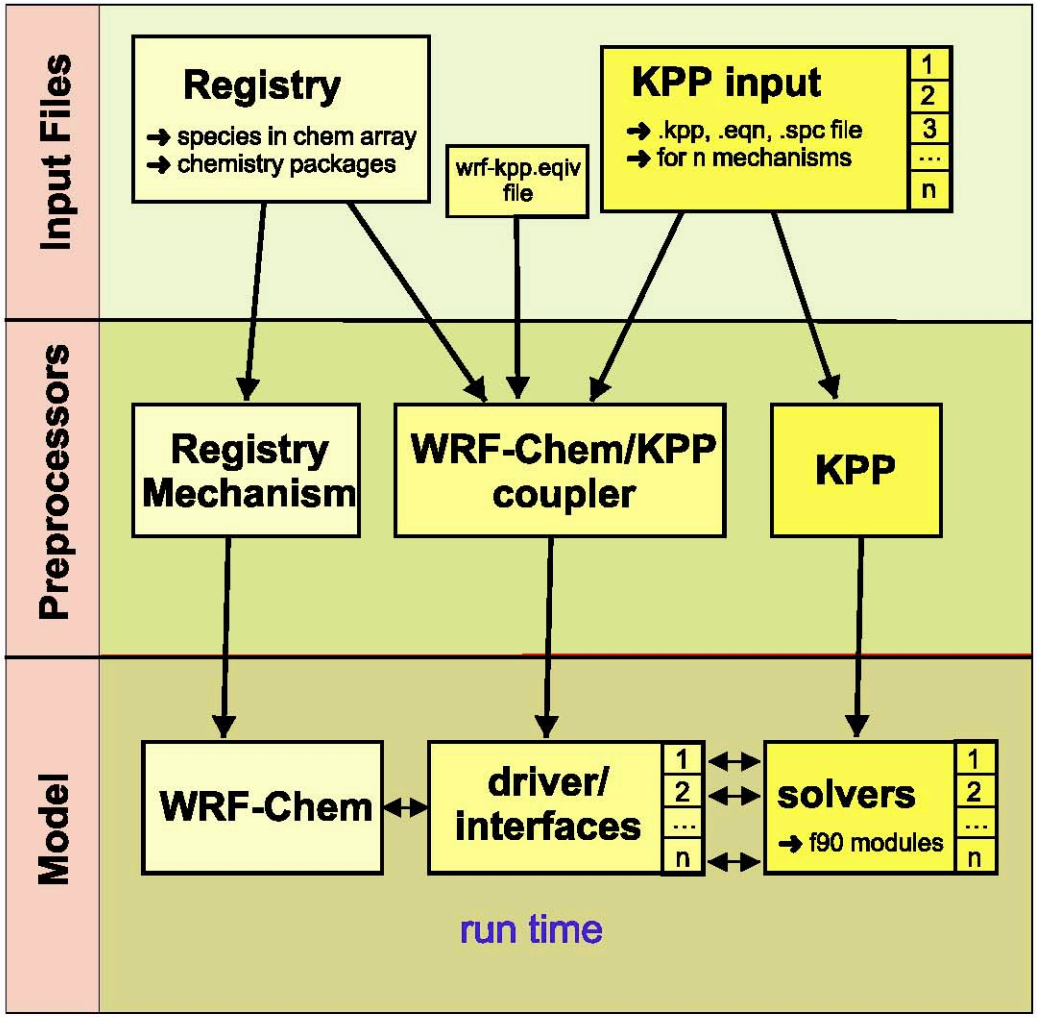
WKC (WRF-Chem/KPP Coupler)

- write f90 interfaces between WRF-Chem and KPP generated code

- minimum modifications to WRF-Chem, almost everything in one directory and also minimum changes in KPP

- script to automatically compile/run KPP during WRF compile time

WKC (WRF-Chem/KPP Coupler)



(From Salzmann and Lawrence, 2006 WRF user workshop)

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
ho                     OH
```

Pre-defined variables in WKC

	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)

4. How to add or update mechanism with KPP?

1) Add new chemistry package in /WRFV3/Registry/registry.chem.

--> If there are new species and/or photolysis rates defined in the mechanism, those need to be added in chemistry array in the registry file.

2) Make a new subdirectory in /WRFV3/chem/KPP/mechanisms.

3) Change directory to the new directory and generate KPP input files.

4) Modify programs (e.g., case statement) in /chem in order to incorporate the new mechanism

5) Run WKC

6) Further modification

5. Example: update rate constants of RACM-MIM (Geiger et al., Atmos. Environ., 2003) following JPL 2006 report etc.

1) Add new chemistry package in /WRFV3/Registry/registry.chem.

For example, new mechanism “racm_esrlsorg_kpp”

package racm_esrlsorg_kpp chem_opt==108 -
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,so4aj,so4ai,nh4aj,nh4ai,no3aj,no3ai,orgaro1j,orgaro1i,orgaro2j,orgaro2i,orgalk1j,orgalk1i,orgole1j,orgole1i,orgba1j,orgba1i,orgba2j,orgba2i,orgba3j,orgba3i,orgba4j,orgba4i,orgpaj,orgpai,ecj,eci,p25j,p25i,antha,seas,soila,nu0,ac0,corn

2) Make a new subdirectory in /WRFV3/chem/KPP/mechanisms

The name of this directory should be the same as the package name in the Registry without the "_kpp" suffix.

(e.g.) /WRFV3/chem/KPP/mechanisms/racm_esrlsorg

3) Change directory to the new directory and link some files like in chem/KPP/mechanisms/racm_mim:

ln -s ../racm_mim/atmos_red .

ln -s ../racm_mim/racm_mim.eqn ./racm_esrlsorg.eqn

ln -s ../racm_mim/racm_mim.spc ./racm_esrlsorg.spc

ln -s ../racm_mim/racm_mim_wrfkpp.equiv ./racm_esrlsorg_wrfkpp.equiv

copy ../racm_mim.kpp ./racm_esrlsorg.kpp

copy ../racm_mim.def ./racm_esrlsorg.def

4) Update reaction sets and rate constants etc.

racm_esrlsorg.eqn

```
#EQUATIONS {}
 {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) ;
 {024:001} O3P+M{O2}=O3               : (C_M *6.00D-34*(TEMP/300.0)**(-2.4)) ;
 {025:002} O3P+O3=M {2O2}             : ARR2( 8.00D-12 , 2060.0_dp, TEMP) ;
 {026:003} O1D + M = O3{+O2}          : .78084*ARR2(2.15D-11 , -110.0_dp, TEMP);
 {027:004} O1D + M = O3P{+O2}         : .20946*ARR2( 3.30D-11 , -55.0_dp , TEMP ) ;
 {028:005} O1D+H2O=HO+HO              : ARR2( 1.63D-10 , -60.0_dp, TEMP ) ;
 {029:006} O3+HO=HO2{+O2}             : ARR2( 1.70D-12 , 940.0_dp, TEMP ) ;
 {030:007} O3+HO2=HO{+2.0 O2}         : ARR2( 1.0D-14 , 490.0_dp, TEMP ) ;
 {031:008} HO+HO2=H2O{+O2}            : ARR2( 4.80D-11 , -250.0_dp, TEMP ) ;
 {032:009} H2O2+HO=HO2+H2O            : 1.8D-12 ;
 {033:010} HO2+HO2=H2O2{+O2}          : (3.5D-13*EXP(430./TEMP) + 1.7D-33* C_M *EXP(1000./TEMP)) ;
 {034:011} HO2+HO2+H2O=H2O2+H2O{+O2}    : (4.9D-34* EXP(2630./TEMP)+ 2.38D-54* C_M *EXP(3200./TEMP)) ;
 {035:012} O3P+NO=NO2                 : TROE( 9.00D-32 , 1.5_dp , 3.00D-11 , 0.0_dp , TEMP, C_M) ;
```

4) Update equation sets and rate constants etc.

racm_esrlsorg.def

```
#include atoms_red
#include ./racm_esrlsorg.spc
#include ./racm_esrlsorg.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=2.4E-14_dp * EXP(460._dp/TEMP)
   k2=2.7E-17_dp * EXP(2199._dp/TEMP)
   k3=6.5E-34_dp * EXP(1335._dp/TEMP) * c_m

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


END FUNCTION k46
```

5) Modify programs in /chem in order to include the new mechanism.

chem_driver.F

chemics_init.F

module_input_chem_data.F

mechanism_driver.F

aerosol_driver.F

dry_dep_driver.F

module_bioemi_megan2.F

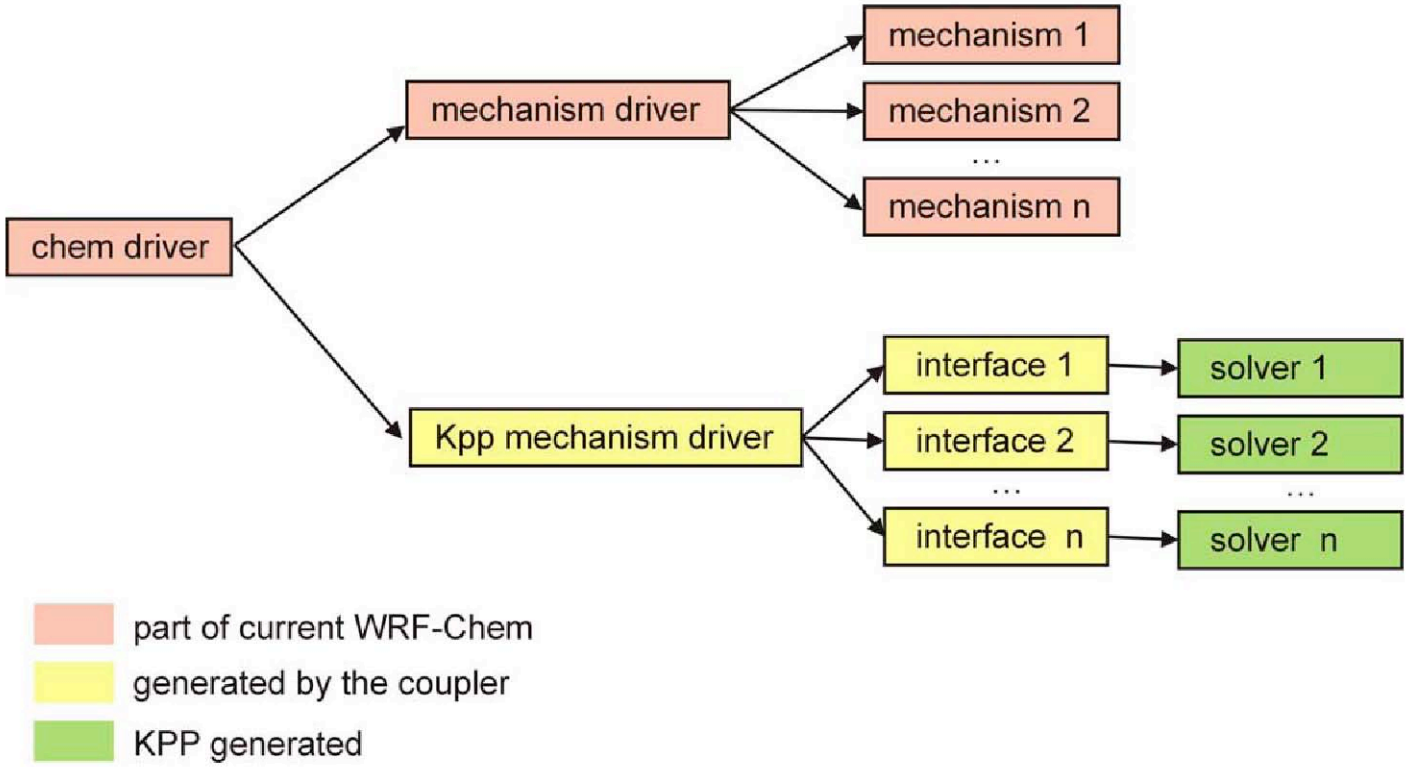
module_ftuv_driver.F

(e.g.) case (RADM2SORG_KPP,…RACM_ESRLSORG_KPP)

6) Run the WRF-Chem compiler (coupler)

For KPP installation, programs yacc and flex are needed.

set environment variable: setenv WRF_KPP 1



After running the WRF-Chem compiler

/WRFV3/chem/:

module_kpp_racm_esrlsorg_Integr.F
module_kpp_racm_esrlsorg_Integr.f90
module_kpp_racm_esrlsorg_Integr.o
module_kpp_racm_esrlsorg_interface.F
module_kpp_racm_esrlsorg_interface.f90
module_kpp_racm_esrlsorg_interface.o
module_kpp_racm_esrlsorg_interf.mod
module_kpp_racm_esrlsorg_Jacobian.F
module_kpp_racm_esrlsorg_Jacobian.f90
module_kpp_racm_esrlsorg_Jacobian.o
module_kpp_racm_esrlsorg_JacobianSP.F
module_kpp_racm_esrlsorg_JacobianSP.f90
module_kpp_racm_esrlsorg_JacobianSP.o
module_kpp_racm_esrlsorg_Parameters.F
module_kpp_racm_esrlsorg_Parameters.f90
module_kpp_racm_esrlsorg_Parameters.o
module_kpp_racm_esrlsorg_Precision.F
module_kpp_racm_esrlsorg_Precision.f90
module_kpp_racm_esrlsorg_Precision.o
module_kpp_racm_esrlsorg_Update_Rconst.F
module_kpp_racm_esrlsorg_Update_Rconst.f90
module_kpp_racm_esrlsorg_Update_Rconst.o
racm_esrlsorg_integrator.mod
racm_esrlsorg_jacobian.mod
racm_esrlsorg_jacobiansp.mod
racm_esrlsorg_parameters.mod
racm_esrlsorg_precision.mod
racm_esrlsorg_updaterconstwrf.mod

After running the WRF-Chem compiler

/WRFV3/chem/KPP/mechanisms/racm_esrlsorg:

atoms_red* racm_esrlsorg_Jacobian.f90 racm_esrlsorg_Monitor.f90
Makefile@ racm_esrlsorg_JacobianSP.f90 racm_esrlsorg_Parameters.f90
Makefile_racm_esrlsorg racm_esrlsorg.kpp* racm_esrlsorg_Precision.f90
racm_esrlsorg.def* racm_esrlsorg_LinearAlgebra.f90 racm_esrlsorg_Rates.f90
racm_esrlsorg.eqn racm_esrlsorg_Main.f90 racm_esrlsorg.spc*
racm_esrlsorg_Function.f90 racm_esrlsorg.map
racm_esrlsorg_Update_Rconst.f90
racm_esrlsorg_Global.f90 racm_esrlsorg_mex_Fun.f90 racm_esrlsorg_Util.f90
racm_esrlsorg_Initialize.f90 racm_esrlsorg_mex_Jac_SP.f90
racm_esrlsorg_wrfkpp.equiv*
racm_esrlsorg_Integrator.f90 racm_esrlsorg_Model.f90

After running the WRF-Chem compiler

/WRFV3/chem/KPP/inc/racm_esrlsorg:

extra_args_to_update_rconst_racm_esrlsorg.inc
extra_args_update_rconst_racm_esrlsorg.inc
extra_decls_update_rconst_racm_esrlsorg.inc
kpp_mechd_a_racm_esrlsorg.inc
kpp_mechd_b_racm_esrlsorg.inc
kpp_mechd_e_racm_esrlsorg.inc
kpp_mechd_ia_racm_esrlsorg.inc
kpp_mechd_ib_racm_esrlsorg.inc
kpp_mechd_ibu_racm_esrlsorg.inc
kpp_mechd_l_racm_esrlsorg.inc
kpp_mechd_u_racm_esrlsorg.inc

7) Further modification

(e.g.) .inc files in the chem/KPP/inc directory.

These files can be edited manually to implement the actual coupling and the changes will be retained when the coupler is re-run.

For example, RACM_ESRL chemistry & SORGAM aerosol routines could be coupled through .inc files.

kpp_mechd_ia_racm_esrlsorg.inc --> module_kpp_racm_esrlsorg_interface.F
--> module_aerosols_sorgam.F

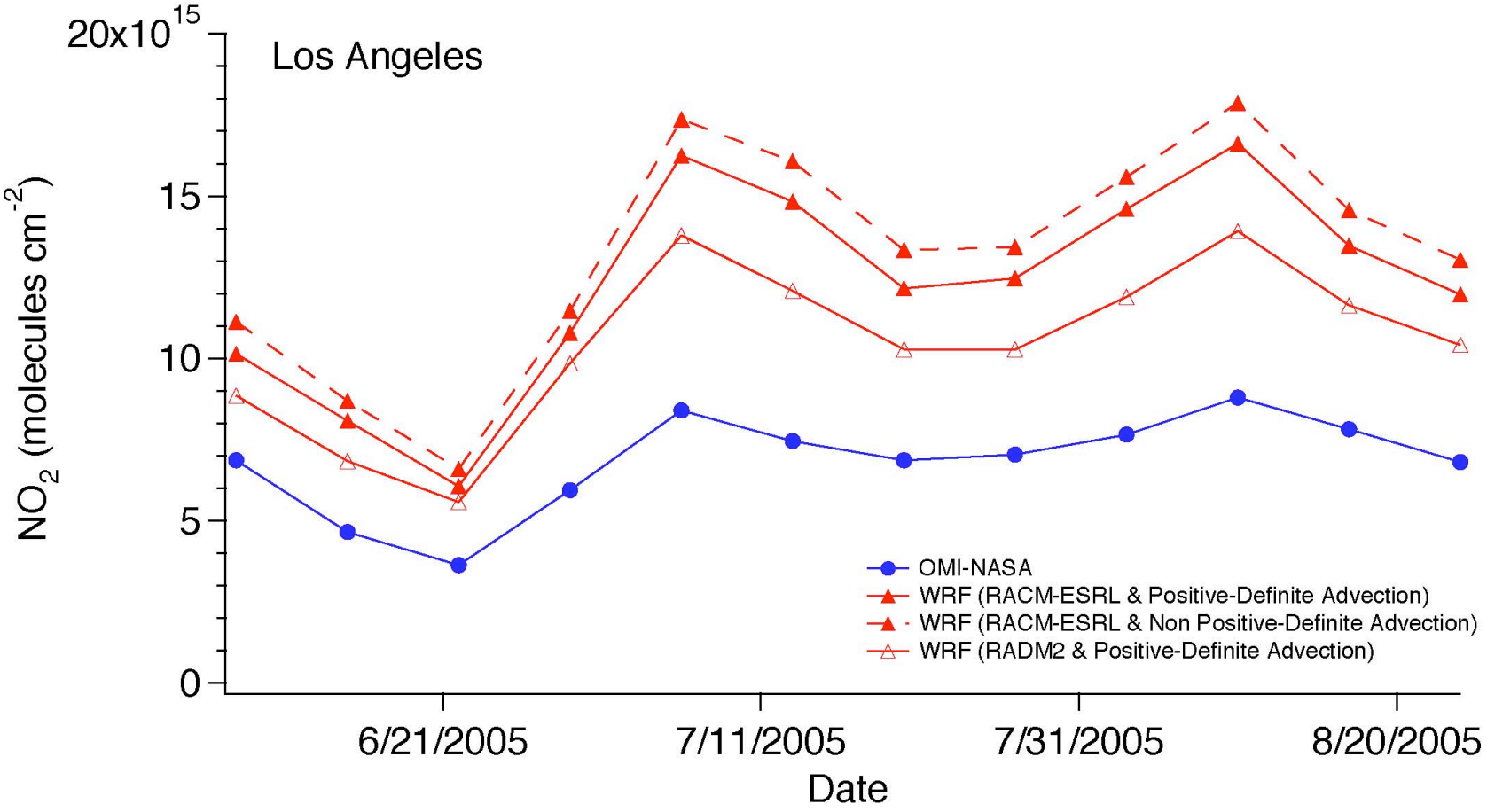
kpp_mechd_ia_racm_esrlsorg.inc:

```
if(p_nu0.gt.1)then
 rxylho =   RCONST(74)     !ARR(7.30e-12,-355.0,t_phy(i,k,j)))
 rtolho =   RCONST(73)     !ARR(1.81e-12,-355.0,t_phy(i,k,j)))
 rcslho =   RCONST(75)     !ARR(6.00e-11,0.0,t_phy(i,k,j)))
 rcslno3 =  RCONST(97)     !ARR(2.20e-11,0.0,t_phy(i,k,j)))
 rhc8ho =   RCONST(65)     !ARR(1.64e-11,125.0,t_phy(i,k,j)))
 roliho =   RCONST(68)     !ARR(1.33e-11,-500.0,t_phy(i,k,j)))
 rolino3 =  RCONST(100)    !ARR(8.64e-13,-450.0,t_phy(i,k,j)))
 rolio3 =   RCONST(108)    !ARR(4.40e-15,845.0,t_phy(i,k,j)))
 roltho =   RCONST(67)     !ARR(5.72e-12,-500.0,t_phy(i,k,j)))
 roltno3 =  RCONST(99)     !ARR(1.79e-13,450.0,t_phy(i,k,j)))
 rolto3 =   RCONST(107)    !ARR(4.33e-15,1800.0,t_phy(i,k,j)))
 rapiho =   RCONST(71)     !ARR(1.21e-11,-444.0,t_phy(i,k,j)))
 rapino3 =  RCONST(103)    !ARR(1.19e-12,-490.0,t_phy(i,k,j)))
 rapio3 =   RCONST(111)    !ARR(1.01e-15,736.0,t_phy(i,k,j))) !! 732!!
 rlimho =   RCONST(72)     !ARR(1.71e-10,0.0,t_phy(i,k,j)))
 rlimno3 =  RCONST(104)    !ARR(1.22e-11,0.0,t_phy(i,k,j)))
 rlimo3 =   RCONST(112)    !ARR(2.00e-16,0.0,t_phy(i,k,j)))
```

```
    PRDROG(PXYL)  = rxylho * var(ind_xyl)*var(ind_ho)
    PRDROG(PTOL)  = rtolho * var(ind_tol)*var(ind_ho)
    PRDROG(PCSL1) = rcslho * var(ind_csl)*var(ind_ho)
    PRDROG(PCSL2) = 0.50_dp * rcslno3* var(ind_csl)*var(ind_no3)
    PRDROG(PHC8)  = rhc8ho * var(ind_hc8)*var(ind_ho)
    PRDROG(POLI1) = roliho * var(ind_oli)*var(ind_ho)
    PRDROG(POLI2) = rolino3 * var(ind_oli)*var(ind_no3)
    PRDROG(POLI3) = rolio3 * var(ind_oli)*var(ind_o3)
    PRDROG(POLT1) = roltho * var(ind_olt)*var(ind_ho)
    PRDROG(POLT2) = roltno3 * var(ind_olt)*var(ind_no3)
    PRDROG(POLT3) = rolto3 * var(ind_olt)*var(ind_o3)
    PRDROG(PAPI1) = rapiho * var(ind_api)*var(ind_ho)
    PRDROG(PAPI2) = rapino3 * var(ind_api)*var(ind_no3)
    PRDROG(PAPI3) = rapio3 * var(ind_api)*var(ind_o3)
    PRDROG(PLIM1) = rlimho * var(ind_lim)*var(ind_ho)
    PRDROG(PLIM2) = rlimno3 * var(ind_lim)*var(ind_no3)
    PRDROG(PLIM3) = rlimo3 * var(ind_lim)*var(ind_o3)
    DO n = 1, LDROG
      VDROG3( i,k,j, n ) =  oconv * PRDROG( n ) * DTSTEPC
      VDROG3( i,k,j,n  ) = MAX( 0., VDROG3( i,k,j, n ) )
    ENDDO
endif
```

module_aerosols_sorgam.F

Results from RACM_ESRLSORG



6. Suggestions

- Caution!

Check if the added mechanisms work with pre-existing initial and boundary conditions, emissions, photolysis rates, dry deposition rates, and wet deposition rates etc.

- Refer WKC documentation:

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

- Questions/suggestions:

wrfchemhelp.gsd@noaa.gov