

A Tutorial on Chemkin

Niket Kaisare

(11-12-2007)

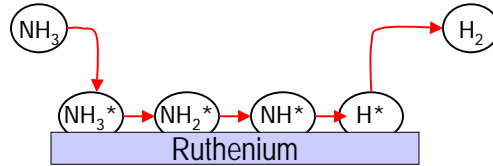
Indian Institute of Technology – Madras

Original Presentation: Dr. Ashish Mhadeshwar (June 2004)

Why use Chemkin?

1. Efficient handling of large reaction mechanisms
2. Generalized framework, simple and standard inputs
3. Minimization of common mistakes
4. Common platform for distributing new products

Example: Ammonia Decomposition on Ruthenium



Requirements

We intend to solve a combined flow-diffusion-reaction problem

1. Reaction stoichiometry and rate constants
2. Reaction thermo-chemistry
3. Thermodynamic properties
4. Transport properties (diffusion, viscosity and thermal conductivity)

Example: Ammonia Decomposition on Ruthenium

Matrix of stoichiometric coefficients

1. $\text{N}_2 + 2^* \leftrightarrow 2\text{N}^*$
2. $\text{H}_2 + 2^* \leftrightarrow 2\text{H}^*$
3. $\text{NH}^* + ^* \leftrightarrow \text{N}^* + \text{H}^*$
4. $\text{NH}_2^* + ^* \leftrightarrow \text{NH}^* + \text{H}^*$
5. $\text{NH}_3^* + ^* \leftrightarrow \text{NH}_2^* + \text{H}^*$
6. $\text{NH}_3 + ^* \leftrightarrow \text{NH}_3^*$

	R_1	R_2	R_3	R_4	R_5	R_6
N^*	2		1			
H^*		2	1	1	1	
NH^*			-1	1		
NH_2^*				-1	1	
NH_3^*					-1	1

Kinetic parameters for each reaction

$$r_j = A_{0j} \left(\frac{T}{T_{ref}} \right)^{\beta_j} e^{-E_j/RT} \prod a_{kj}^m$$

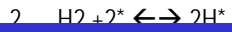
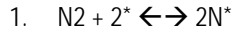
Reaction thermo-chemistry ΔH , ΔG , ΔS

Thermodynamic and Transport properties

$$c_p, \hat{H}, \rho; \\ k_g, D_{ik}, \mu$$

Example: Ammonia Decomposition on Ruthenium

Matrix of stoichiometric coefficients



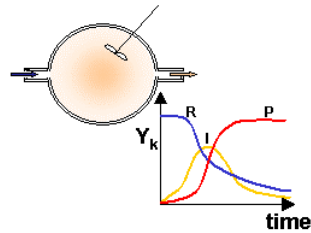
	R_1	R_2	R_3	R_4	R_5	R_6
N^*	2		1			
H^*		2	1	1	1	

1. Prone to errors
2. Detection of errors and debugging is not straightforward
3. Adding / removing reactions from mechanism is cumbersome

Reaction thermo-chemistry ΔH , ΔG , ΔS

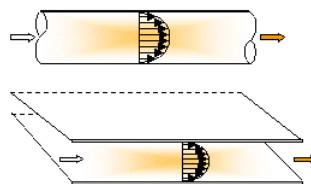
Thermodynamic and Transport properties $c_p, \hat{H}, \rho;$
 k_g, D_{ik}, μ

Utilities



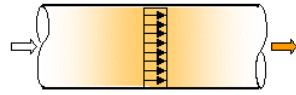
AURORA/SENKIN

AURORA predicts the time-evolution or steady state of a **well mixed reactor**.

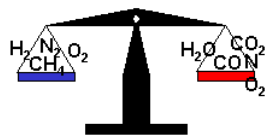


CRESLAF simulates laminar, chemically reacting, **boundary-layer flow** in cylindrical or planar channels.

Utilities

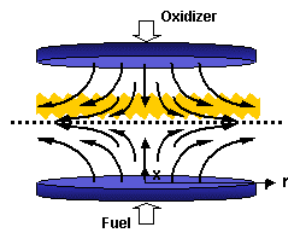


PLUG simulates a **plug-flow reactor** with gas-phase and surface chemistry.

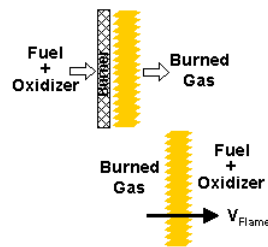


EQUIL calculates the **equilibrium state of systems** containing ideal gas mixtures or ideal solutions.

Utilities

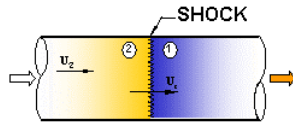


OPPDIF predicts the structure of **opposed-flow diffusion** flames in cylindrical or planar coordinate systems.

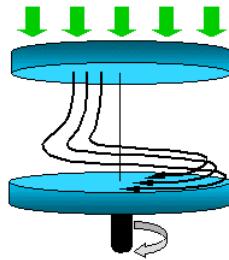


PREMIX predicts the structure of steady, laminar, 1D, **pre-mixed flames**.

Utilities



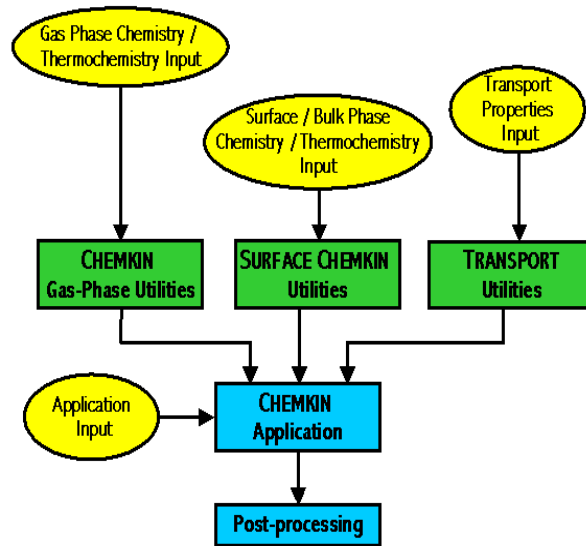
SHOCK: A program for predicting chemical behavior behind incident and reflected shock waves.



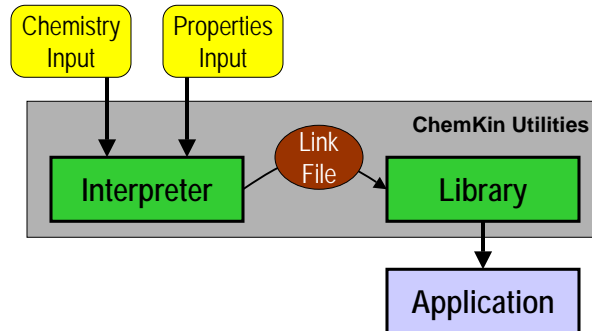
SPIN: A program for modeling one-dimensional, rotating-disk or stagnation-flow chemical-vapor-deposition reactors.

Components of Chemkin Suite

1. Chemkin
2. Surface Chemkin
3. Transport



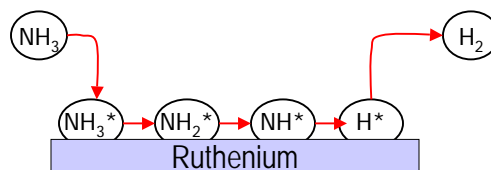
Components of Chemkin Suite



	CHEMKIN	SURFACE	TRANSPORT
Interpreter	ckinterp.x	skinterp.x	tranfit.x
Library	cklib.f	sklib.f	tranlib.f
Chemistry Input	fort.15	surf.inp	---
Physical Properties Input	thermdat		trandat
Link File	cklink	sklink	tplink

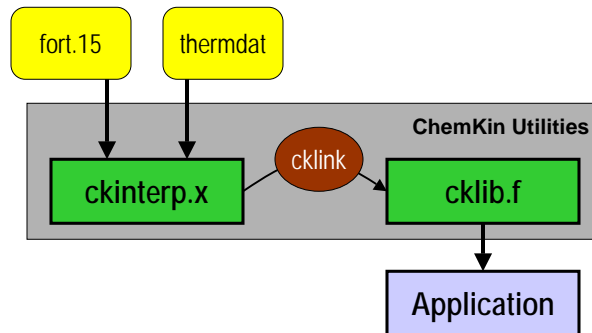
Note: Surface Chemkin and Transport Libraries require cklink for species information

Example: Ammonia Decomposition on Ruthenium



1. **CHEMKIN:** To define gas phase species and thermodynamics
2. **SURFACE CHEMKIN:** To define surface species and reactions
3. **TRANSPORT:** To define gas phase transport properties (diffusion, viscosity and thermal conductivity) *if applicable*

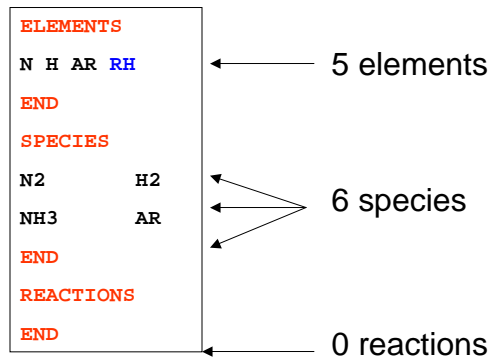
Defining Gas Phase Conditions in Chemkin



Steps:

- Create *fort.15* (gas phase reaction mechanism)
 - Input of elements, species, and reactions in *fort.15*
- Input of thermodynamic database: *thermdat*
- Compile *ckinterp.f* to create *ckinterp.x*
- Run *ckinterp.x* to create *fort.16* and *cklink* (**MUST READ fort.16**)
- Compile *cklib.f* to create *cklib.o* and link with your code

1. Chemkin inputs: Sample *fort.15*



Chemkin peculiarity:

Catalyst should be declared with gas phase species

2. Chemkin inputs: Thermodynamic database *thermdat*

Low Mid High

THERMO
300.000 1000.000 5000.000

```
NH3      L 7/88N  1H  3      G 200.000 3500.000 1000.000  1
3.85746029E+00 4.41437026E-03-2.21481404E-06 5.23490188E-10-4.72084164E-14  2
-4.87591660E+04 2.27163806E+00 2.35677352E+00 8.98459677E-03-7.12356269E-06  3
2.45919022E-09-1.43699548E-13-4.83719697E+04 9.90105222E+00  4
```

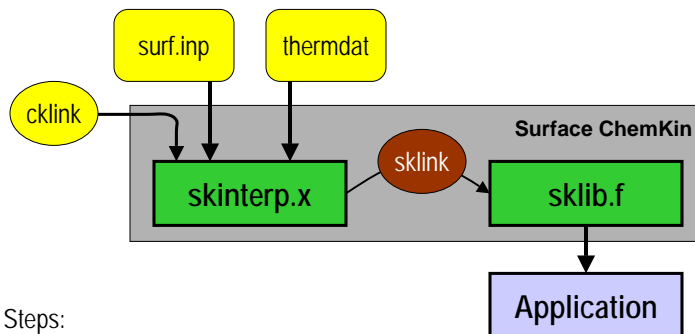
Chemkin peculiarity:

Fixed format that must be followed for every new entry

3. Check output file *fort.16* for errors

Run *ckinterp.x*. Two output files are created: *cklink* and *fort.16*

Defining Surface Conditions in Chemkin



Steps:

- Create *surf.inp* (surface reaction mechanism)
 - Input surface species and reactions
- Input of thermodynamic database: *thermdat*
- Compile *skinterp.f* to create *skinterp.x*
- Run *skinterp.x* to create *surf.out* and *sklink* (**MUST READ *surf.out***)
- Compile *cklib.f* to create *cklib.o* and link with your code

1. Surface Chemkin inputs: Sample surf.inp

```

SITE/SURFACE/      SDEN/1.66058e-9/
  H(S)   N(S)   NH(S)
  NH2(S) NH3(S) RU(S)
END

BULK RU(B)/12.37/
END

REACTIONS      kcal/mole
H2 + 2RU(S) => 2H(S) + 2RU(B)      1.0      0.0      1.9
  STICK
2H(S) + 2RU(B) => H2 + 2RU(S)      6.022e21  0.0      23.7

N2 + 2RU(S)   => 2N(S) + 2RU(B)      1.0      0.0      6.2
  STICK
2N(S) + 2RU(B) => N2 + 2RU(S)      6.022e21  0.0      50.3

NH(S) + RU(S) => N(S) + H(S) + RU(B) 6.022e19  0.0      5.8
COV/N(S) 0.0 0.0 15.4/
N(S) + H(S) + RU(B) => NH(S) + RU(S) 6.022e19  0.0      37.3
COV/N(S) 0.0 0.0 -19.7/
...

```

Annotations:

- site density (mol/cm²s) → SDEN/1.66058e-9/
- surface species → H(S), N(S), NH(S), NH2(S), NH3(S), RU(S)
- units → RU(B)/12.37/
- reaction → H2 + 2RU(S) => 2H(S) + 2RU(B)
- sticking keyword → STICK
- Coverage dependent parameters → COV/N(S) 0.0 0.0 15.4/

$$k = A_0 \left(\frac{T}{T_{ref}} \right)^\beta \exp\left(-\frac{E}{RT} \right)$$

1. Surface Chemkin inputs: Peculiarities in surf.inp

Site density is a 3 element vector (gas, surface, bulk). Surface site density, i.e., $sden(2)$ is typically used in the calculations, other two are zero.

For every consumed vacant surface site, a bulk site is produced. However, for reactions in the category of $A(S) + B(S) \rightarrow C(S) + D(S)$, no vacant surface sites or bulk sites are involved.

Reactions are written in irreversible manner, due to unavailability of thermodynamic database (specific heats) for surface species.

Units of pre-exponential factors are dependent on the type of reaction in the mechanism. The pre-exponential factor should be defined such that the reaction rate is calculated in mol/cm^2sec .

2. Surface Chemkin inputs: Thermodynamic database *thermdat*

```

RU(S)      81292RU 1                      S 300.00 2000.00 1000.00 1
0.02318546E+02 0.01929426E-01-0.01682130E-05 0.04131402E-09-0.16692935E-14 2
-0.07634172E+04-0.09802473E+02 0.15056368E+01 0.09217769E-01-0.01955590E-03 3
0.02020330E-06-0.07251403E-10-0.07226661E+04-0.06689818E+02 4
N(S)      TPIS79N 1                      S 300.000 3000.000 1000.000 1
0.00000E+00 0.00000E-00 -0.00000E-00 -0.00000E-00 0.00000E-00 2
-0.00000E+00 -0.00000E+00 -0.00000E+00 0.00000E-00 0.00000E-00 3
-0.00000E-00 0.00000E-00 -0.00000E+00 0.00000E+00 4
    
```

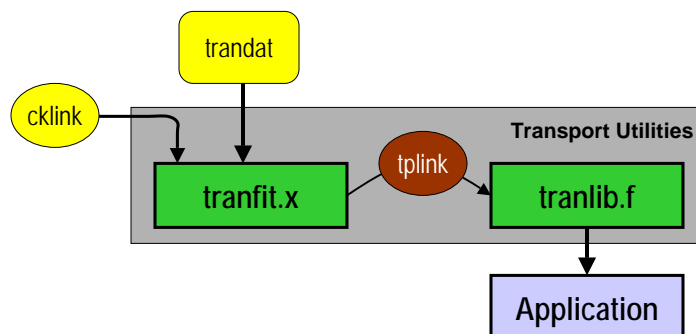
Follows same format as earlier

Can be provided in surf.inp or thermdat

Thermodynamic inputs for surface species are usually not available. Put all specific heats as zero, since all heats of reactions are calculated using BOC

Thermodynamic consistency is therefore important

Defining Surface Conditions in Chemkin



Steps:

- Input of transport database: *thermdat*
- Compile *tranfit.f* and *dmath.f* to create *tranfit.x*
- Run *tranfit.x* to create *tplink*
- Compile *tranlib.f* to create *tranlib.o* and link with your code

1. Transport inputs: Transport database *trandat* (optional)

AR	0	136.500	3.330	0.000	0.000	0.000
C	0	71.400	3.298	0.000	0.000	0.000 ! *
C2	1	97.530	3.621	0.000	1.760	4.000
C2O	1	232.400	3.828	0.000	0.000	1.000 ! *
CN2	1	232.400	3.828	0.000	0.000	1.000 ! OIS

species molecular index L-J potential well depth L-J collision diameter dipole moment polarizability rotational relaxation collision number

Interfacing with Application codes

Typical structure of Chemkin subroutines:

```

subroutine ckcpbs(t, yk, ickwrk, rckwrk, cpmean)
           Name Input(s)      Work arrays      Output(s)
  
```

Interfacing with Application codes: Chemkin Mnemonics

- ck** = Chemkin
- sk** = Surface Chemkin
- mc** = Transport
- iwk** = integer work array
- rwk** = real work array
- cwk** = character work array

Interfacing with Application codes: Dimensionalize Work Arrays

```
! *** File identifiers
integer, parameter :: linc = 25, linksk = 26, linkmc = 35, lout = 6

! *** Dimensionalizing work arrays
integer, parameter :: leniwrk = 6000, lenrwrk = 6000, lencwk = 6000
integer, parameter :: lsiwk = 6000, lsrwk = 6000, lscwk = 6000
integer, parameter :: lenimc = 6000, lenrmc = 6000

integer          :: ickwrk(leniwrk), iskwrk(lsiwk), imcwrk(lenimc)
double precision :: rckwrk(lenrwrk), rskwrk(lsrwk), rmcwrk(lenrmc)
character (len=16) :: cckwrk(lencwk), cskwrk(lscwk)
```

Interfacing with Application codes: Initialization

```
! *** Open chemkin link files
open (linc, file='cklink', form='unformatted')
open (linkmc, file='tplink', form='unformatted')
open (linksk, file='sklink', form='unformatted')

! *** Initialize chemkin
call ckinit (leniwrk, lenrwrk, lencwk, linc, lout, ickwrk, rckwrk, cckwrk)
call mcinit (linkmc, lout, lenimc, lenrmc, imcwrk, rmcwrk)
call skinit (lenisk, lenrsk, lencsk, linksk, lout, iskwrk, rskwrk, cskwrk)
```

Interfacing with Application codes: Using subroutines

```
!.....tutorial
open (unit=394, file='tutorial.out', status='unknown')
write(394,*)'Inlet mole fractions'
write(394,39)xin
call ckxty(xin, ickwrk, rckwrk, yin)
write(394,*)'Inlet mass fractions'
write(394,39)yin
```

```
Inlet mole fractions
0.300000 0.600000 0.100000 0.000000
Inlet mass fractions
0.743363 0.106195 0.150442 0.000000
```

Interfacing with Application codes: Using subroutines

```
write(394,*)'Initial coverages'  
write(394,39)(act(ii),ii=kgmax+1,ksmax-1)  
call skrat (p,temper,act,sden,iskwrk,rskwrk,sdot,sitdot)  
write(394,*)'Rates of species production'  
write(394,39)sdot
```

pressure

activity

Gas: mole fractions
Surface: coverage
Bulk: 1.0

Site density

sden(2) is the site density
of the catalyst surface

```
Initial coverages  
0.10000E-05 0.99990E+00 0.10000E-05 0.98000E-04  
Rates of species production  
-0.53416E-05 -0.21331E-05 0.45925E-05 0.00000E+00 0.10678E-10  
0.21331E-05 -0.45925E-05 0.24594E-05 -0.24594E-05
```

Recent modifications in Surface Chemkin

1. UBI-QEP implementation using a new subroutine (subroutine `ubi`) internally called from subroutine `skrrop`
2. Modification of `sklib.f` to incorporate
 - Molecular beam effects (subroutines `skatcz`, `skden`)
 - Large reaction mechanisms (subroutine `skinu`)
 - Replacement of temperature exponents using a new subroutine (subroutine `skrbex`)
 - Modification of Arrhenius temperature exponent formula (subroutine `skrrop`)

Useful links

Reaction Design:

<http://www.reactiondesign.com>

History of chemkin:

<http://egweb.mines.edu/rjkee/>

Overview of chemkin:

<http://www.sandia.gov/1100/CVDwww/chemkin.htm>

Equilibrium calculations:

<http://www.gaseq.co.uk/>

GRI-Mech:

http://www.me.berkeley.edu/gri_mech/

Sensitivity, reaction path, and principal component analysis modules:

<http://www.chem.leeds.ac.uk/Combustion/links.htm>

<http://www.cstl.nist.gov/div836/836.03/xsenkplot/index.html>

<http://www.chem.leeds.ac.uk/Combustion/kinalc.htm>

Useful links

Cantera - MATLAB based Chemkin:

<http://rayleigh.cds.caltech.edu/~goodwin/cantera/index.html>

Burcat's extended thermodynamic database:

<http://garfield.chem.elte.hu/Burcat/burcat.html>

Chemical reaction network toolbox:

<ftp://ftp.che.rochester.edu/pub/feinberg/>

EGLIB – A multicomponent transport software:

<http://www.cmap.polytechnique.fr/www.eglib/>

Caltech gas phase mechanism library:

<http://www.galcit.caltech.edu/EDL/mechanisms/library/library.html>

Software aid for developing mechanisms (unit conversion, deleting species, etc.):

<http://www.chem.leeds.ac.uk/Combustion/mechmod.htm>

Useful links

Information about Equil, Aurora, Senkin, Premix, Spin, Creslaf, Surftherm:
<http://www.me.umn.edu/courses/me8646/chemkin.html>

A general link for chemical engineers:
<http://www.che.ufl.edu/www-che/topics/software.html>