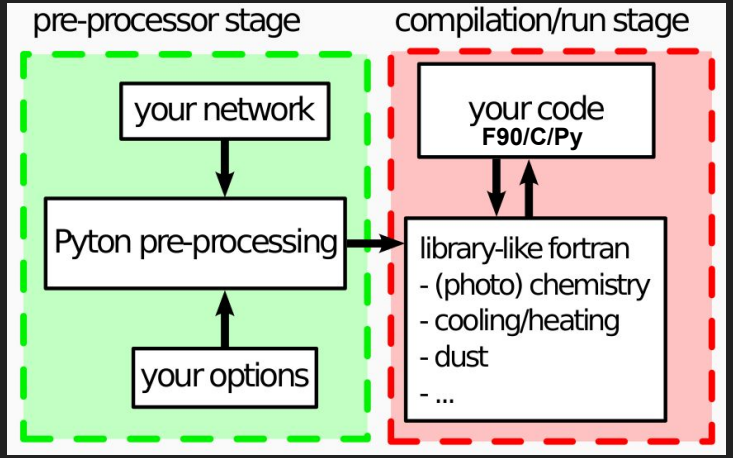
Chemical networks with KROME

(getting started)

Tommaso Grassi (USM/LMU)

KROME needs a chemical network

```
$ python3 krome
**********************************
WELCOME TO KROME
**********************************

ERROR: you must define -n FILENAME or -network FILENAME, where
FILENAME is the reaction file!
```




```
$ python3 krome -n my_network.dat
```

?

(answer in this presentation)

Choose a chemical network (it's problem-dependent)

***GIGO**: your model cannot be better than the quality of your data*

1. Use one of the pre-built KROME networks (networks/ folder, with caution)
2. Use the KIDA → KROME converter tool to select reactions from standard database (automatic selection is dangerous)
3. Find a chemical network in literature (recommended, updates?)
4. Prepare your own network based on databases, literature, experts, and pre-existing networks (ideal, time consuming)

A simple example

$$\mathrm{H^+ + e^- \rightarrow H + \gamma}$$

$$\frac{\mathrm{d}n_\mathrm{H}}{\mathrm{d}t} = k(T)\, n_{\mathrm{H^+}}\, n_{\mathrm{e^-}}$$

KIDA KINETIC DATABASE FOR ASTROCHEMISTRY

Home Species Download References Help

H+ + e-

Your query may contain lot of results.
Try some of these tips to filter results:

REACTANTS ○ PRODUCTS ○ BOTH ●

ASTRO ○ PLANETO ○ BOTH ●

ION + NEUTRAL ● NEUTRAL ○

SEARCH

SPECIES 2 | UNIMOLECULAR REACTIONS 3 | BIMOLECULAR REACTIONS 1 | TERMOLECULAR REACTIONS 0 | SURFACE REACTIONS 0

1 result(s)

Type	Reaction			α	β	γ	T (K)	Formula	Evaluation
ER	$H^+ + e^-$	→	H + Photon	3.50E-12	-7.00E-1	0.00E+0	10-280	Modified Arrhenius equation	

Arrhenius

$$k(T) = \alpha \left(\frac{T}{300\ \mathrm{K}}\right)^{\beta} \exp\left(-\frac{\gamma}{T}\right)$$

A simple example / 2

$$\mathrm{H^+ + e^- \rightarrow H + \gamma}$$

α	β	γ	T (K)
3.50E-12	-7.00E-1	0.00E+0	10-280

Build by KROME

$$\frac{\mathrm{d}n_{\mathrm{H}}}{\mathrm{d}t} = k(T)\, n_{\mathrm{H^+}}\, n_{\mathrm{e^-}}$$

$$k(T) = \alpha \left(\frac{T}{300\,\mathrm{K}}\right)^{\beta} \exp\left(-\frac{\gamma}{T}\right)$$

```
@format: idx,R,R,P,Tmin,Tmax,rate
1, H+, E, H, 10, 280, 3.5e-12 * (Tgas / 3e2)**(-0.7)
```

Service variables

```
@var: T32=Tgas/3e2
1, H+, E, H, 10, 280, 3.5e-12 * T32**(-0.7)
```

speed-up

Custom functions

$$k(T) = \alpha \left(\frac{T}{300\,\mathrm{K}}\right)^{\beta} \exp\left(-\frac{\gamma}{T}\right)$$

```
1, H+, E, H, 10, 280, k_arr(3.5d-12, -0.7d0, 0d0, Tgas)
```

Defined as a F90 function in a dedicated file

(1) $\mathrm{H} + \mathrm{e}^- \rightarrow \mathrm{H}^+ + 2\mathrm{e}^-$ — Janev et al. 1987 (2.1.5)

$$k_1 = \exp[-32.71396786 + 13.536556\ln(T) - 5.73932875\ln(T)^2 + 1.56315498\ln(T)^3 - 0.2877056\ln(T)^4 + 3.48255977\times10^{-2}\times\ln(T)^5 - 2.63197617\times10^{-3}\times\ln(T)^6 + 1.11954395\times10^{-4}\ln(T)^7 - 2.03914985\times10^{-6}\ln(T)^8]\ \mathrm{cm}^3\ \mathrm{s}^{-1}.$$

(2) $\mathrm{H}^+ + \mathrm{e}^- \rightarrow \mathrm{H} + \gamma$ — Our fit to data given by Ferland et al. (1992)

$$k_2 = \exp[-28.6130338 - 0.72411256\ln(T) - 2.02604473\times10^{-2}\ln(T)^2 - 2.38086188\times10^{-3}\ln(T)^3 - 3.21260521\times10^{-4}\ln(T)^4 - 1.42150291\times10^{-5}\ln(T)^5 + 4.98910892\times10^{-6}\ln(T)^6 + 5.75561414\times10^{-7}\ln(T)^7 - 1.85676704\times10^{-8}\ln(T)^8 - 3.07113524\times10^{-9}\ln(T)^9]\ \mathrm{cm}^3\ \mathrm{s}^{-1}.$$

A simple example / 3

$$\mathrm{H} + \mathrm{CR} \rightarrow \mathrm{H}^+ + \mathrm{e}^-$$

$$\frac{\mathrm{d}n_\mathrm{H}}{\mathrm{d}t} = k_1\, n_{\mathrm{H}^+}\, n_{\mathrm{e}^-} - k_2\, n_\mathrm{H}$$

$$k_2 = \alpha\, \zeta$$

Type	Reaction			α	β	γ	UV Field	Formula	Evaluation
CR	H + CR	→	$H^+ + e^-$	4.60E-1	0.00E+0	0.00E+0		Cosmic-ray ionization	?
Phot	H + Photon	→	$H^+ + e^-$	0.00E+0	0.00E+0	3.00E+0	Draine	Photo (Draine)	✓

Preprocessor stage in your network

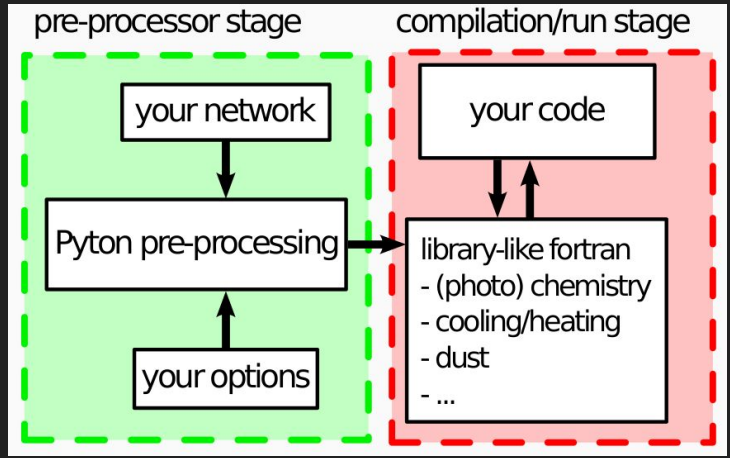
```
@format: idx,R,P,P,rate
@common: user_crate
2, H, H+, E, 0.46 * user_crate
```

model parameter

Change the value at runtime from your code

```
call krome_set_user_crate(1d-15)
```

[1/s]

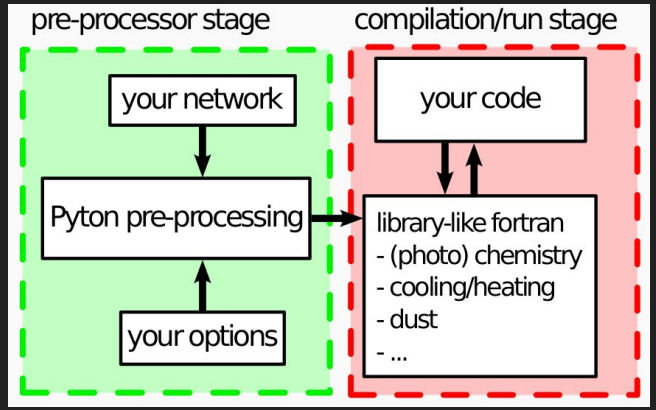


A simple example (network recap)

```
@var: T32=Tgas/3e2

@common: user_crate


@format: idx,R,R,P,Tmin,Tmax,rate

1, H+, E, H, 10, 280, 3.5e-12 * T32**(-0.7)


@format: idx,R,P,P,rate

@CR_start
@storeOnce_start
2, H, H+, E, 0.46 * user_crate
@storeOnce_stop
@CR_stop
```

For CR heating

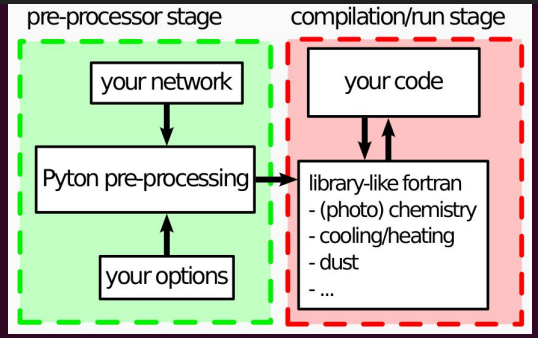
To save evaluations



Pre-processing *simple.ntw* (python)

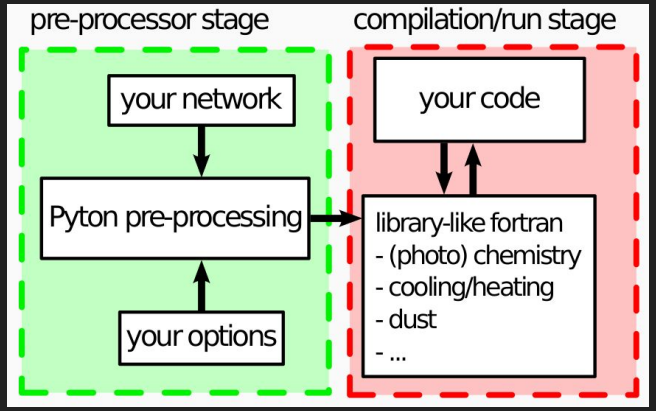
```
tgrassi@multivac:~/repos/krome$ python3 krome -n=simple.ntw
*****************************
      WELCOME TO KROME
*****************************
Any key to ignore q to quit...
Reading from file "simple.ntw"...
copied H.gfe
checking sinks/sources...
checking recombinations...
done!
Enthalpy OK!

ODEs needed: 7
Reactions found: 2
Species found: 3
Species list, reactions, and info saved in build/info.log
Species index initialization for gnuplot in build/species.gps
Heating cooling index init for gnuplot in build/heatcool.gps
Reactions saved in build/reactions.log
ODE partition: [3 atom/mols]  + [1 CR] + [1 PHOT] + [1 Tgas] + [1 dummy] = 7 ODEs
ODEs list: E, H, H+, CR, g, Tgas, dummy

Max number of reactants: 2
Max number of products: 2
```

Network checks



Runtime stage (fortran)

```
program test
  use krome_main !use krome (mandatory)
  use krome_user !use utility (for krome_idx_* constants and others)
  implicit none
  integer,parameter::nsp=krome_nmols !number of species (common)
  real*8::Tgas,dt,x(nsp),spy

  spy = 3.65d2 * 2.4d1 * 3.6d3 !seconds per year

  call krome_init() !init krome (mandatory)

  x(:) = 0d0 !default abundances
  x(krome_idx_H) = 1d0 !hydrogen initial abundance

  Tgas = 270. !gas temperature (K)
  dt = 1d6 * spy !time-step (s)

  call krome_set_user_crate(1d-12)

  !call the solver
  print *, x(krome_idx_H), x(krome_idx_Hj), x(krome_idx_e)
  call krome(x(:), Tgas, dt) !call KROME
  print *, x(krome_idx_H), x(krome_idx_Hj), x(krome_idx_e)

  print *,"Test OK!"

end program test
```



```
tgrassi@multivac:~/repos/krome/build$ make gfortran
gfortran -ffree-line-length-none -w -c opkda2.f -o opkda2.o
gfortran -ffree-line-length-none -w -c opkda1.f -o opkda1.o
gfortran -ffree-line-length-none -w -c opkdmain.f -o opkdmain.o
gfortran -ffree-line-length-none -c krome_commons.f90 -o krome_commons.o
gfortran -ffree-line-length-none -c krome_constants.f90 -o krome_constants.o
gfortran -ffree-line-length-none -c krome_user_commons.f90 -o krome_user_commons.o
gfortran -ffree-line-length-none -c krome_fit.f90 -o krome_fit.o
gfortran -ffree-line-length-none -c krome_getphys.f90 -o krome_getphys.o
gfortran -ffree-line-length-none -c krome_gadiab.f90 -o krome_gadiab.o
gfortran -ffree-line-length-none -c krome_grfuncs.f90 -o krome_grfuncs.o
gfortran -ffree-line-length-none -c krome_phfuncs.f90 -o krome_phfuncs.o
gfortran -ffree-line-length-none -c krome_subs.f90 -o krome_subs.o
gfortran -ffree-line-length-none -c krome_stars.f90 -o krome_stars.o
gfortran -ffree-line-length-none -c krome_dust.f90 -o krome_dust.o
gfortran -ffree-line-length-none -c krome_photo.f90 -o krome_photo.o
gfortran -ffree-line-length-none -c krome_tabs.f90 -o krome_tabs.o
gfortran -ffree-line-length-none -c krome_coolingGH.f90 -o krome_coolingGH.o
gfortran -ffree-line-length-none -c krome_cooling.f90 -o krome_cooling.o
gfortran -ffree-line-length-none -c krome_heating.f90 -o krome_heating.o
gfortran -ffree-line-length-none -c krome_ode.f90 -o krome_ode.o
gfortran -ffree-line-length-none -c krome_user.f90 -o krome_user.o
gfortran -ffree-line-length-none -c krome_reduction.f90 -o krome_reduction.o
gfortran -ffree-line-length-none -c krome.f90 -o krome.o
gfortran -ffree-line-length-none -c test.f90 -o test.o
gfortran opkda2.o opkda1.o opkdmain.o krome_commons.o krome_constants.o krome_user_c
 krome_phfuncs.o krome_subs.o krome_stars.o krome_dust.o krome_photo.o krome_tabs.o
_user.o krome_reduction.o krome.o test.o -o test -ffree-line-length-none
tgrassi@multivac:~/repos/krome/build$ ./test
   1.0000000000000000        0.0000000000000000        0.0000000000000000
  0.70623921291217606       0.29363554124160135       0.29363554124160129
 Test OK!
```

```
!H+ + E -> H
if(Tgas.GE.10d0 .and. Tgas.LT.280d0) then
  k(1) = small + (3.5e-12 &
       * T32**(-0.7))
end if

!H -> H+ + E
k(2) = small + (0.46 * user_crate)
```

```
!E
dn(idx_E) = &
    -k(1)*n(idx_Hj)*n(idx_E) &
    +k(2)*n(idx_H)

!H
dn(idx_H) = &
    +k(1)*n(idx_Hj)*n(idx_E) &
    -k(2)*n(idx_H)

!H+
dn(idx_Hj) = &
    -k(1)*n(idx_Hj)*n(idx_E) &
    +k(2)*n(idx_H)
```

H, H+, e-

Additional examples

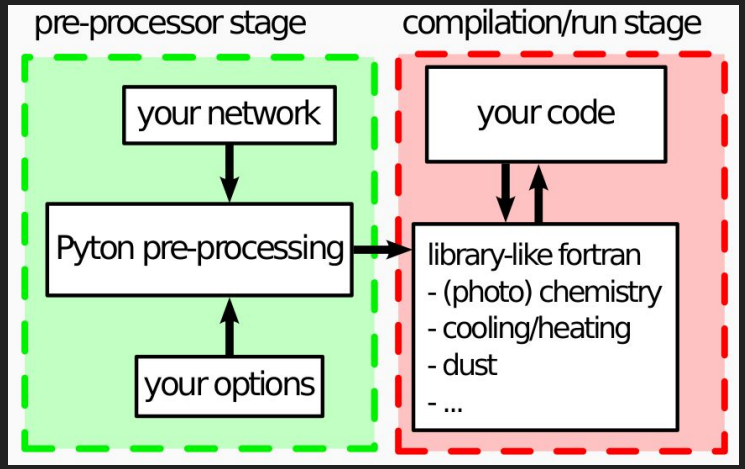
```
#PHOTOREACTIONS THIN WITH Av
@format:idx,R,P,P,rate

@storeOnce_start
# FROM GLOVER+2009, https://arxiv.org/abs/0907.4081
224,H-,H,E,7.1d-7*exp(-.5*user_Av)
225,H2+,H,H+,1.1d-9*exp(-1.9*user_Av)
227,H3+,H2,H+,4.9d-13*exp(-1.8*user_Av)
228,H3+,H2+,H,4.9d-13*exp(-2.3*user_Av)
229,C,C+,E,3.1d-10*exp(-3.*user_Av)
230,C-,C,E,2.4d-7*exp(-.9*user_Av)
231,CH,C,H,8.7d-10*exp(-1.2*user_Av)
232,CH,CH+,E,7.7d-10*exp(-2.8*user_Av)
233,CH+,C,H+,2.6d-10*exp(-2.5*user_Av)
234,CH2,CH,H,7.1d-10*exp(-1.7*user_Av)
235,CH2,CH2+,E,5.9d-10*exp(-2.3*user_Av)
236,CH2+,CH+,H,4.6d-10*exp(-1.7*user_Av)
237,CH3+,CH2+,H,1d-9*exp(-1.7*user_Av)
238,CH3+,CH+,H2,1d-9*exp(-1.7*user_Av)
239,C2,C,C,1.5d-10*exp(-2.1*user_Av)
240,O-,O,E,2.4d-7*exp(-.5*user_Av)
241,OH,O,H,3.7d-10*exp(-1.7*user_Av)
242,OH,OH+,E,1.6d-12*exp(-3.1*user_Av)
243,OH+,O,H+,1d-12*exp(-1.8*user_Av)
244,H2O,OH,H,6d-10*exp(-1.7*user_Av)
245,H2O,H2O+,E,3.2d-11*exp(-3.9*user_Av)
254,O2,O2+,E,5.6d-11*exp(-3.7*user_Av)
255,O2,O,O,7d-10*exp(-1.8*user_Av)
256,CO,C,O,2.d-10*exp(-3.53*user_Av)
@storeOnce_stop
```

```
@noTab_start
#DUST FREEZE-OUT/EVAPORATION
@var:tabTdust = get_table_Tdust(n(:))
@ice:CO,freezeout,krate_stickSi(n(:),idx_CO,tabTdust)
@ice:CO,evaporation,krate_evaporation(n(:),idx_CO,tabTdust)
@ice:H2O,freezeout,krate_stickSi(n(:),idx_H2O,tabTdust)
@ice:H2O,evaporation,krate_evaporation(n(:),idx_H2O,tabTdust)

#NON-THERMAL EVAPORATION, Yield=1e-3, Gnot=1.78 is Draine Flux
@ice:CO,evaporation,krate_nonthermal_evaporation(idx_CO,1.78d0, user_Av, user_crate, 1d-3)

@noTab_stop
```

```
@format:idx,R,R,P,P,P,rate
@var:invTe=1d0/Te
@var:logT=log10(Tgas)
@var:invsqrT=1d0/sqrt(Tgas)
@var:kl11 = 1d1**(-27.029d0+3.801d0*logT-29487d0*invT)
@var:kh11 = 1d1**(-2.729d0-1.75d0*logT-23474d0*invT)
@var:ncr11 = 1d1**(5.0792d0*(1d0-1.23d-5*(Tgas-2d3)))
@var:ax11=1.d0/(1.d0+(Hnuclei/(ncr11+1d-40)))
@noTabNext
11,H2,He,H,H,He,kh11**(1.-ax11)*kl11**ax11
```

Recap

- How to prepare a simple chemical network
- Search into a database / literature
- Pre-process your chemical network
- Evolve a chemical network in time



KROME

better science through chemistry