Microphysics and chemistry in hydrodynamical numerical simulations

Part II: Microphysics

Thursday 29 October, 2015
Gandalf School

Stefano Bovino

Hamburger Sternwarte

Highlights from yesterday's talk

- chemistry is a mess!
- solve the ODEs system is computationally expensive (stiff)
- it requires accurate implicit solvers (high-order, adaptive step-size)
- build a chemical network is not an easy task
  - accuracy of rates
  - availability of rates
  - how to retrieve the rates
- chemistry is strongly coupled with other physical processes
- it's difficult to find a public package for the chemistry which is flexible, accurate, and well optimized

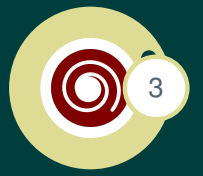
KROME

T. Grassi, **SB**+ MNRAS, 2014

KROME
better science through chemistry

- KROME is a package which helps users to build their own microphysics
- KROME is open source
- KROME is a pre-processor (python)
- KROME is flexible and can be customized
- KROME allow users to choose between efficiency and accuracy
- to embed KROME into an external code requires a simple call
  - `call krome(x(:), Tgas, dt)`
- www.kromepackage.org
- VERY IMPORTANT:
  - we have implemented an auto-generator of networks
  - just give the species, types of reactions
  - But I personally, strongly discourage this!

How to get KROME
via bitbucket




```
$ git clone https://bitbucket.org/tgrassi/krome.git
Cloning into 'krome'...
remote: Counting objects: 4193, done.
remote: Compressing objects: 100% (2063/2063),
    done.
remote: Total 4193 (delta 2675), reused 3278
    (delta 2094)
Receiving objects: 100% (4193/4193), 16.81 MiB |
    673.00 KiB/s, done.
Resolving deltas: 100% (2675/2675), done.
Checking connectivity... done.
```

- you will generate a folder named "krome"

Explore KROME
through the files


```
$ ls
README.md       data              outtest.md5
alltest.py      gpl-3.0.txt       patches
argparse.py     krome             solver
build           kromelib.py       src
changelog.txt   kromeobj.py       tests
clean           networks          tools
custom.dat      options_example   wizard
```

- folders
- python files-the core of the package
- executable-pre-processor
- other files (e.g. clean to clean up your working dir)

Tabula rasa

Forget that KROME exists

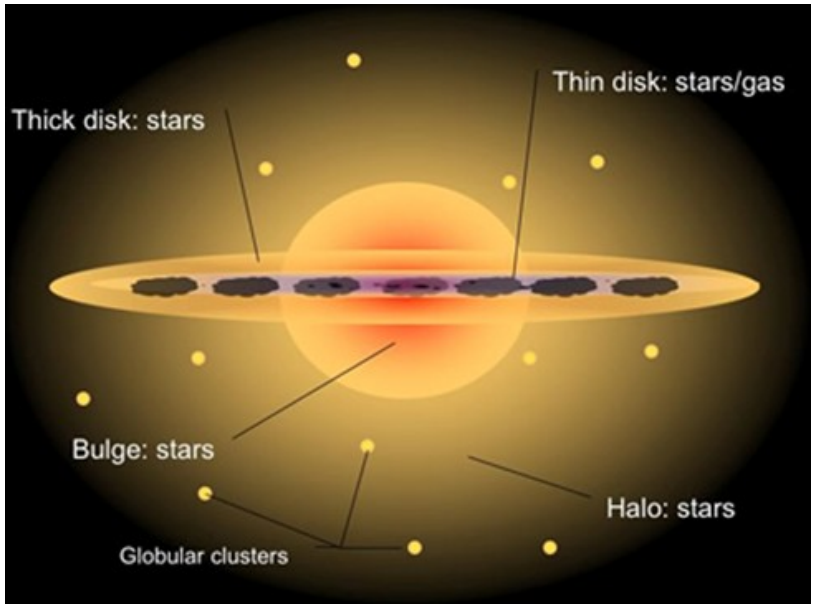
Let’s start with a practical example

simulating an isolated galaxy

Let's start with a practical example (cont'd)
simulating an isolated galaxy

Physical/Hydro parameters:

- Isolated disc galaxy of 4.4 kpc at $z = 3$
- Composition:
  - NFW DM halo
  - stellar bulge, $M_{Bulge} = 1.8 \times 10^9 M_\odot$
  - stellar and gaseous disc, $M_{disc} = 8.8 \times 10^9 M_\odot$
- Virial mass $M_{vir} = 2.2 \times 10^{11} M_\odot$
- Disc gas fraction = 30%
- Disc scale length 1.1 kpc
- Kroupa IMF, 2Gyr-old stars
- Star formation
- Feedback: SNe type II, Ia, wind
- Some Radiative Transfer!
- Metal enrichment by SNe



Let's start with a practical example (cont'd)
simulating an isolated galaxy

- Goal: follow the atomic-to-molecular transition
- or WORST, obtain information on CII abundance

- Do I need chemistry and microphysics?
  - YES, cooling affects the star formation rate (indirectly)
  - H_2 model to know when we reach the fully molecular stage
  - ADVANCED: need a network for CII
- What else?
  - Heating, Photochemistry, Dust...

Typical physical conditions:

- $10^{-8} \leq n_{tot} \leq 10^{4}$ cm^{-3}
- $10 \leq T \leq 10^{8}$ K

Chemical network
first you want to solve kinetics

H_2 network we need:

- 9 species + $\sim$ 40 reactions (including photochemistry)
  - if you need CII, you will end with min. 16 species and $\sim$ 60-70 reactions, without including molecules like CO

Boring/tricky/dangerous?

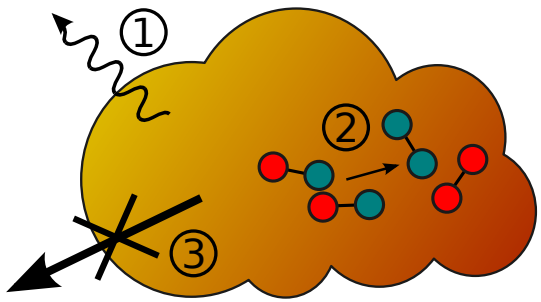
- write down the rate equations (hard coding)
- if you want to remove or add a reaction becomes boring
- for large network almost impossible
- develop a solver or set the parameters for an existing solver

If you are lucky there is some chemistry in your hydro-code already implemented, suitable for your application (not so common)

Cooling
mechanism

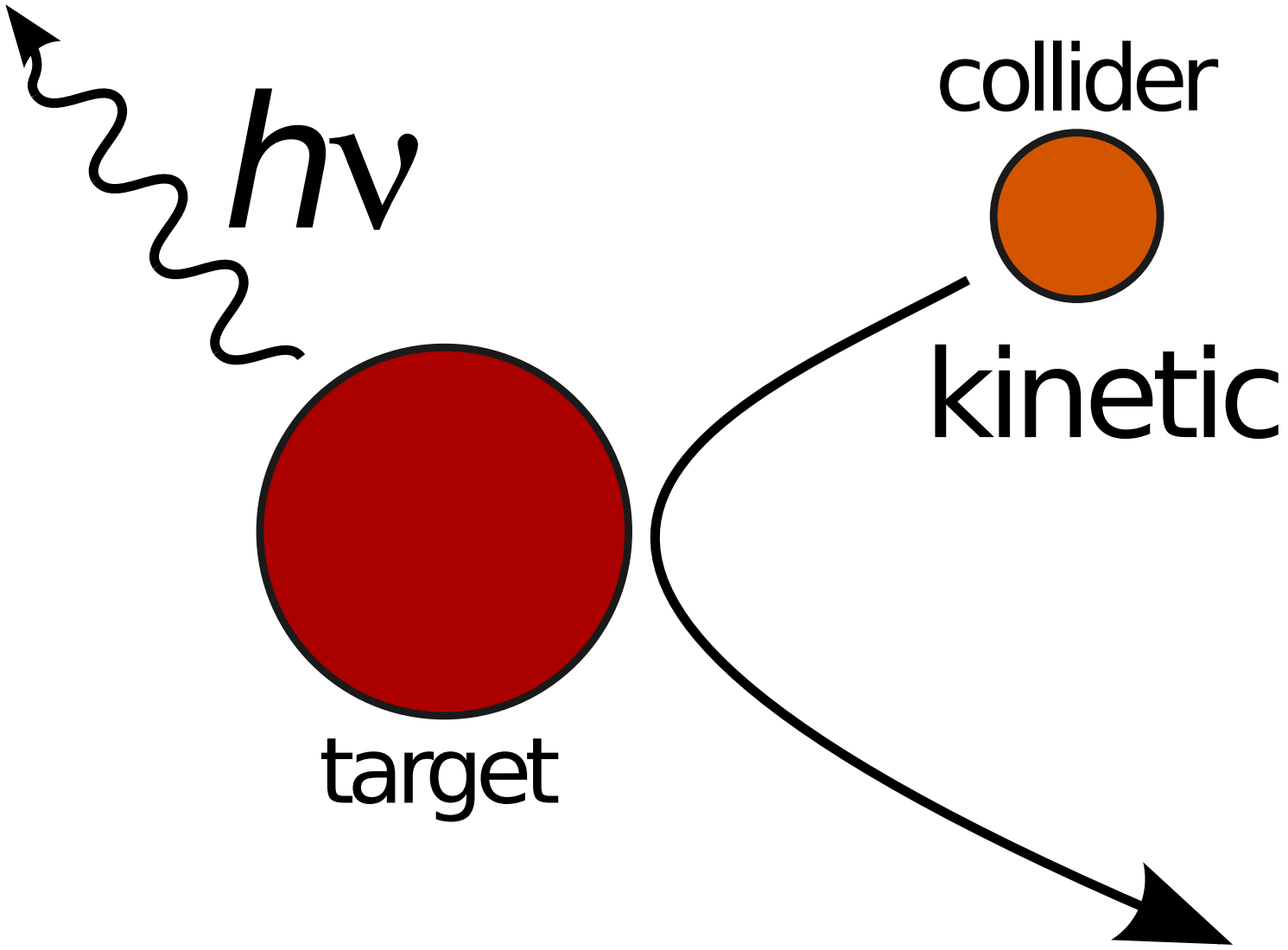
convert kinetic energy in something else!

1. radiative loss
2. endothermic reactions
3. gas flows
4. gas-dust interaction



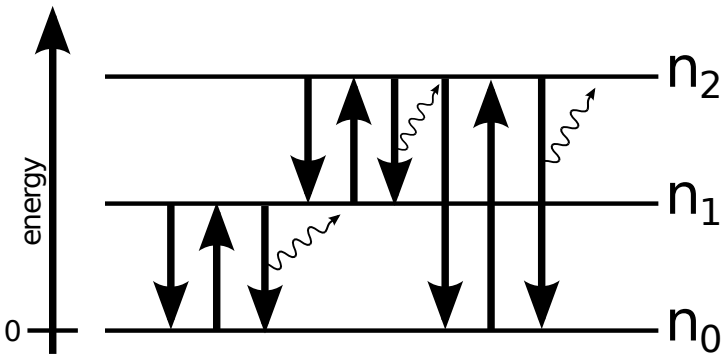
Cooling example: collisional

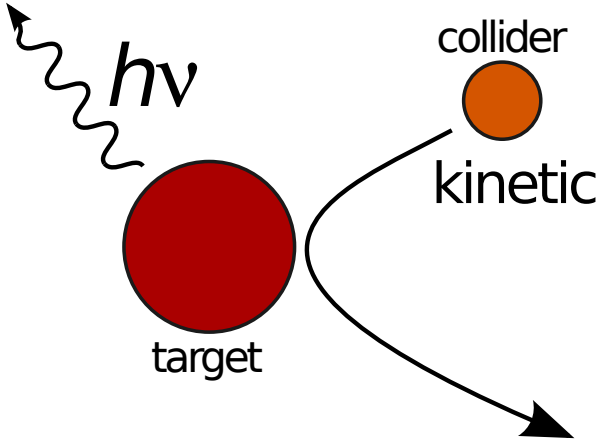
a prototype



Cooling example: collisional

very messy!





general expression (N levels)

$$\dot{n}_i = n_i \left(\sum_{j<i} A_{ij} + \sum_{i \neq j} \sum_k n_{ck} C_{ij}^{(k)} \right) + \left(\sum_{j>i} n_j A_{ij} + \sum_{i \neq j} n_j \sum_k n_{ck} C_{ji}^{(k)} \right)$$

$$\Lambda(n, T) = \sum_i n_i \sum_{j<i} \Delta E_{ij} A_{ji}$$

How to solve - Multilevel system (general)

Full Matrix

$$\begin{pmatrix} 1 & \cdots & 1 \\ M_{1,1} & \cdots & M_{1,N-1} \\ \vdots & \ddots & \vdots \\ M_{N-1,1} & \cdots & M_{N-1,N-1} \end{pmatrix} \times \begin{pmatrix} n_1 & \cdots & n_{N-1} \end{pmatrix} = \begin{pmatrix} n_{\mathrm{tot}} \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

where

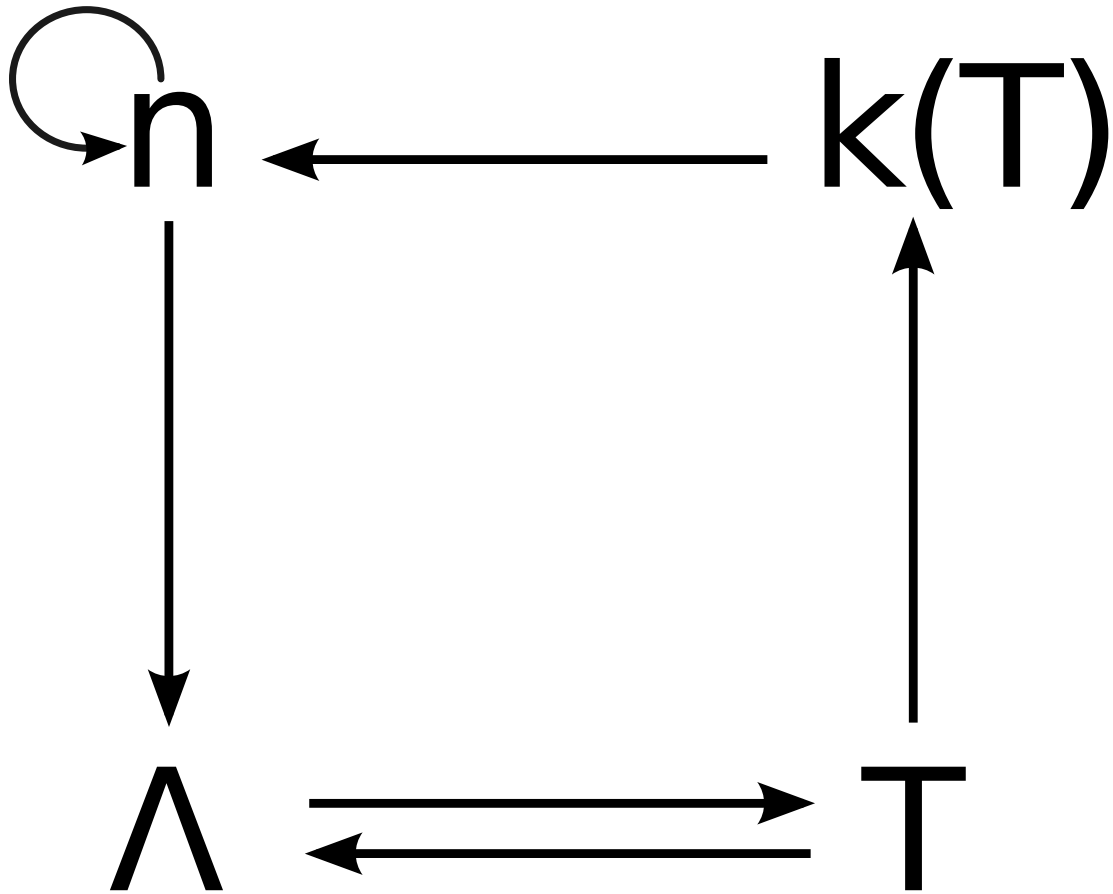
$$M_{l,m} = g_{l,m}(A_{ij}, C_{ij}, \{n_k\})$$

Linear solver

- 2 levels: analytic (fast)
- 3 levels: analytic (fast)
- $>$ 3 levels: LAPACK (fast for $N \gg 3$)

Cooling problems!

$\dot{n}_i$ and $\dot{T}$ must be solved together



Cooling - Necessary atomic data

E_i - Energy of the ith level

- Needed to get $\Delta E_{ij} = E_i - E_j$ (erg or K)
- Easy to retrieve (e.g. NIST, literature), accurate
- Constants

A_{ij} - Einstein's coefficients

- Probability of spontaneous radiative de-excitations (1/s)
- Easy to retrieve (e.g. NIST, literature), accurate
- Constants

$C_{ij}^{(k)}(T)$ - collisional (de)excitation rate coefficient

- Not-so-easy to retrieve (e.g. database and literature search), accuracy?
- Temperature dependence ($\rightarrow$ functions, $\rightarrow$ limits)
- Collider dependent $^{(k)}$ (CI-H_2, CI-H, CI-e)

Problems - Recap

Recap on problems

- Search for atomic data
- How to solve linear systems? (many caveats not shown here)
- Convert messy equations into efficient code (e.g. tables for $C_{ij}^{(k)}$)
- Avoid typos in atomic data (less trivial than expected)
- Adding a collider later could be troublesome

Cooling - simple approach
not always available

What about fitting functions (or cooling eq/non-eq tables)?

- usually functions of T and of the chemical composition
- OK when parameters are independent
- PRO: very fast (depends)
- CON: less accurate (depends)
- CON: not always valid (time-scales)
- CON: difficult to update with new atomic data (re-build machinery)

Examples

- H_2 from S. Glover (colliders H, H^+, e^-, H_2, He)
- Metal cooling tables from Oppenheimer et al. 2013

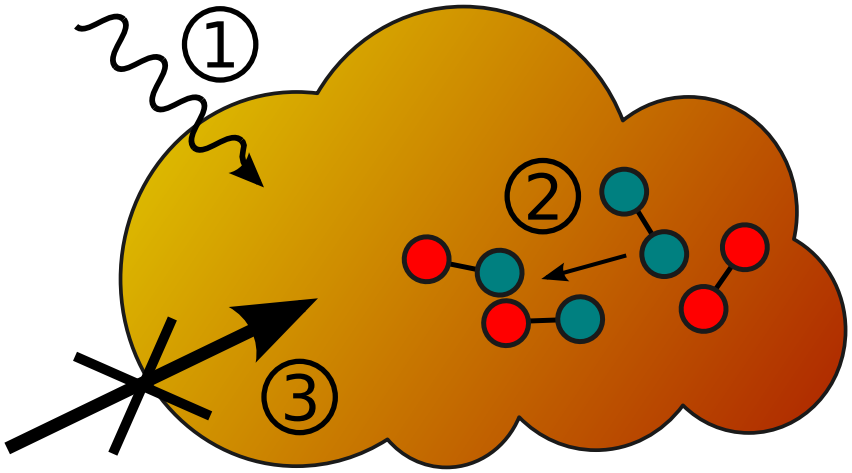
Cooling processes
overview

Going back to your galaxy simulation!

- atomic cooling (e.g. Ly-alpha cooling by H)
- molecular cooling (H_2)
- chemical cooling (endothermic reactions, formation of H_2)
- dust-gas thermal cooling
- metal cooling (e.g. CI, CII, OI)

- simple fit
- on the fly
- tables (eq. or non-eq)

Heating Mechanisms



Template process

something else $\rightarrow$ kinetic energy

Interesting processes

1. photochemistry
2. exothermic reactions
3. gas compression

Heating Mechanisms (cont'd)

There are different ways to heat the gas in the ISM, i.e. to transfer energy into the gaseous medium

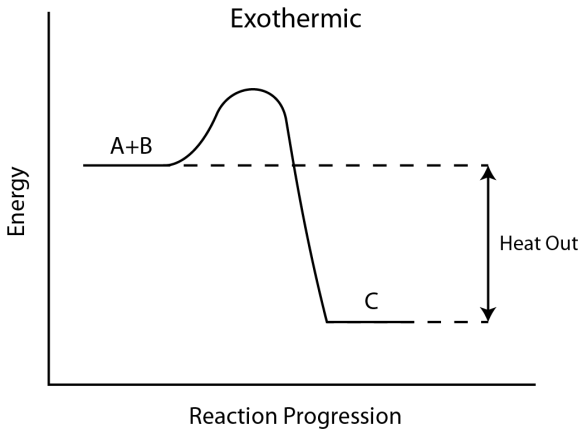
- via photons/electrons
- via chemical reactions
- via dust grains
- via dynamical processes

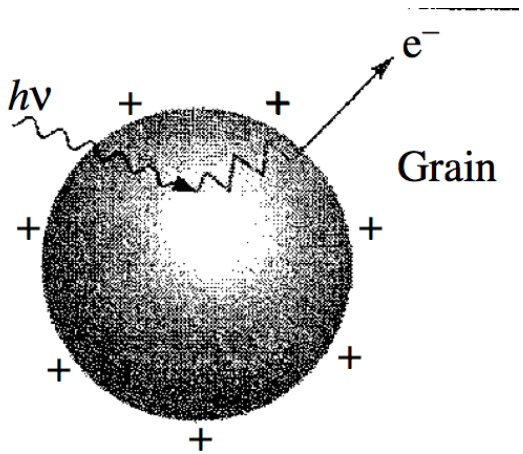
Main processes:

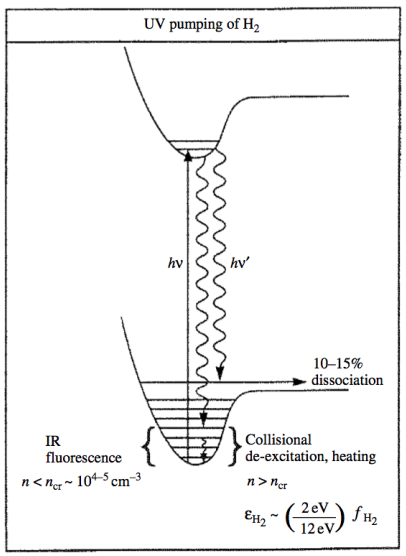
- elastic collisions
- collisional de-excitations

Heating processes overview
for our hypothetical galaxy simulation

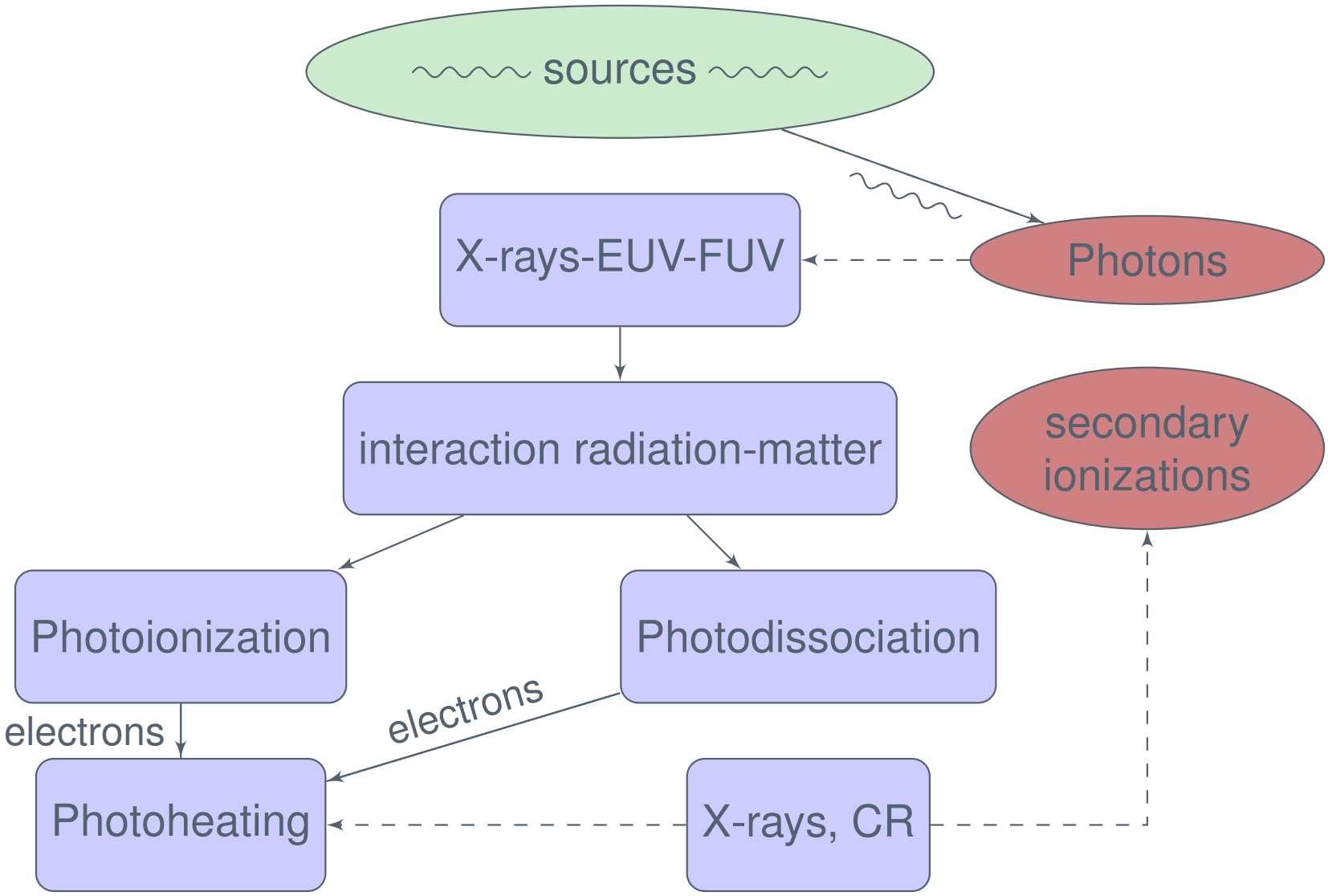
- photo-heating (e.g. H, He ionizations, H_2 photodissociation)
- H_2 UV pumping (Solomon process, LW bands)
- photo-electric heating by dust
  - $E > 6$ eV
  - electron detachment
  - Coulomb losses $\rightarrow$ gas heating
- chemical heating



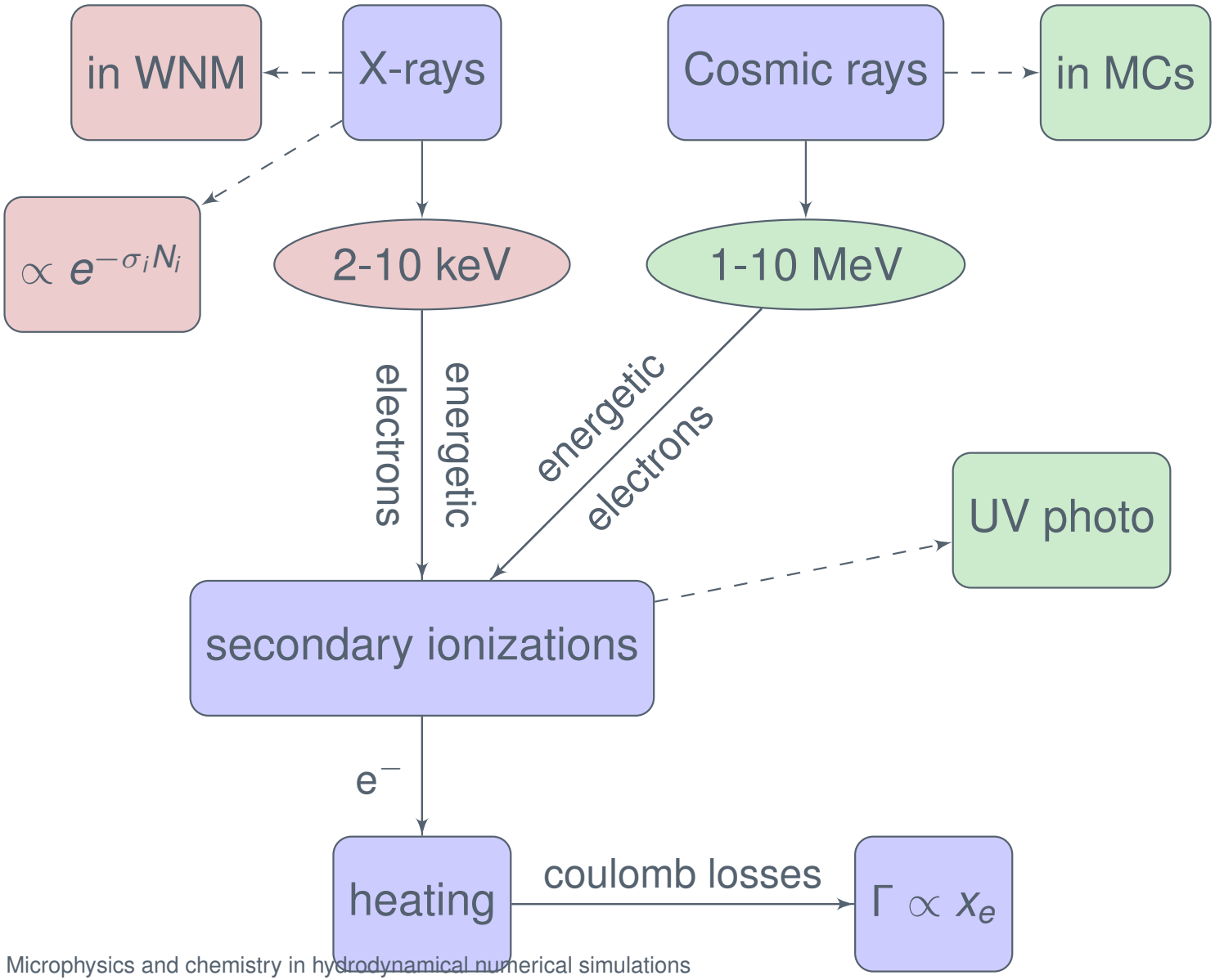




Interaction radiation-matter
a schematic summary



Secondary ionizations
some more information



Photochemistry - Equations first

Photochemistry process

$$\mathrm{H} + \gamma \xrightarrow{\sigma_0(E)} \mathrm{H}^+ + e^- \quad (1)$$

$$\mathrm{OH} + \gamma \xrightarrow{\sigma_1(E)} \mathrm{OH}^+ + e^- \quad (2)$$

$$\mathrm{OH} + \gamma \xrightarrow{\sigma_2(E)} \mathrm{O} + \mathrm{H} \quad (3)$$

Photochemistry equations (thin)

$$k_{ph} = \frac{4\pi}{h} \int_{E_0}^{\infty} \frac{I(E)\sigma(E)}{E} dE \quad (4)$$

$$H_{ph} = \frac{4\pi}{h} \int_{E_0}^{\infty} \frac{I(E)\sigma(E)}{E} (E - E_0)\eta(E) dE \quad (5)$$

$$\Gamma^i_{ph} = H^i_{ph} n_i \quad (6)$$

Photochemistry - Equations first

Photochemistry equations (thin)

$$k_{ph} = \frac{4\pi}{h}\int_{E_0}^{\infty}\frac{I(E)\sigma(E)}{E}dE \tag{7}$$

$$H_{ph} = \frac{4\pi}{h}\int_{E_0}^{\infty}\frac{I(E)\sigma(E)}{E}(E-E_0)\eta(E)dE \tag{8}$$

$$k_{ph} = \left[\mathrm{s}^{-1}\right] \tag{9}$$

$$H_{ph} = \left[\frac{\mathrm{erg}}{\mathrm{s}}\right] \tag{10}$$

$$E = [\mathrm{eV}] \tag{11}$$

$$I(E) = \left[\frac{\mathrm{eV}}{\mathrm{cm}^2\,\mathrm{s}\,\mathrm{Hz}\,\mathrm{sr}}\right] = \left[\frac{\mathrm{eV}}{\mathrm{cm}^2\,\mathrm{sr}}\right] \tag{12}$$

$$\sigma(E) = \left[\mathrm{cm}^2\right] \tag{13}$$

Photochemistry - bins

$$k_{ph} = \frac{4\pi}{h}\int_{E_0}^{\infty}\frac{I(E)\sigma(E)}{E}dE \quad (14)$$

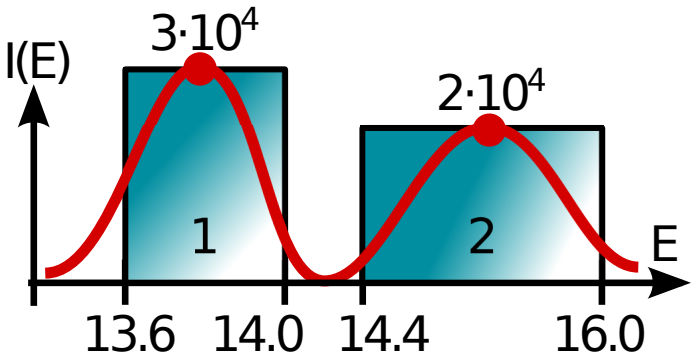
$$H_{ph} = \frac{4\pi}{h}\int_{E_0}^{\infty}\frac{I(E)\sigma(E)}{E}(E-E_0)\eta(E)dE \quad (15)$$

Discretization by bins

$$k_{ph} = \frac{4\pi}{h}\sum_{i=1}^{N_{bins}}\frac{I_i\sigma_i}{\langle E_i\rangle}(E_i^{right}-E_i^{left}) \qquad \langle E_i\rangle > E_0$$

$$H_{ph} = \frac{4\pi}{h}\sum_{i=1}^{N_{bins}}\frac{I_i\sigma_i}{\langle E_i\rangle}(E_i^{right}-E_i^{left})(\langle E_i\rangle - E_0)$$

$$\langle E_i\rangle = \frac{E_i^{right}+E_i^{left}}{2}$$

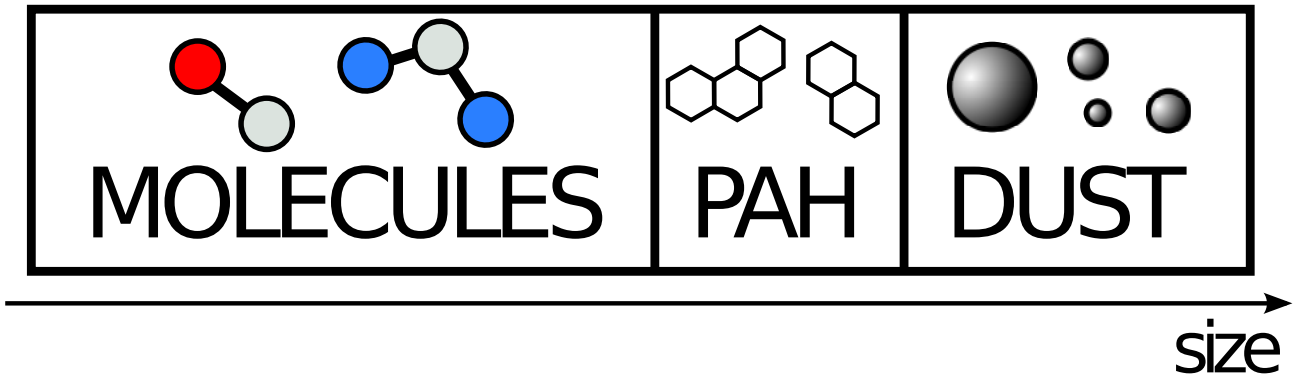


Photochemistry
it's a mess

Problems

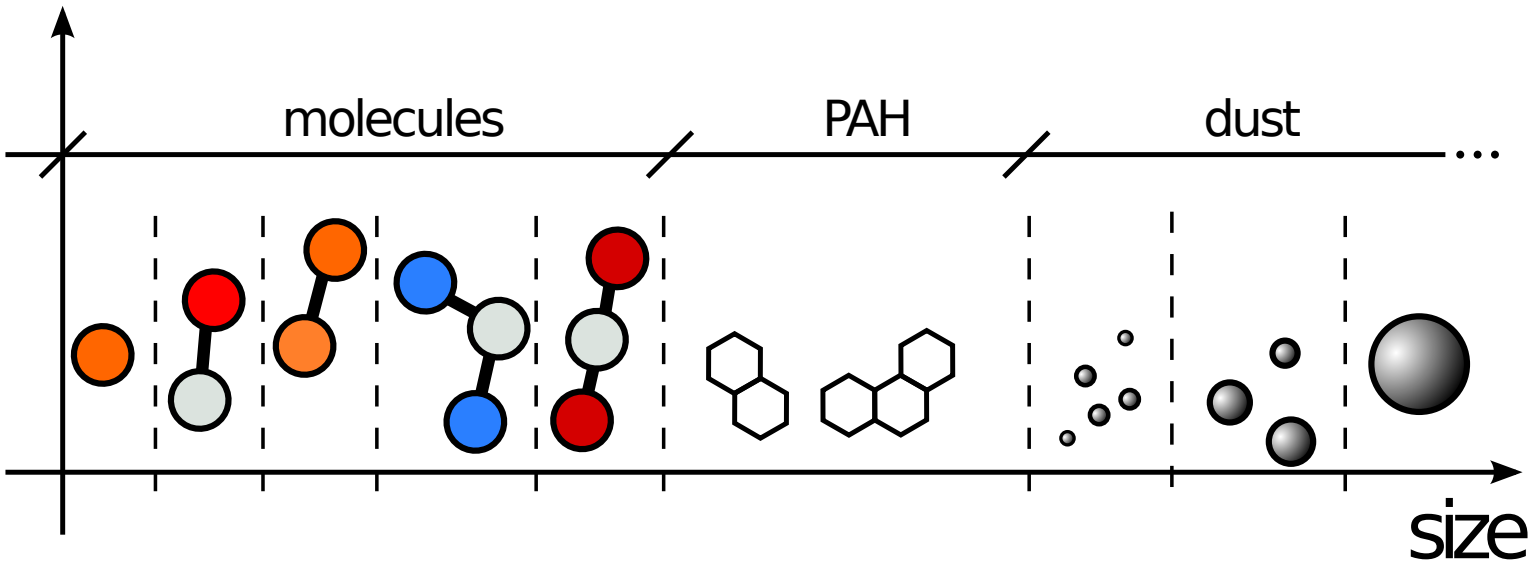
- collect cross-sections
- coupling with chemistry, dust, and thermal evolution
- coupling with radiative transfer scheme (consistently)

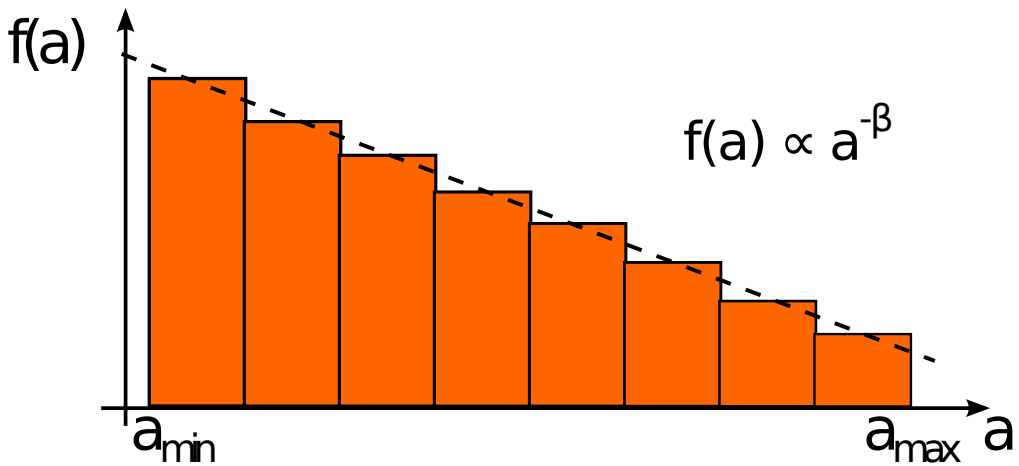
Dust physics
(see also Richard's talk)



- different compositions and size
- different optical properties
- interact with the gas
- interact with radiation
- catalyze formation of molecules
- generally speaking $\rightarrow$ quite messy!

Dust physics
how to model





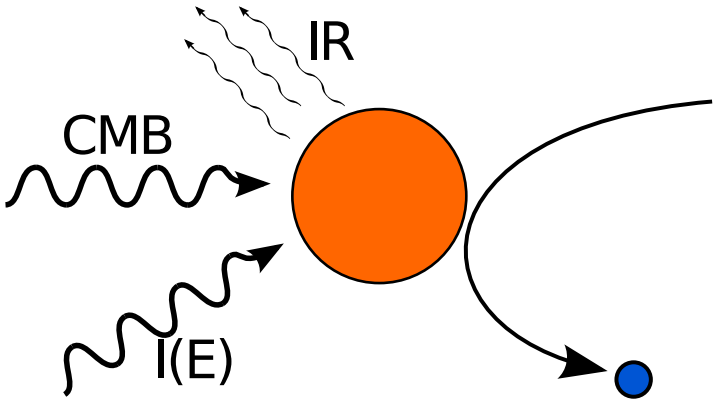
β = 3.5, Mathis, Rumpl, and Nordsieck 1977

Dust physics
how to model

$$\Lambda_{g\to d}(a, T_d) = 2\pi a^2 n_g n_d v_g k_B (T_g - T_d)$$

Dust formation and destruction depends on

- the grain size ($\propto \pi a^2$)
- dust and gas temperature
- sticking coefficient



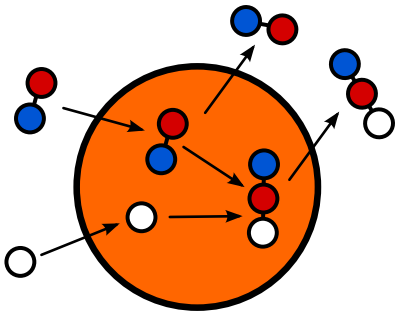
Gas-dust-radiation interaction

$$\Gamma_{\rm em} = \Lambda_{\rm g\to d} + \Gamma_{\rm CMB} + \Gamma_{\rm abs}$$

$$\int Q(a,E)B[E,T_d(a)]dE = \int Q(a,E)I(E)dE + \int Q(a,E)B(E,T_{CMB})dE + \Lambda_{g\to d}$$

To be solved bin-by-bin to obtain the dust temperature!!!

and H_2 formation on dust
and heating



$$H + H + GR \rightarrow H_2 + GR + 4.48\ eV$$

- most relevant process
- energy distributed as following
  - 0.2 eV as kinetic energy
  - 4.2 eV in roto-vibrational state of H_2
  - heating of grain negligible

$$\Gamma^d_{H_2} = R_f(0.2 + 4.2\epsilon)n_{tot}n_H$$

WARNING: different rates available

H_2 formation on dust
and heating

$$
\begin{aligned}
^{1}R_f &= 3 \times 10^{-17} n_{\rm tot} n_{\rm H} \\
^{2}R_f &= 3 \times 10^{-17} Z/Z_{\odot} C_{\rho} n_{\rm tot} n_{\rm H} \\
^{3}R_f &= 3 \times 10^{-17} \frac{T_2^{0.5} f_a}{1 + 0.4(T_2 + T_{d_2})^{0.5} + 0.2 T_2 + 0.09 T} n_{\rm tot} n_{\rm H} \\
^{4}R_d &= 0.5 v_g \pi a^2 \varepsilon_{\rm H_2} S(T, T_d) n_{\rm H} n_d
\end{aligned}
$$

[1] Jura 1975
[2] Gnedin+ 2009
[3] Hollenbach& McKee 1979
[4] Cazaux & Tielens 2004, state-of-the-art

Final ingredients to cook our galaxy

RECIPE

- minimal H_2 chemical network (easiest part)
- cooling
  - metal cooling
  - H_2 cooling
  - dust cooling
  - atomic cooling
  - chemical cooling
- heating
  - photo-heating
  - photo-electric heating by dust
  - chemical heating
- dust
  - dust-gas interaction (cooling)
  - most important: H_2 catalysis on dust
- icing on the cake $\rightarrow$ shielding (by molecules and dust)!

How this is in KROME

Options file: my_options

```
-n networks/react_galax[5]
-heating CHEM,PHOTO,PHOTODUSTNET,H2PUMPING
-cooling ATOMIC,CHEM,H2,Z_EXTENDED
-conserve
-photoBins=100
-dustTabs=H2,COOL,HM2012
-compact
```

Run KROME, effective python pre-processor

```
./krome -options=my_options
```

Build folder

In `build/` you find the F90 routines to be coupled with your hydro-code

[5] the network is already part of the package but users should carefully check that this is updated; KROME is not a network provider.

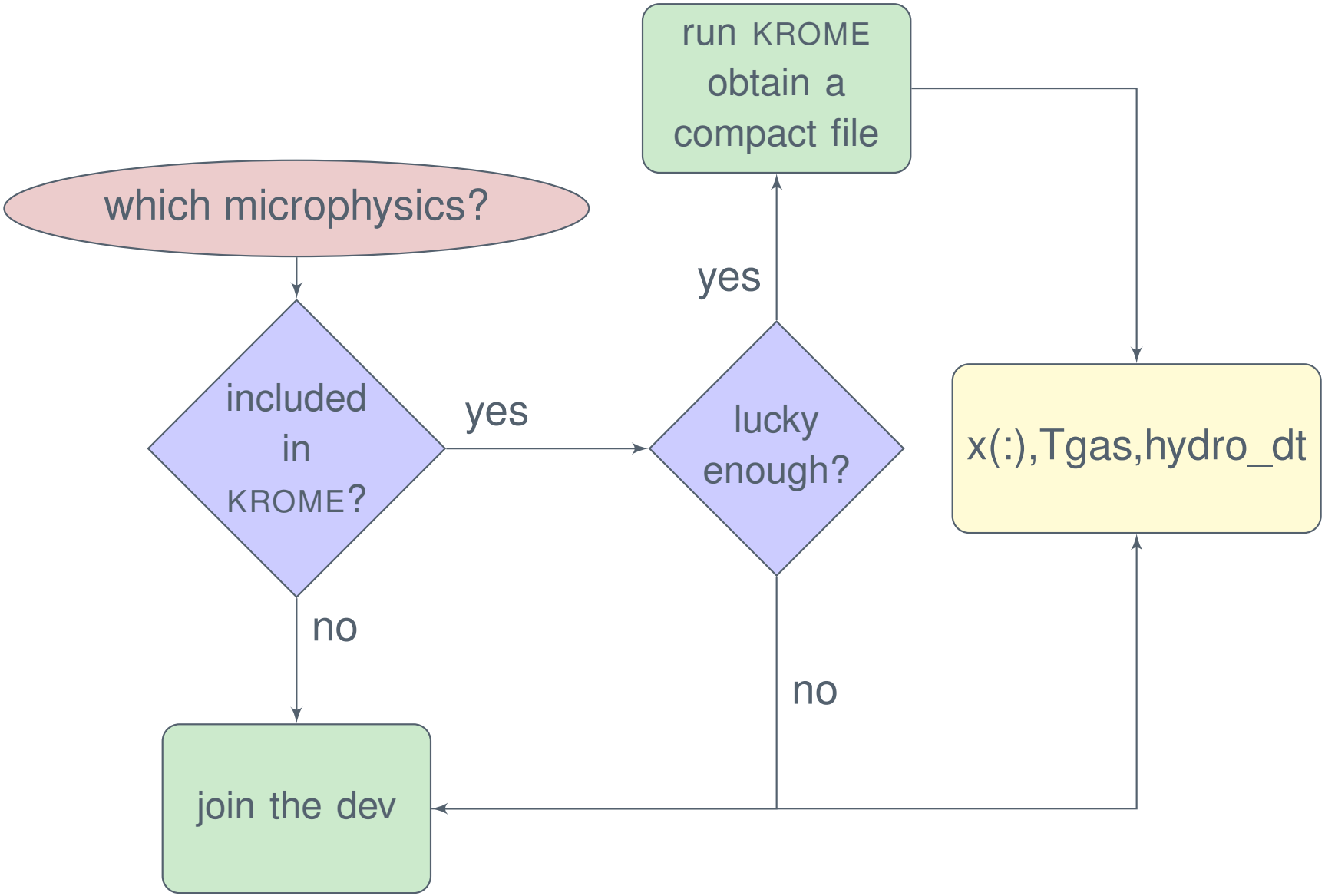
Look into the modules
krome_ode module


```
k(:) = coe_tab(n(:)) !compute coefficients

   !H2+
   dn(1) = &
      -k(1)*n(idx_H2j)*n(idx_Hk) &
      -k(2)*n(idx_H2j)*n(idx_H) &
      +k(3)*n(idx_Hk)*n(idx_Hj)

   !H-
   dn(2) = &
      -k(1)*n(idx_H2j)*n(idx_Hk) &
      -k(3)*n(idx_Hk)*n(idx_Hj)

   !H2
   dn(3) = &
      +k(1)*n(idx_H2j)*n(idx_Hk) &
      +k(2)*n(idx_H2j)*n(idx_H)
```

Look into the modules
krome_main module


```
!*****************************
   !init DLSODES (see DLSODES manual)
   neq = nspec !number of eqns
   liw = size(iwork)
   lrw = size(rwork)
   iwork(:) = 0
   rwork(:) = 0.d0
   itol = 4 !both tolerances are arrays
   rtol(:) = 1.000000d-04 !relative tolerance
   atol(:) = 1.000000d-20 !absolute tolerance
   icount_max = 100 !maximum number of iterations
  itask = 1
   iopt = 0

   !MF=
   ! = 222 internal-generated JAC and sparsity
   ! = 121 user-provided JAC and internal generated sparsity
   ! = 22 internal-generated JAC but sparsity user-provided
   ! = 21 user-provided JAC and sparsity
   MF = 222
   !end init DLSODES
```

A quick summary
on how to include KROME in your simulations

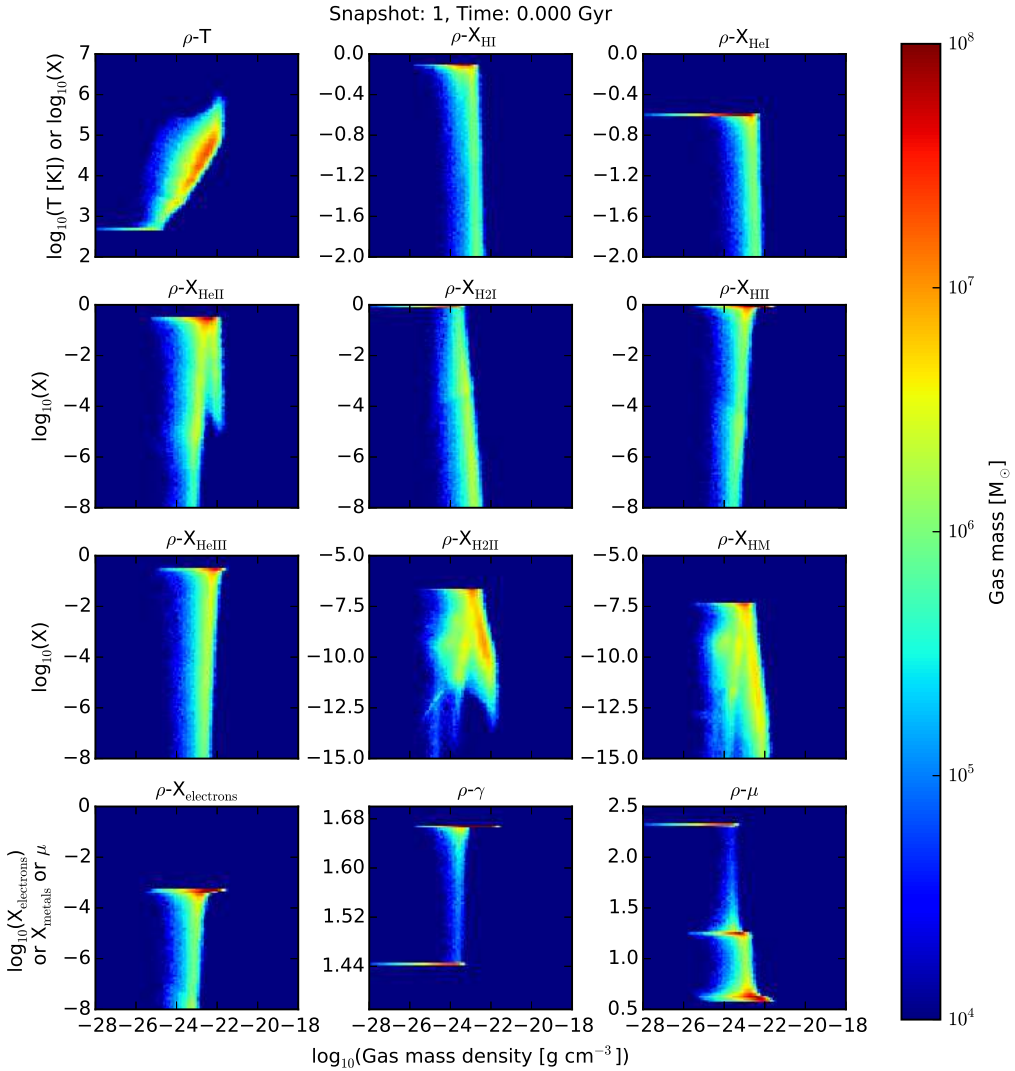


A quick summary
on how to include KROME in your simulations

- enable options you need
- run KROME
- inlcude the krome modules
  - use krome
  - use krome_user
- call krome_init()
  - once and for all
- be sure to have initialized the species (make use of krome_nmols and other common arrays)
- change your Makefile accordingly
  - solver files (opkd*.F)
  - krome routine (`krome_all`)

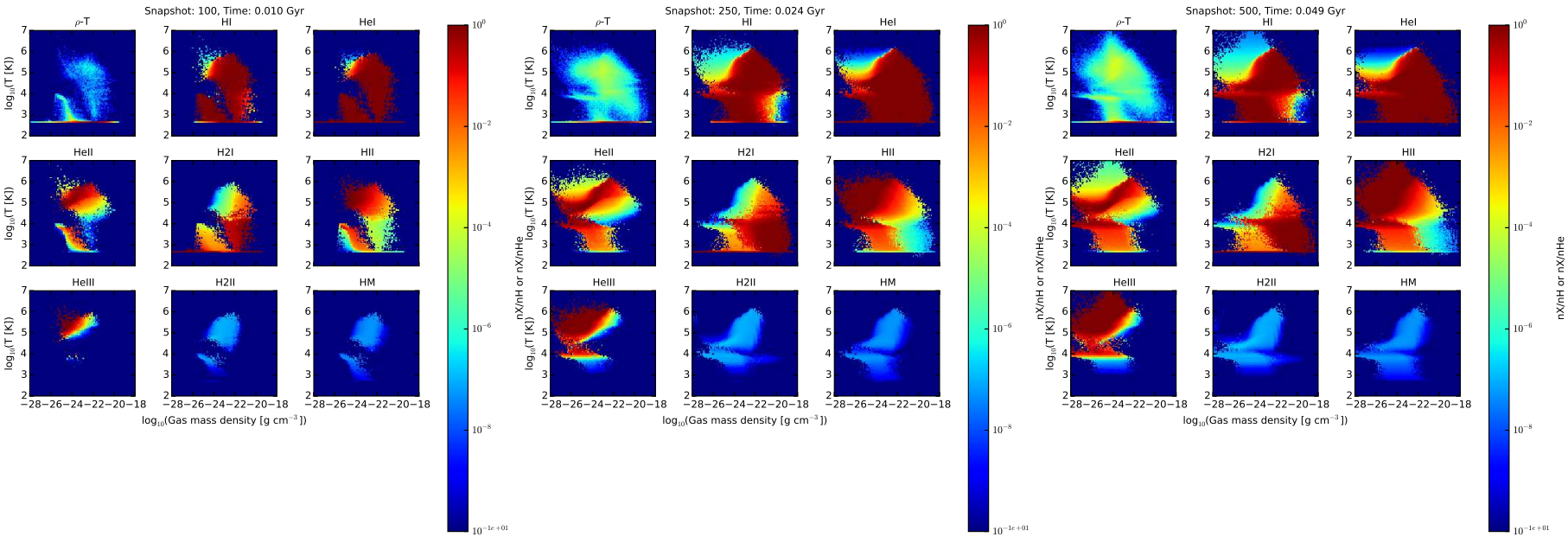
Very preliminary test
run GASOLINE (SPH) + KROME

Initial conditions (no UV background)



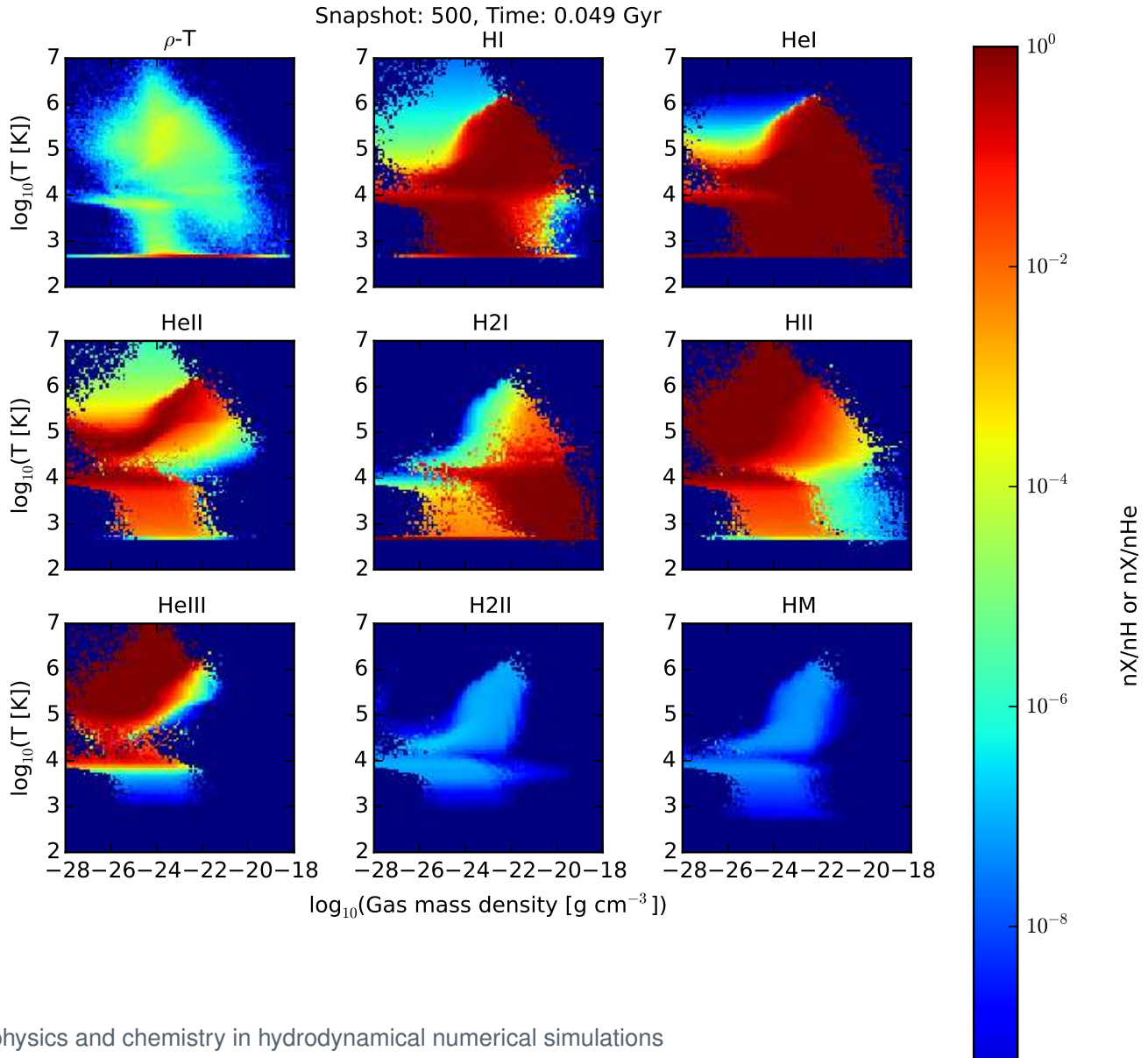
Very preliminary test

Evolution, first 50 Myr (relaxation)



Very preliminary test

run GASOLINE (SPH) + KROME



Final remarks

- modelling astrophysical environments is not an easy task
- chemistry in most cases is out of equilibrium
- chemistry/microphysics is CPU demanding but necessary
- KROME helps users to build a proper chemical model for astrophysical applications
  - open-source
  - in continuous development
  - extended based on the users needs
  - comprehensive and accurate microphysics included
  - flexible
- chemistry is strongly coupled with hydro and radiation

If you are interested in KROME

- join the KROME community
  - `www.kromepackage.org`
  - `http://bitbucket.org/tgrassi/krome`
- participate to one of the KROME schools ($\sim$ organized yearly)
  - 1st in Göttingen October 2014
  - 2nd in Copenhagen July 2015
  - 3th TBD (Florence, Italy?) 2016 or 2017
- contact me or Tommaso
  - stefano.bovino@uni-hamburg.de
  - tommaso.grassi@gmail.com

Thank you for your attention!

DFG Deutsche Forschungsgemeinschaft