Cooling processes in KROME: an overview

Tommaso Grassi
tgrassi@nbi.dk

Centre for Star and Planet Formation

July 2015

Aims of this talk

1. Understand that cooling processes are complicated
2. Realize that KROME saves your day

Chemical network = Cauchy's problem:

$$\frac{\mathrm{d}n_i}{\mathrm{d}t} = \overbrace{\sum_{lm} k_{lm}(T)n_l(t)n_m(t)}^{formation} - \overbrace{n_i(t)\sum_j k_{ij}(T)n_j(t)}^{destruction} \qquad [\times N]$$

$$J_{ij} = \frac{\partial^2 n_i}{\partial t \partial n_j} \qquad [N \times N]$$

- $n_i(t=0) = \hat{n}_i$
- $\sum_i n_i(t)\, m_i = \mathrm{const}$

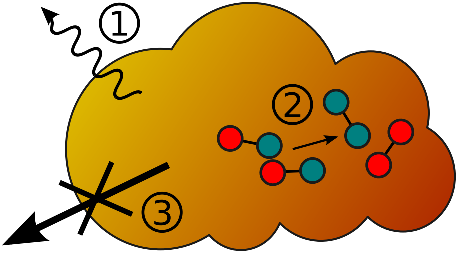
Expanding Cauchy

- $k_1(T)$ is a function of gas temperature!
- Hydro codes need temperature variation from chemistry/microphysics

$$\frac{\mathrm{d}T}{\mathrm{d}t} = f(?) \quad (1)$$

$$\frac{\partial^2 T}{\partial t \partial n_i} = ????? \quad (2)$$

- What is the coupling between $\mathrm{d}T/\mathrm{d}t$ and T?
- What is the coupling between $\mathrm{d}T/\mathrm{d}t$ and n_i?

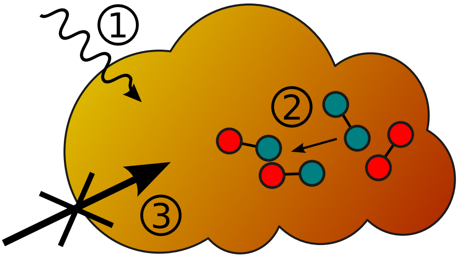


Template process

kinetic energy → something else

Interesting processes

1. radiative loss (thermal and non-thermal)
2. endothermic reactions
3. ~~gas expansion~~

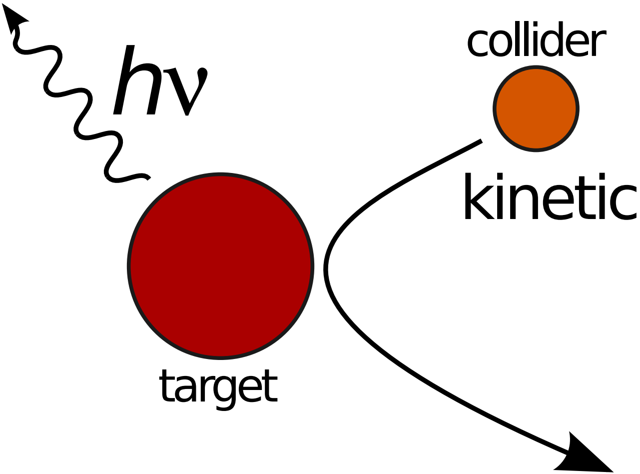


Template process

something else → kinetic energy

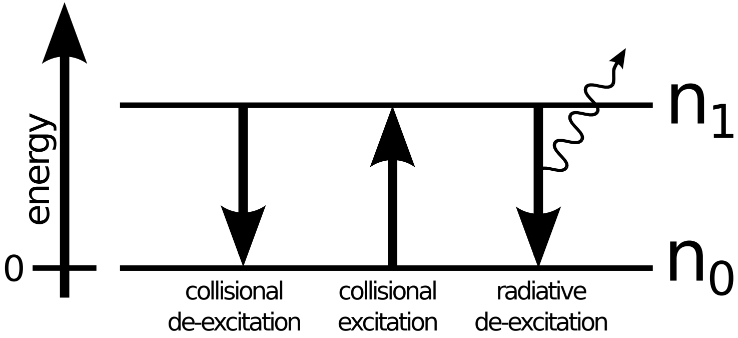
Interesting processes (see Stefano's talk)

1. photochemistry
2. exothermic reactions
3. ~~gas compression~~



Prototype: metal atom (e.g. carbon) excited by gas particle collisions (e.g. H)

KROME Bootcamp 2015 - Two-level system



Two-levels ODE system

$$\begin{cases} \dot{n}_0 &= -n_0 n_c C_{01} + n_1 A_{10} + n_1 n_c C_{10} \\ \dot{n}_1 &= n_0 n_c C_{01} - n_1 \left(A_{10} + n_c C_{10}\right) \end{cases}$$

Looking for steady state

$$\begin{cases} 0 = -n_0 n_c C_{01} + n_1 \left(A_{10} + n_c C_{10}\right) \\ 0 = n_0 n_c C_{01} - n_1 \left(A_{10} + n_c C_{10}\right) \end{cases}$$

Looking for steady state

$$\begin{cases} 0 = -n_0 n_c C_{01} + n_1 \left(A_{10} + n_c C_{10}\right) \\ 0 = n_0 n_c C_{01} - n_1 \left(A_{10} + n_c C_{10}\right) \end{cases}$$

Looking for steady state

$$\begin{cases} 0 = -n_0 n_c C_{01} + n_1 \left(A_{10} + n_c C_{10}\right) \\ n_{\rm tot} = n_0 + n_1 \end{cases}$$

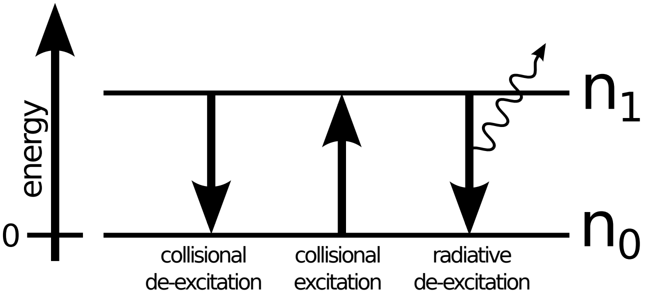
Looking for steady state

$$\begin{cases} 0 = -n_0 n_c C_{01} + n_1 \left(A_{10} + n_c C_{10}\right) \\ n_{\rm tot} = n_0 + n_1 \end{cases}$$

System solution

$$\begin{cases} n_1 = n_{\rm tot} \frac{C_{01} n_c}{(C_{01}+C_{10}) n_c + A_{10}} \\ n_0 = n_{\rm tot} - n_1 \end{cases}$$

good news: $\propto n_{\rm tot}$
bad news: $\propto \phi(n_c)$



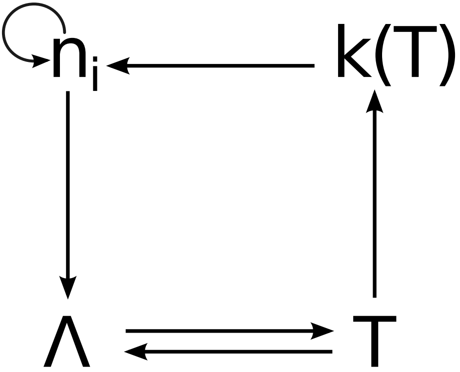
Total cooling

$$\Lambda_{\mathrm{2levels}} = n_1 \Delta E_{10} A_{10} \text{ erg/cm}^3\text{/ s}$$

$$\Lambda_{\mathrm{2levels}} = n_{\mathrm{tot}} f(n_c, T)$$

wait a moment... T?

Collisional rate coefficients are functions of T! $\rightarrow C_{ij}(T)$



$\dot{n}_i$ and $\dot{T}$ must be solved together!
DLSODES: *"and a splendid time(step) is guaranteed for all"*

e.g. collisions with H and e$^-$

$$\dot{n}_0 = -n_0\left[n_\mathrm{H} C^\mathrm{H}_{01} + n_\mathrm{e} C^\mathrm{e}_{01}\right] + n_1\left[A_{10} + (n_\mathrm{H} C^\mathrm{H}_{10} + n_\mathrm{e} C^\mathrm{e}_{10})\right]$$

$$n_\mathrm{tot} = n_0 + n_1$$

General expression

$$\dot{n}_0 = -n_0 \sum_k n_{ck} C^{(k)}_{01} + n_1 \left(A_{10} + \sum_k n_{ck} C^{(k)}_{10}\right)$$

$$n_1 = n_\mathrm{tot} \frac{\sum_k C^{(k)}_{01} n_{ck}}{\sum_k n_{ck}\left(C^{(k)}_{01} + C^{(k)}_{10}\right) + A_{10}}$$

$$\Lambda = n_\mathrm{tot} f(\{n_{ck}\}, T)$$

KROME Bootcamp 2015 - Necessary atomic data

E_i - Energy of the ith level

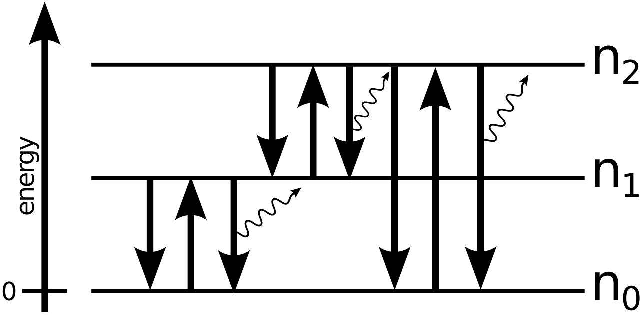
- Needed to get $\Delta E_{ij} = E_i - E_j$ (erg or K)
- Easy to retrieve (e.g. NIST, literature), accurate
- Constants

A_{ij} - Einstein's coefficients

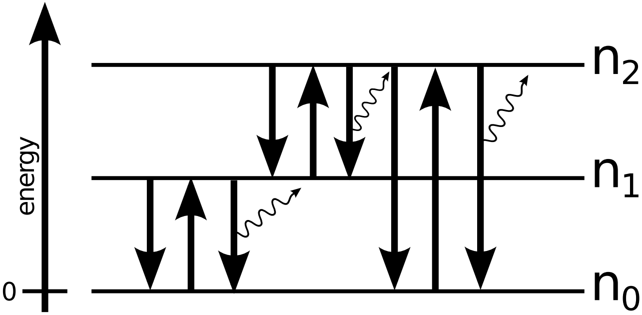
- Probability of spontaneous radiative de-excitations (1/s)
- Easy to retrieve (e.g. NIST, literature), accurate
- Constants

$C_{ij}^{(k)}(T)$ - 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 $^{(k)}$



$$\dot{n}_0 = -n_0 \sum_k n_{ck} \left(C_{01}^{(k)} + C_{02}^{(k)} \right) + n_1 \left(A_{10} + \sum_k n_{ck} C_{10}^{(k)} \right) + n_2 \left(A_{20} + \sum_k n_{ck} C_{20}^{(k)} \right) \quad (3)$$



$$\begin{aligned}\dot{n}_1 &= n_0 \sum_k n_{ck} C_{01}^{(k)} - n_1 \left[A_{10} + \sum_k n_{ck} \left(C_{10}^{(k)} + C_{12}^{(k)} \right) \right] \\ &+ n_2 \left[A_{21} + \sum_k n_{ck} \left(C_{21}^{(k)} + C_{20}^{(k)} \right) \right] \end{aligned} \quad (4)$$

Complete system

$$\begin{cases} 0 & = & \dot{n}_0 = -n_0 \sum_k n_{ck} \left(C_{01}^{(k)} + C_{02}^{(k)} \right) + n_1 \left(A_{10} + \sum_k n_{ck} C_{10}^{(k)} \right) \\ & + & n_2 \left(A_{20} + \sum_k n_{ck} C_{20}^{(k)} \right) \\ 0 & = & \dot{n}_1 = n_0 \sum_k n_{ck} C_{01}^{(k)} - n_1 \left[A_{10} + \sum_k n_{ck} \left(C_{10}^{(k)} + C_{12}^{(k)} \right) \right] \\ & + & n_2 \left[A_{21} + \sum_k n_{ck} \left(C_{21}^{(k)} + C_{20}^{(k)} \right) \right] \\ n_{\mathrm{tot}} & = & n_0 + n_1 + n_2 \end{cases}$$

$$\Lambda = n_1 \Delta E_{10} A_{10} + n_2 \left(\Delta E_{20} A_{20} + \Delta E_{21} A_{21} \right)$$

Full Matrix

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

where

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

Linear solver in KROME

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

General expression (N levels)

$$
\begin{aligned}
\dot{n}_i &= -n_i\left(\sum_{j<i} A_{ij} + \sum_{i\neq j}\sum_k n_{ck} C_{ij}^{(k)}\right) \\
&+ \left(\sum_{j>i} n_j A_{ji} + \sum_{i\neq j} n_j \sum_k n_{ck} C_{ji}^{(k)}\right) \qquad (5)\\
\Lambda &= \sum_i n_i \sum_{j<i} \Delta E_{ij} A_{ij} \qquad (6)
\end{aligned}
$$

OK, this is a mess...

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

KROME saves your day(s)!

Fitting cooling functions

- usually functions of T and of the chemical composition
- OK when parameters are independent
- PRO: very fast (it depends)
- CON: less accurate (it depends)
- CON: difficult to update with new atomic data (re-build machinery)
- Example: H_2 from Simon Glover (colliders H,H^+,e^-,H_2,He)

Physical considerations

$$\Lambda_{cont} = 4\sigma_{SB}\,\kappa\,\rho_{tot}\,T^4$$

```
> ./krome -n mynetwork.ntw -cooling=?
Available coolings are: ATOMIC, H2, HD, DH, DUST, FF, H2GP98, COMPTON, EXPANSION,
CIE, CONT, CHEM, CO, Z_CIE, Z_CIENOUV, CI, SiI, FeI, OI, CII, OII, SiII, FeII
```

- Atomic collisional (H, He)
- Metals collisional (C, O, Si, Fe)
- Molecular collisional (H_2, HD, CO)
- Chemical (Endothermic, Collisional Induced Emission)
- Thermal (Continuum)
- Non thermal (Free-free)
- Dust
- Gas dynamics (Expansion, Compression)

```
> ./krome -n mynetwork.ntw -cooling=?
Cooling CI, SiI, FeI, OI, CII, OII, SiII, FeII available from data/coolZ.dat
```

Species (0, 1, 2)[a]	$\frac{E_{ij}}{k}$ (K)[b]	λ_{μ}[c]	A_{ij} (s^{-1})[e]	γ^{e}_{ij} ($cm^{-3}\ s^{-1}$)[f]	γ^{H}_{ij} ($cm^{3}\ s^{-1}$)[f]
O I ($^3P_2$, $^3P_1$, $^3P_0$)	2.3(2)	63.1	9.0(−5)	1.4(−8)	$9.2(-11)T_2^{0.67}$
	3.3(2)	44.2	1.0(−10)	1.4(−8)	$4.3(-11)T_2^{0.80}$
	9.8(1)	145.6	1.7(−5)	5.0(−9)	$1.1(-10)T_2^{0.44}$

[b] For three-level systems, the energies listed are E_{10}, E_{20}, and E_{21}, respectively.

Custom metal cooling

```
##############
#OXYGEN I
metal:O
#level n: energy (K), degeneracy g
level 0: 0.e0, 5
level 1: 230.e0, 3
level 2: 330e0, 1

#Aij
1 -> 0, 9.d-5
2 -> 0, 1.7d-5
2 -> 1, 1.0d-10

#collisional excitation rates
H, 1, 0, 9.2d-11*(T2)**(.67)
H, 2, 0, 4.3d-11*(T2)**(.80)
H, 2, 1, 1.1d-10*(T2)**(.44)
```

How to add cooling in KROME

```
> ./krome -n mynetwork.ntw -cooling=H2,CII,OI    [-coolFile=data/coolZ.dat]
```

```
x(:) -> INITIAL ABUNDANCES
Tgas -> INITIAL TEMPERATURE
dt   -> TIME-STEP

call krome(x(:),Tgas,dt)

x(:) -> UPDATED ABUNDANCES
Tgas -> UPDATED TEMPERATURE
```

$$\begin{cases} \frac{\mathrm{d}n_i}{\mathrm{d}t} = \sum_{f\in form} R_f(n_i, T) - \sum_{d\in destr} R_d(n_i, T) \\ \frac{\mathrm{d}T}{\mathrm{d}t} = -COOL_{H2}(n_i, T) - COOL_{CII}(n_i, T) - COOL_{OI}(n_i, T) + HEAT \end{cases}$$

What we learned today

1. Understand that cooling processes are complicated
2. Realize that KROME saves your day

Thank you for your attention!

"The only legitimate use of a computer is to play games"
(Eugene Jarvis)

http://kromepackage.org/
http://kromepackage.org/bootcamp