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 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!
$ 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"
| $ 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)
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, $M_{\text{Bulge}} = 1.8 \times 10^9 M_\odot$
  - Stellar and gaseous disc, $M_{\text{disc}} = 8.8 \times 10^9 M_\odot$
- Virial mass $M_{\text{vir}} = 2.2 \times 10^{11} M_\odot$
- 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)
  - H₂ model to know when we reach the fully molecular stage
  - ADVANCED: need a network for CII

- What else?
  - Heating, Photochemistry, Dust…

Typical physical conditions:
- \(10^{-8} \leq n_{tot} \leq 10^4 \text{ cm}^{-3}\)
- \(10 \leq T \leq 10^8 \text{ K}\)
Chemical network
first you want to solve kinetics

H₂ network we need:
- 9 species + ~ 40 reactions (including photochemistry)
  - if you need CII, you will end with min. 16 species and ~ 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)
convert kinetic energy in something else!

1. radiative loss
2. endothermic reactions
3. gas flows
4. gas-dust interaction
Cooling example: collisional
a prototype

$h\nu$
target

Stefano Bovino | Microphysics and chemistry in hydrodynamical numerical simulations
Cooling example: collisional
very messy!

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

\[ \Lambda(n, T) = \sum_i n_i \sum_{j<i} \Delta E_{ij} A_{ij} \]
How to solve - Multilevel system (general)

Full Matrix

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

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 \(N \gg 3\))
$\dot{n}_i$ and $\dot{T}$ must be solved together
Cooling - Necessary atomic data

$E_i$ - Energy of the $i$th 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)
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 $C^{(k)}_{ij}$)
▶ Avoid typos in atomic data (less trivial than expected)
▶ Adding a collider later could be troublesome
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
Going back to your galaxy simulation!

- atomic cooling (e.g. Ly-alpha cooling by H)
- molecular cooling (H₂)
- chemical cooling (endothermic reactions, formation of H₂)
- 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 → kinetic energy

Interesting processes

1. photochemistry
2. exothermic reactions
3. gas compression
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

---

Stefano Bovino | Microphysics and chemistry in hydrodynamical numerical simulations
Interaction radiation-matter
a schematic summary

sources

X-rays-EUV-FUV

interaction radiation-matter

Photoionization

Photodissociation

Photoheating

X-rays, CR

Photons

secondary ionizations

Stefano Bovino | Microphysics and chemistry in hydrodynamical numerical simulations
Secondary ionizations

in WNM

$\propto e^{-\sigma_i N_i}$

X-rays

2-10 keV

electrons

Cosmic rays

1-10 MeV

electrons

UV photo

secondary ionizations

heating

e$^-$
coulomb losses

$\Gamma \propto x_e$
Photochemistry - Equations first

Photochemistry process

\[ H + \gamma \xrightarrow{\sigma_0(E)} H^+ + e^- \quad (1) \]
\[ OH + \gamma \xrightarrow{\sigma_1(E)} OH^+ + e^- \quad (2) \]
\[ OH + \gamma \xrightarrow{\sigma_2(E)} O + H \quad (3) \]

Photochemistry equations (thin)

\[ k_{ph} = \frac{4\pi}{h} \int_{E_0}^{\infty} \frac{l(E)\sigma(E)}{E} dE \quad (4) \]
\[ H_{ph} = \frac{4\pi}{h} \int_{E_0}^{\infty} \frac{l(E)\sigma(E)}{E} (E - E_0)\eta(E) dE \quad (5) \]
\[ \Gamma_{ph}^i = H_{ph}^i n_i \quad (6) \]
Photochemistry - Equations first

Photochemistry equations (thin)

\[ k_{ph} = \frac{4\pi}{h} \int_{E_0}^{\infty} \frac{I(E)\sigma(E)}{E} dE \]  \hspace{1cm} (7)

\[ H_{ph} = \frac{4\pi}{h} \int_{E_0}^{\infty} \frac{I(E)\sigma(E)}{E} (E - E_0)\eta(E) dE \]  \hspace{1cm} (8)

\[ k_{ph} = \left[ \text{s}^{-1} \right] \]  \hspace{1cm} (9)

\[ H_{ph} = \left[ \frac{\text{erg}}{\text{s}} \right] \]  \hspace{1cm} (10)

\[ E = \left[ \text{eV} \right] \]  \hspace{1cm} (11)

\[ I(E) = \left[ \frac{\text{eV}}{\text{cm}^2 \text{s Hz sr}} \right] = \left[ \frac{\text{eV}}{\text{cm}^2 \text{sr}} \right] \]  \hspace{1cm} (12)

\[ \sigma(E) = \left[ \text{cm}^2 \right] \]  \hspace{1cm} (13)
Photochemistry - bins

\[ k_{ph} = \frac{4\pi}{h} \int_{E_0}^{\infty} \frac{I(E)\sigma(E)}{E} \, dE \]  \hspace{1cm} (14)

\[ H_{ph} = \frac{4\pi}{h} \int_{E_0}^{\infty} \frac{I(E)\sigma(E)}{E} (E - E_0)\eta(E) \, dE \]  \hspace{1cm} (15)

Discretization by bins

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

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

\[ \langle E_i \rangle = \frac{E_i^{right} + E_i^{left}}{2} \]
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 → quite messy!
Dust physics
how to model

\[ f(a) \propto a^{-\beta} \]

\( \beta = 3.5, \text{Mathis, Rumpl, and Nordsieck 1977} \)

Stefano Bovino | Microphysics and chemistry in hydrodynamical numerical simulations
Dust physics
how to model

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

Dust formation and destruction depends on

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

**Gas-dust-radiation interaction**

\[ \Gamma_{em} = \Lambda_{g \rightarrow d} + \Gamma_{CMB} + \Gamma_{abs} \]

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

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

Stefano Bovino | Microphysics and chemistry in hydrodynamical numerical simulations
and \( H_2 \) formation on dust

and heating

\[
H + H + \text{GR} \rightarrow H_2 + \text{GR} + 4.48 \text{ eV}
\]

- 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

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

WARNING: different rates available
H$_2$ formation on dust
and heating

\begin{align*}
1 R_f &= 3 \times 10^{-17} n_{\text{tot}} n_H \\
2 R_f &= 3 \times 10^{-17} \frac{Z}{Z\odot} C_\rho n_{\text{tot}} n_H \\
3 R_f &= 3 \times 10^{-17} \frac{T^0.5 a}{1 + 0.4(T_2 + T_{d2})^{0.5} + 0.2 T_2 + 0.09 T} n_{\text{tot}} n_H \\
4 R_d &= 0.5 v_g \pi a^2 \varepsilon_{H_2} S(T, T_d) n_H n_d
\end{align*}

\begin{enumerate}
\item Jura 1975
\item Gnedeni+ 2009
\item Hollenbach& McKee 1979
\item Cazaux & Tielens 2004, state-of-the-art
\end{enumerate}
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 → shielding (by molecules and dust)!
How this is in KROME

Options file: my_options

- n networks/react_galax
- 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

5the 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

\[
k(:, :) = \text{coe-tab}(n(:, :)) \text{ !compute coefficients}
\]

! 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

1. Which microphysics?
2. Included in KROME?
   - Yes
      - Lucky enough?
        - Yes: run KROME, obtain a compact file: $x(:, T_{gas}, \text{hydro}_dt$
        - No: join the dev
   - No: join the dev
A quick summary on how to include KROME in your simulations

- enable options you need
- run KROME
- include 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_nmol and other common arrays)
- change your Makefile accordingly
  - solver files (opkd*.F)
  - krome routine (krome_all)

Stack Correctly
Initial conditions (no UV background)
Very preliminary test

Evolution, first 50 Myr (relaxation)
Very preliminary test
run GASOLINE (SPH) + KROME

Stefano Bovino | Microphysics and chemistry in hydrodynamical numerical simulations
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 (∼ 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!