Reduction methods for astrochemical networks

T. Grassi

"La Sapienza" University of Rome - Group of Theoretical Chemistry

October 2012

Outline

- Introduction
- Computational problem
- Methods overview
- Flux-based method
- Large network example
- Conclusions

ISM chemistry

Why chemistry is important (in ISM numerical simulations)

- Needed to compute metal/molecular cooling → SF
- Opacity
- Comparison with observations
- ...

Why chemistry is troublesome (in ISM numerical simulations)

- very CPU demanding
- has a non-linear behavior
- chemical networks are complex
- needs accurate rates

Dealing with complexity - Numerical framework

Typical framework:

e.g. Hydrodynamical simulations (1D, 3D, shock, PDR, . . .)

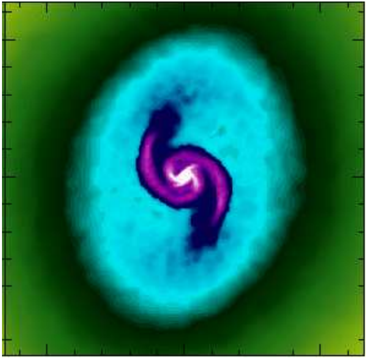
particle $\equiv$ unit of gas ($\approx 10^4$ M$_\odot$)

Each gas particle computes:

- hydrodynamics (e.g. SPH)
- gravity (e.g. tree)

Each gas particle updates:

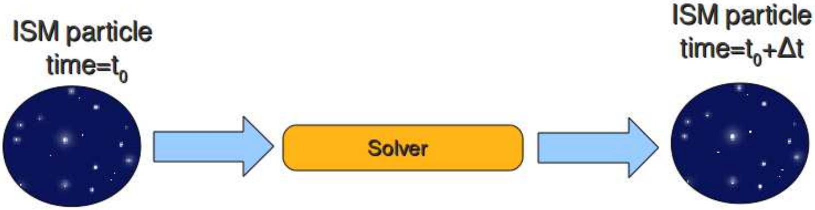
- chemistry (H, H_2, e^-, . . .)
- gas temperature (Λ, Γ)
- dust (ρ, T, dn/da, . . .)



Merlin, Grassi, et al. 2009 A&A

Dealing with complexity - Numerical framework

How to update a gas particle at a given time-step
($\gtrsim 10^6$ particles (gas units) $> 10^4$ time-steps)



- e.g. 1D-hydro: chemistry (large network)
- $\gtrsim$ 90% of the CPU time

Dealing with complexity - The chemical network

Chemical network = Cauchy's problem:

$$\frac{\mathrm{d}n_i}{\mathrm{d}t} = \overbrace{\sum_{lm} k_{lm}(T)n_l(t)n_m(t)}^{formation} - \overbrace{\sum_{j} k_{ij}n_i(t)n_j(t)}^{destruction}$$

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

e.g. A+B$\rightleftharpoons$ C+D

$\dot{n}_\mathrm{A} = -k_\mathrm{AB}\, n_\mathrm{A}\, n_\mathrm{B} + k_\mathrm{CD}\, n_\mathrm{C}\, n_\mathrm{D}$
$\dot{n}_\mathrm{C} = +k_\mathrm{AB}\, n_\mathrm{A}\, n_\mathrm{B} - k_\mathrm{CD}\, n_\mathrm{C}\, n_\mathrm{D}$

Problem:

Solve an Ordinary Differential Equations System (ODE)

Dealing with complexity - Methods/1

“Pure” computational strategies

- Efficient solver: DVODE/DLSODES (up to $\times 100$)
- Interpolate rate tables (up to $\times 5$)
- Particle Buffering (up to $\times 10$)
- Custom compiler optimization
- Know your code: profiling
- Good programming practices (save divisions, avoid casting, ...)

Dealing with complexity - Methods/2

Buffering method or “The solver always* rings twice”

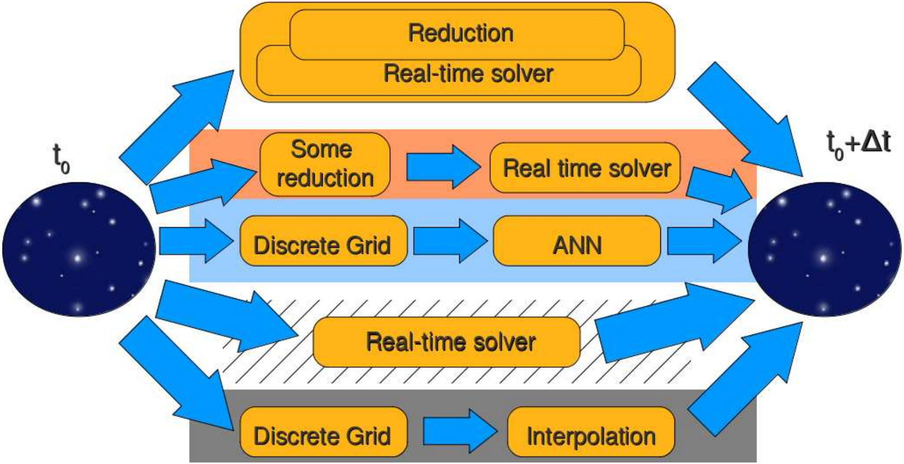
- calls to the solver are cpu demanding
- same initial particle conditions lead to same particle evolution
- store already computed chemical evolutions in a buffer
- criterion of similarity: $s_j = \sqrt{\sum_i \left(B_{ij} - x_i\right)^2 / N} < \xi_b$

```
$x \leftarrow$ initial conditions
for $j$ = particle in buffer $B$ do
  if $(s_j < \xi_b)$ then $\hat{x} = (\hat{x}_j - x_j)\,\mathrm{d}t/\mathrm{d}t_j + x$
endfor
if (not found) then
  $\hat{x} \leftarrow solver(x)$
  add $\{x, \hat{x}\}$ to buffer $B$
endif
```

*often

Dealing with complexity - Methods/3

Problem: too many reactants (>400) and too many reactions (>4000)!
Methods to solve ODE System:



Dealing with complexity - Flux-based reduction /1

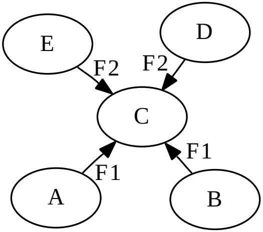
Flux-based reduction

- Flux: $F_i = k_i\, n_q\, n_p$
- RHS term: $\dot{n}_i = \sum_{j=1}^{N} s_j F_j$
- Determining less “active” reactions @ given sub-steps
- i.e. neglect $F_i | F_i < \zeta\, F_{\mathrm{max}}$
- reduced RHS term: $\dot{n}_i = \sum_{j=1}^{M} s_j F_j$ where $M < N$

Two-reactions example

- A+B→C $\quad F_1 = k_1\, n_{\mathrm{A}}\, n_{\mathrm{B}}$
- D+E→C $\quad F_2 = k_2\, n_{\mathrm{D}}\, n_{\mathrm{E}}$

- $\dot{n}_{\mathrm{C}} = F_1 + F_2 \qquad N = 2$
- $\dot{n}_{\mathrm{C}} = F_2 \qquad M = 1\ (n_{\mathrm{A}} = 0)$



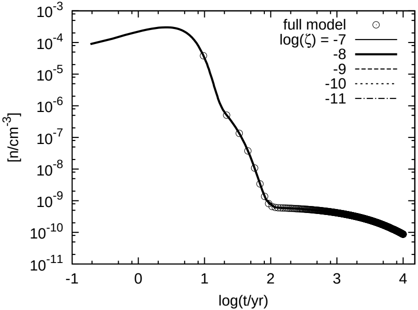
Dealing with complexity - Flux-based reduction /2

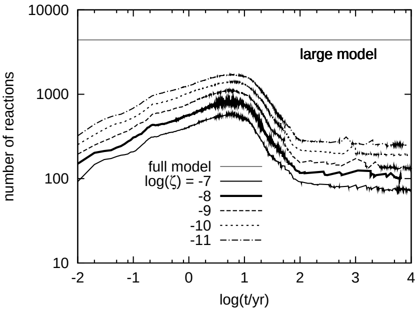
Large model example

- ≈Wakelam&Herbst 2008
- species: 451
- reactions: 4399
- one-zone
- no cooling
- $T = 10$ K
- $\zeta_{\rm CR} = 1.3 \times 10^{-17}$ s^{-1}
- $A_v = 10$

Dealing with complexity - Flux-based reduction /3

Results $x_{\mathrm{C}}(t)$:





speed-up: ×10

(details in Grassi,Bovino et al. 2012 MNRAS)

Conclusions

Dealing with complexity

- Computational astrochemistry needs reduction methods
- “Pure” computational methods are required
- Other strategies allow large simulations
- Flux-based method is the best choice*

Next talk by Stefano Bovino: *a priori* methods!

*for the moment

Thank you for your attention!

Also thanks to: F.A.Gianturco, D. Schleicher, S. Bovino