Using KROME on-the-fly and as a postprocessing tool for star formation simulations

Daniel Seifried
I. Physikalisches Institut, University of Cologne

KROME Computational school
18.9.2014, Göttingen

Chemistry in present day SF

- Heard about chemistry in simulation of POPIII star formation (e.g. talk by D. Schleicher
  - „Relatively“ simple: includes only light atoms, (almost) no metals
  - Computational costs are moderate
  - Applied many times in the past
- For present day star formation (SF) metals + dust chemistry have to be included
- This makes chemistry unproportionally more expensive
  - Number of rate equations could be up to N^2 (N = number of species)
  - reduction due to selection of most important reactions
- Even for the most abundant (and simple) molecule CO
  - ~ 40 species
  - ~ 300 reaction

FLASH code

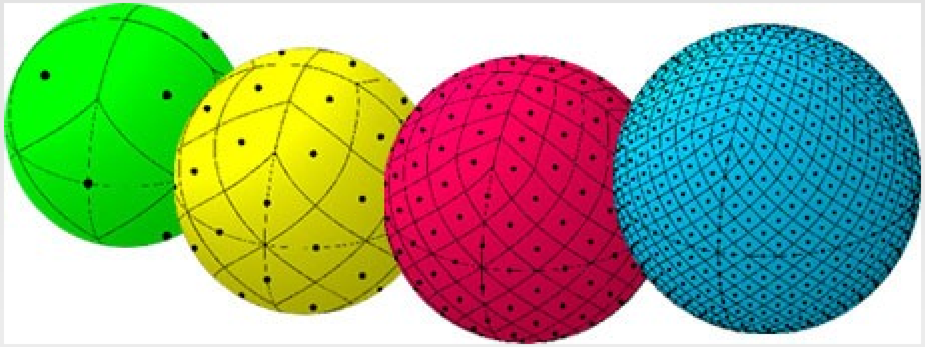
- Astrophysical code to simulate 3D, magneto-hydrodynamical problems
- Uses Adaptive-Mesh-Refinement to resolve regions of interest with higher spatial resolution
- Block structure: Simulation domain divided in blocks/patches consisting of 8^3 cells
    - A block resides completely on one CPU (reduced communication)
    - Each block can be divided into 8 smaller blocks with half the linear size
- FLASH is designed in a modular fashion:
    - Each module covers a certain physical process
    - Modules can be used individually or in combination

FLASH modules

- Self-gravity:
  - Multigrid
  - Tree-code (by Richard Wünsch, usually faster by a factor of a few)
- Sink particles
  - Lagrangian particles accreting/ejecting mass
  - Interacting only gravitationally with gas
- Stellar feedback models (coupled to sinks)
  - Protostellar outflows, supernovae, stellar winds
- Radiative feedback of ionising and non-ionising radiation (optically thin gas)
- Tree-Col: for (self-) shielding of ionising radiation

Tree-Col

- Tree-Col developed by P. Clark and R. Wünsch (Clark et al., 2012):
  - Calculates the mean optical depth / column density for each cell
- Makes use of the Healpix tool:
  - Divides sphere in regions of equal size
  - Calculates column density along each direction
  - Averages over all directions
  - Usually already 12 pixels are sufficient to recieve accuracy of 10%

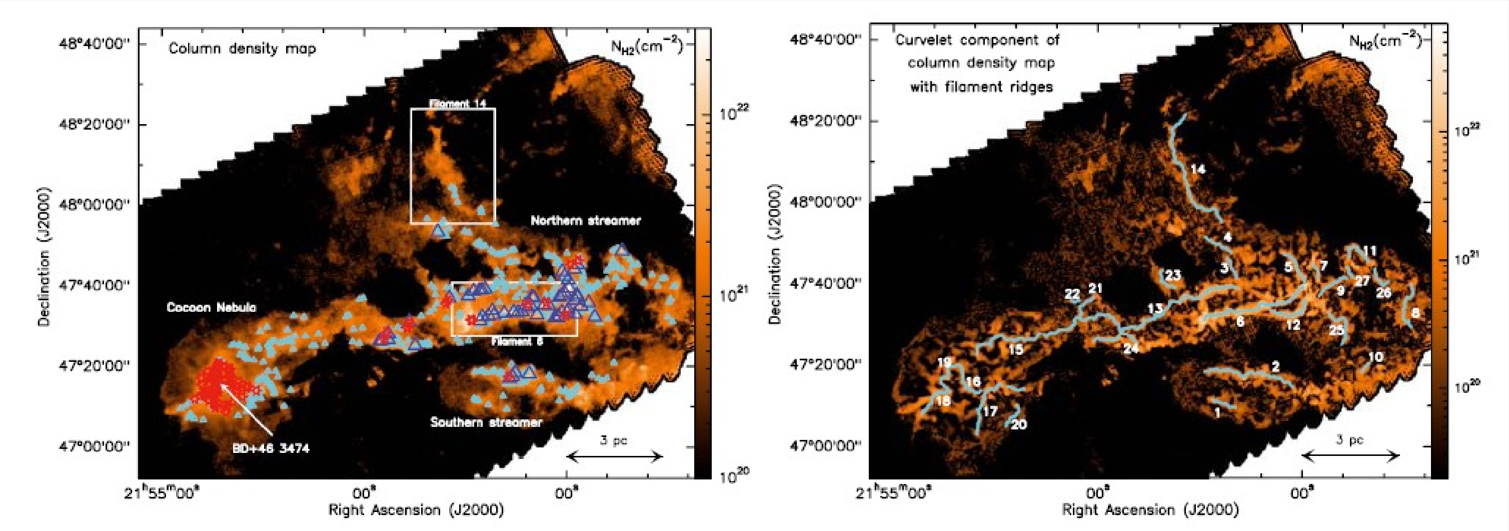


Picture taken from http://healpix.jpl.nasa.gov/

- Column density is essential for many chemical rates to obtain proper ionisation rates by incident radiation

Star formation in interstellar filaments

- Filaments seem to be everywhere: „Filamentology“
- SF takes places in dense cores lining up along filaments
- Typical properties:
  - width of 0.1 pc
  - pervaded by magnetic fields
  - flat inner part, at larger distances density falls off as r^-2



Arzoumanian et al. 2011

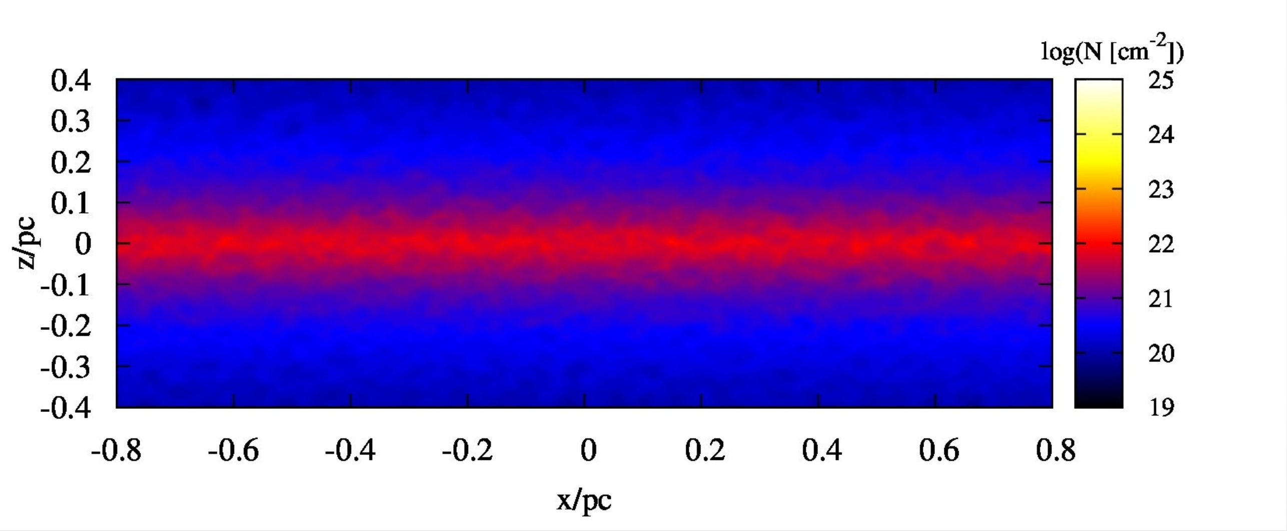
Star formation in interstellar filaments

- Open questions:
  - How are filaments formed (not covered here)?
  - What sets the fragmentation of filaments?
  - Where and at which rate does SF take place?
  - How do simulated filaments appear in observations (dust + molecular line emission)
  - → needs information about chemical abundances (KROME)!

- We plan to tackle the last three questions (although not in detail this talk)

Simulation setup

- Initial conditions (from observation):
  - Mass per length: 25 and 75 M_{sun}/pc
  - Central density of ~ 10^{-19} g/cm^{-3} , T = 15K
  - Without and with magnetic fields
    - Perpendicular and parallel to filaments, strength: 40 muG
  - Turbulent motions with M_{rms} ~ 1
  - Width ~ 0.1 pc, length 1.6 pc



Physics / chemistry applied

- Run with FLASH4, using
  - (sink particles, not yet, but applicable without any further modifications)
  - Spatial resolution of 40 AU
  - Self-gravity

- Aim: Following SF process over ~ 100 kyr – 200 kyr

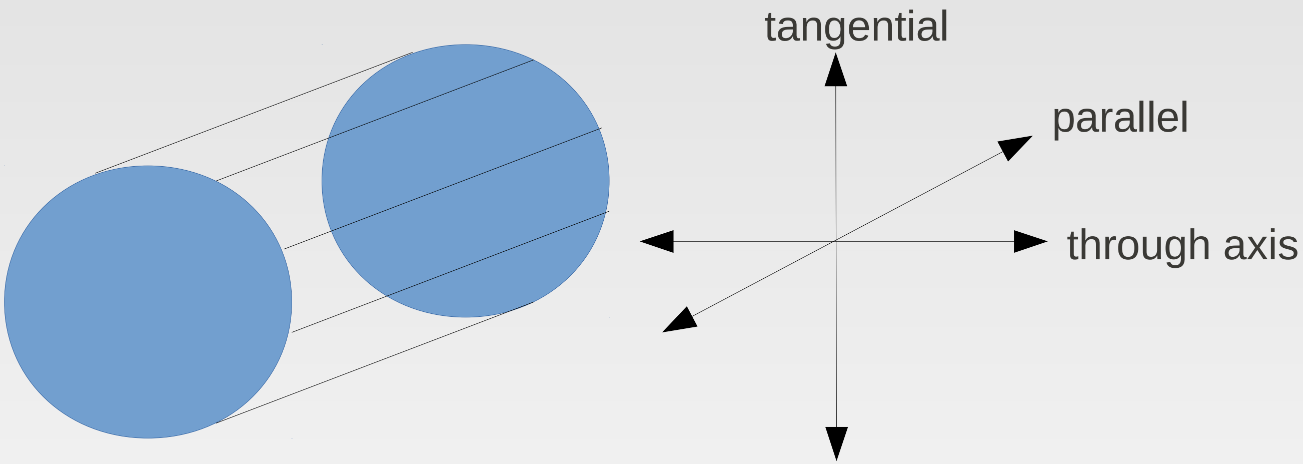
- As a reference run: simulations without chemistry, isothermal EOS
- Including chemistry: KROME network for CO formation
- We use the react_COthin network

Chemistry

- Starting with ionized carbon (CII)
- 42 species, 278 reactions, including CO, HCO+, H_2O
- H_2 formation on dust in parametrised form, dust temperature set to 10 K
  - call krome_set_user_Tdust(10.)
- Ionisation by incident radiation (CR), ionisation rate set to $1.3 \cdot 10^{-17}$ s^{-1}
  - call krome_set_user_crate(1.3e-17)
- KROME Heating and cooling mechanism:
  - -cooling H2, CHEM, CIE, CI, CII, OI, OII, SiI, SiII
  - -heating CHEM, PHOTODUST, (CR could be included)
- As well as own defined dust cooling
  - Goldsmith ApJ 557, Eq. 15
  - Does not require the usage of dust within KROME (memory saving)
  - Similar to KROME method, but integrated over dust particles sizes

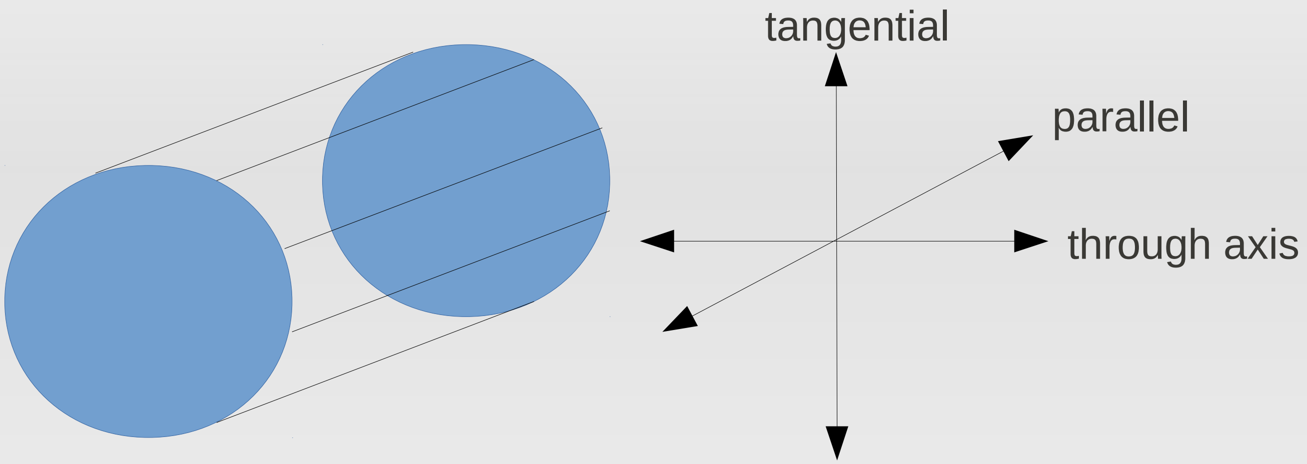
What about the ionising radiation?

- We do not use Tree-Col, but a simplified analytical proxy for the column density ∑ / extinction Av
- Consider a cylindrical symmetric setup



- Consider 6 rays: 2 parallel to axis, 2 „through“ axis, 2 „tangential“ to structure
- For a given density profile: ∑ along each direction can be calculated
  - Use initial density distribution throughout the simulation
  - Final ∑ is simply the average of the 6 rays

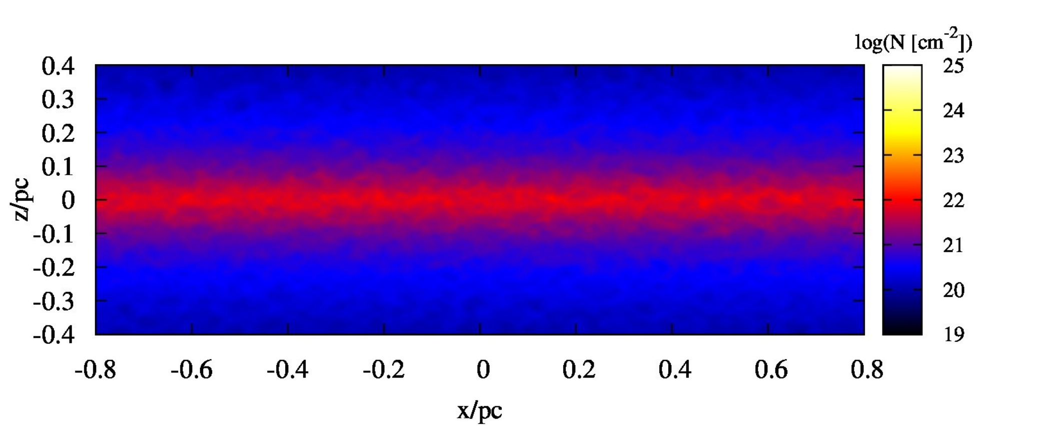
What about the ionising radiation?



- To set the exctinction: call krome_set_user_Av(„Σ [1/cm^2]“ / 1.87d21)
- For the H_2 self-shielding we assume that all hydrogen is in H_2
  - In code adapt: user_H2self = fselfH2(„Σ [1/cm^2]“ * 0.5, 1d5)
- Caveats:
  - H_2 self-shielding overestimated
  - No constant density profile over time
  - Average over only 6 directions

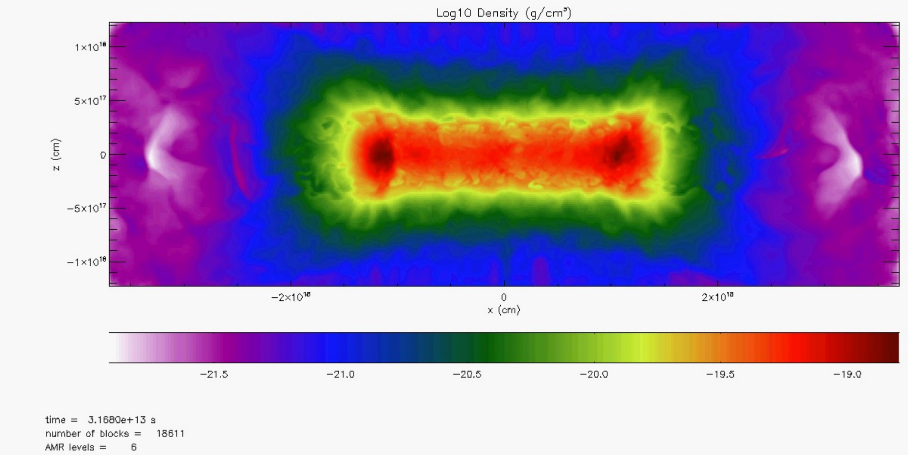
Results

- Some technical details
  - Simulation runs on SuperMUC at LRZ/Garching
  - Use of 500 blocks/CPU, standard queue (~ 1.5 GB memory / CPU)
  - Run for 20 h on 240 CPUs for the first 1 Myr, ~ 4800 CPU-h
  - About 4 min for each timestep, ~ 300 steps in total
  - About 8 times slower than without chemistry

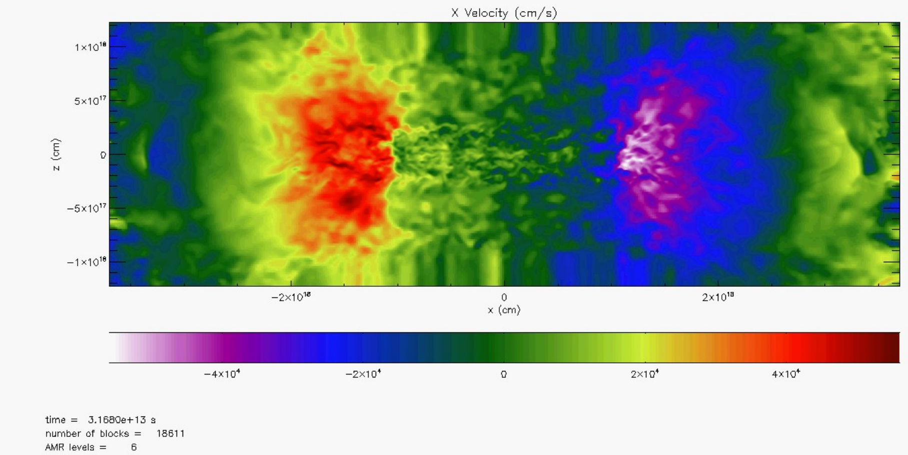


Time evolution

Density



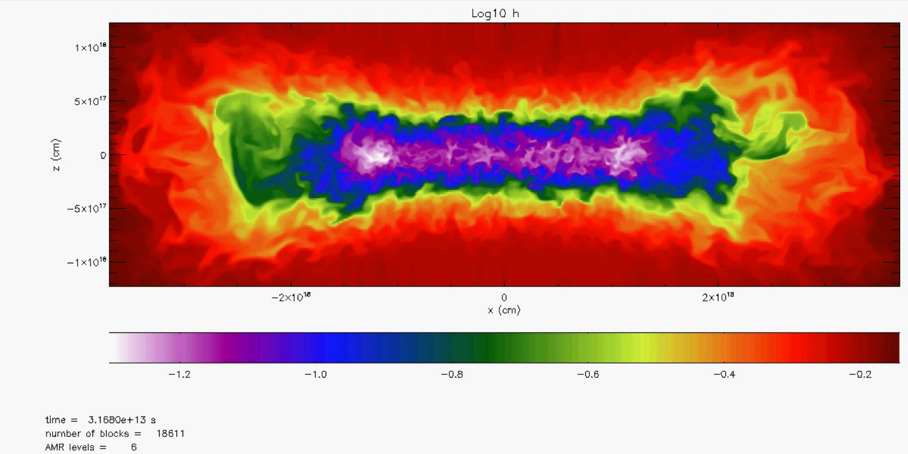
Velocity



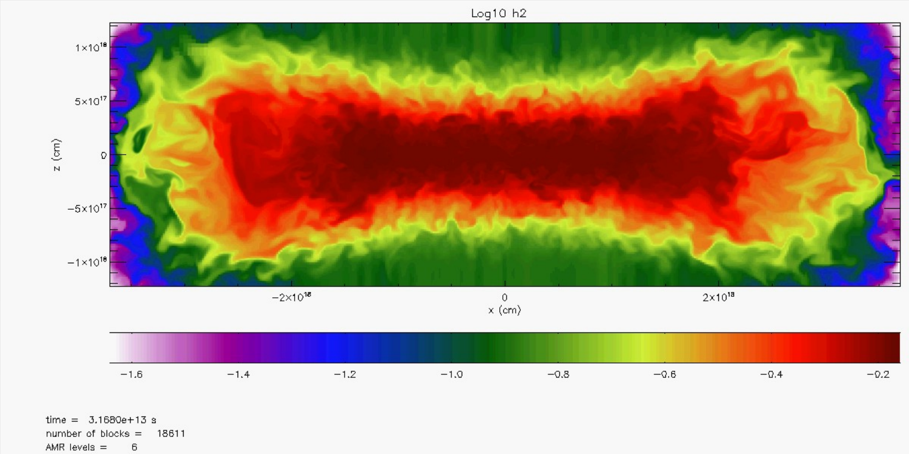
- Edge-on collapse, condensations form first at outer parts
- In general for first 1 Myr: evolution similar to reference run without chemistry

Results

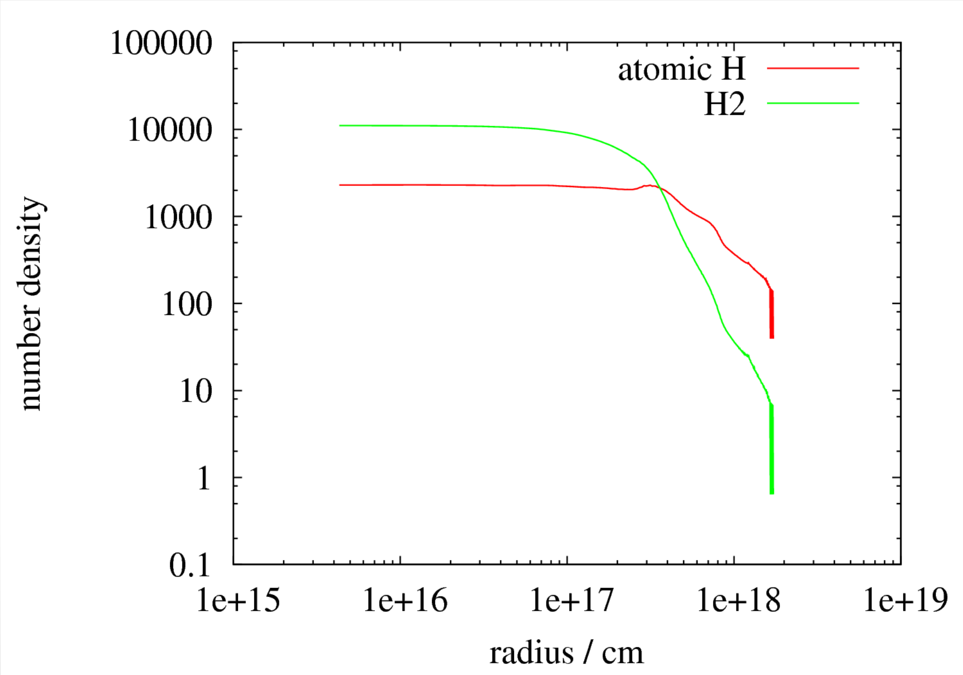
Atomic hydrogen



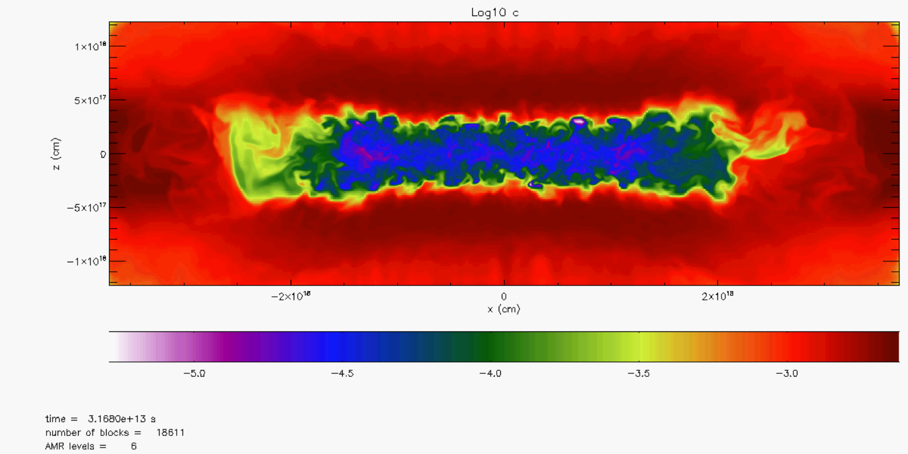
Molecular hydrogen



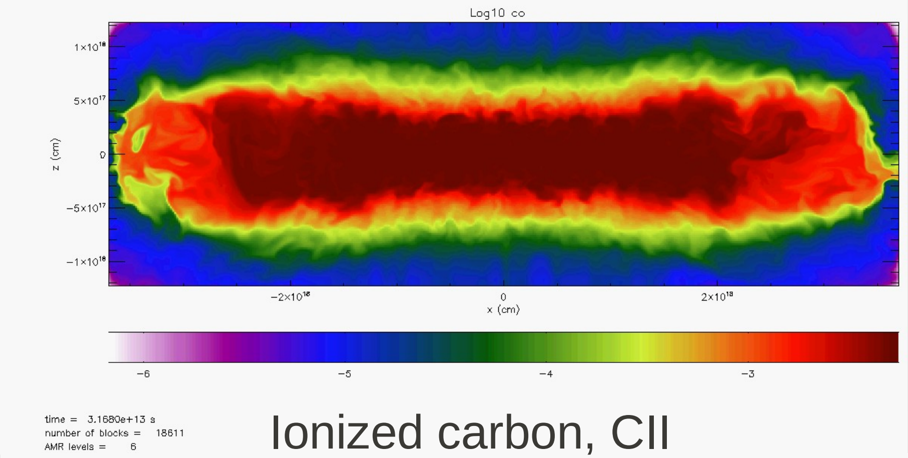
- In dense part: hydrogen almost completely in H2
- Gradial conversion H → H_2 along radial direction



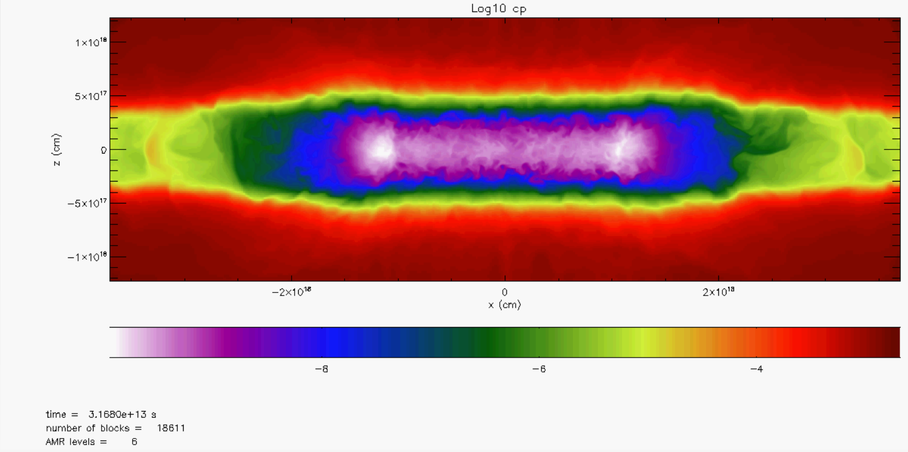
Atomic carbon

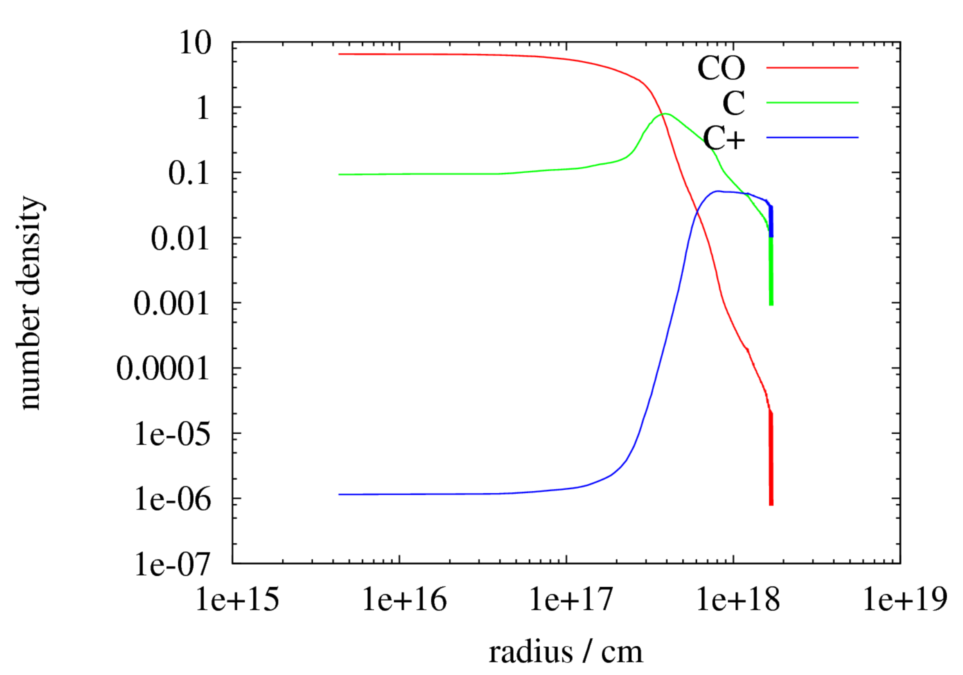


CO



Ionized carbon, CII



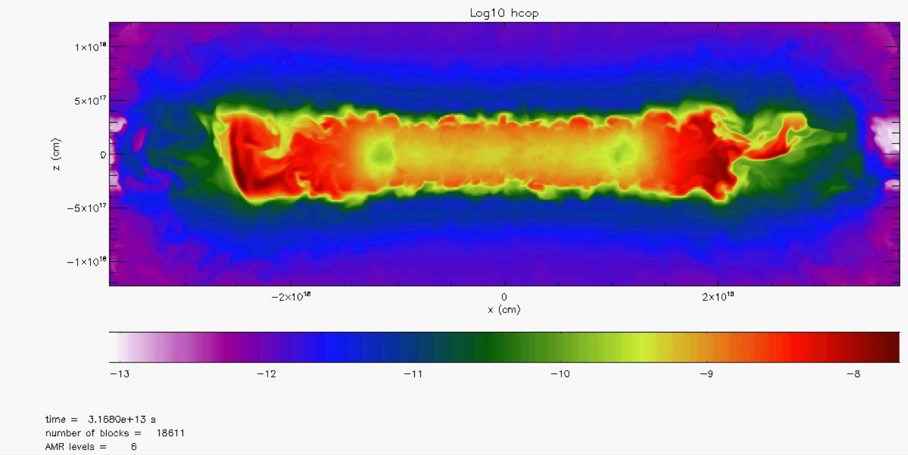


- Conversion from C^+ → C → CO
  - Blocking of ionising radiation

Results

- Also other more complex species like HCO+ seem to be reproduced reasonable well
- However: n(HCO^+)/n(H2) ~ 10^{-11}
- Observations: 10^{-9}
- Missing something:
  - Larger ionizing flux?
  - Larger network?

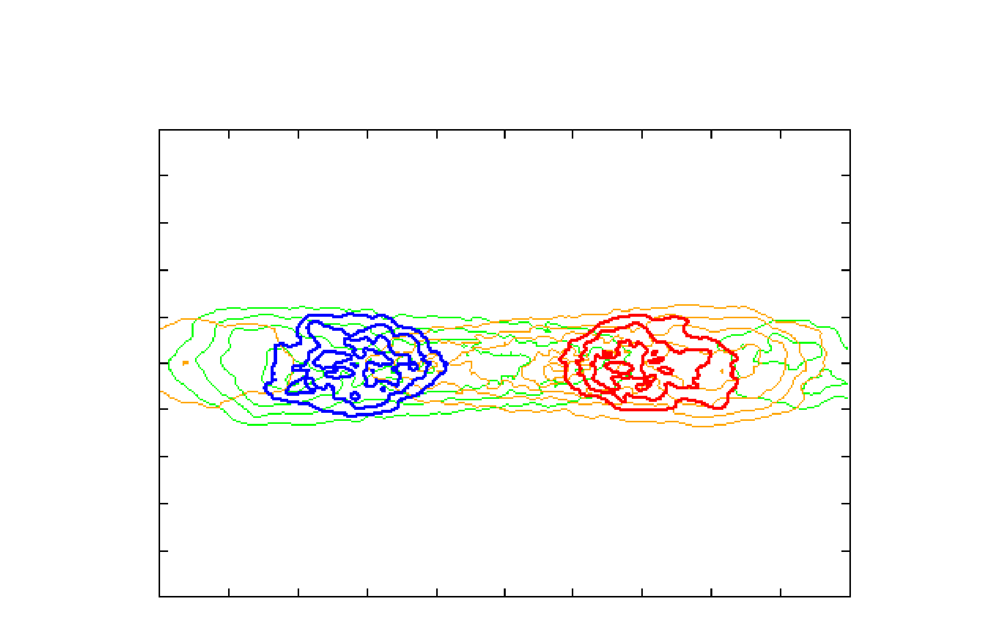
HCO^+

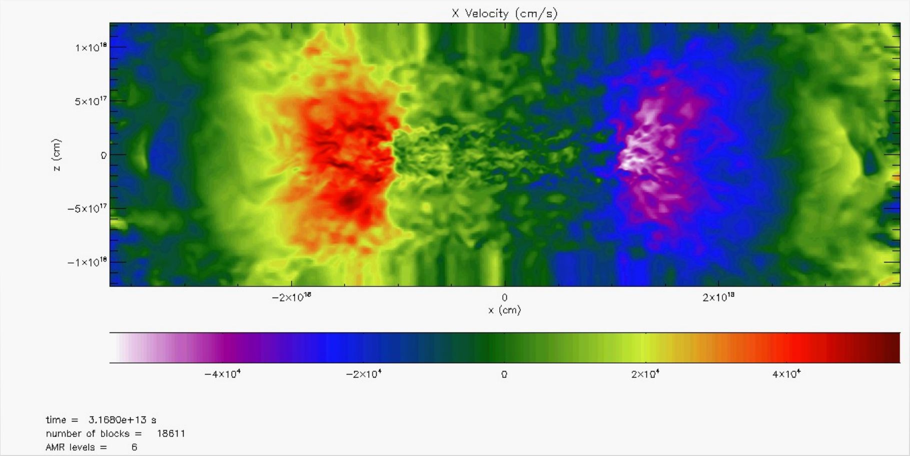


Results

Usage of data for line transfer calculations:

- Required for comparison with observations
- Done in a postprocessing step
- CO-channel maps (RADMC-3D) reflect velocity structure





Intermediate summary

- The previous result show that
  - In principle the usage of a complex chemical network is possible „on-the-fly“
  - On „normal“ machine (memorywise)
  - However, computationally very expensive
  - Reasonable results for most species

Chemistry via Postprocessing

- → What if we run the simulation without chemistry and apply KROME in a postprocessing step
- **Main question 1:** how long do we iterate KROME, i.e. for how long do we let the chemical network evolve?
  - Until a chemical equilibrium is reached?
  - As long as simulation has run to that point?
- **Main question 2:** How to model the thermal gas properties properly?

Chemistry via Postprocessing

- We use same network (react_COthin) and same (spatial dependent) proxy for extinction (Av) as before
- No cooling applied, dust and gas temperature set constant during KROME loop
- KROME postprocessing frontend:
  - Read in all data of current snapshot
  - Hand over density, **temperature** and Av to the main postprocessing routine (here called EquilibriumChemistry.F90)
  - Further parameters:
    - Z (metallicity),
    - dt_start (starting timestep)
    - tmax (time until KROME is iterated)
    - dtmax (optional)

Chemistry via Postprocessing

- subroutine EquilibriumChemistry(rho, T, Z, dt_start, tmax, n, ion_rate, opt_depth, dtmax)
- Determine hydrogen density n(idx_H) from rho
- call krome_scale_Z(n,Z)
- n(idx_Cj) = n(idx_C) , carbon is ionized rather than neutral
- n(idx_C) = 1.e-40
- n(idx_E) = krome_get_electrons(n(:))
- call krome_set_user_crate(ion_rate)
- call krome_set_user_Av(opt_depth)
- call krome_set_user_Tdust(10.)

Main iteration loop

```
dtC= dt_start, ttot = 0

do while(ttot<tmax)
 ttot=ttot+dtC

  call krome(n(:),T,dtC)
```

- Relative changes in abundances:

```
  diff(:) = (n(:)-nold(:))/n(:)
  where(n .eq. 0) diff=0. ! just to avoid infinities and Nans in diff
```

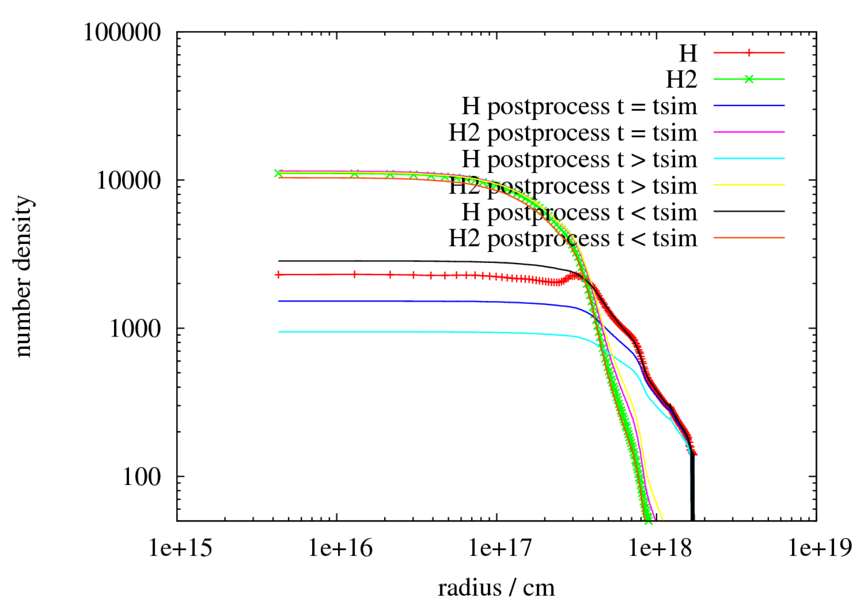
- Next, we check whether we can use a larger timestep in next iteration
- threshold of 0.3 is variable, of course

```
  if(maxval(diff) .le. 0.3 .and. minval(diff) .ge. -0.3) dtC=min(2.*dtC,dtmax)

  nold(:)=n(:)

 end do
```

- Write out whatever species you like!

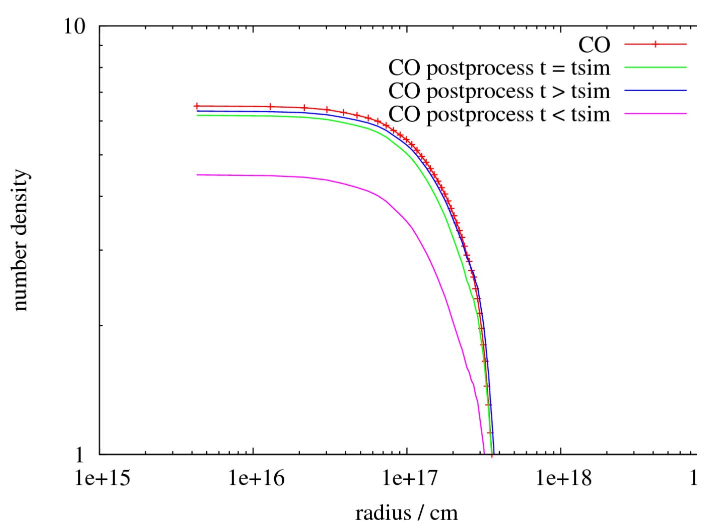
Chemistry via Postprocessing

- OpenMP parallised around call to EquilibriumChemistry.F90
- First test:
  - For radially average data, i.e. density, temperature as a function of distance from symmetry axis of filament
  - For t_iteration (tmax) = t_sim: agreement within factor of 2
  - Slight dependence of atomic H density on iteration time
    - n(H) decreases at t goes up

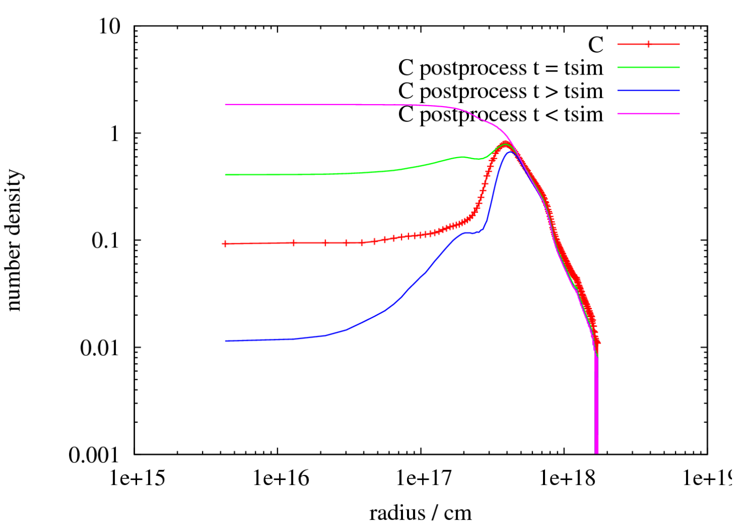


Chemistry via Postprocessing

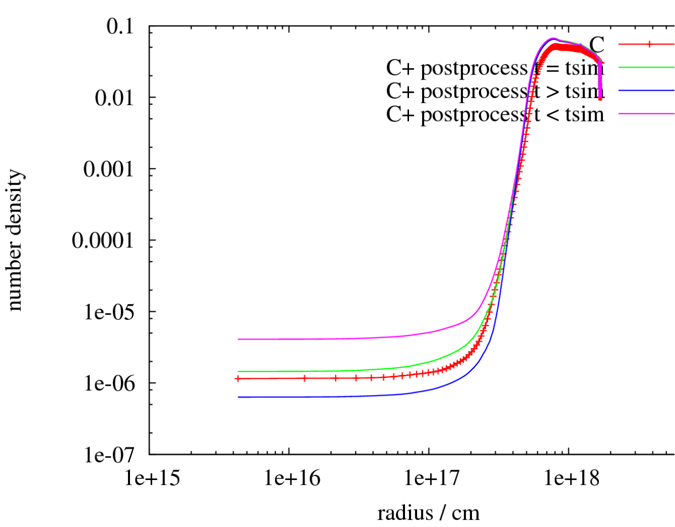
CO



atomic C



ionized C



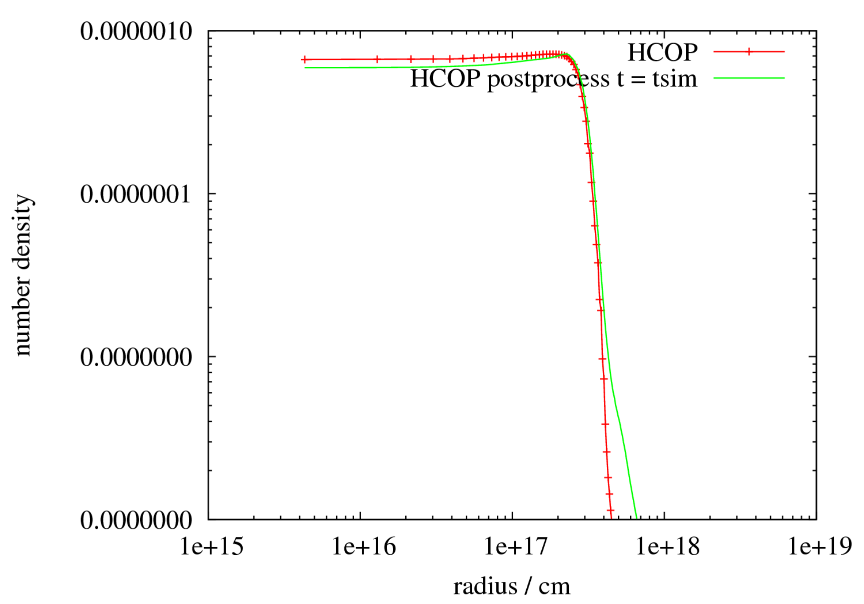
- Strong dependence of abundances on integration time
  - Factor of 2 in integration time gives large differences > 10 x

- Integration time is a crucial parameter
- Still: For t_integration = t_sim reasonable agreement

Chemistry via Postprocessing

- Another source of error might be the averaging before postprocessing
- Apply postprocessing to „unaveraged“ data
- OpenMP parallelised around call to EquilibriumChemistry.F90
- First test:
  - For about 10 Mio cells: about 15 h runtime with 80 threads
  - probably speed up through clever subcycling

Chemistry via Postprocessing



Even more complex molecules like HCO^+ are reproduced well in postprocessing step!

- In general: agreement within a factor of a few for:
  - $H, H_2, C, CI, CII, HCO^+$
- in principle KROME as postprocessing tool usable
- But: careful testing required for
  - each network
  - different physical situtations

Conclusions

- KROME can by used „on-the-fly“ even with a complex network
  - 40 species, 300 reaction
  - Runs on standard machines with 1.5 GB memory / CPU
  - Slow down by a factor of 8
- Applied to a collapsing filament
  - Come up with a proxy for optical depth
  - Reasonable results for carbon bearing species

Conclusions

- KROME can by used „on-the-fly“ even with a complex network
  - 40 species, 300 reaction
  - Runs on standard machines with 1.5 GB memory / CPU
  - Slow down by a factor of 8
- Applied to a collapsing filament
  - Come up with a proxy for optical depth
  - Reasonable results for carbon bearing species
- KROME as a postprocessing tool: it seems to work
  - Iteration time is a critical parameter
  - Reasonable agreement within a factor of 2
  - Check beforehand by means of a reference run
  - Applicability might depend on network

Thank you for your attention!