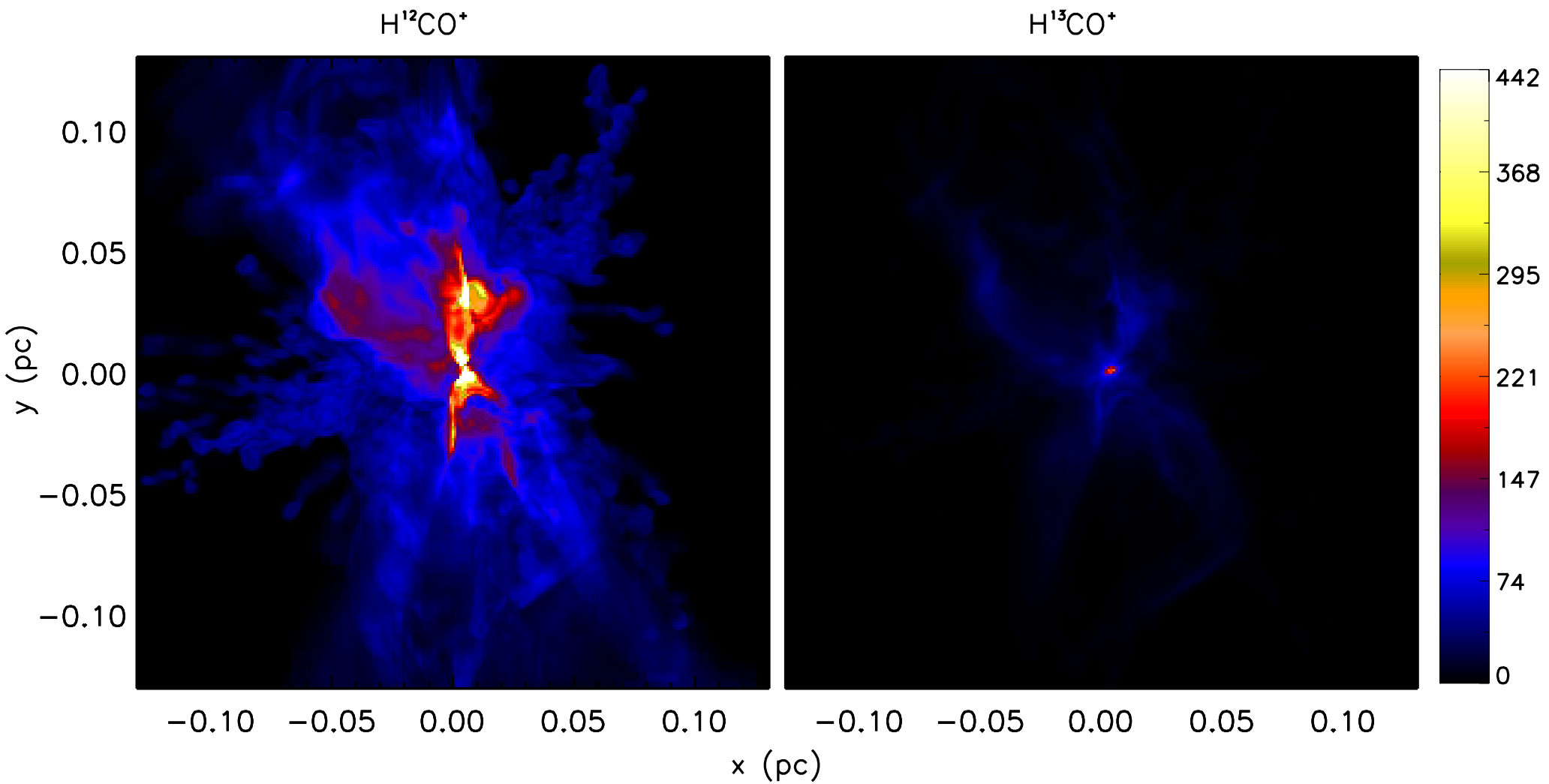
Gas Line Transfer with RADMC-3D

Stella Offner
HiPACC

Aug 5, 2013



Integrated HCO^+ isotopologue intensity (Offner & Arce in prep).

Outline

- Why model lines?
- Molecule basics
- RADMC-3D
- Non-LTE Line Methods
- Project

Outline

- Why model lines?
- Molecule basics
- RADMC-3D
- Non-LTE Line Methods
- Project

Why model molecular lines?

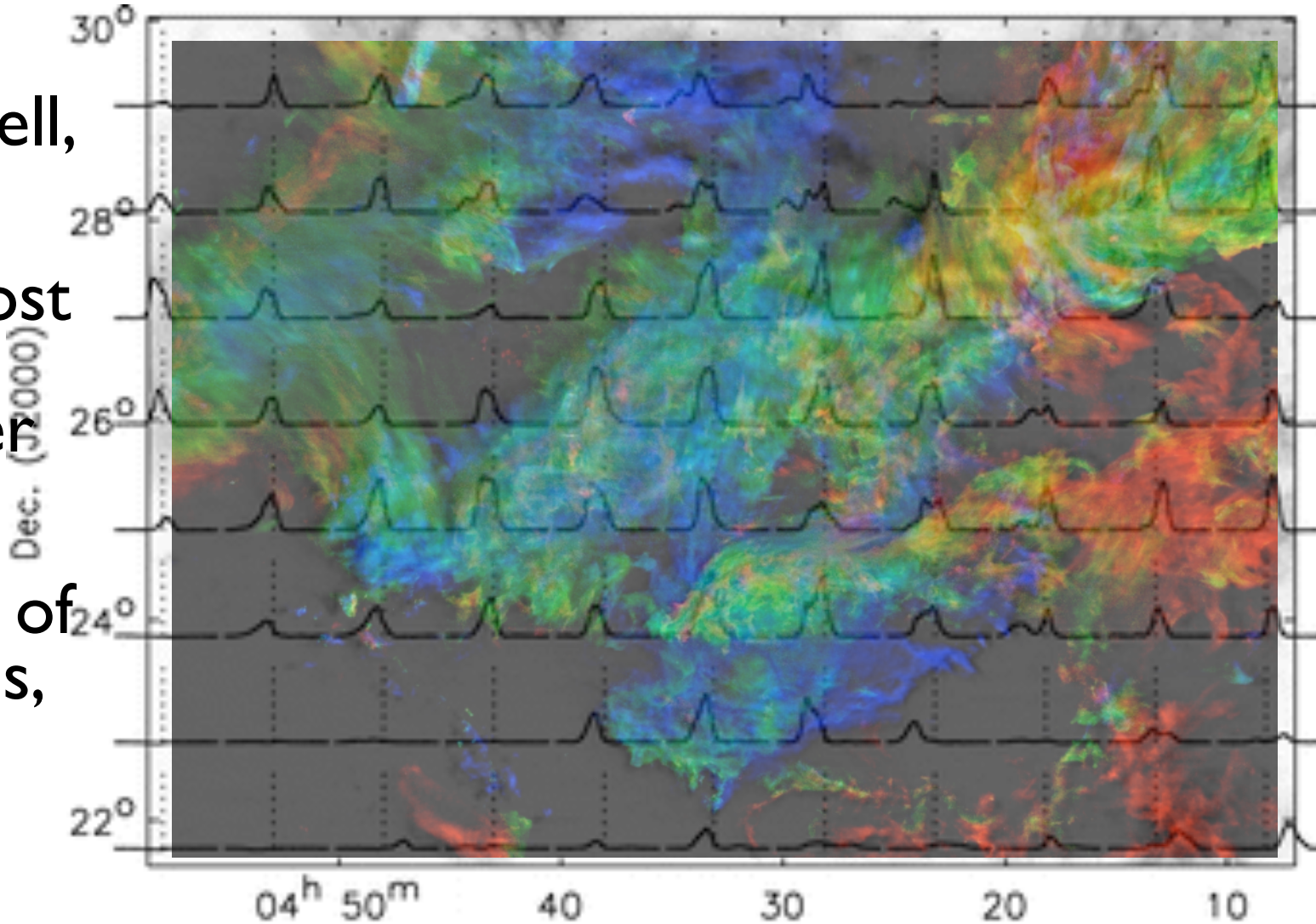
- Molecular clouds are, well, molecular
- CO is the most common molecule after H_2
- Direct probe of gas kinematics, density and temperature

^{12}CO axis range is 0-15 km/

Narayanan et al. 2008

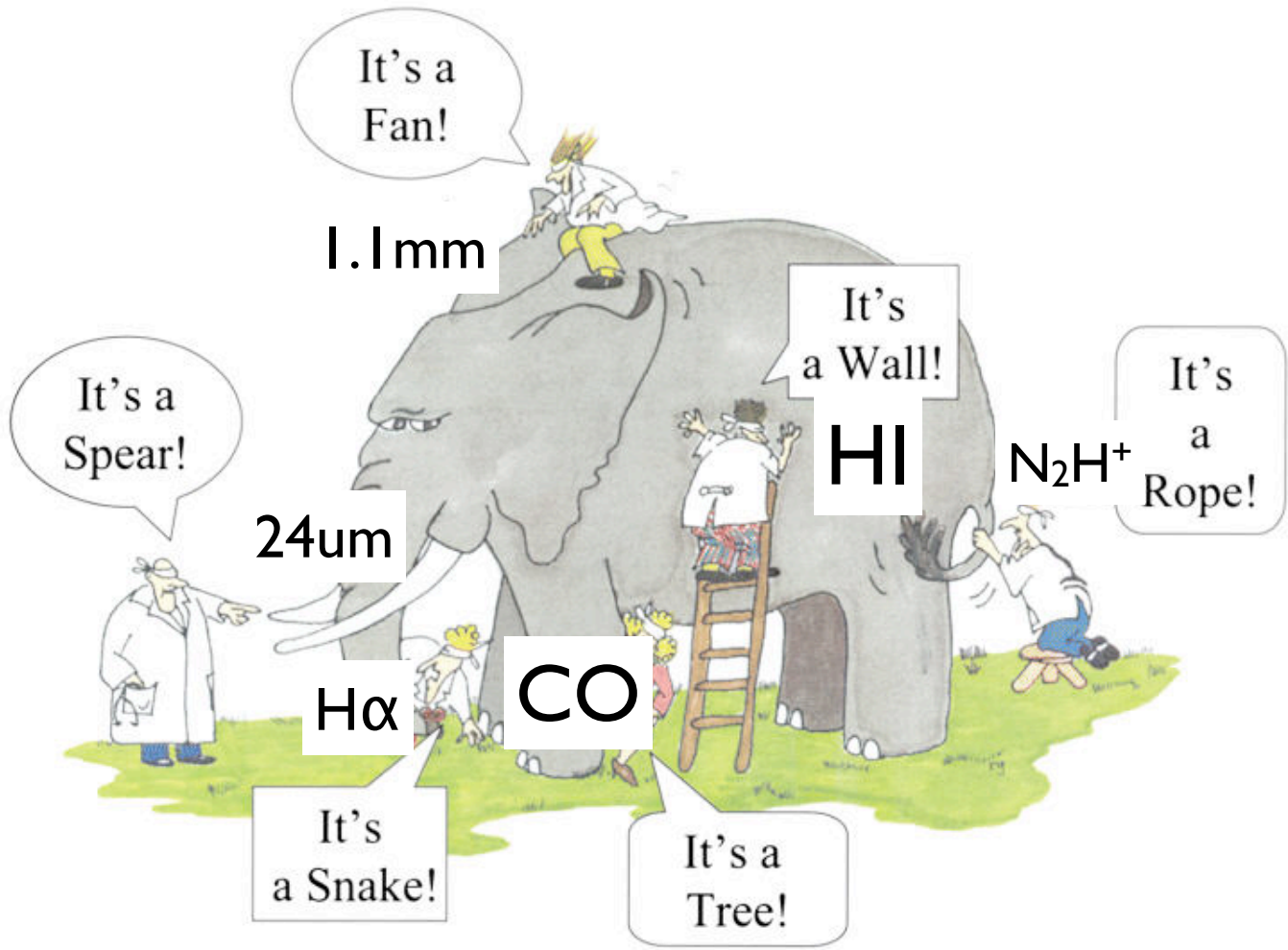
Image: Mark Heyer

Taurus Molecular Cloud



Why model molecular lines?

- Molecular clouds are, well, molecular
- CO is the most common molecule after H_2
- Direct probe of gas kinematics, density and temperature

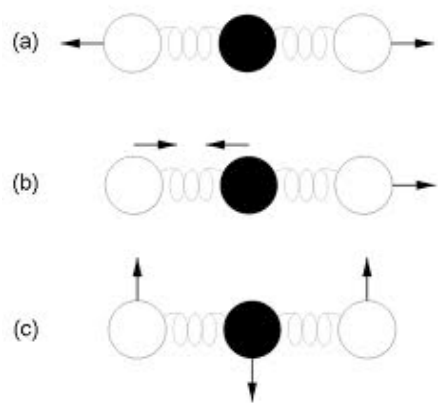


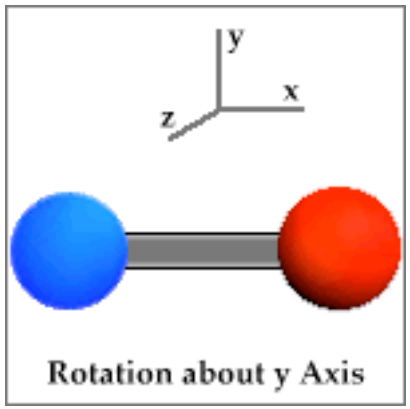
Outline

- Why model lines?
- Molecule basics
- RADMC-3D
- Non-LTE Line Methods
- Project

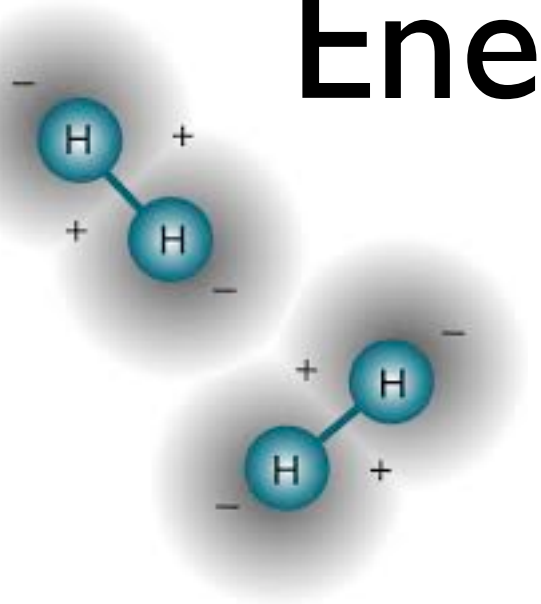
Molecule Basics

- Electronic bands (optical and UV): H_2 Lyman and Werner bands near 1100A
- Vibrational transitions (NIR-MIR) CO v=1-0 at 4.6 um
- Rotational transitions (MIR-FIR): CO J=1-0 at 2.6mm

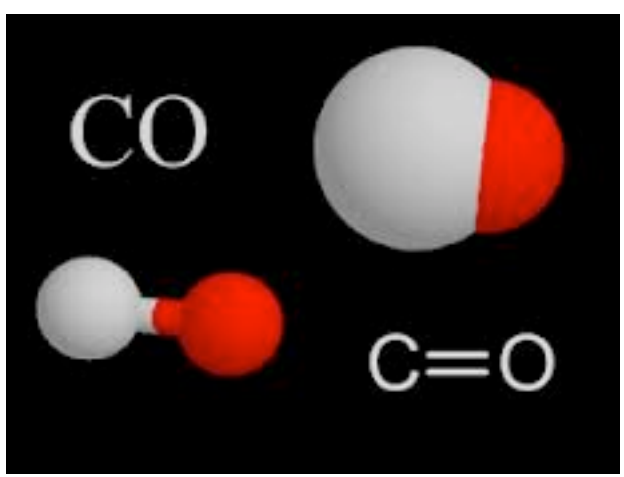




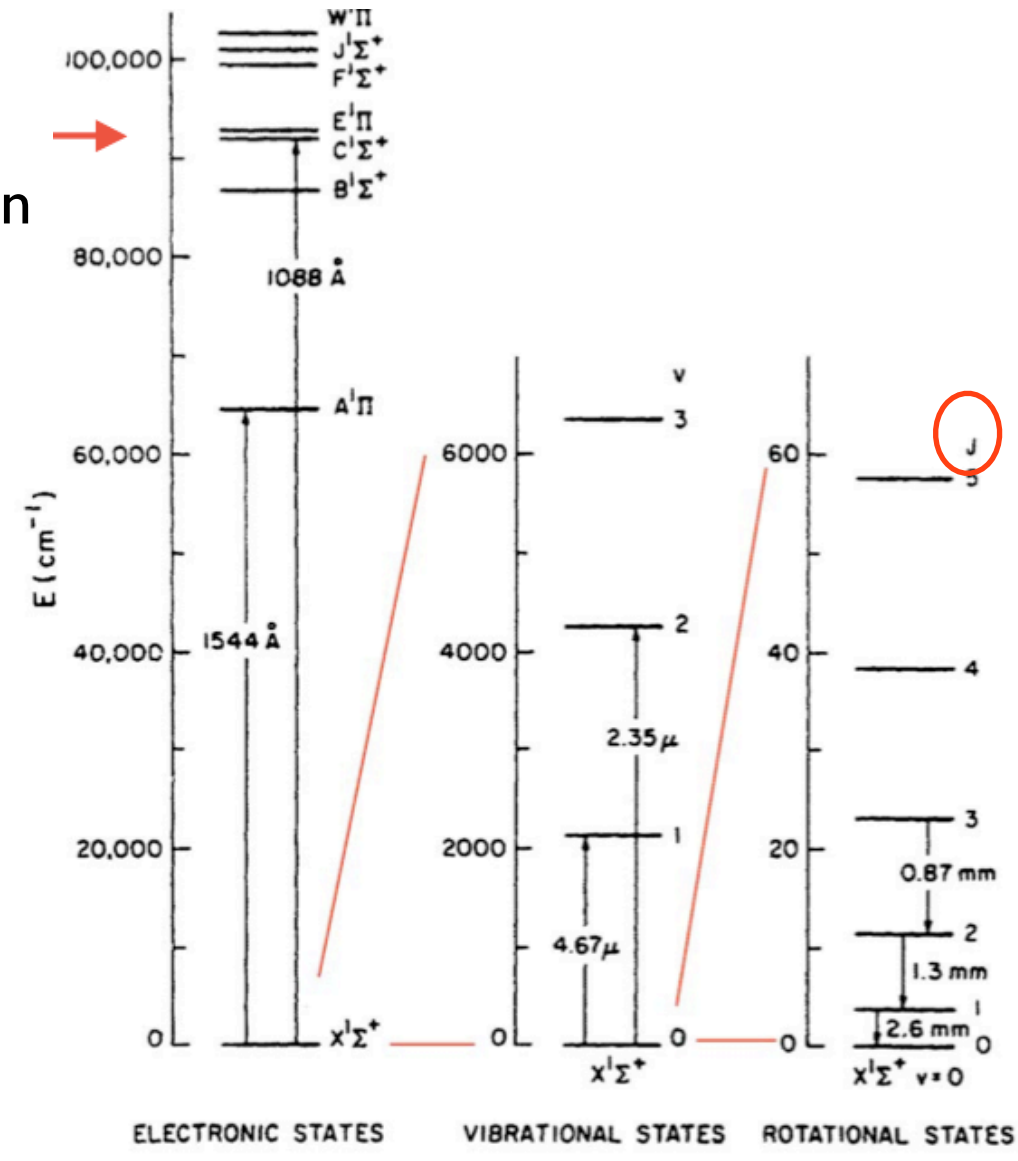
Energy Levels of CO



- H_2 has no dipole moment so CO is more useful



FUV dissociation starts



Critical Density

- Density at which collisions balance spontaneous emission:
- Above the critical density, transition is strongly excited and approaches LTE
- Molecules with different n_{crit} probe different densities
- To determine most useful tracer, consider: n_{crit}, gas temperature, optical depth

$$n_{crit} \approx A_{ij} / \langle \nu \sigma_{ij} \rangle$$

ν = velocity

σ= cross-section

$$n_{crit,CO(1\text{-}0)} \approx 700 \text{ cm}^{-3}$$

$$n_{crit,HCN(1\text{-}0)} \approx 10^5 \text{ cm}^{-3}$$

or ask an observer

Find the molecules: http://home.strw.leidenuniv.nl/~moldata/

LAMDA

Leiden Atomic and Molecular Database

Atomic datafiles | Molecular datafiles | Data format | RADEX

Atomic datafiles

C C^+ O

Molecular datafiles

CO	CS	HCl
OCS	SO	SO_2
SiO	SiS	SiC_2
HCO^+	N_2H^+	HCS^+
HC_3N	HCN	HNC
C_3H_2	H_2O	H_2CO
OH	CH_3OH	NH_3
HDO	H_3O^+	HNCO
NO	CN	CH_3CN
O_2	HF	

The aim of this project is to provide users of radiative transfe codes with the basic atomic and molecular data needed for the excitation calculation. Line data of a number of astrophysically interesting species are summarized, including energy levels, statistical weights, Einstein A-coefficients and collisional rate coefficients. Available collisional data from quantum chemical calculations and experiments are in some cases extrapolated to higher energies.

Currently the database contains data for 3 atomic and 29 molecular species. In addition, several isotopomers and deuterated versions are available. Work is currently underway to add more datafiles. We encourage comments from the users in order to improve and extend the database.

This database should form an important tool in analyzing observations from current and future infrared and (sub)millimetre telescopes. Databases such as these rely

Radiative Transfer Equation I

Change in specific intensity,I, along a ray = emission - absorption

emissivity opacity/absorption

$$\frac{dI_\nu(\omega)}{ds} = j_\nu(\omega) - \alpha_\nu(\omega) I_\nu(\omega)$$

w= direction

Equation for non-overlapping lines*:

$$\frac{dI_{ij}(\omega,\nu)}{ds} = j_{ij}(\omega,\nu) - \alpha_{ij}(\omega,\nu) I_{ij}(\omega,\nu)$$

N=number density
ni=fraction on level i

For line emission:
where i=upper level, j=lower level

$$j_{ij}(w\,\nu) = \frac{h\nu}{4\pi} N n_i A_{ij} \varphi_{ij}(\omega,\nu)$$

$$\alpha_{ij}(\omega,\nu) = \frac{h\nu}{4\pi} N (n_j B_{ji} - n_i B_{ij}) \varphi_{ij}(\omega,\nu)$$

g=statistical weights of the levels

"Einstein A and B Coefficients":

$$A_{ij} = \frac{2h\nu_{ij}^3}{c^2} B_{ij} \qquad B_{ji} g_j = B_{ij} g_i$$

Radiative Transfer Equation II

Line Profile Function:
(For 0 velocity, independent of direction)

$$\tilde{\varphi}_{ij}(\nu) = \frac{c}{a_{\rm tot}\nu_{ij}\sqrt{\pi}} \exp\left(-\frac{c^2(\nu-\nu_{ij})^2}{a_{\rm tot}^2\nu_{ij}^2}\right)$$

Linewidth Thermal:

$$a_{\rm tot} = a_{\rm therm} = \sqrt{\frac{2kT_{\rm gas}}{m_{\rm mol}}}$$ =sound speed

Thermal+Non-thermal:

$$a_{\rm tot} = \sqrt{a_{\rm turb}^2 + a_{\rm therm}^2} = \sqrt{a_{\rm turb}^2 + \frac{2kT_{\rm gas}}{m_{\rm mol}}}$$

"microturbulence"

Line Profile Function:
Non-zero velocities

$$\varphi_{ij}(\omega,\nu) = \tilde{\varphi}_{ij}\left(\nu(1-\vec{\omega}\cdot\vec{v}/c) - \nu_{ij}\right)$$

*RADMC doesn't include overlapping lines in non-LTE

Outline

- Why model lines?
- Molecule basics
- RADMC-3D
- Non-LTE Line Methods
- Project

http://www.ita.uni-heidelberg.de/~dullemond/software/radmc-3d/

RADMC-3D: A new multi-purpose radiative transfer tool

C.P. Dullemond

(main developer: dullemond@uni-heidelberg.de)

Introduction

I have built a new Monte Carlo radiative transfer code called RADMC-3D. It is the 3-D version of the RADMC code. Or better: it is a complete bottom-up rewrite of RADMC. It is written in Fortran 90 (as opposed to F77 for RADMC), it can treat 1-D, 2-D and 3-D geometries in cartesian or spherical coordinates, it has various options for gridding including grid refinement (AMR). And it has a Graphical User Interface for rendering images from a model, written in the IDL language. It has a number of example model setups, including one for a protoplanetary disk model (also with a GUI).

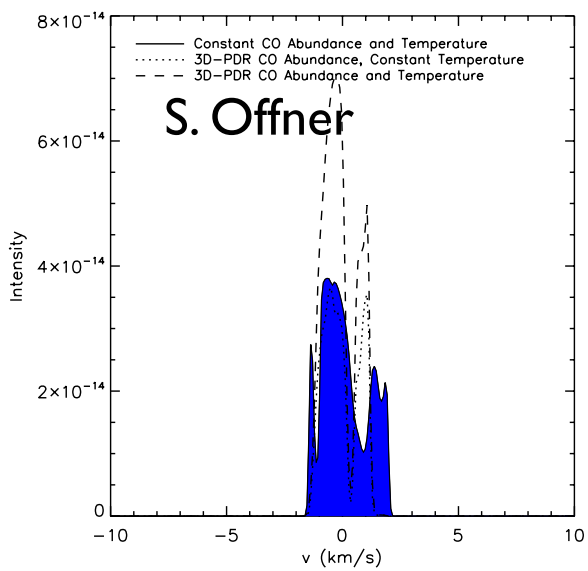
Capabilities and planned capabilities

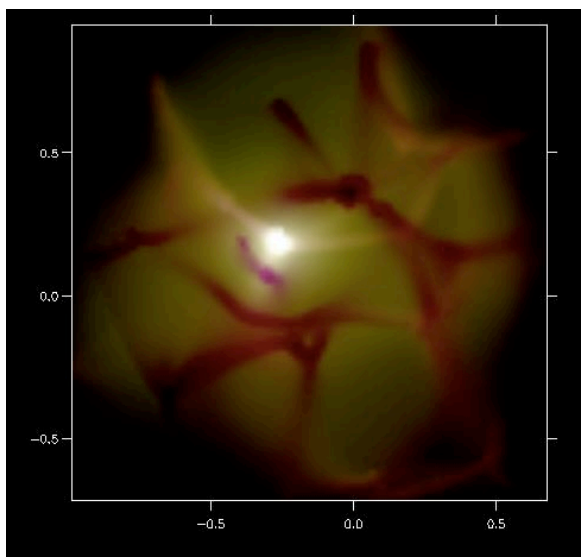
- Dust continuum radiative transfer:
  - Computation of dust temperature and scattering [DONE]
  - Any number of dust species, each having its own spatial distribution [DONE]
  - Anisotropic scattering (several methods to deal with large data volume) [DONE]
  - Polarized scattering, Stokes vectors [DONE/BETA]
  - Modified Random Walk (MRW) method for very high optical depths [DONE/BETA]
- Gas line transfer: [DONE]
  - LTE or local optically thin non-LTE populations [DONE]
  - LVG (Sobolev approximation) mode for level populations [DONE]
  - "Doppler catching method" for dealing with models with large velocity gradients [DONE]
  - Full non-LTE ALI radiative transfer [planned for 2013/2014]
- Cartesian coordinates (3D) [DONE]
- Spherical coordinates (1D spherical symmetry, 2D axial symmetry or fully 3D) [DONE]
- Adaptive Mesh Refinement (oct tree or patch-based, works in both cartesian and spherical coordinates) [DONE]
- Both ascii and binary I/O (choice by user) [DONE]

Also: A. Juhasz, R. Shetty, T. Peters, B. Commercon

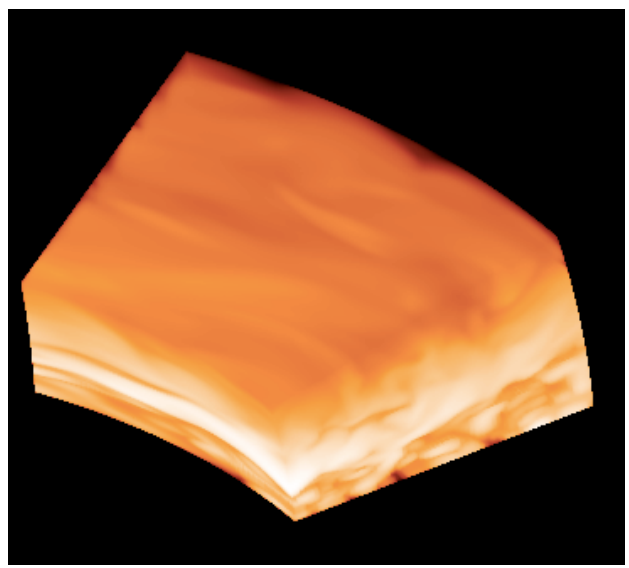
Things RADMC Can Do

- Compute dust temperature (Monte Carlo, tomorrow)
- Compute spectrum or SED (dust or lines)
- Compute an image (raytracing)
- Compute the local radiation field inside the model (monochromatic MC)
- Volume rendering & movie mode

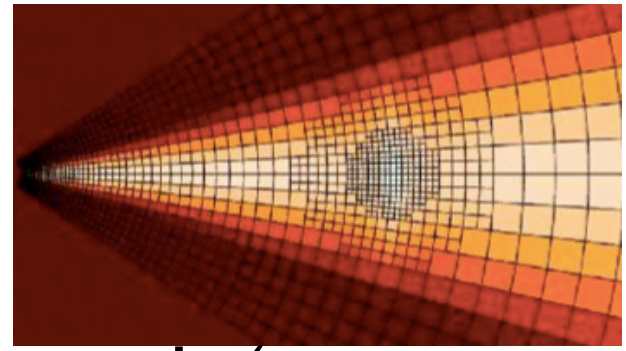




M. Flock

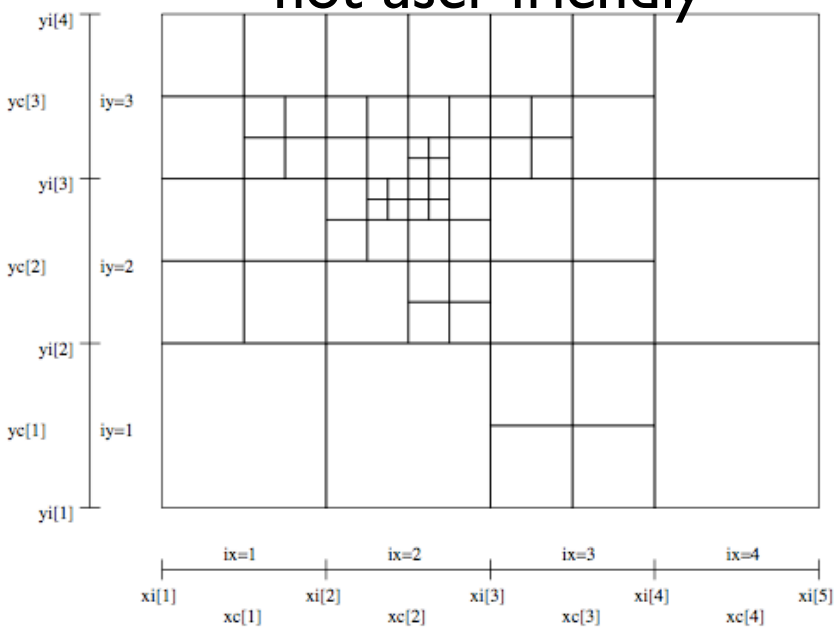


RADMC Grid

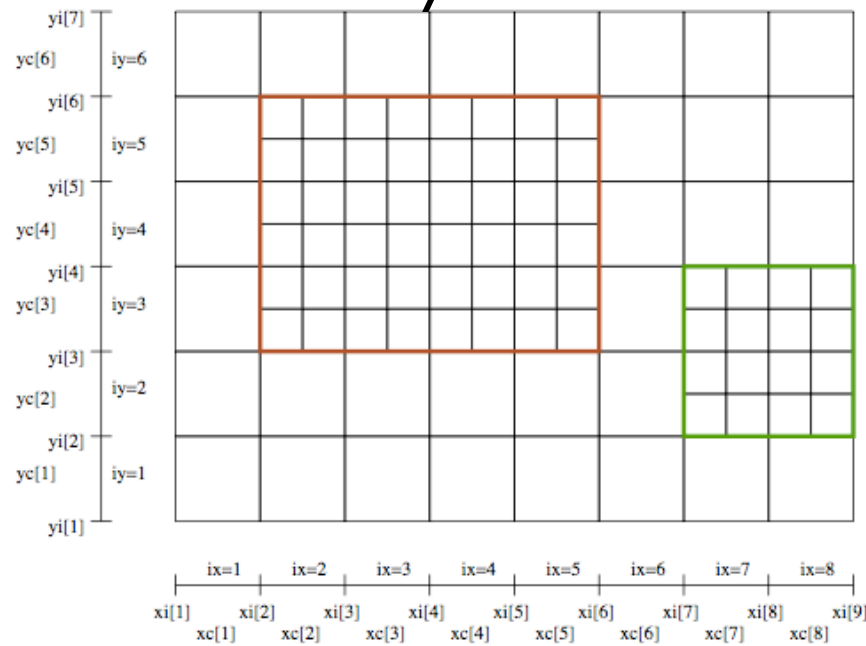


- Cartesian or Spherical (1D, 2D, 3D)
- Spherical can be logarithmic
- Regular grid (grid style 0) or adaptive mesh (oct-tree, grid style 1; "layer style", grid style 10)

oct-tree (FLASH, RAMSES)
"not user friendly"



layer grid (redundant)
"easier to handle by the human brain"



lines_mode

- Local Thermodynamic Equilibrium (1, default)
  - populations computed as a function of gas temperature (collisions dominate)

$$\frac{j_\nu}{\alpha_\nu} = B_\nu(T) \qquad n_i/n_j = g_i/g_j\ e^{-h\nu/kT}$$

- User-defined populations (2)
- Non-LTE: Large Velocity Gradient (Sobolev) (3)
  - requires gas velocities and temperatures
  - calculates angle-averaged velocity gradients

Outline

- Why model lines?
- Molecule basics
- RADMC-3D
- Non-LTE Line Methods
- Project

Non-LTE

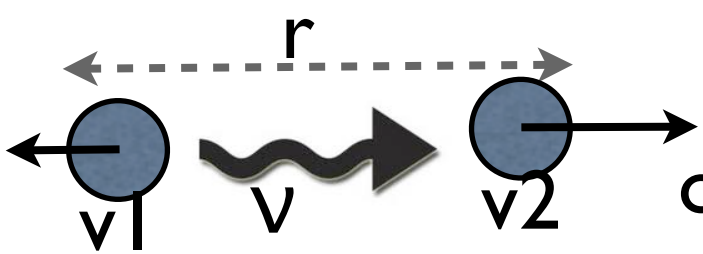
Cij=collisional rate,
depends on collision partner

- In statistical equilibrium, solve equation of detailed balance:

$$\sum_{j>i}[f_j A_{ji} + (f_j B_{ji} - f_i B_{ij})\bar{J}_{ji}] - \sum_{j<i}[f_i A_{ij} + (f_i B_{ij} - f_j B_{ji})\bar{J}_{ij}] + \sum_{j}[f_j C_{ji} - f_i C_{ij}] = 0.$$

- Non-LTE source function is not B(T):

$$\frac{j_\nu}{\alpha_\nu} = S_{ij} = \frac{f_i A_{ij}}{f_j B_{ji} - f_i B_{ij}}$$



Interact
or Escape?

Non-LTE

Cij=collisional rate, depends on collision partner

- In statistical equilibrium, solve equation of detailed balance:

$$\sum_{j>i}[f_j A_{ji} + (f_j B_{ji} - f_i B_{ij})\bar{J}_{ji}] - \sum_{j<i}[f_i A_{ij} + (f_i B_{ij} - f_j B_{ji})\bar{J}_{ij}] + \sum_{j}[f_j C_{ji} - f_i C_{ij}] = 0.$$

- Non-LTE source function is not B(T):

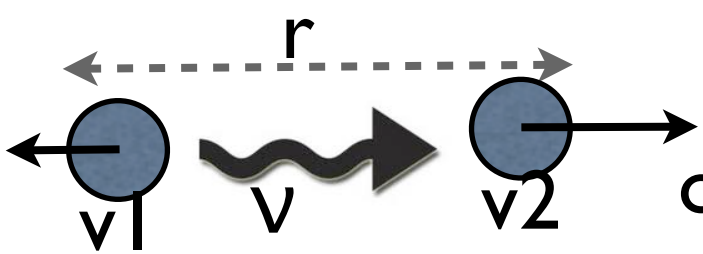
$$\frac{j_\nu}{\alpha_\nu} = S_{ij} = \frac{f_i A_{ij}}{f_j B_{ji} - f_i B_{ij}}$$

- Escape probability of a photon: If (v1-v2)/dr is large, photon is unlikely to be absorbed

$$\beta = \frac{1}{\tau}\int_0^\tau \mathrm{e}^{-\tau'}\mathrm{d}\tau' = \frac{1-\mathrm{e}^{-\tau}}{\tau}$$

$$\tau = \frac{c^3}{8\pi\nu_{ij}^3}\frac{A_{ij}n}{1.065\langle|\mathrm{d}v/\mathrm{d}r|\rangle}\left(f_j\frac{g_i}{g_j} - f_i\right)$$

Effective optical depth

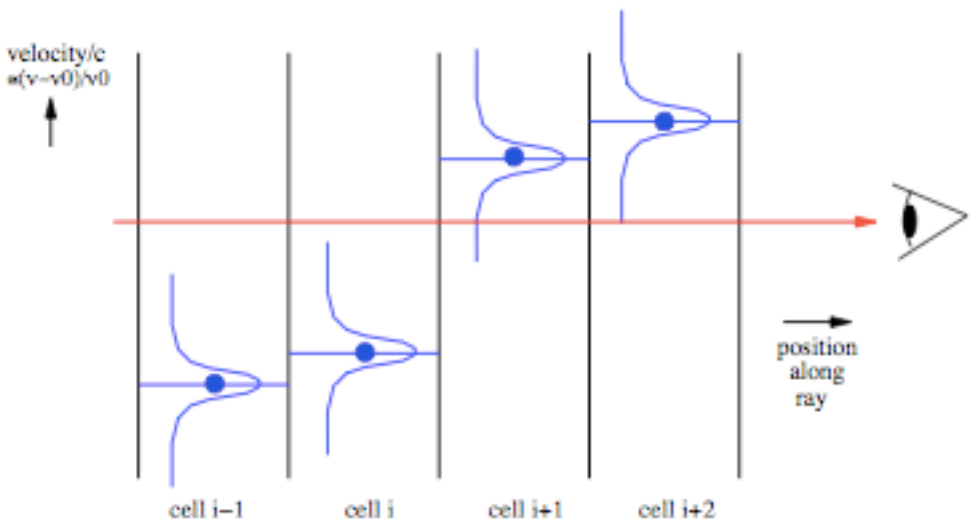


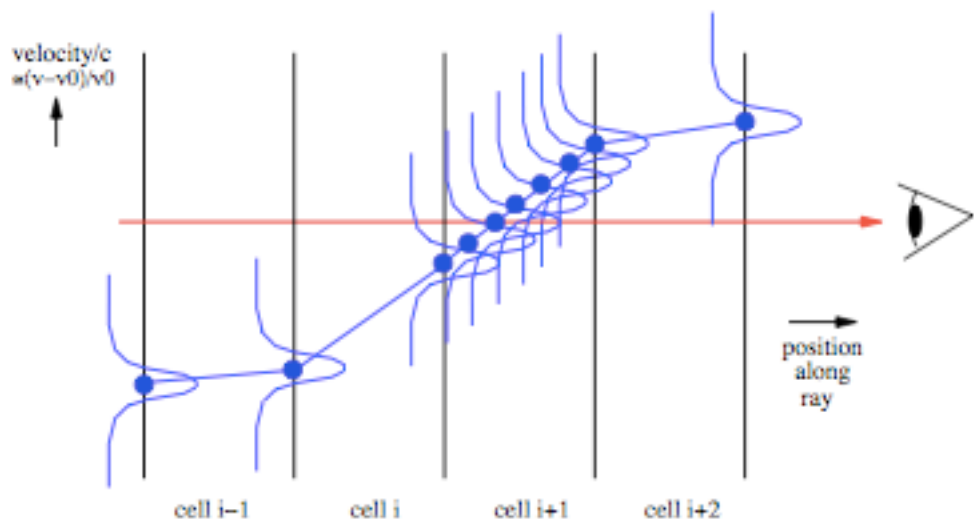
Interact or Escape?

Doppler Catching

- Account for discontinuous velocities (e.g. grid setup)
- Detects jumps and integrates in sub-steps
- doppcatch = 0.2 = doppler shift over each step is not more than 0.2 x local linewidth
- Second order; much slower

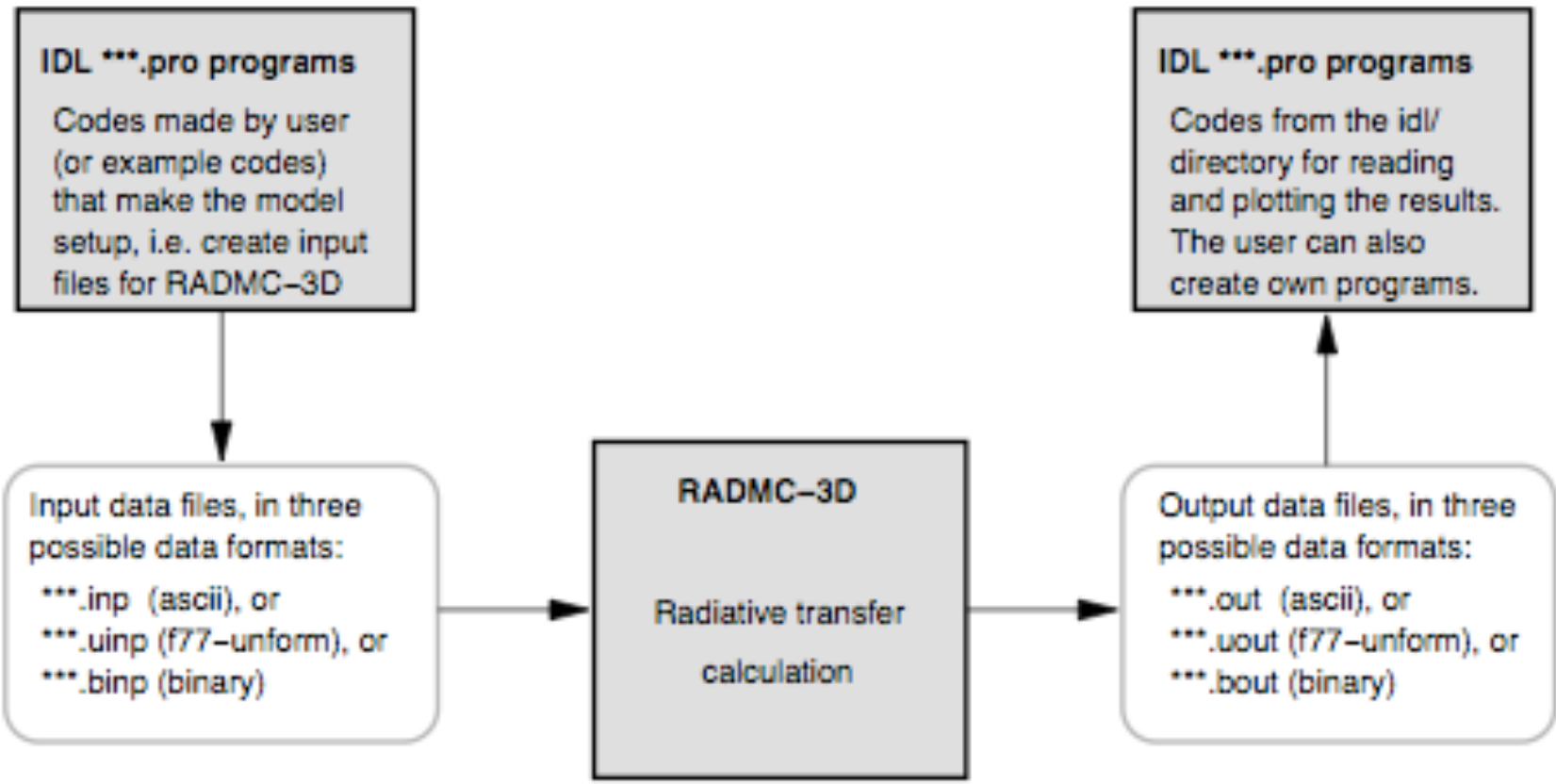
Should I use doppler catching?





RADMC Schematic

Input files can be created using any coding language



RADMC Inputs

- For every grid point: molecule (e.g. CO) densities, temperature, position, velocities
- collisional partner number density (e.g. H2)
- level populations (if not internally computed)
- microturbulence
- molecular line info (e.g. molecule_co.inp)
- wavelength range (for output spectra)
- dust opacity, dust temperature
- radiative transfer parameters (lines_mode, maxJbariter, lvgconvergeto; nphot, dust_to_gas)

By default: dust thermal emission, extinction and scattering is included

Other details

- Not currently parallelized
- 256^3 most reasonable basegrid
- A number of other features are under development (full non-LTE)
- Directly interface with *FLASH*, *RAMSES*, *ZEUS*, *PLUTO*
- Alternative code: Line Modeling Engine (*LIME*), Christian Brinch, http://www.nbi.dk/~brinch/lime.php

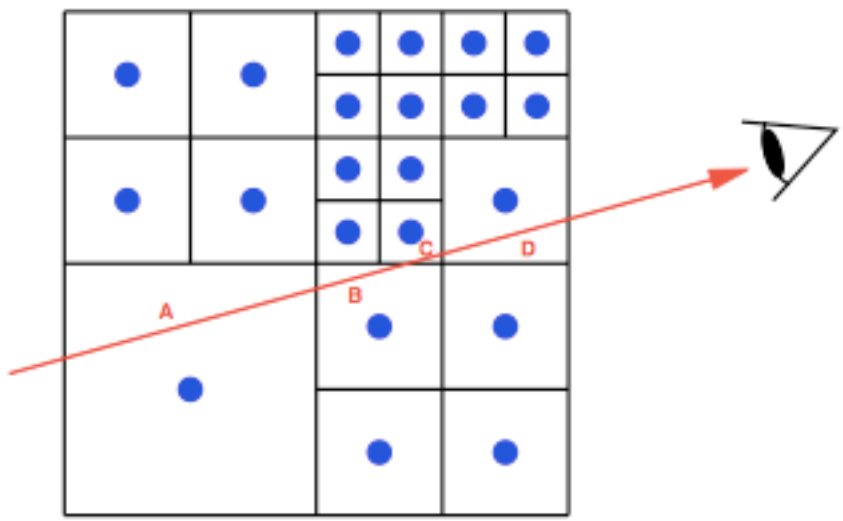
Raytracing

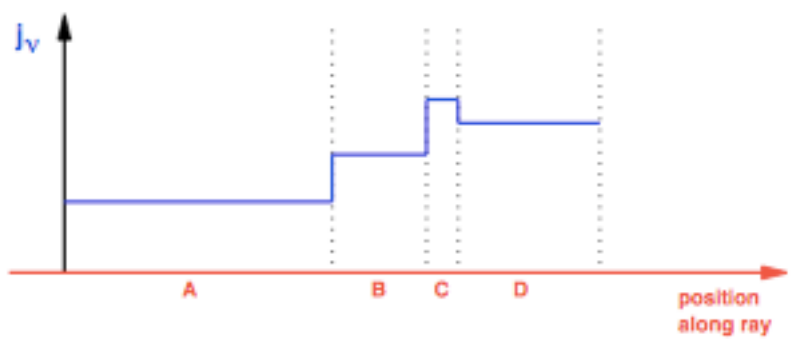
- Default: images and spectra are first order (source term and opacity are constant over cell)

$$I_{\text{result}} = I_{\text{start}} e^{-\tau} + (1 - e^{-\tau}) S \quad \tau = \alpha \Delta s$$

- Can also do second order (e.g. Doppler catching)

1st order





Sample IDL Setup Code

```
@natconst
; Grid parameters
nx      = 64L
ny      = 64L
nz      = 64L
sizex   = 100*AU
sizey   = 100*AU
sizez   = 100*AU
; Make grid with size interpreted as half-width size here
xi      = -sizex + 2*sizex*dindgen(nx+1)/(1.d0*nx)
yi      = -sizey + 2*sizey*dindgen(ny+1)/(1.d0*ny)
zi      = -sizez + 2*sizez*dindgen(nz+1)/(1.d0*nz)
xc      = 0.5 * ( xi[0:nx-1] + xi[1:nx] )
yc      = 0.5 * ( yi[0:ny-1] + yi[1:ny] )
zc      = 0.5 * ( zi[0:nz-1] + zi[1:nz] )
xx      = rebin(xc,nx,ny,nz)
yy      = transpose(rebin(yc,ny,nx,nz),[1,0,2])
zz      = transpose(rebin(zc,nz,ny,nx),[2,1,0])
rrcyl   = sqrt(xx^2+yy^2)
;Write the grid file
openw,1,'amr_grid.inp'
printf,1,1                    ; iformat
printf,1,0                    ;AMR grid style  (0=regular grid, no AMR)
printf,1,0                    ; Coordinate system
printf,1,0                    ; gridinfo
printf,1,1,1,1                 ; Include x,y,z coordinate
printf,1,nx,ny,nz               ; Size of grid
for i=0,nx do printf,1,xi[i]    ; X coordinates (cell walls)
for i=0,ny do printf,1,yi[i]    ;Y coordinates (cell walls)
for i=0,nz do printf,1,zi[i]    ; Z coordinates (cell walls)
close,1
```

The People Who Stare at Code

examples: run_lines_lte_3d_1
problem_setup.pro

```
;
;Write the wavelength_micron.inp file
; For continuum transfer
;(if no dust, don't need to worry about this)
;
lambda1 = 0.1d0
lambda2 = 7.0d0
lambda3 = 25.d0
lambda4 = 1.0d4
n12     = 20
n23     = 100
n34     = 30
lam12   = lambda1 * (lambda2/lambda1)^(dindgen(n12)/(1.d0*n12))
lam23   = lambda2 * (lambda3/lambda2)^(dindgen(n23)/(1.d0*n23))
lam34   = lambda3 * (lambda4/lambda3)^(dindgen(n34)/(1.d0*(n34-1.d0)))
lambda  = [lam12,lam23,lam34]
nlam    = n_elements(lambda)
;
openw,1,'wavelength_micron.inp'
printf,1,nlam
for ilam=0,nlam-1 do printf,1,lambda[ilam]
close,1
;
```

The People Who Stare at Code

For line transfer, can set up a separate finer grid: camera_wavelength_micron.inp

examples: run_lines_lte_3d_1
problem_setup.pro

```
;
;Write the gas temperature file
;
;Also: numberdens_co.inp, gas_velocity.inp,
;microturbulence.inp
;dust_temperature.inp, dust_density.inp
openw,1,'gas_temperature.inp'
printf,1,1                    ; Format number
printf,1,nx*ny*nz             ; Nr of cells
for iz=0,nz-1 do begin
   for iy=0,ny-1 do begin
      for ix=0,nx-1 do begin
         printf,1,gastemp[ix,iy,iz]
      endfor
   endfor
endfor
close,1
```

The People Who Stare at Code

examples: run_lines_lte_3d_1
problem_setup.pro

```
;
;Write the lines.inp control file
;
openw,1,'lines.inp'
printf,1,'1'                    format
printf,1,'1'                    # of species
printf,1,'co    leiden    0    0    0'  ->Looks for molecule_co.inp
close,1                             in leiden format, no collisional partner
;
```

The People Who Stare at Code

examples: run_lines_lte_3d_1
problem_setup.pro

```
;
;Write the lines.inp control file
;
openw,1,'lines.inp'
printf,1,'1'                    format
printf,1,'1'                    # of species
printf,1,'co    leiden    0    0    1'  ->Looks for molecule_co.inp
printf, 1, 'h2'                    in leiden format, no collisional partner
close,1
;              Need to make a density file for each collision partner:
                                numberdens_h2.inp
```

The People Who Stare at Code

examples: run_lines_lte_3d_1
problem_setup.pro

```
;
;Write the lines.inp control file
;
openw,1,'lines.inp'
printf,1,'2'                    format
printf,1,'1'                    # of species
printf,1,'co    leiden    0    0    1'   ->Looks for molecule_co.inp
printf, 1, 'h2'                          in leiden format, no collisional partner
close,1
;
```

Need to make a density file for each collision partner:
numberdens_h2.inp

The People Who Stare at Code

```
;Write the radmc3d.inp control file
;
openw,1,'radmc3d.inp'
printf,1,'nphot = 1000000'
printf,1,'scattering_mode_max = 0'
```

lines_mode default = 1 (lte)
See Page 116-118 in manual for other options

RADMC will assume dust is included if dustopac.inp is present unless you tell it not to include it:
incl_dust = 0

Command Line Call

- Many options are specified in the command line call (See pages 141-145 in the manual)

Create an image at CO (1-0) transition) -> image.out:

radmc3d image lambda 2600.757 incl 60 phi 30

or

radmc3d image iline 1 incl 60 phi 30

Source and examples can be found at:
/home/soffner/radmc-3d/version_0.35/
See also ../manual

Outline

- Why model lines?
- Molecule basics
- RADMC-3D
- Non-LTE Line Methods
- Project

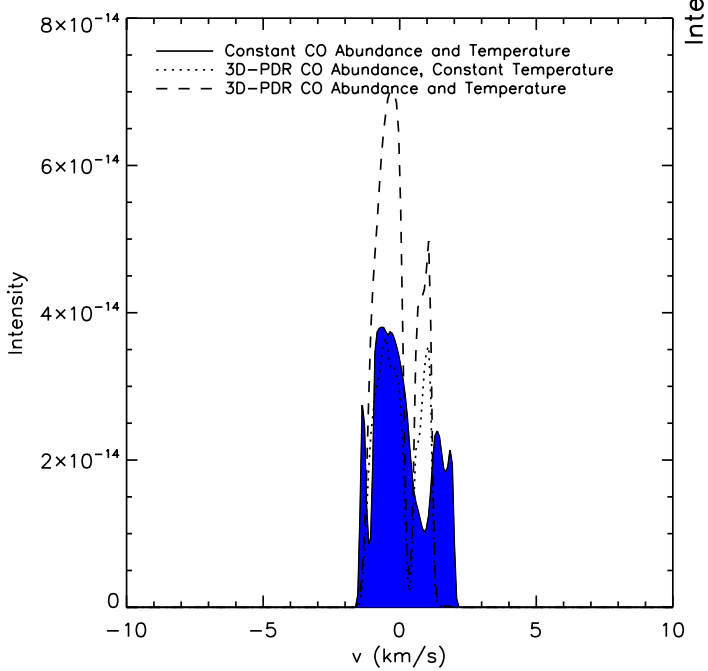
Suggested Project

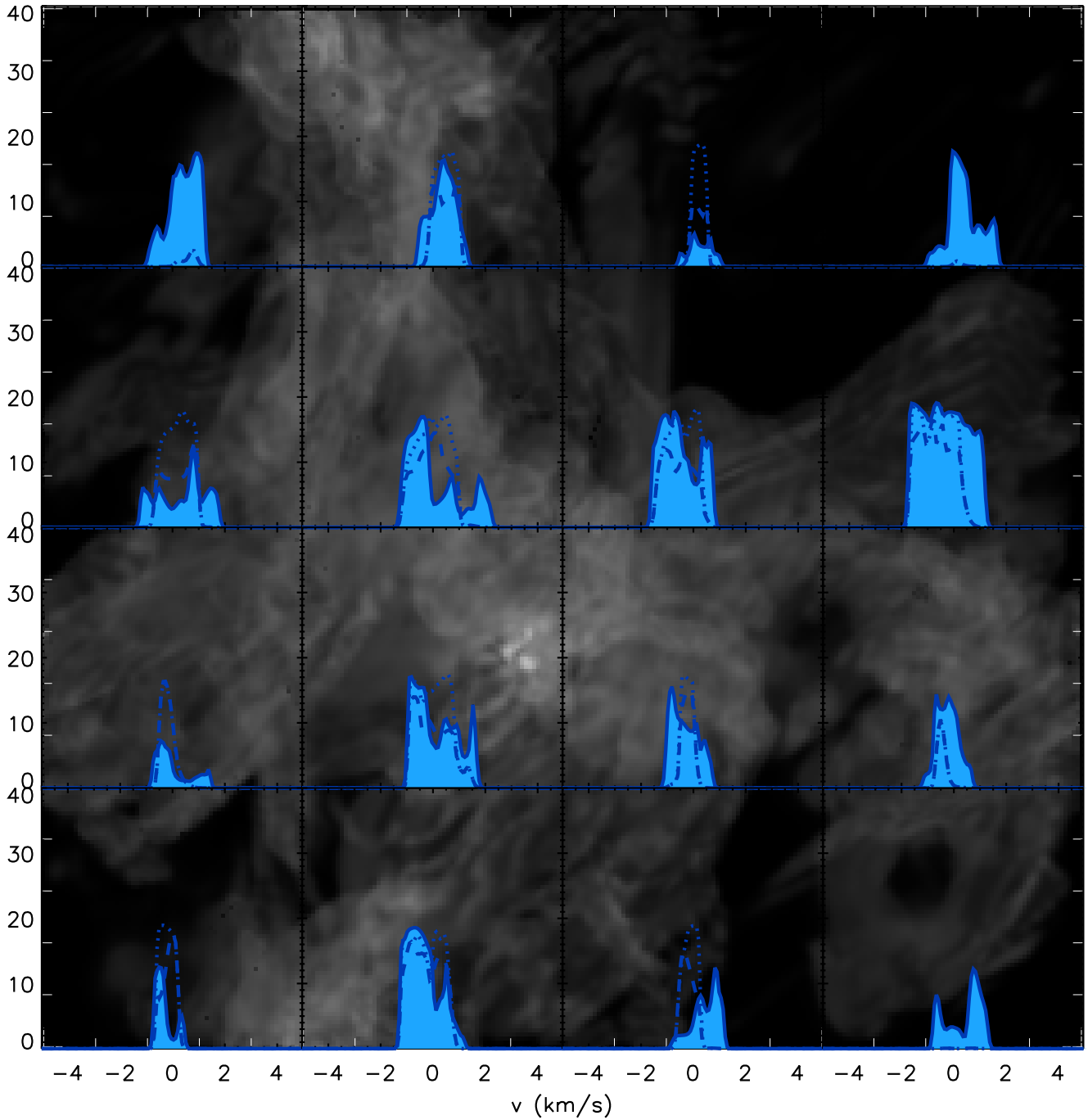
- Pick a molecule from the Leiden database
- Make a setup file for RADMC based on Athena data (I suggest <=128^3)
- Produce a synthetic molecular emission spectra or cube for one transition for that molecule
- Examples at: /home/soffner/radmc-3d/version_0.35/examples

A word about chemistry

Assuming a constant abundance may be too simplistic:

Offner, Bisbas, Viti & Bell in prep.





Useful References

- Journal articles: Dullemond 2011 (RADMC), Dullemond 2012 (RADMC-3D), Shetty et al. 2011 (LVG); Van der Tak et al. 2007 (RADEX), Schoier et al. 2005 (LAMDA)
- Books: *Physics of the Interstellar and Intergalactic Medium*, Bruce Draine
- dullemond@uni-heidelberg.de