Methods for Optimising MC

Tim Harries
University of Exeter

Variance reduction/ optimisation

- Monte Carlo simulations are stochastic and any quantity estimated using MC methods is noisy
- We can improve the estimates by increasing the number of photon packets
- **Or** we can use our knowledge of the physics to improve the signal-to-noise in our simulations
- And use our knowledge of physics to improve the speed of our simulations

Say we have two stars of luminosities L1 and L2 and we want to emit photon packets from our stars. Say we have N photon packets, the pseudo-code might look like:

```
p = L1 / (L1 + L2)
do loop from 1 to N
   r = uniform random deviate [0,1]
   if (r < p) then
      emit from star 1
   else
      emit from star 2
   endif
 end do
```

But we don't need to use the random number here, as we know what fractions of the photons are emitted by L1 and L2.

```
p = L1 / (L1 + L2)
do loop for i = 1 to N
   if (i < N*p) then
        emit from star 1
    else
        emit from star 2
    endif
end do
```

The advantage here is that we haven't introduced unnecessary MC noise…

But what if the objects were very different luminosities? say L1 was a star and L2 was a planet so p ~ 0.9999

```
p = L1 / (L1 + L2)
do loop from 1 to N
   r = uniform random deviate [0,1]
   if (r < p) then
      emit from star
   else
      emit from planet
   endif
 end do
```

In this case the planet signal is very noisy since p is close to unity - can we do better? We could instead produce more packets from the planet than indicated by the luminosity ratio, but give these packets a lower **weight**

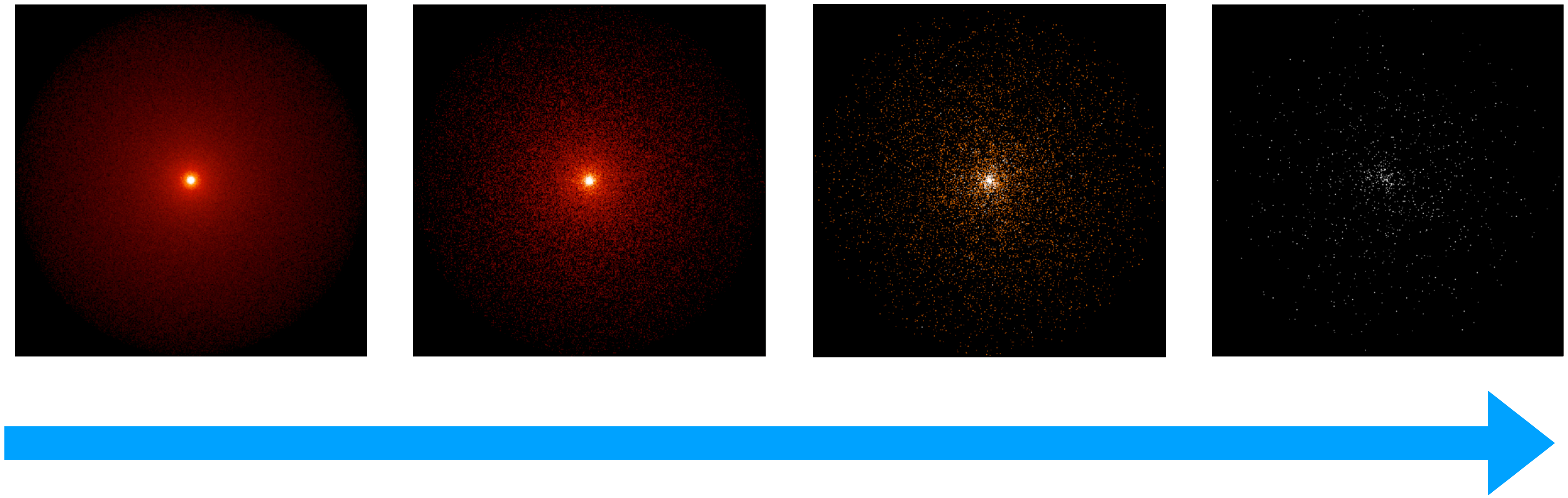
Let's make the probability of photon packets coming from the star c. We then need to weight the photon packets appropriately.

```
p = L1 / (L1 + L2)
c = some fraction
do loop for i = 1 to N
   r = uniform random deviate [0,1]
   if (r < c) then
      w = p / c; emit from star
   else
      w = (1-p)/(1-c); emit from planet
   endif
 end do
```

```
N(star) = c * N * w = N * p
N(planet) = (1-c) * N * w = N * (1-p)
```

So we could set c=0.5, which means that we'd get an equal number of photon packets from the star and the planet, but of course the packets emitted by the planet have a lower weight (conserving energy!)

Let’s take this concept further. Let’s look at scattering in an envelope. If the envelope is very optically-thin most photons will pass straight though - but we might be interested in the signal from the scattered photons:

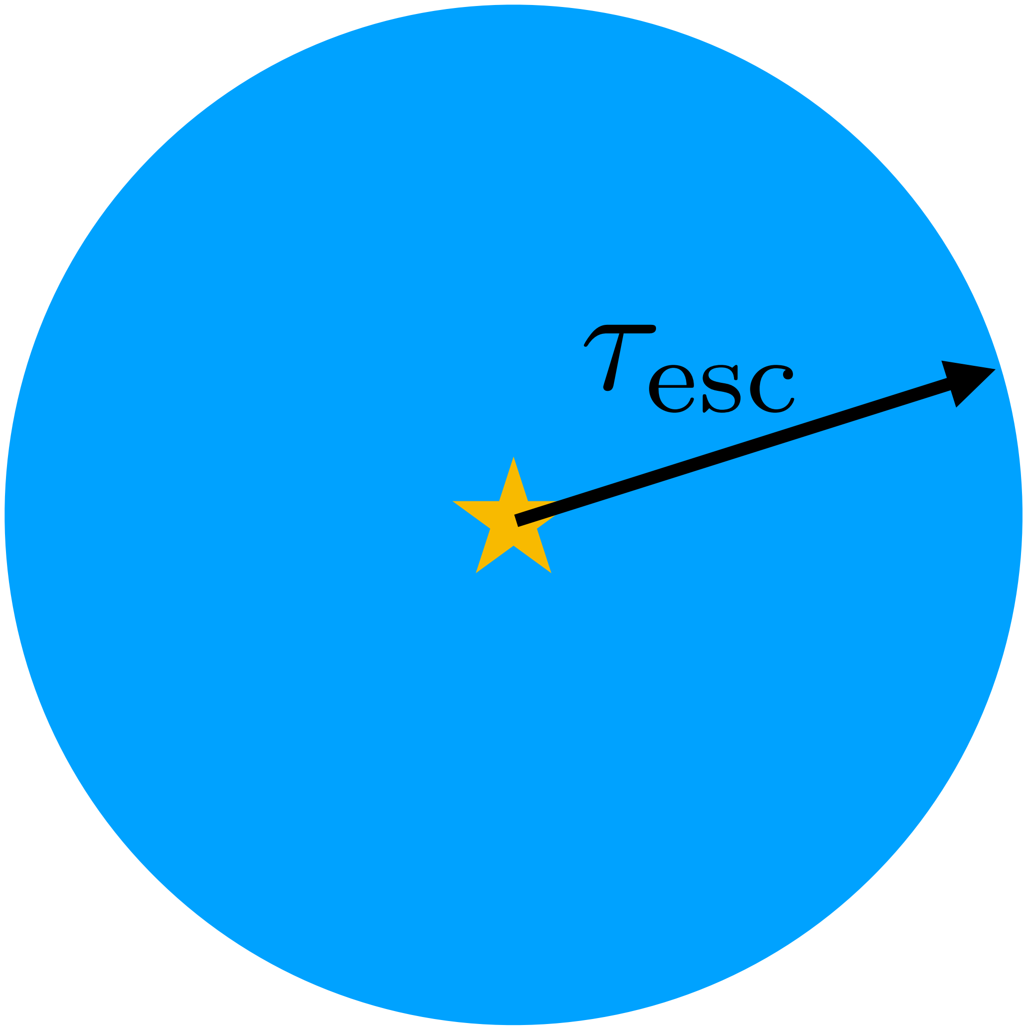


Decreasing envelope density, but same number of packets

The signal-to-noise in the rightmost image is disastrously low - how do we overcome this problem?

Image credit: Tom Robitaille

Forced first scattering:



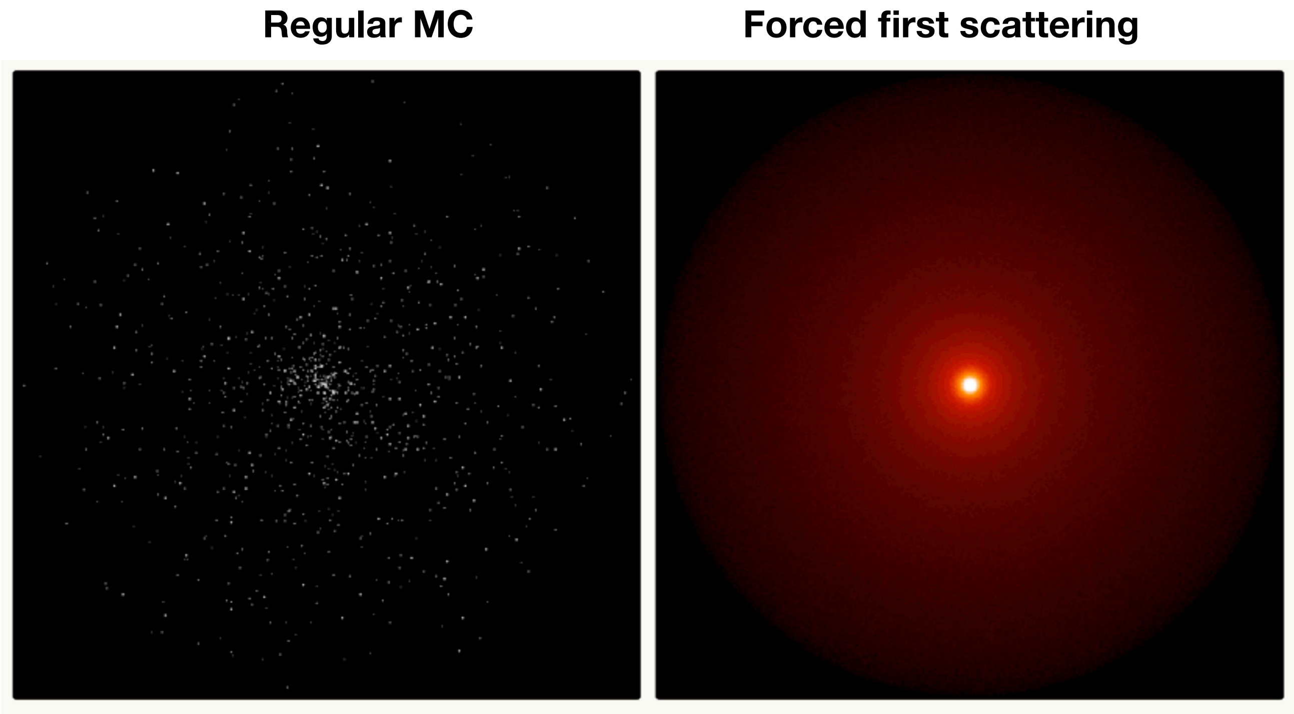
$$w = 1 - e^{-\tau_{\rm esc}}$$

$$r = {\rm rand}[0, 1]$$

Force photon to scatter between 0 and tau_esc

$$\tau = -\log(1 - rw)$$

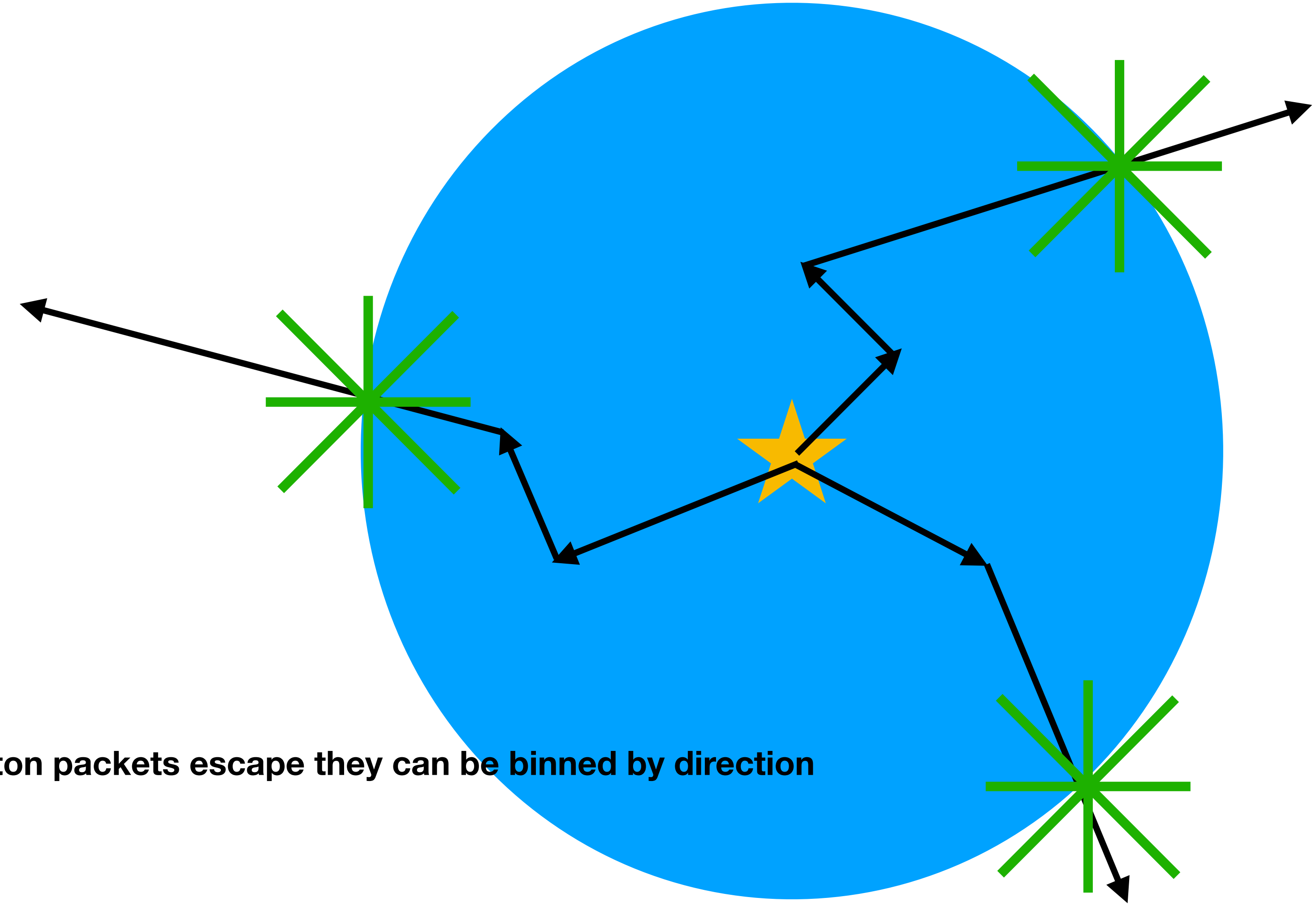
and weight packet by w.



These models have the same number of photon packets, but in the RH image all photon packets contribute to the scattered light

Image credit: Tom Robitaille

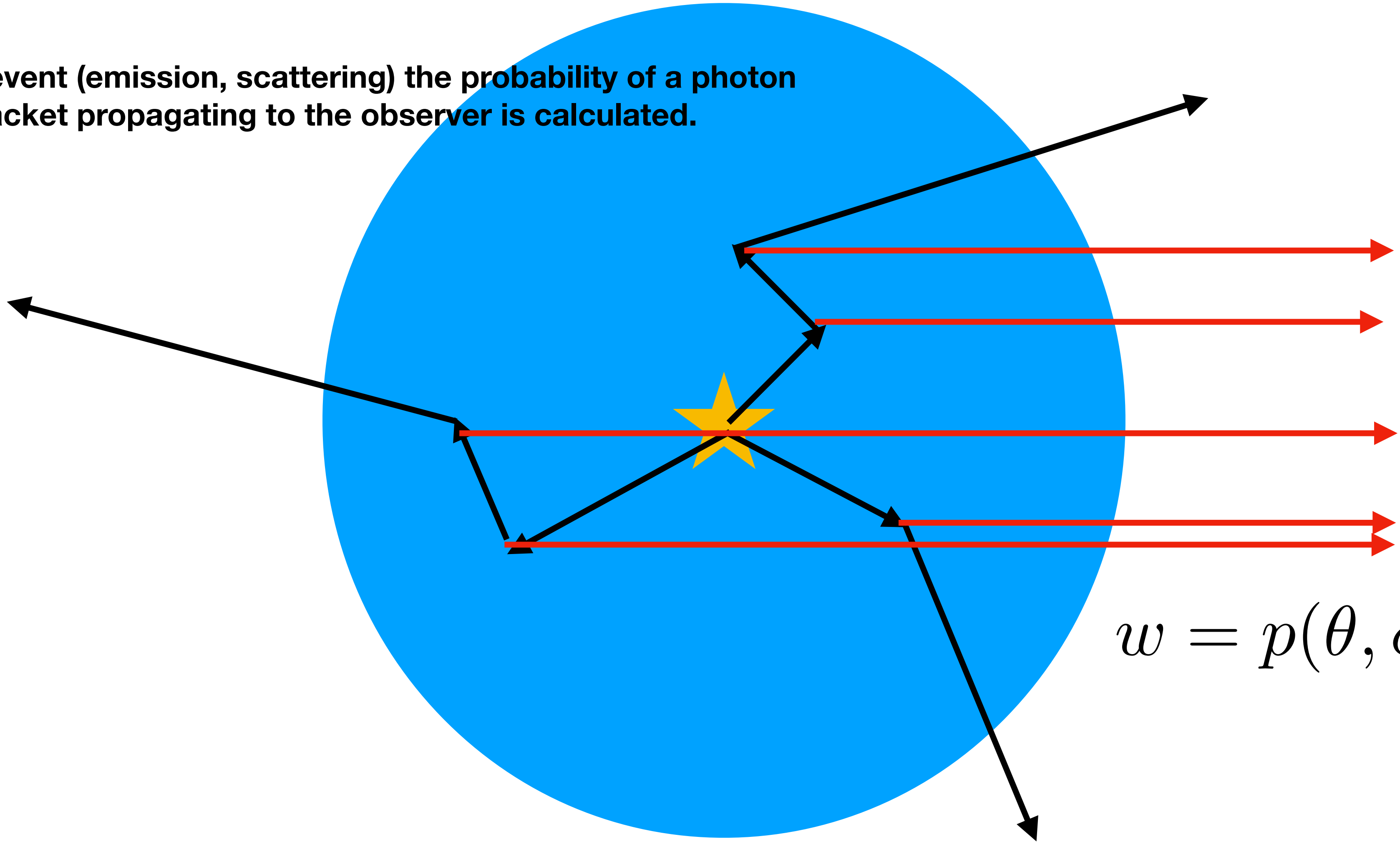
Creating images and SEDS



As photon packets escape they can be binned by direction

Peeling off

At each event (emission, scattering) the probability of a photon packet propagating to the observer is calculated.



$$w = p(\theta, \phi)e^{-\tau_{\rm obs}}$$

What if medium is very optically thick?

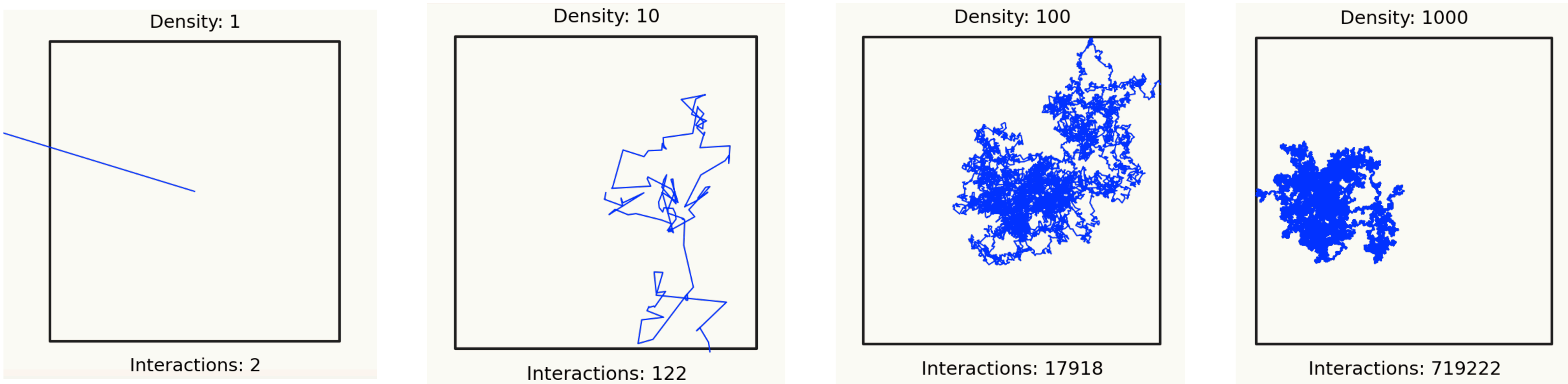
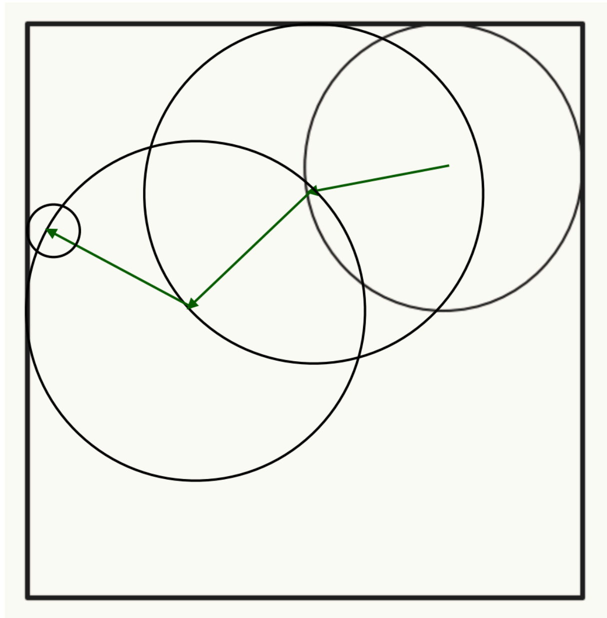


Image credit: Tom Robitaille

Fortunately the probability that a photon is still inside a homogeneous sphere after time t can be determined analytically.

$$P(t) = 2\sum_{n=1}^{\infty}(-1)^{n+1}y^{n^2} \qquad y = \exp\left(-Dct\left(\frac{\pi}{R_0}\right)^2\right)$$

where D is a diffusion coefficient and R_0 is the radius of the sphere



We can then move a photon to the spherical surface, replacing millions of random walk steps by a single one

Min et al. (2009), A&A, 497, 155
Robitallie (2010), A&A, 520, 70

Image credit: Tom Robitaille

Path stretching

$$p(\tau) = e^{-\tau}$$

Remember we are choosing the path length that photons move from this PDF, which is biased towards small tau

$$q(\tau) = \frac{1}{2}\left(e^{-\tau} + \alpha e^{-\alpha\tau}\right)$$

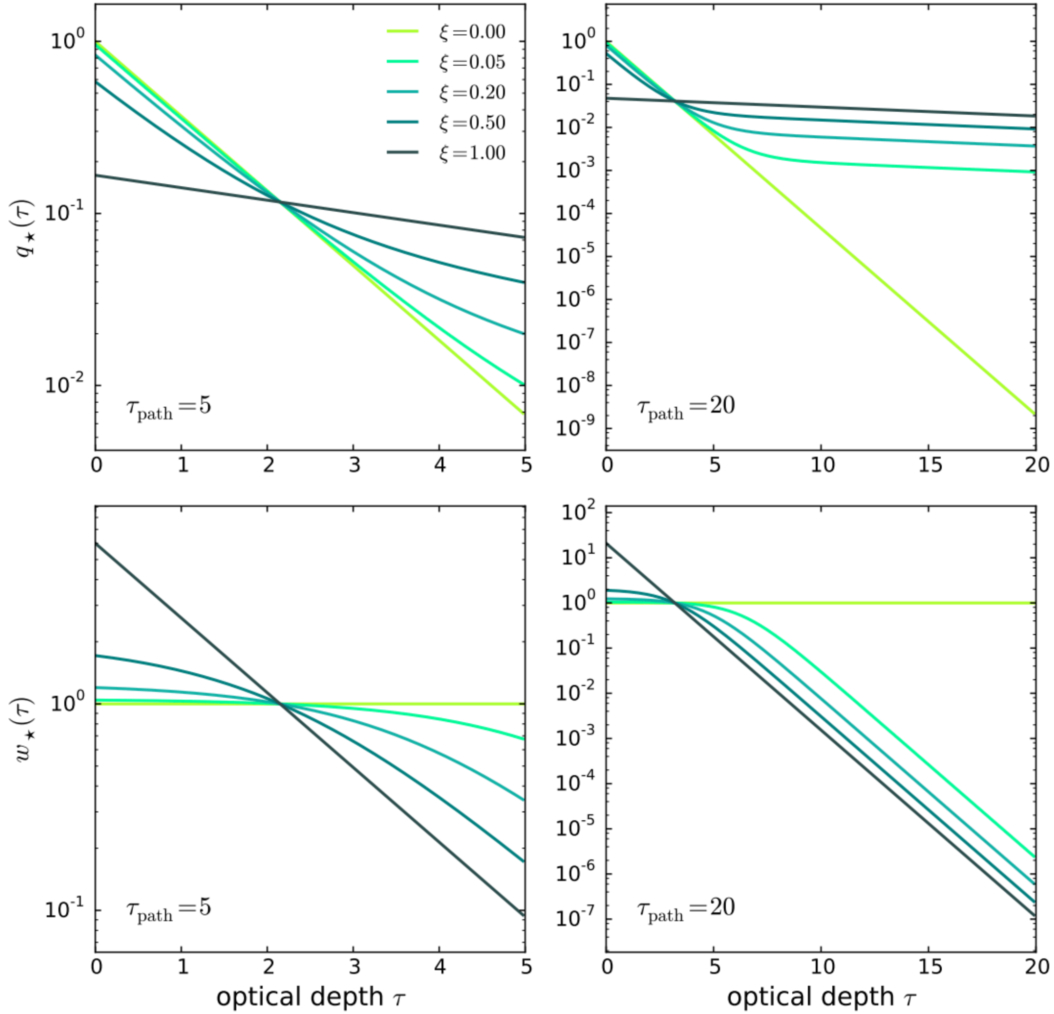
We could modify this to make larger taus more probable

where $$\alpha = \frac{1}{1+\tau_{\rm path}}$$

we must re-weight the packet by

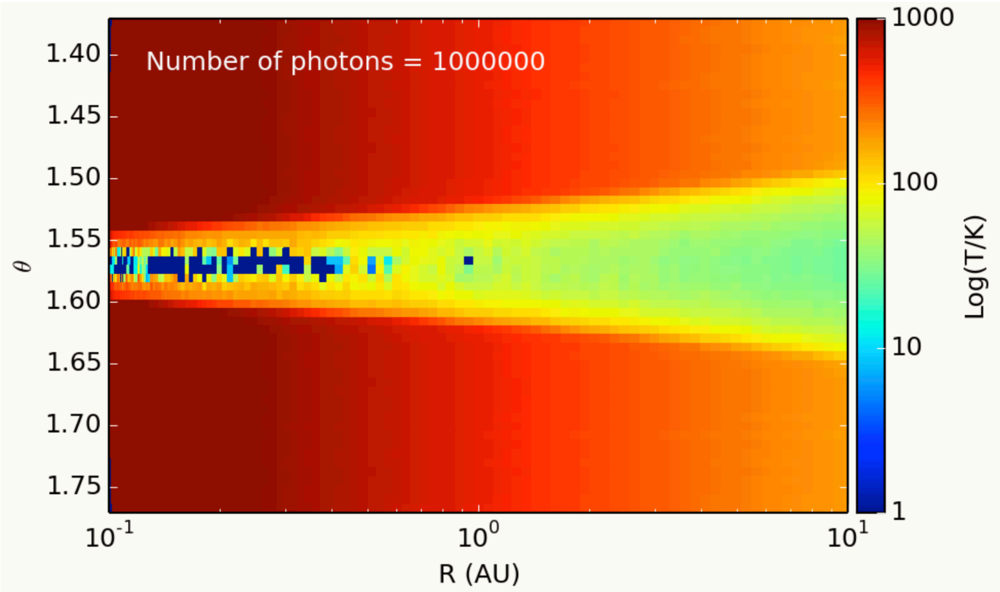
$$w = \frac{p(\tau)}{q(\tau)}$$

Baes et al. 2016, A&A , 590, 55



Baes et al. 2016, A&A , 590, 55

In some cases photon packets just won't penetrate optically-thick regions. Here we can use the diffusion approximation (very fast) with the know temperatures from MC estimates as boundary conditions.



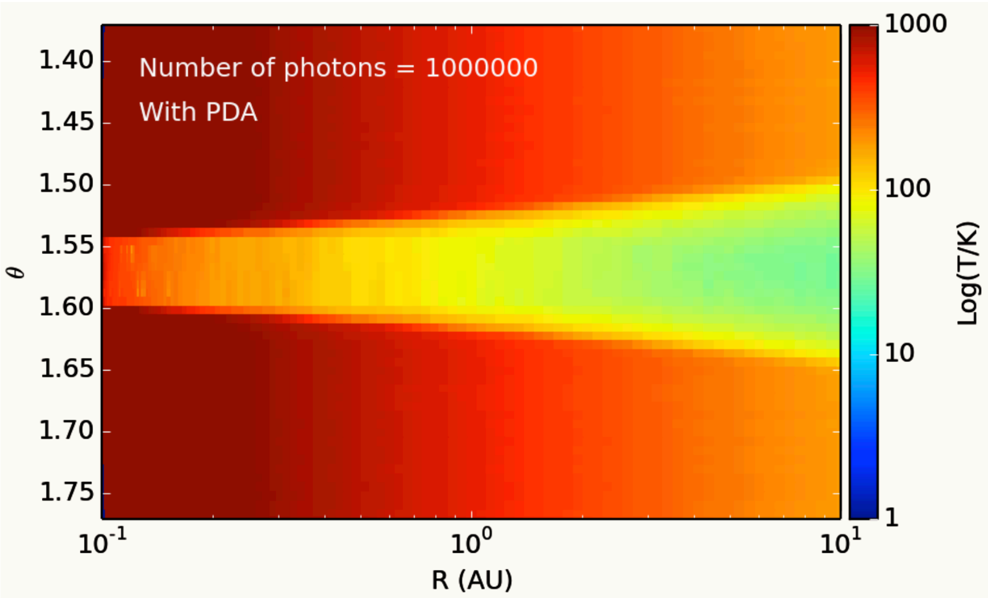
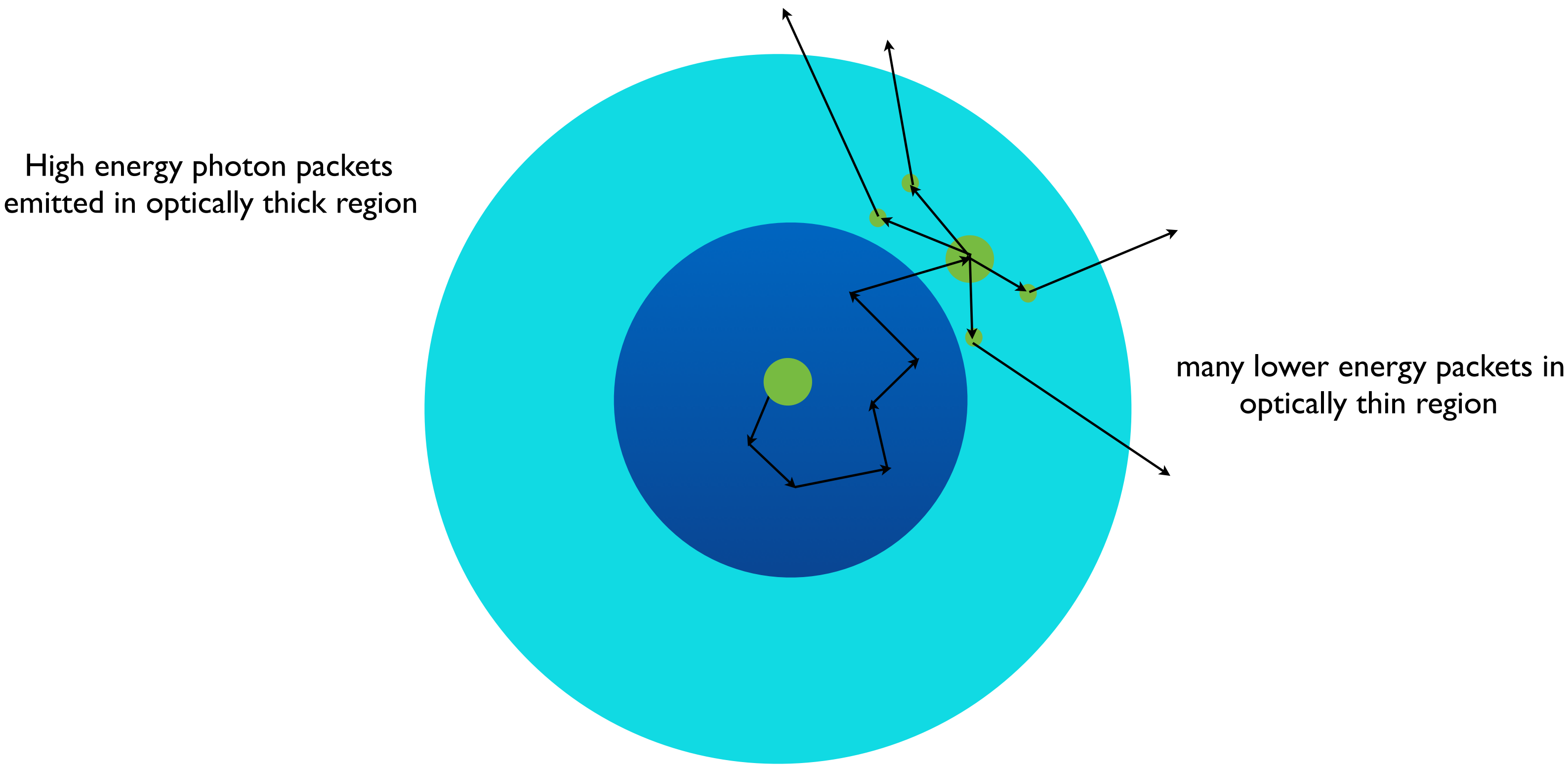
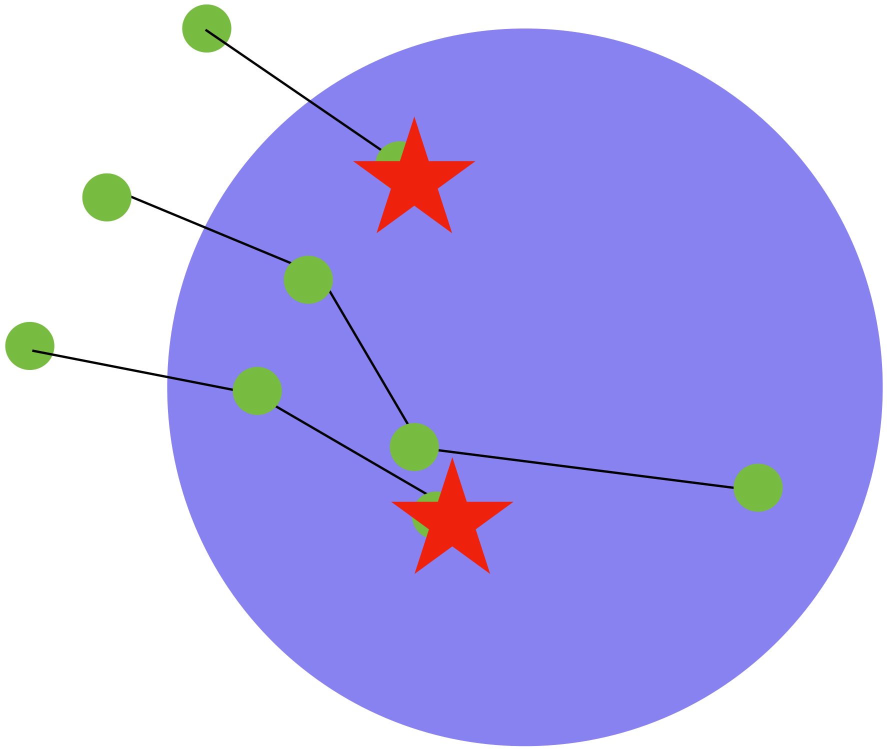


Image credit: Tom Robitaille

Packet splitting



Russian roulette

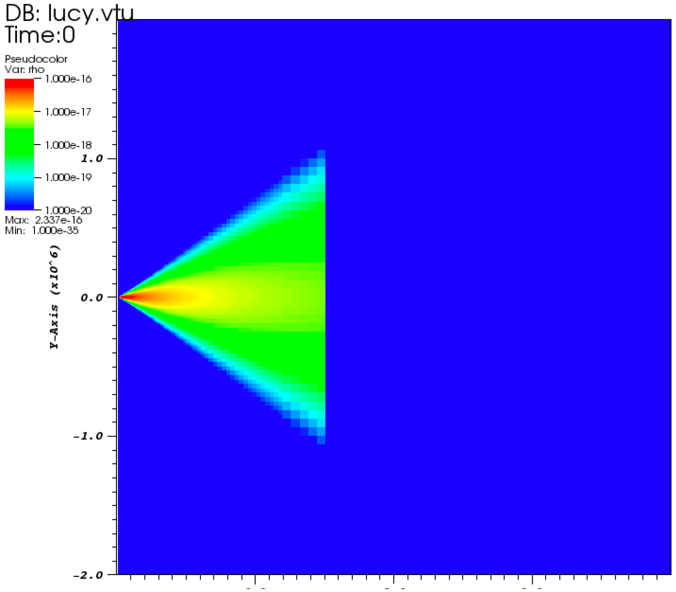


At each step there is a probability p that a photon will be destroyed

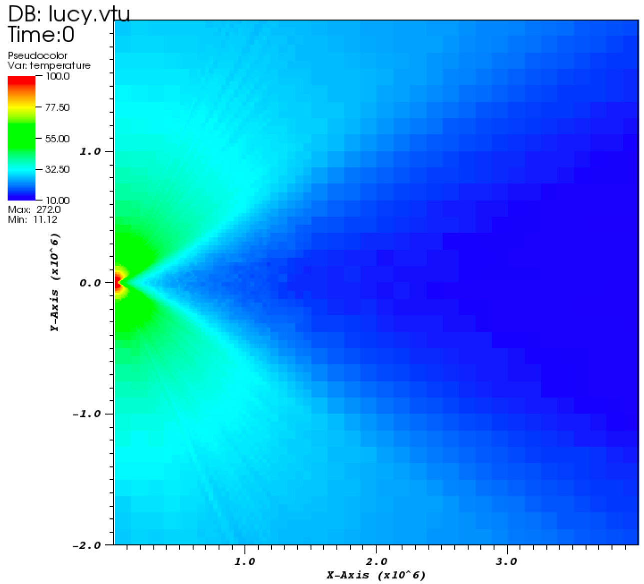
```
r = uniform random deviate [0,1]
if (r < p) then
   kill packet
else
   w_new = w_old / (1-p)
endif
```

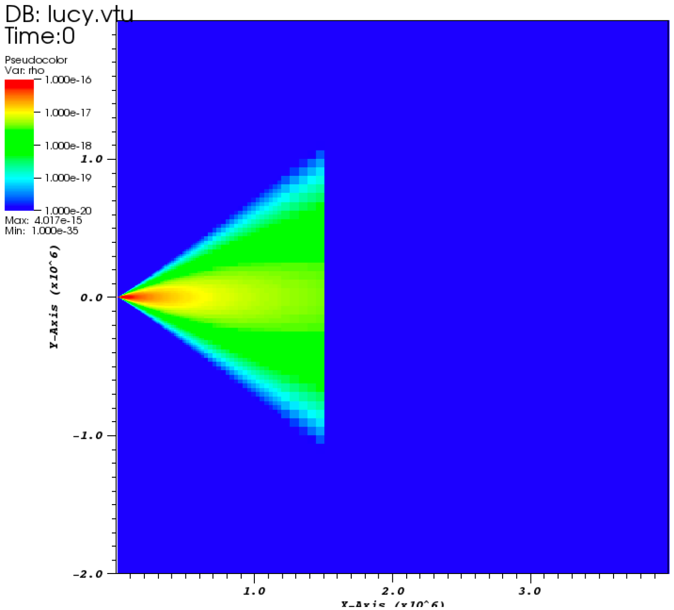
Monte Carlo codes hardly ever crash

They will almost always produce an image or a spectrum…

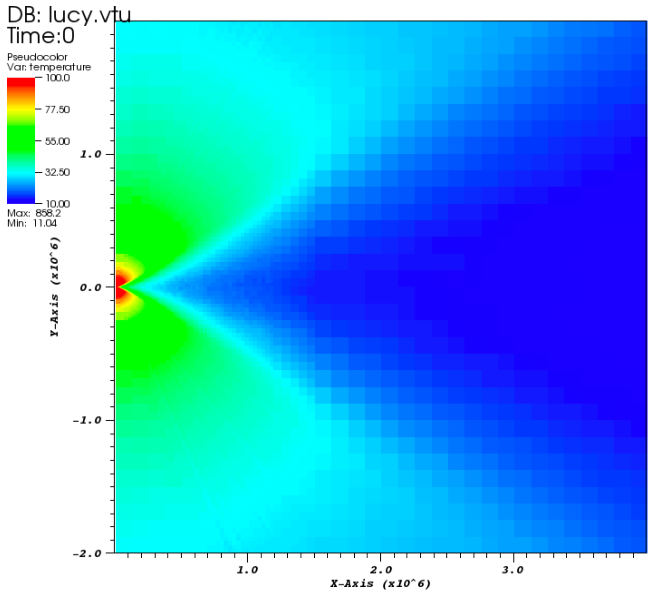


Under resolved

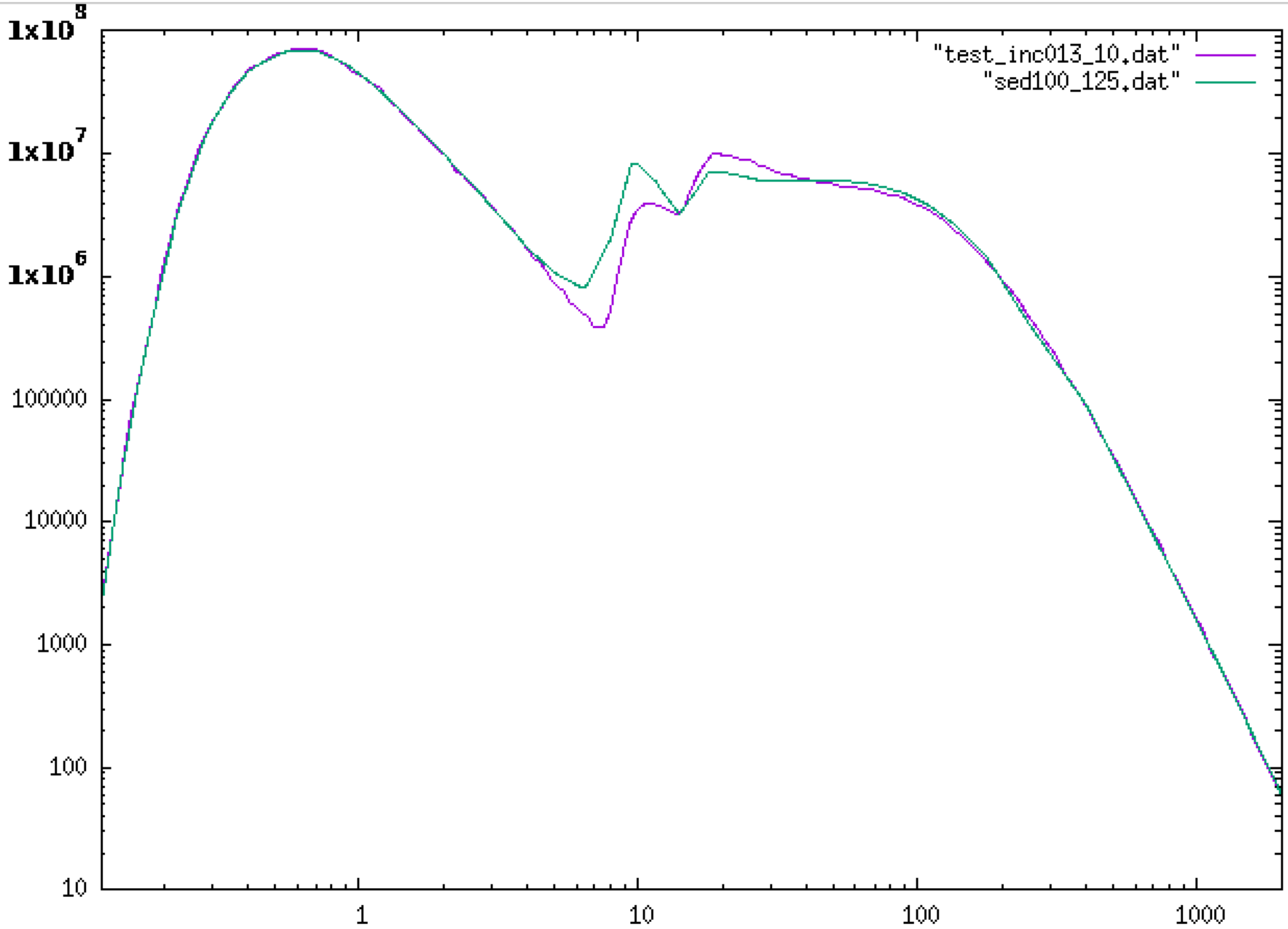




Correct



The under resolved model over-estimates the mid-IR!!



Time-dependent RT, radiation hydrodynamics, and the TORUS code

Tim Harries

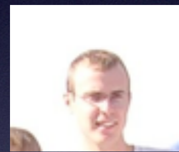
UNIVERSITY OF EXETER

Tim Harries (Exeter)

Tom Haworth (Imperial)

Dave Acreman (Exeter)

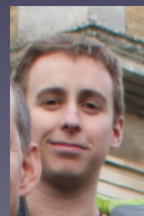
Ryuichi Kurosawa (Netherlands)



Neil Symington (???)

David Rundle (Met Office)

Ahmad Ali (Exeter)

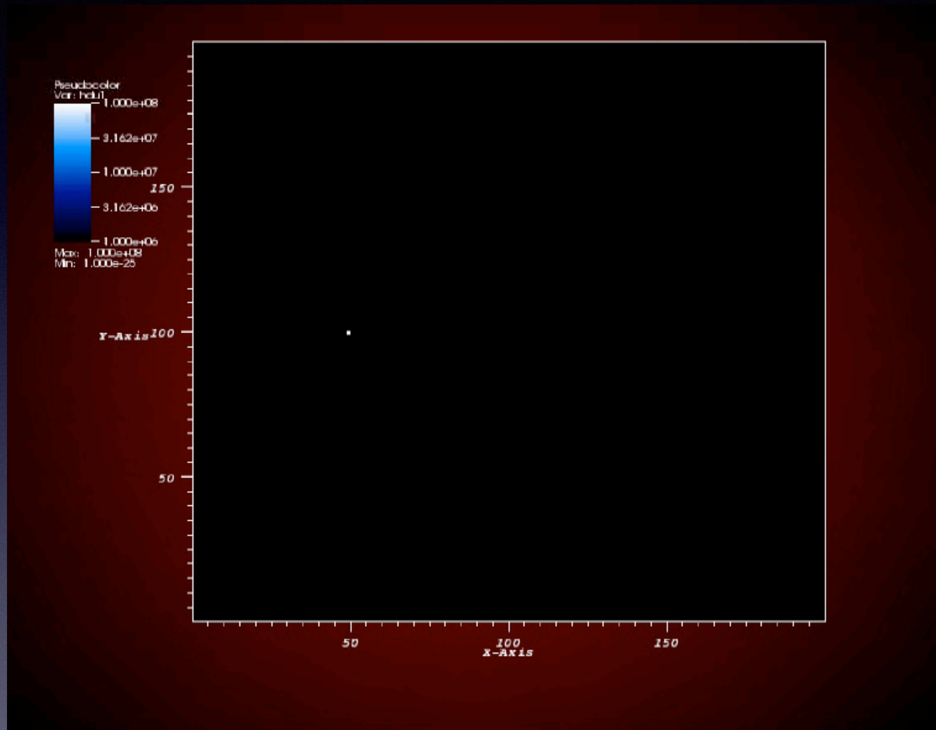


Tom Wilson (Exeter)

Time-dependent RT

- Many interesting phenomena occur out of equilibrium
- Traditionally time-dependent calculations employ flux-limited diffusion
  - Grey
  - Flux-limiter is essentially arbitrary
  - Radiation field can diffuse around obstacles

Harries, 2011, MNRAS, 416, 1500



Pseudocolor
Var: hdu1
1.000e+08
3.162e+07
1.000e+07
3.162e+06
1.000e+06
Max: 1.000e+08
Min: 1.000e-25
150
Y-Axis
100
50
50
100
X-Axis
150

New MC algorithm

- Photon packets are used to determine the radiation energy density
- Photon packet times-of-flight are followed
- Matter interaction terms integrated explicitly
- Method effective in both the optically thick, and crucially, the optically thin (free streaming) limit

Each photon packet has an energy $\epsilon = \frac{L\Delta t}{N}$

A photon packet *i* spends time δt_i in a cell of volume *V*

The packet contributes $\epsilon \frac{\delta t_i}{\Delta t}$ to the energy of the cell

So the energy in the radiation field is

$$\epsilon \sum_{i=1}^{N} \frac{\delta t_i}{\Delta t}$$

So over the duration of the Monte Carlo run the cell has a total energy density

$$u = \frac{1}{V} \frac{1}{\Delta t} \epsilon \sum_{i=1}^{N} \delta t_i$$

We can relate the total path length a photon traverses in a cell to the time via $\delta t_i = \ell_i / c$

$$u = \frac{\epsilon}{\Delta t} \frac{1}{c} \frac{1}{V} \sum_{i=1}^{N} \ell_i$$

The energy density and integrated mean intensity are simply related

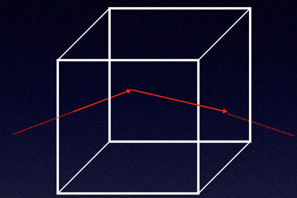
$$u = 4\pi J / c$$

so

$$J = \frac{1}{4\pi} \frac{\epsilon}{\Delta t} \frac{1}{V} \sum_{i=1}^{N} \ell_i$$

$$\dot{E} = 4\pi \int_0^\infty k_\nu B_\nu d\nu \qquad \dot{A} = 4\pi \int_0^\infty k_\nu J_\nu d\nu$$

$$u_{r,\nu} = 4\pi J_\nu d\nu / c$$



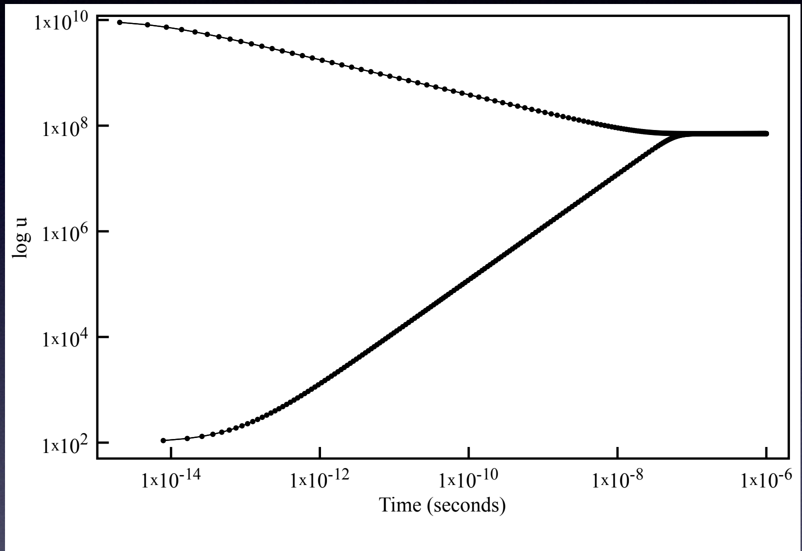
$$u_{r,\nu} = \frac{1}{\Delta t}\frac{1}{V}\frac{1}{c}\sum \epsilon_\nu \ell$$

$$\dot{A} = \frac{1}{V}\frac{1}{\Delta t}\sum k\epsilon\ell$$

$$\dot{u}_r = \dot{E} - \dot{A} \qquad \dot{u}_g = \dot{A} - \dot{E}$$

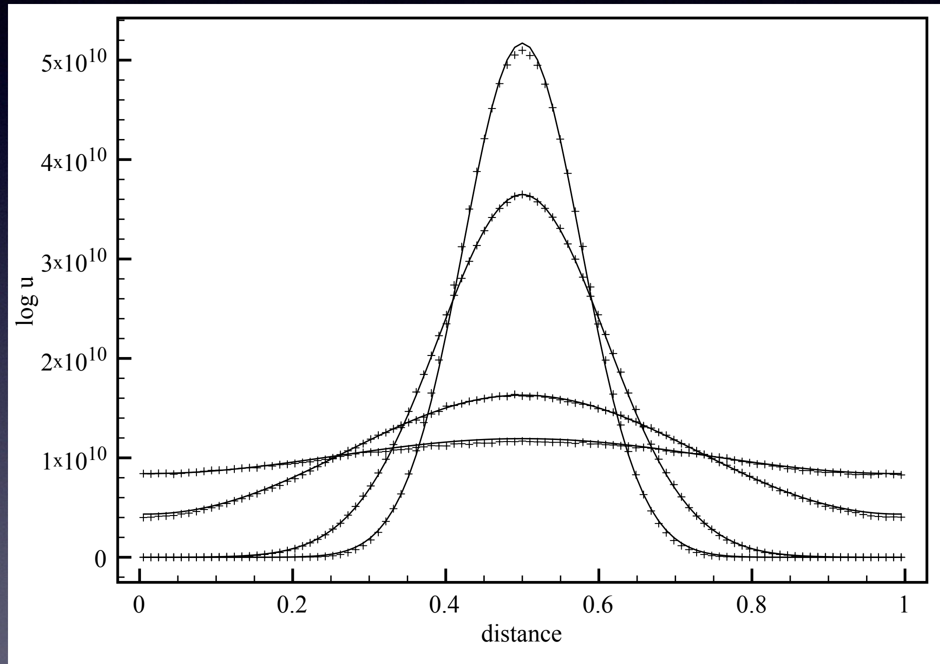
$$u_g^{n+1} = u_g^n + (\dot{A} - \dot{E})\Delta t$$

$$\frac{du_g}{dt} = c\kappa u_r - 4\pi\kappa B(u_g)$$

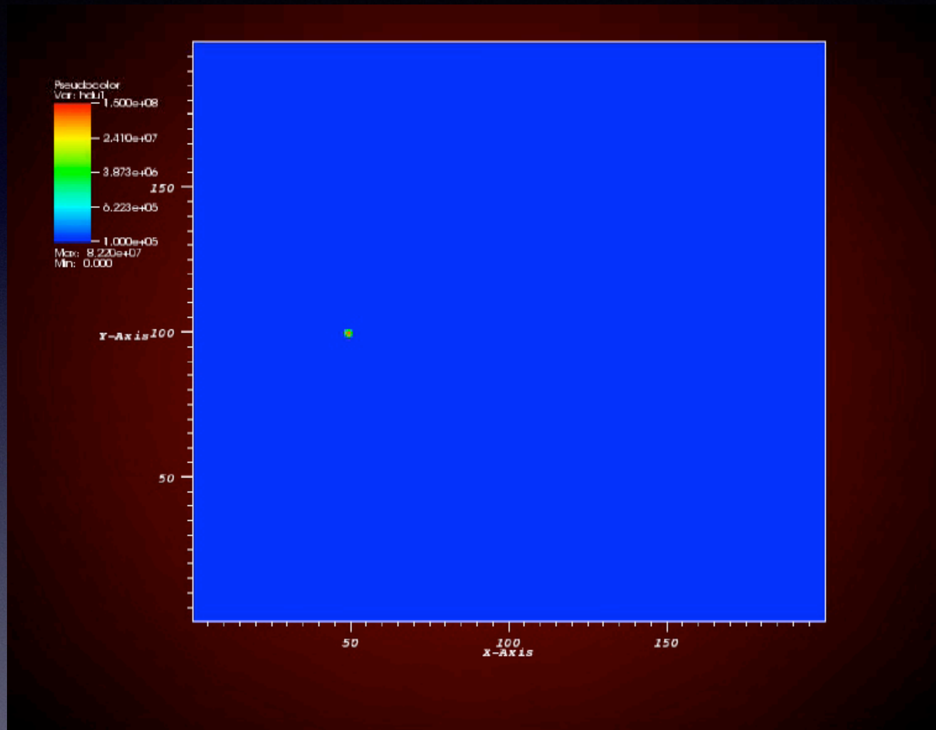


Absorbing gas immersed in radiation field of much higher energy density

$$\frac{du_r}{dt} = -D\frac{d^2u_r}{dx^2} \qquad D = \frac{c}{\kappa}$$



$$u(x,t) = \frac{1}{\sqrt{4\pi Dt}} \exp\left(-\frac{x^2}{4Dt}\right)$$

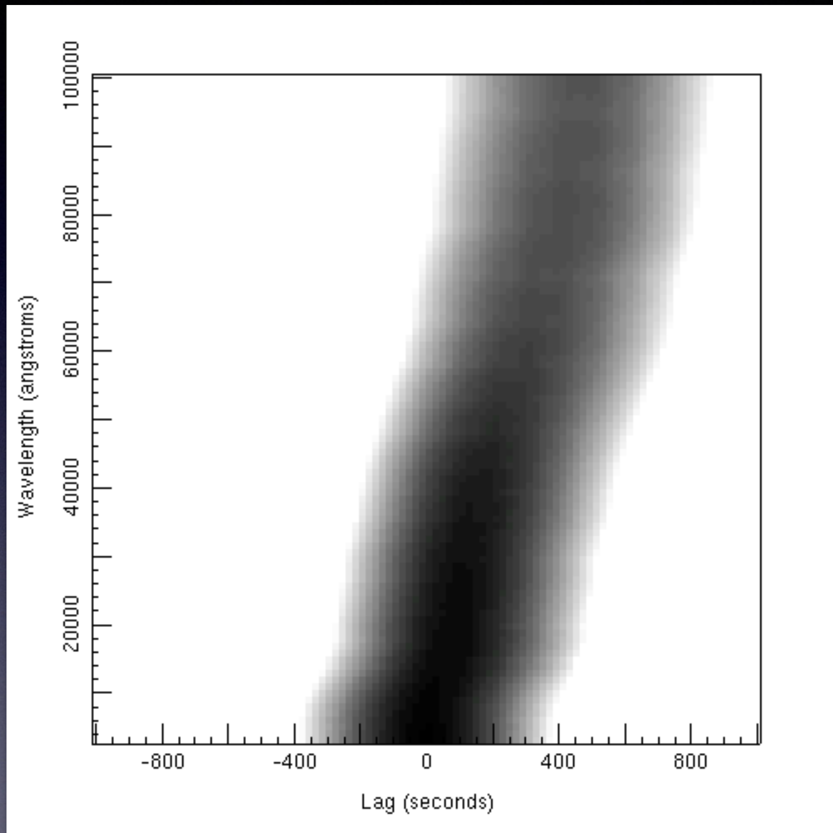

Pseudocolor
Var: hdu1
1.500e+08
2.410e+07
3.873e+06
6.223e+05
1.000e+05
Max: 8.220e+07
Min: 0.000
150
Y-Axis
100
50
50
100
150
X-Axis

An application

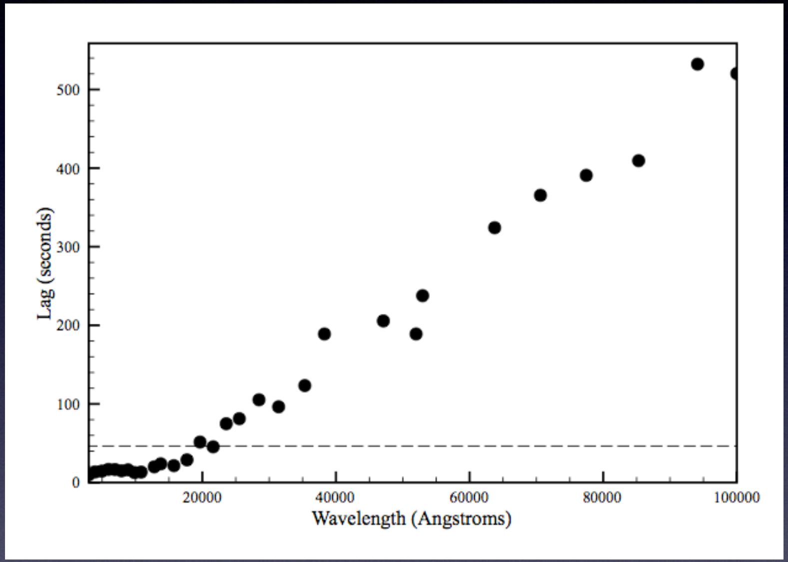
- Standard flared disc (alpha=2.125, beta=1.125, r_{inner}=5 R_*, r_{outer}=300AU, m_{disc}=0.01M_*)
- Illuminated by a typical CTTS (T_{eff} = 4000K, R=2 solar radii)
- Accretion rate sinusoidally varies over a period of 1h (1-5 $\times$ 10^{-8} solar masses per year).

An application

- Additional blue continuum will heat the disc, which will emit more near/mid-IR radiation
- There will be a time-delay between the blue continuum and the disc's response
  - Photon flight time
  - Thermal lag



100000
80000
60000
40000
20000
Wavelength (angstroms)
-800
-400
0
400
800
Lag (seconds)



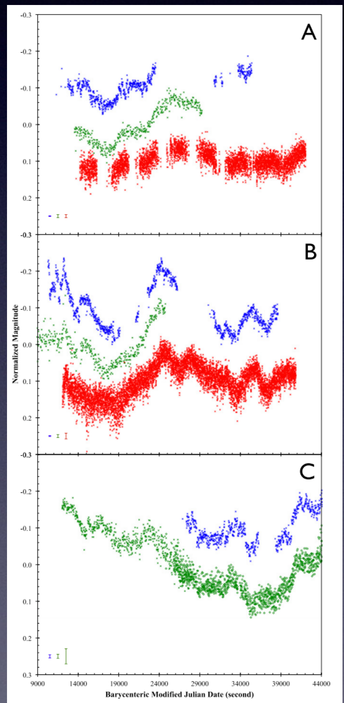
Lag (seconds)
500
400
300
200
100
0
20000
40000
60000
80000
100000
Wavelength (Angstroms)

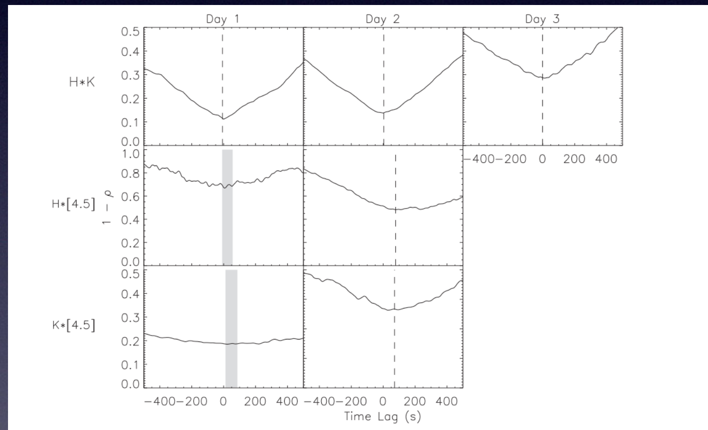
The Astrophysical Journal, 823:58 (15pp), 2016 May 20
doi:10.3847/0004-637X/823/1/58

CrossMark

PHOTO-REVERBERATION MAPPING OF A PROTOPLANETARY ACCRETION DISK AROUND A T TAURI STAR

Huan Y. A. Meng[1,2,3], Peter Plavchan[1,4], George H. Rieke[2,3], Ann Marie Cody[5], Tina Güth[6], John Stauffer[7], Kevin Covey[8], Sean Carey[7], David Ciardi[1], Maria C. Duran-Rojas[9], Robert A. Gutermuth[10], María Morales-Calderón[11], Luisa M. Rebull[7], and Alan M. Watson[12]





A new RHD method

Harries, 2015, *MNRAS*, **448**, 3156

Radiation hydrodynamics

$$\frac{d\rho}{dt} + \rho\nabla \cdot \mathbf{v} = 0$$

Mass conservation

$$\frac{d\rho}{dt} + \nabla \cdot (\rho\mathbf{v}\mathbf{v}) = -\nabla P - \rho\nabla\phi + F_{\mathrm{rad}}$$

Momentum equation

$$\nabla^2\phi = 4\pi G\rho$$

Gravity

$$\frac{dI_\nu}{d\tau_\nu} = S_\nu - I_\nu$$

Radiation transport

Radiation Pressure (I)

momentum per photon packet $$\mathbf{p}_{\text{packet}} = \frac{\epsilon}{c}\hat{\mathbf{u}}$$

The difference in momentum between packets entering and leaving a cell gives net momentum change of a cell

$$\Delta\mathbf{p}_{\text{cell}} = \sum_{m} \mathbf{p}_{\text{packet,in}} - \sum_{n} \mathbf{p}_{\text{packet,out}}$$

m = photon packets entering cell
n = photon packets leaving cell

$$\mathbf{F}_{\text{cell,radiation}} = \frac{\Delta \mathrm{p}_{\text{cell}}}{\Delta t}$$

Radiation Pressure (II)

Alternatively we can use an estimator of the flux to obtain the radiative force on a cell

$$\mathbf{F_{rad}} = \frac{1}{V}\frac{1}{c}\frac{\epsilon}{\Delta t}\sum \ell k_\nu \hat{\mathbf{u}}$$

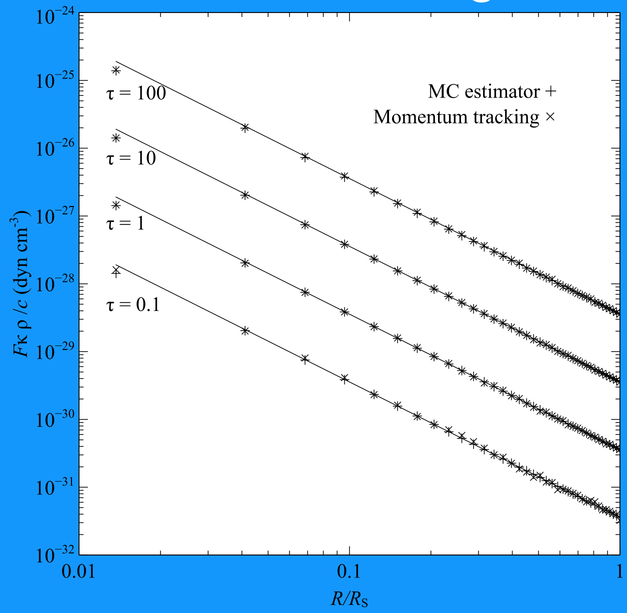
$$k_\nu = k_{\mathrm{abs}} + k_{\mathrm{sca}}(1 - g_\nu)$$

$$g_\nu = \langle \cos\theta \rangle$$

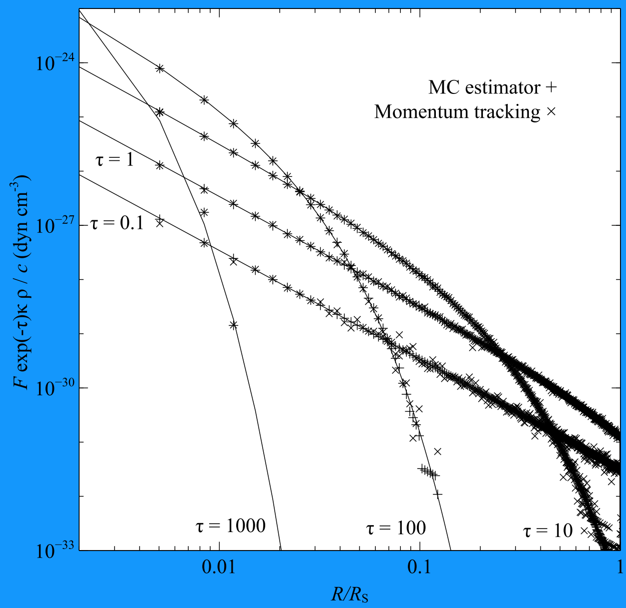
This estimator is better in the optically thin limit.

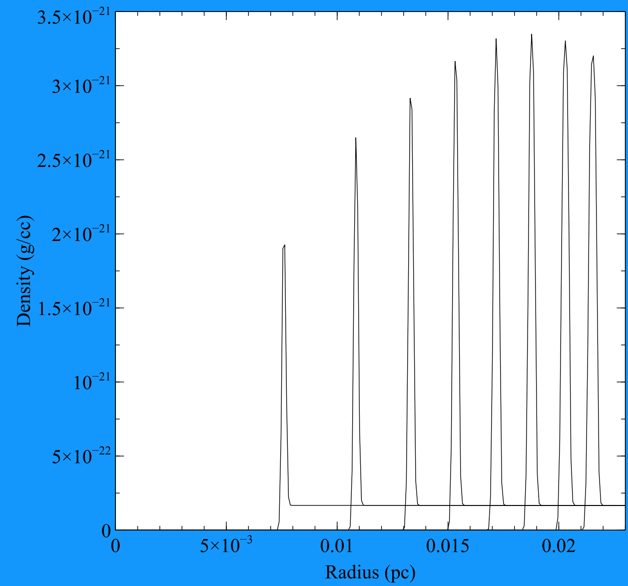
This force is used to update the momentum in the hydrodynamics step

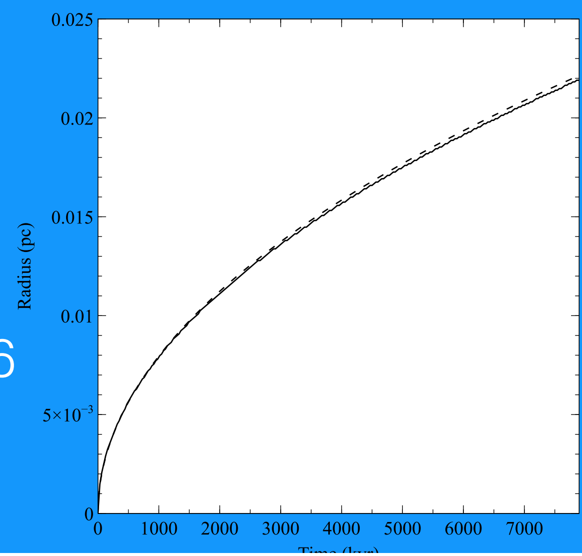
Pure scattering



Pure absorption



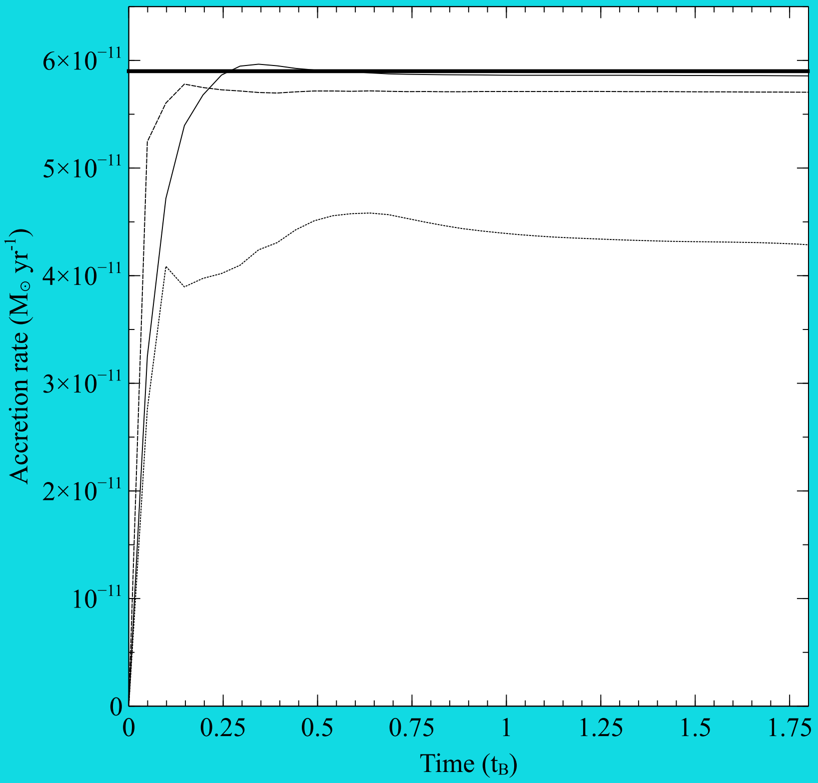




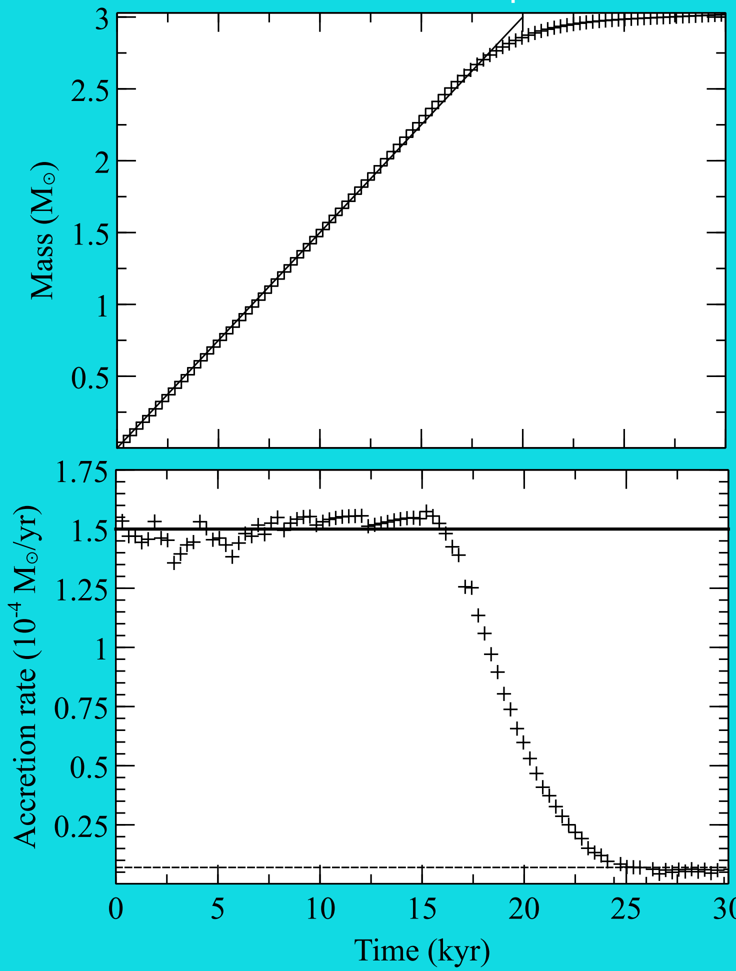
Radiation-driven shell

Harries, 2015, MNRAS, 448, 3156

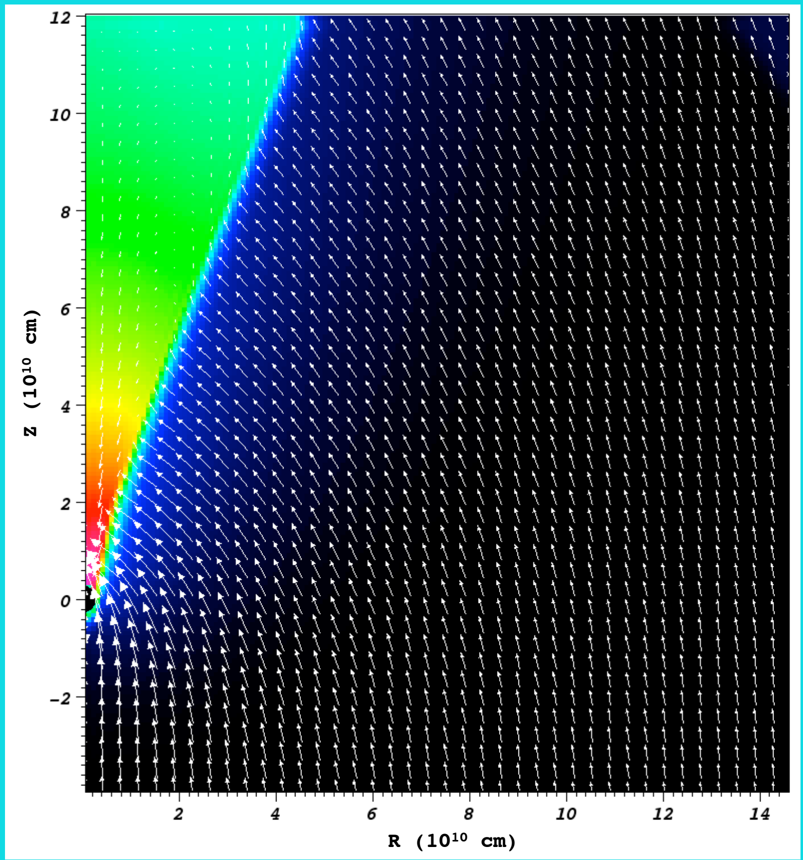
Bondi accretion

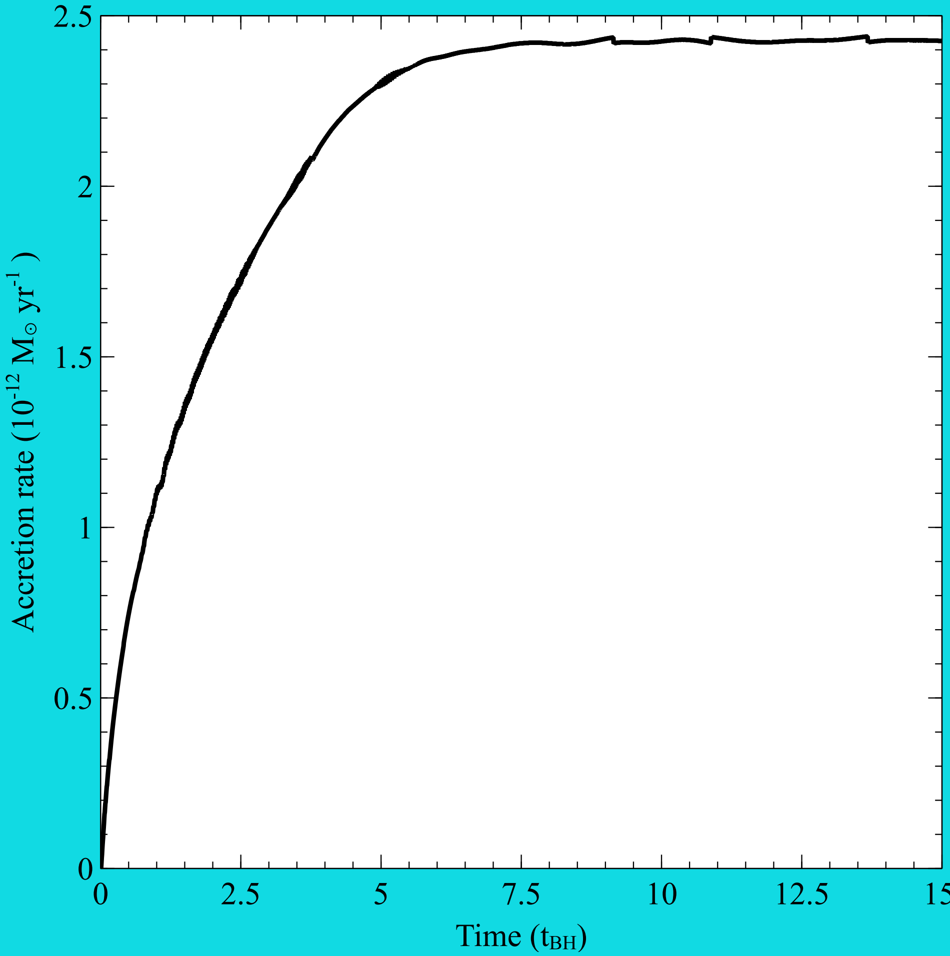


Shu collapse



Bondi-Hoyle accretion



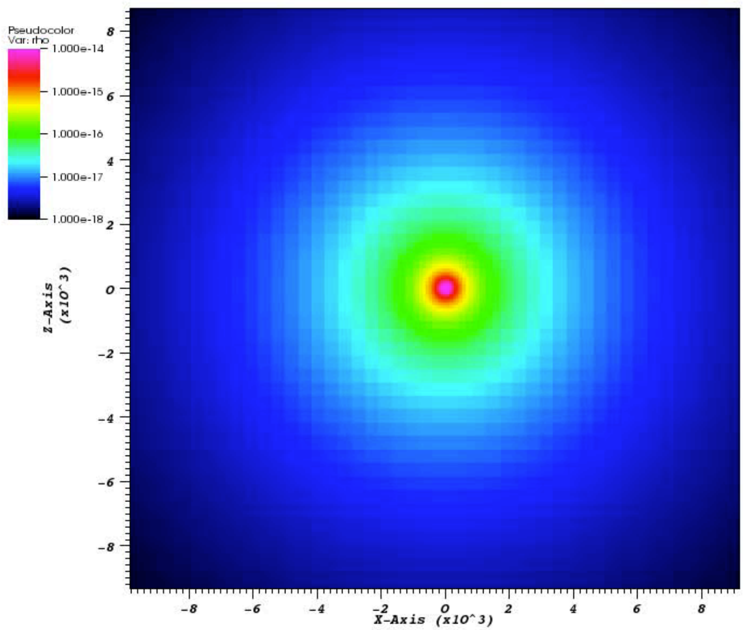


A model of massive star formation

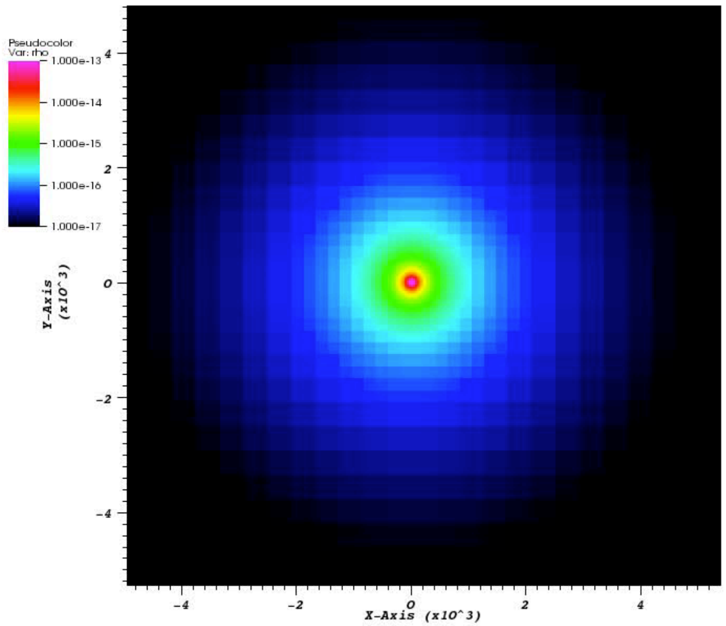
Initial conditions

Parameter	Value
Mass	100 solar
Density profile	r^{-2}
Radius	0.1 pc
rotation, rad s^{-1}	5×10^{-13}
Opacity	MRN sizes, silicate grains
Max resolution	13 AU

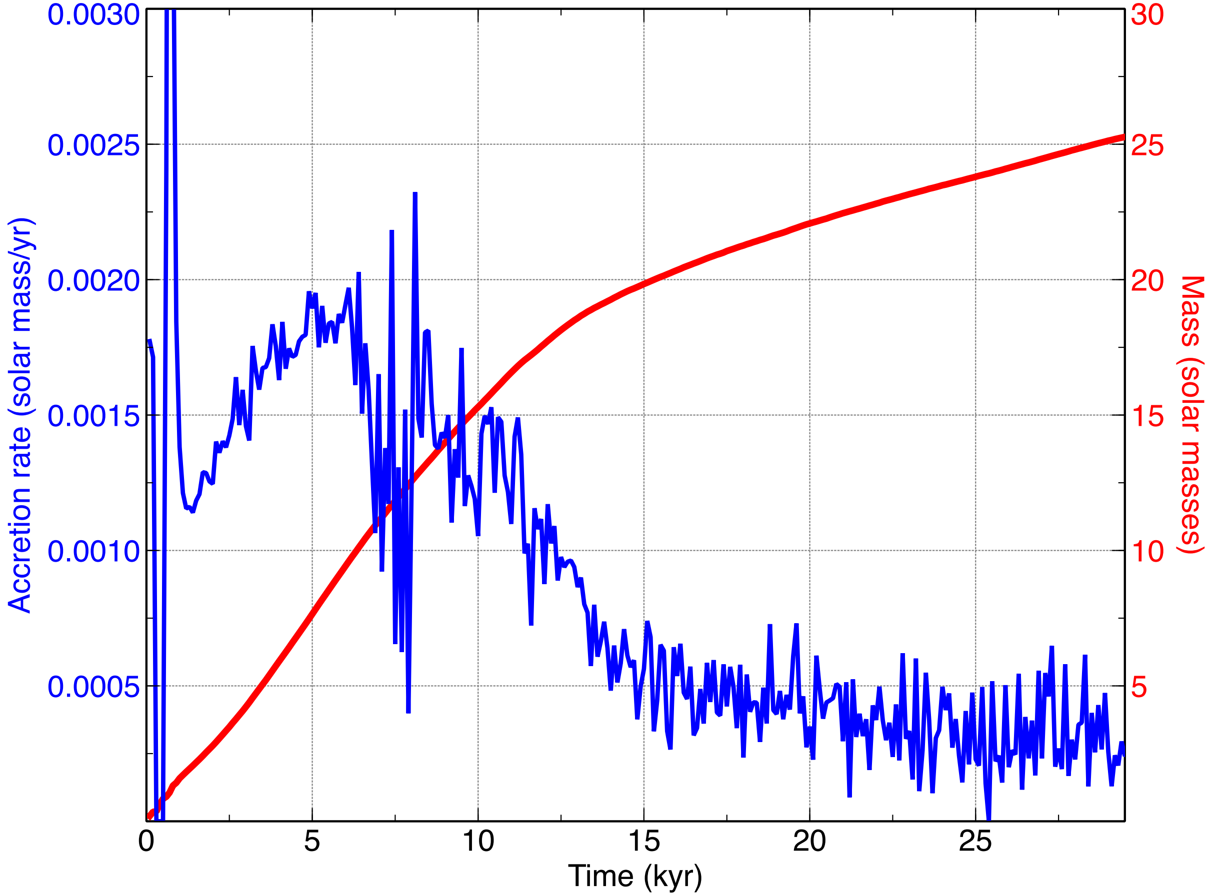
Side on



Midplane

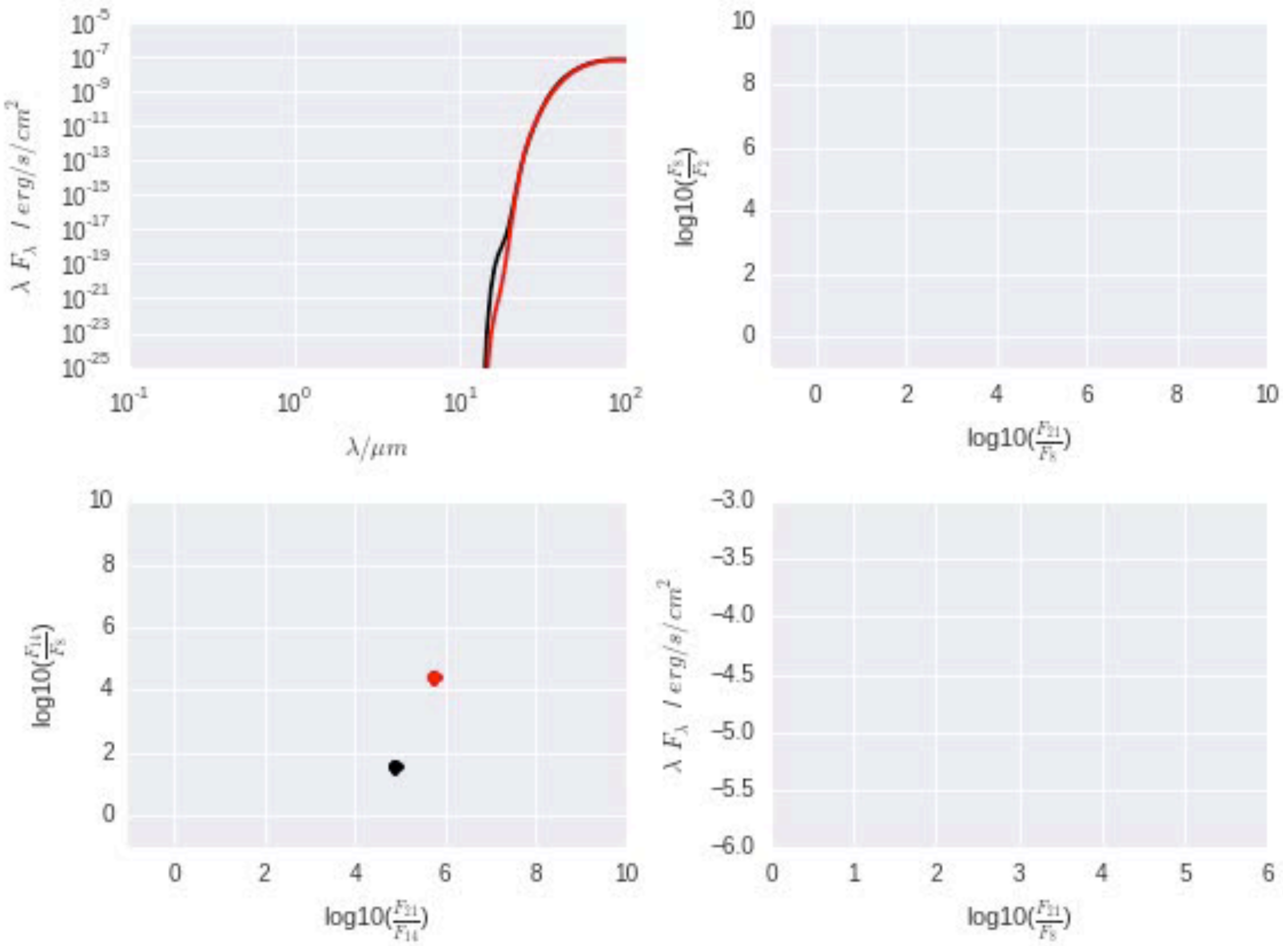


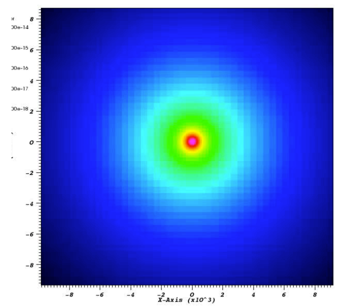
10000 AU

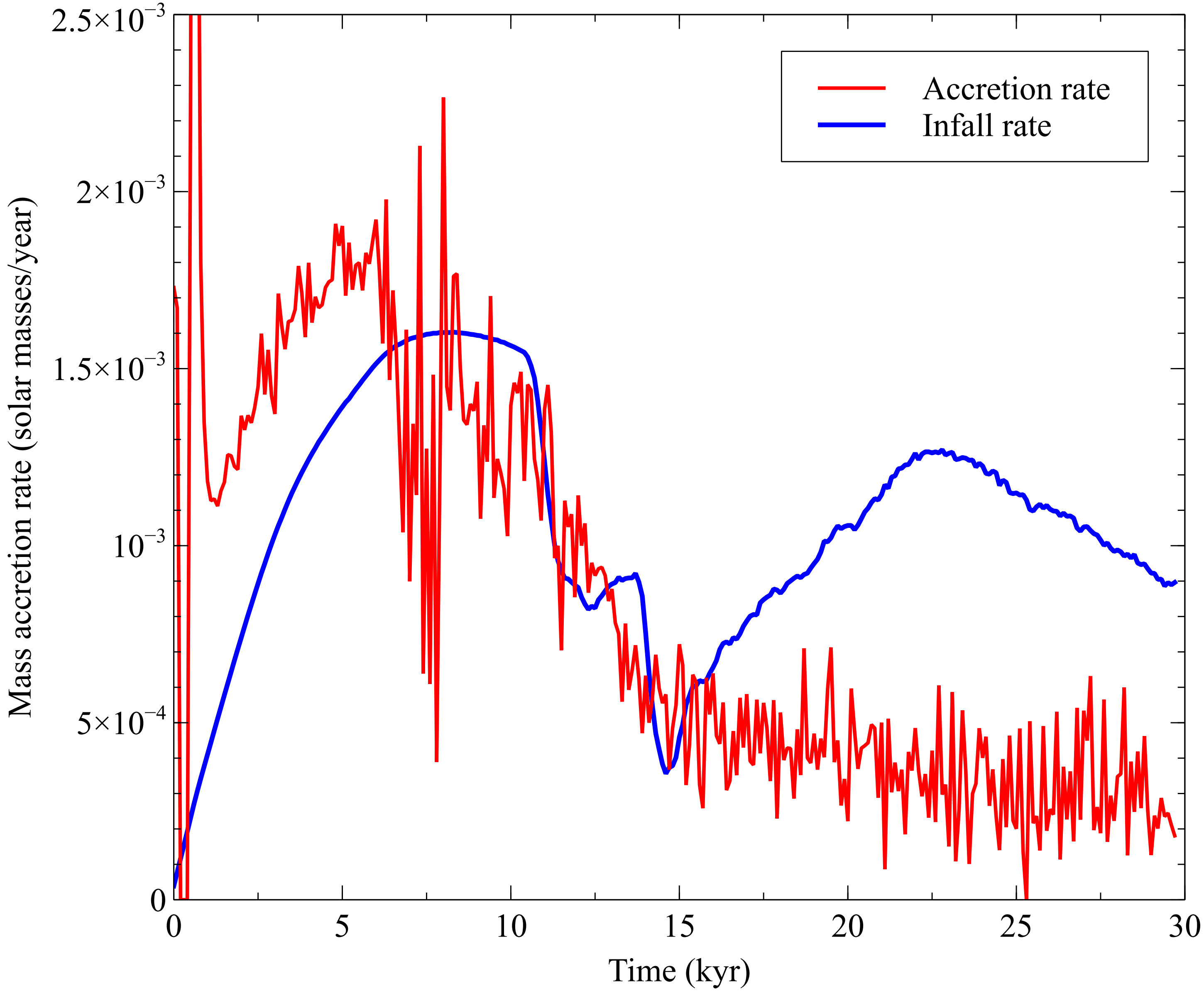

0.0030
0.0025
0.0020
0.0015
0.0010
0.0005
Accretion rate (solar mass/yr)
0
5
10
15
20
25
Time (kyr)
30
25
20
15
10
5
Mass (solar masses)

Comparison with observations

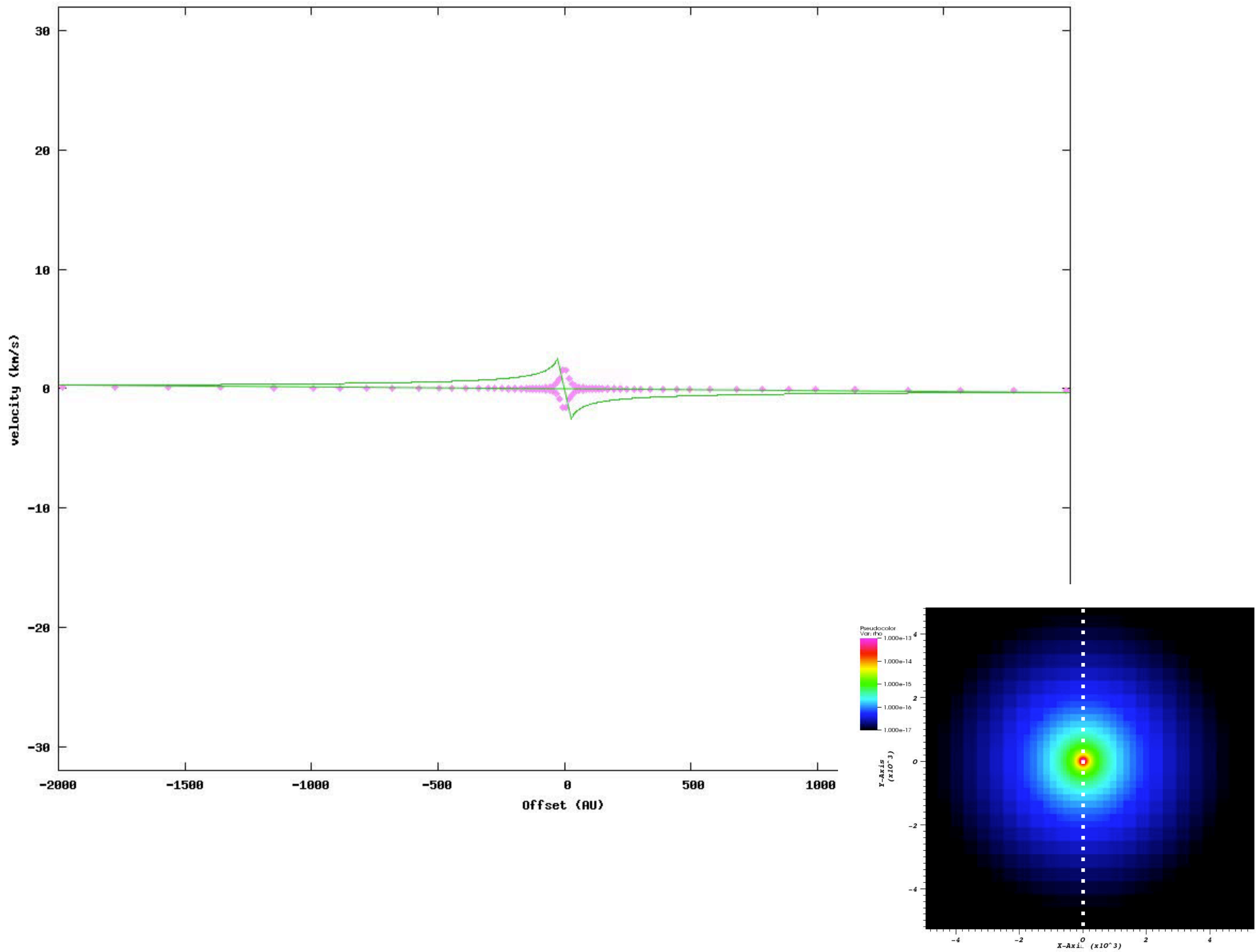
Red MSX Source Survey (Lumsden et al. 2002)







Accretion rate
Infall rate
Mass accretion rate (solar masses/year)
Time (kyr)
2.5×10^{-3}
2×10^{-3}
1.5×10^{-3}
10^{-3}
5×10^{-4}
0
0
5
10
15
20
25
30



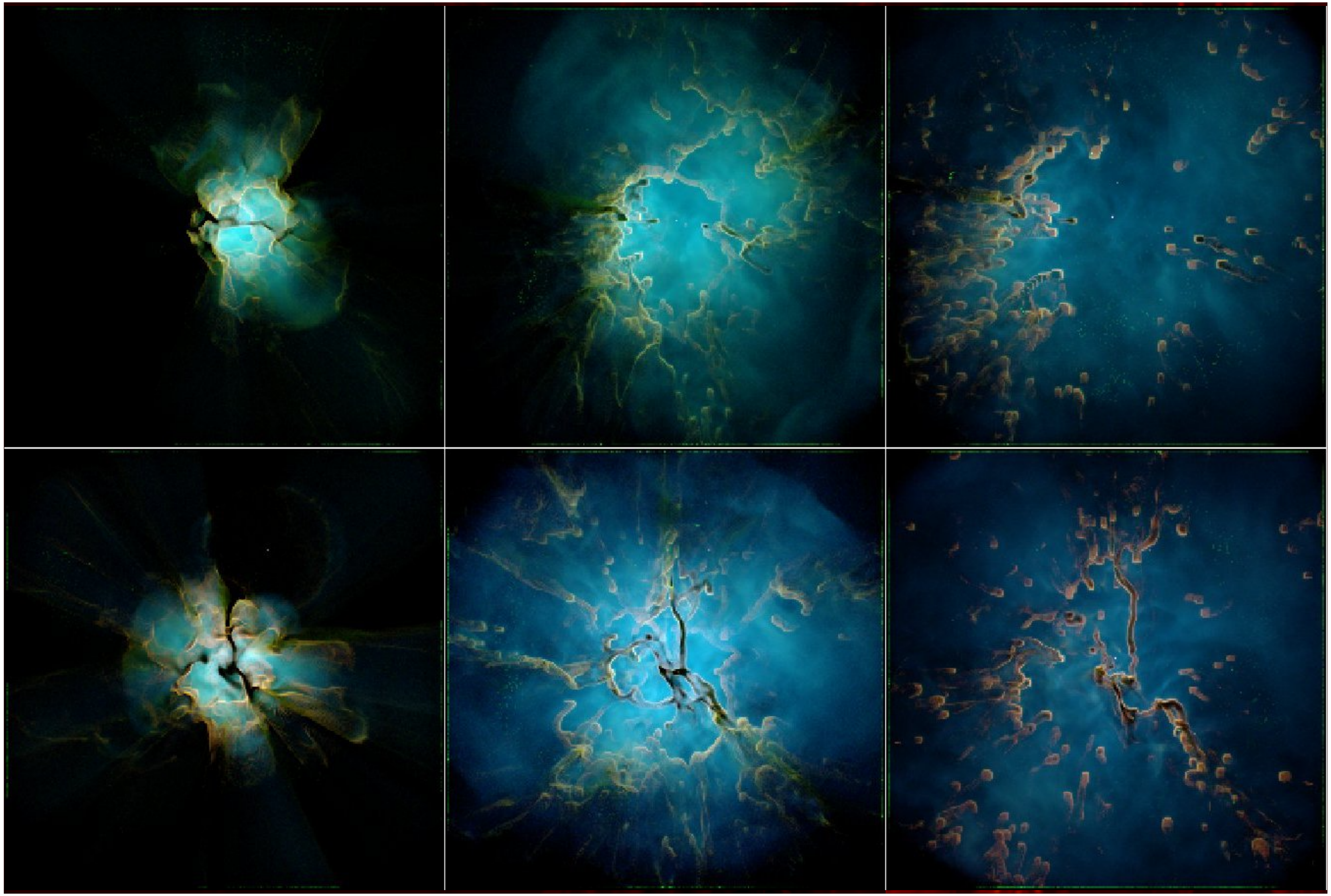
velocity (km/s)
30
20
10
0
-10
-20
-30
-2000
-1500
-1000
-500
0
500
1000
Offset (AU)
Pseudocolor
Var: rho
1.000e-13
1.000e-14
1.000e-15
1.000e-16
1.000e-17
Y-Axis (x10^3)
X-Axis (x10^3)

Dispersal of gas from clusters

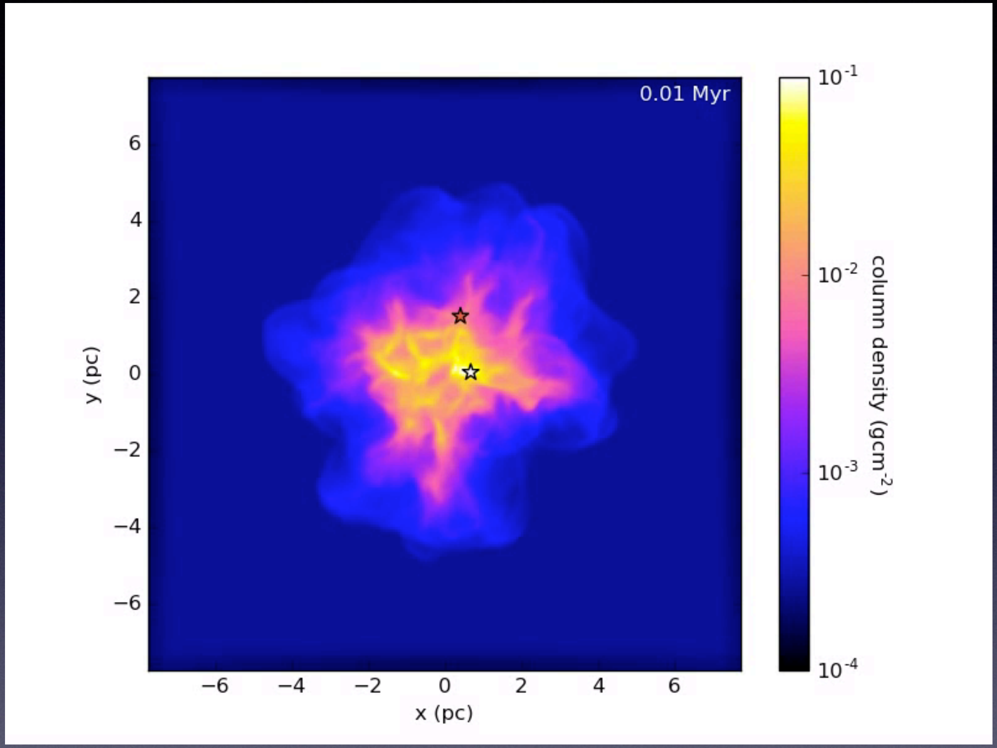
H alpha

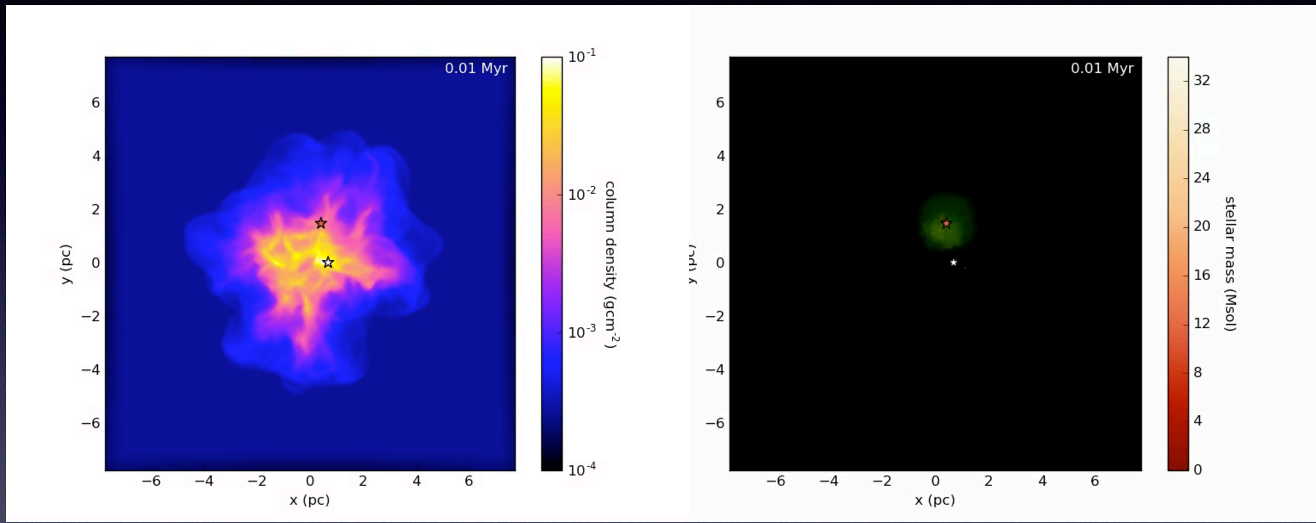
[OIII] 5007

[SII] 6731

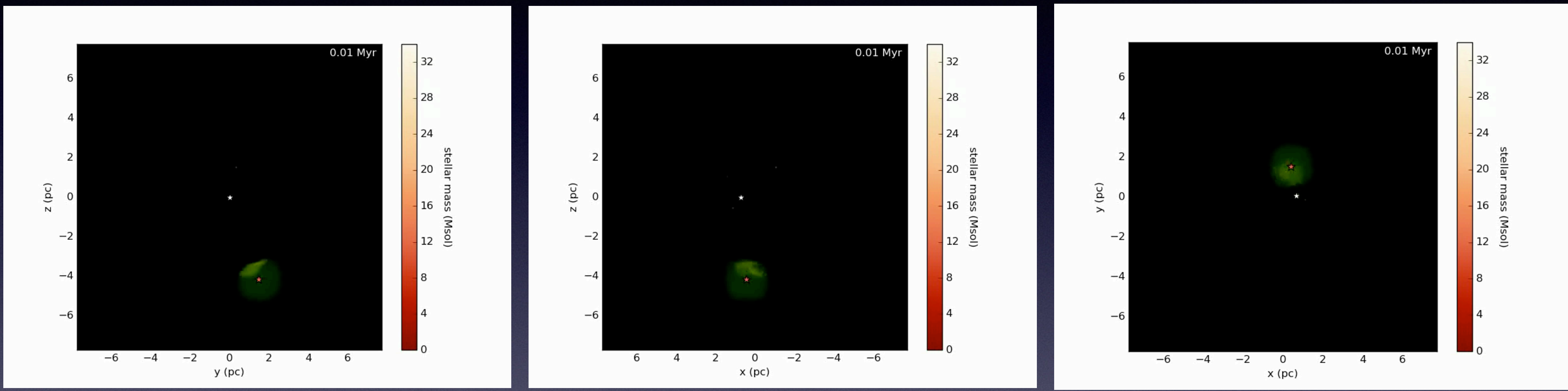


Ali, Harries, Douglas (2018), MNRAS, 477, 5422


0.01 Myr
y (pc)
x (pc)
column density (gcm^{-2})
10^{-1}
10^{-2}
10^{-3}
10^{-4}



0.01 Myr
y (pc)
x (pc)
column density (gcm^{-2})
0.01 Myr
stellar mass (Msol)



0.01 Myr
z (pc)
y (pc)
stellar mass (Msol)
0.01 Myr
z (pc)
x (pc)
stellar mass (Msol)
0.01 Myr
y (pc)
x (pc)
stellar mass (Msol)

Summary

- It is now possible to conduct RHD simulations **with a level of microphysical detail comparable to dedicated RT codes such as Cloudy**
- We rely on the **embarrassingly parallel nature of MC methods**
- We can make **direct comparisons** with observations via synthetic observations

The TORUS code

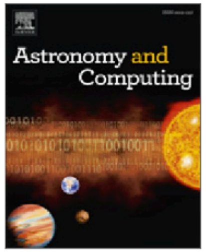
- Flexible tool for computing images and spectra for a wide variety of objects with circumstellar material, e.g.
  - O-star and WR star winds (atomic lines and continuum)
  - Symbiotic binary stars (Raman scattered lines)
  - Classical T Tauri stars (atomic lines and dust continuum)
  - Herbig Ae/Be stars (dust continuum)
  - Stellar clusters (dust continuum)
  - Molecular Clouds (molecular lines and dust continuum)
  - Spiral galaxies (21cm line)

Astronomy and Computing 27 (2019) 63–95

Contents lists available at ScienceDirect

Astronomy and Computing

journal homepage: www.elsevier.com/locate/ascom



Full length article

The TORUS radiation transfer code☆

T.J. Harries [a,*], T.J. Haworth [b], D. Acreman [a], A. Ali [a], T. Douglas [a]

[a] *Department of Physics and Astronomy, University of Exeter, Stocker Road, Exeter EX4 4QL, UK*
[b] *Astrophysics Group, Imperial College London, Blackett Laboratory, Prince Consort Road, London SW7 2AZ, UK*

ARTICLE INFO

Article history:
Received 17 September 2018
Accepted 8 March 2019
Available online 15 March 2019

Keywords:
Radiative transfer
Hydrodynamics
Methods
Numerical

ABSTRACT

We present a review of the TORUS radiation transfer and hydrodynamics code. TORUS uses a 1-D, 2-D or 3-D adaptive mesh refinement scheme to store and manipulate the state variables, and solves the equation of radiative transfer using Monte Carlo techniques. A framework of microphysics modules is described, including atomic and molecular line transport in moving media, dust radiative equilibrium, photoionisation equilibrium, and time-dependent radiative transfer. These modules provide a flexible scheme for producing synthetic observations, either from analytical models or as post-processing of hydrodynamical simulations (both grid-based and Lagrangian). A hydrodynamics module is also presented, which may be used in combination with the radiation-transport modules to perform radiation-hydrodynamics simulations. Benchmarking and validation tests of each major mode of operation are detailed, along with descriptions and performance/scaling tests of the various parallelisation schemes. We give examples on the uses of the code in the literature, including applications to low- and high-mass star formation, cluster feedback, and stellar winds, along with an Appendix listing the refereed papers that have used TORUS.

The TORUS radiation-hydrodynamics code

The input model
- SPH particle list
- Analytical model
- Grid-based hydro model
- Native TORUS RHD

Adaptive mesh refinement
- 1D spherical
- 2D/3D cartesian
- 2D/3D cylindrical

The micro-physics
- Atomic/ Molecular statistical equilibrium
- Radiative equilibrium
- Photoionisation equilibrium
- Hydro-dynamics

Final RT process
- Co-moving frame ray-tracing
- Sobolev/LVG ray-tracing
- Polarised Monte-Carlo transport

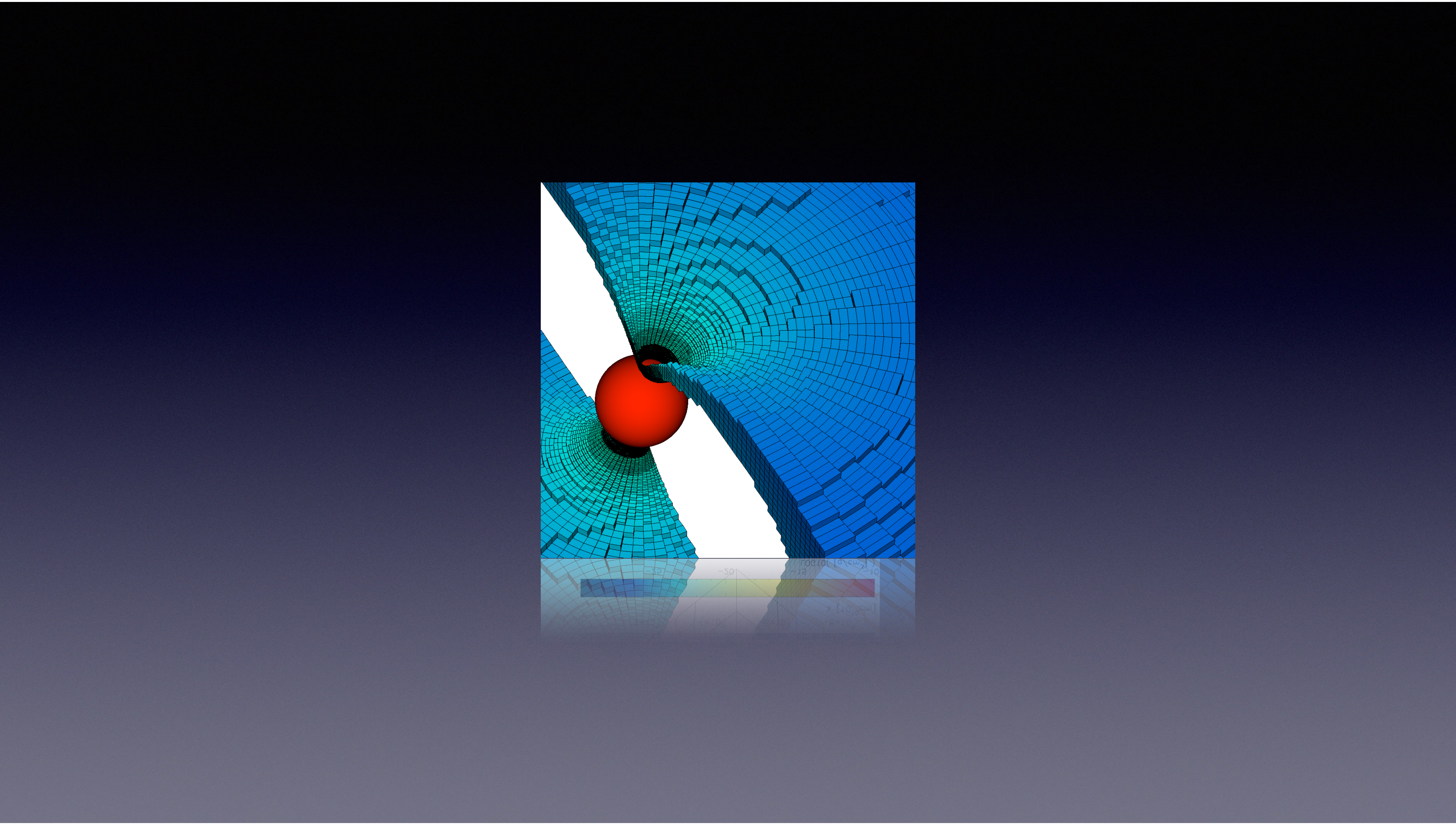
Outputs
- SED
- Line spectrum
- Stokes intensity image
- Data cube

Technical aspects

- Written in modular Fortran 90 (code base is ~250000 lines)
- Stored in git on bitbucket
- Parallelized under MPI and openMP
- Minimal external libraries required (MPI plus cfitsio if you wish to create FITS output)
- Test suite run nightly
- Compiles on a wide variety of architectures

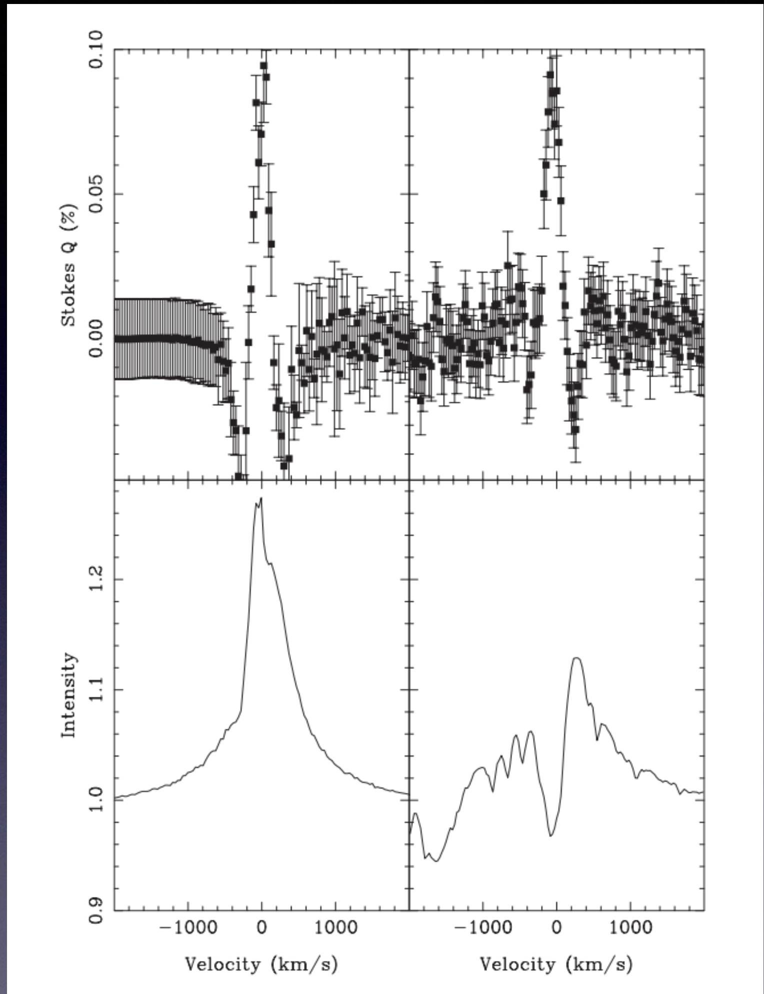
Numerical aspects

- Variables (density, temperature, velocity etc) held on an adaptive mesh
  - Either 3D cartesian (octal tree)
  - or 2D cylindrical (quad tree)
  - or 3D cylindrical polar (mixture of octal/quad)



Science aspects

- Atomic spectral lines
  - Solves statistical equilibrium using either the Sobolev approximation or in the co-moving frame
  - Does **not** currently perform radiative equilibrium for the atomic case (i.e. need a temperature structure)
  - Line transfer followed in all four Stokes intensities (spectropolarimetry)



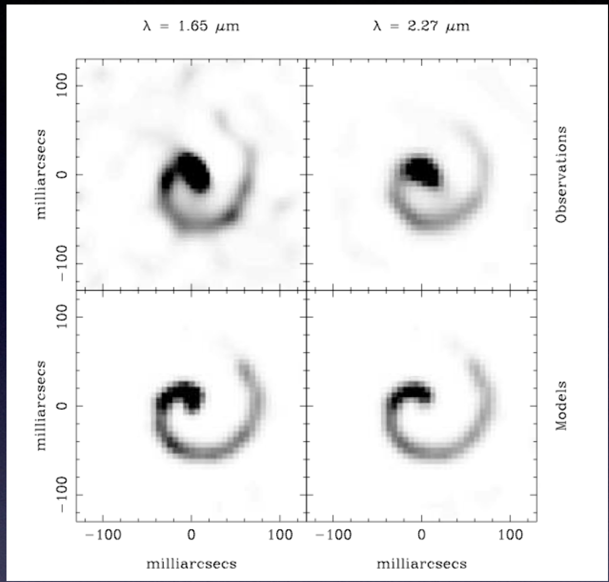
Harries, MNRAS, 2000, 315, 722

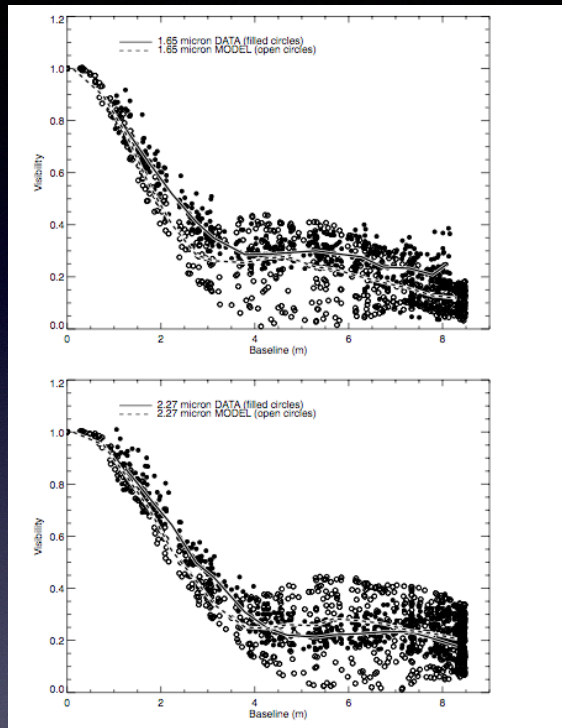
Science aspects

- Photoionization
  - Monte-Carlo estimators for the photoionization rate
  - Full radiative equilibrium inc. dust
  - Similar method to (but not as detailed in atomic physics as) Barbara Ercolano's *Mocassin* code and Kenny's photoionization code

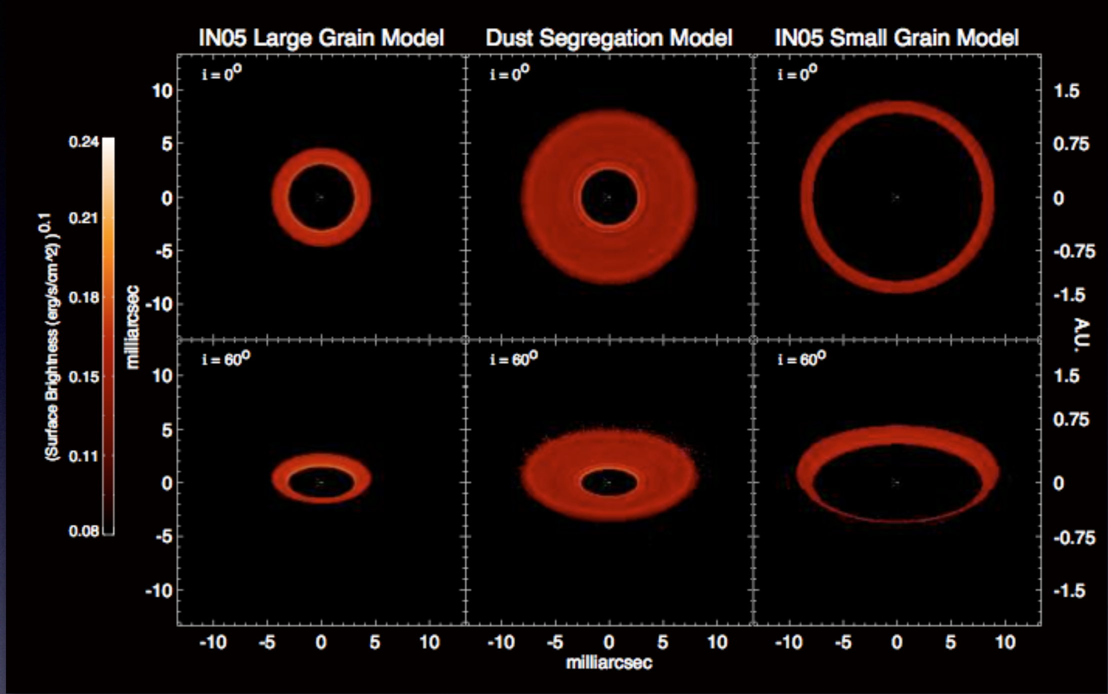
Science aspects

- Dust continuum transfer
  - Radiative equilibrium solving using Lucy's (1999, *A&A*, **344**, 282) Monte-Carlo algorithm
  - Multiple dust species, dust sublimation, vertical hydrostatic equilibrium in discs
  - Stokes intensities followed (polarization images, spectra)





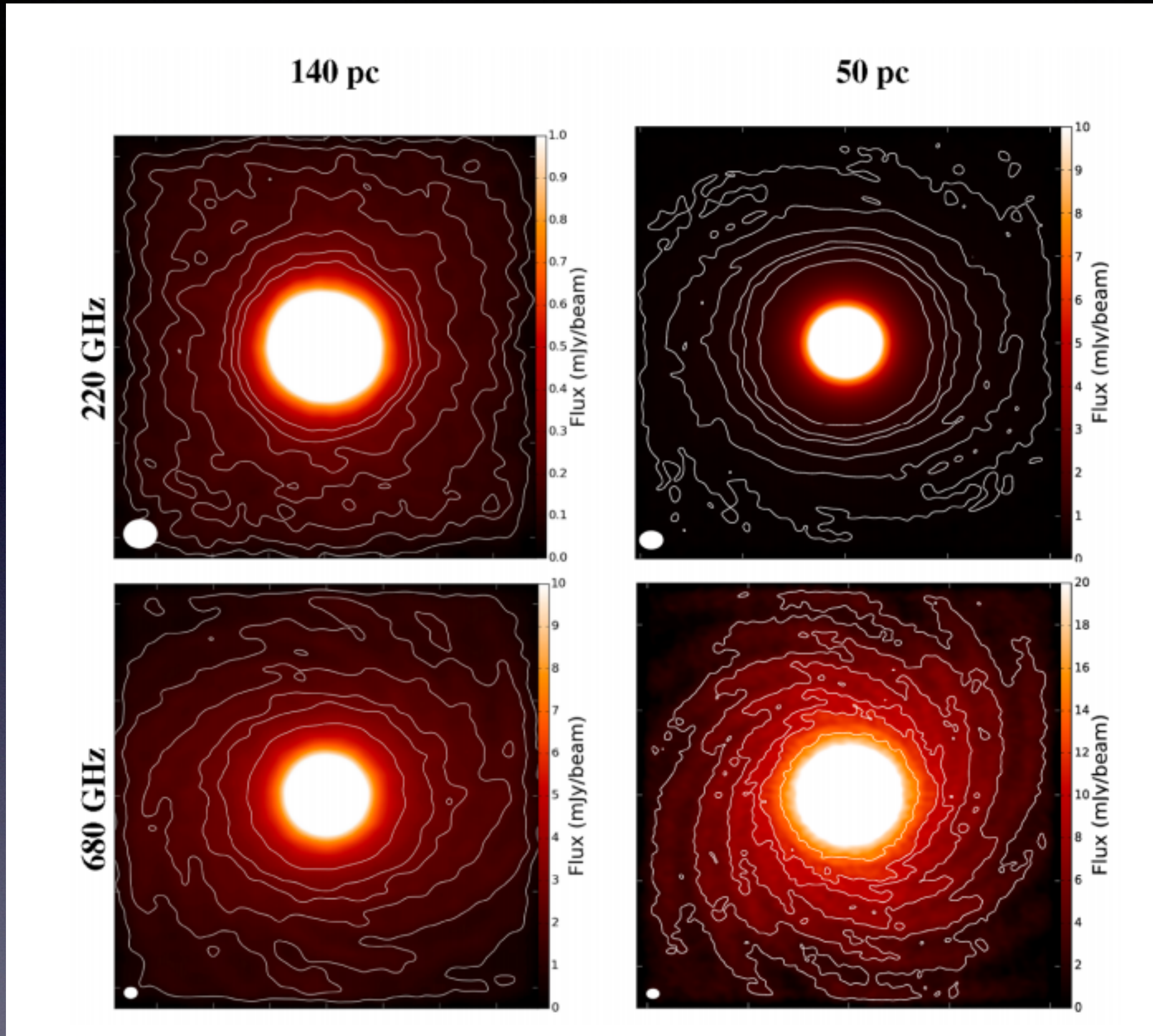
Harries, Monnier, Symington & Kurosawa
(2004)



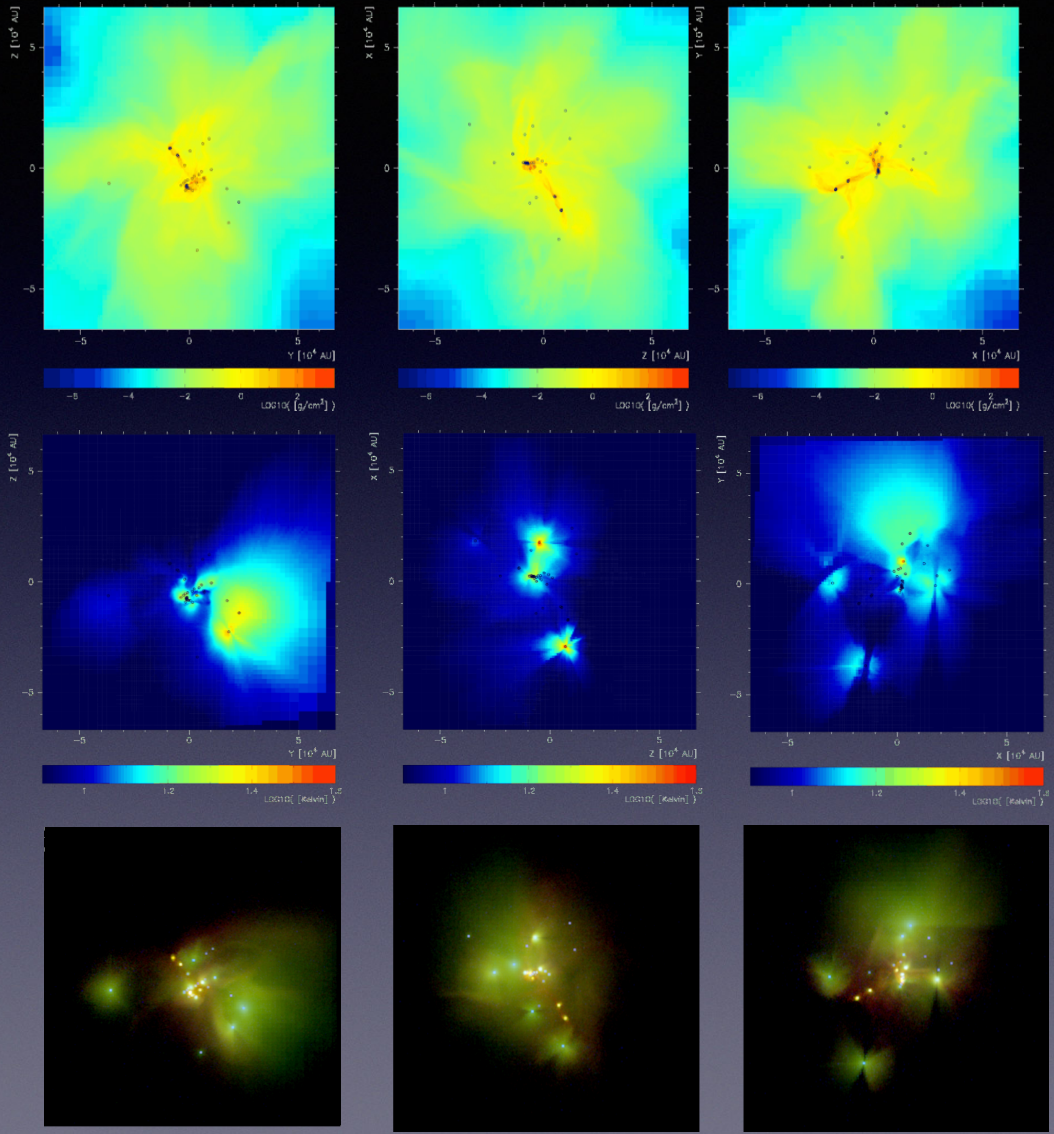
Tannirkulam et al. , 2007, *ApJ*, **661**, 374

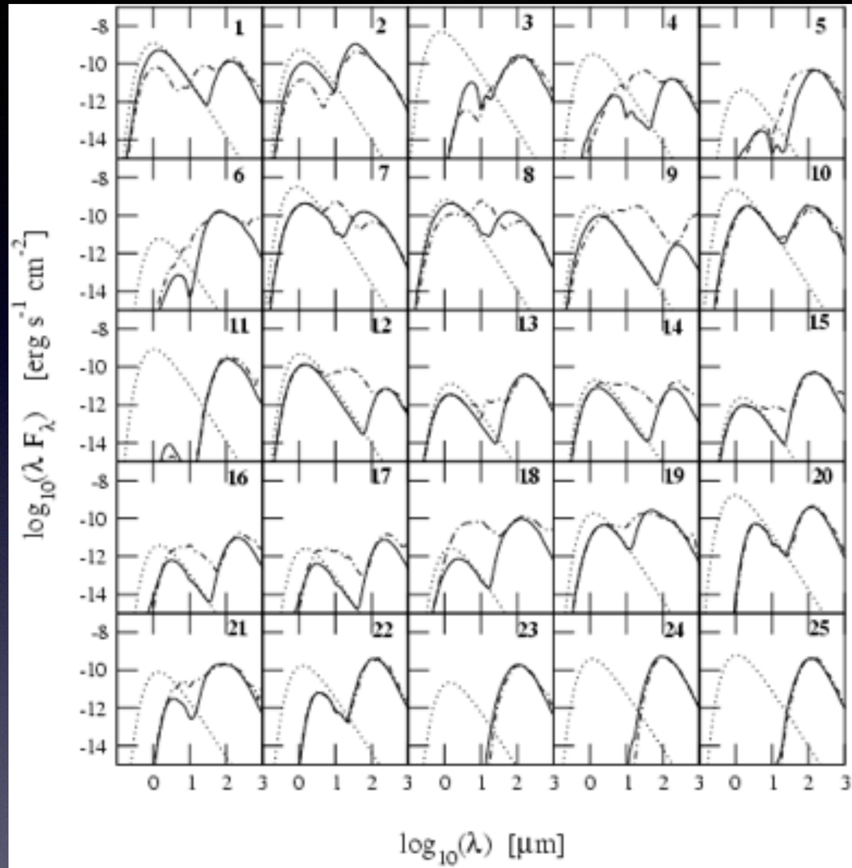
Tannirkulam et al. , 2008, *ApJ*, **677**, 51

Tannirkulam et al. , 2008, *ApJ*, **689**, 513



Hall et al., 2016, MNRAS, 458, 306



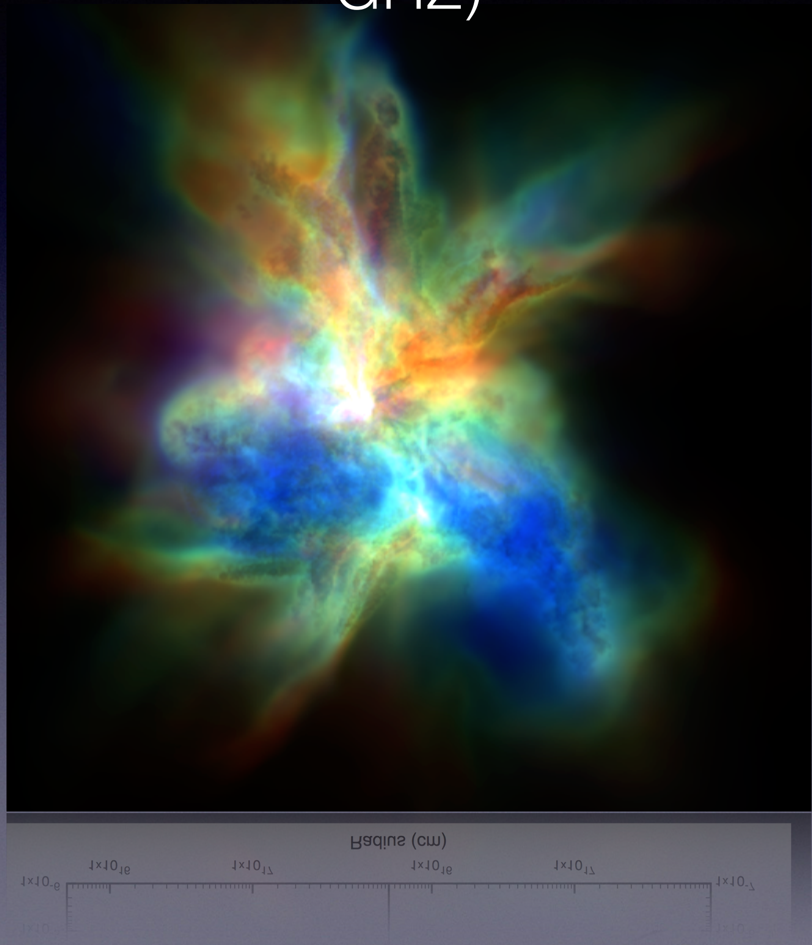


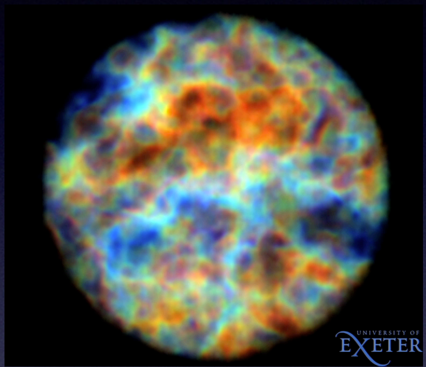
Kurosawa, Harries, Bate & Symington, 2004, MNRAS, 351, 1134

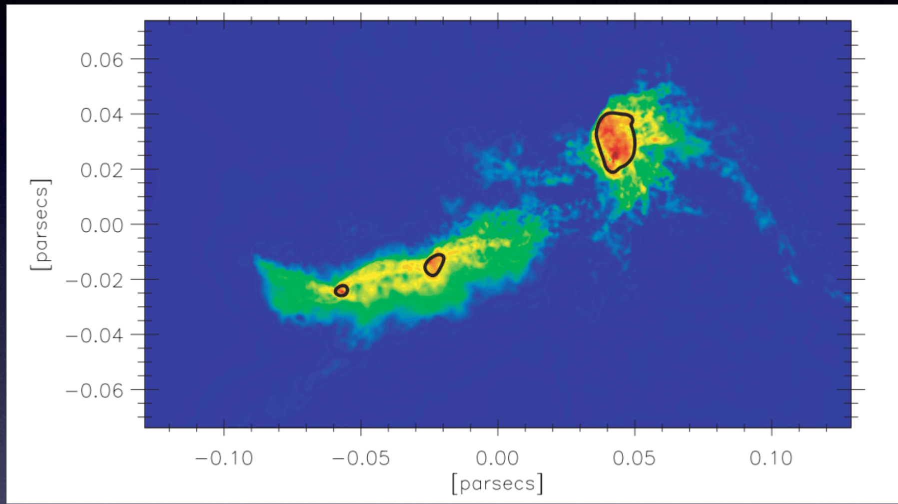
Molecular lines

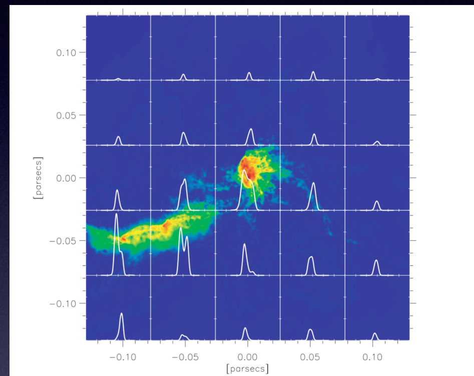
- David Rundle's PhD thesis
- Statistical equilibrium solved using co-moving frame transfer with Monte-Carlo direction sampling (modified version of the MC accelerated lambda iteration method of Hogerheijde & van der Tak 2001)
- Rundle et al. 2010, MNRAS, 407, 986

HCO+ J=1 to 0 (89.2 GHZ)

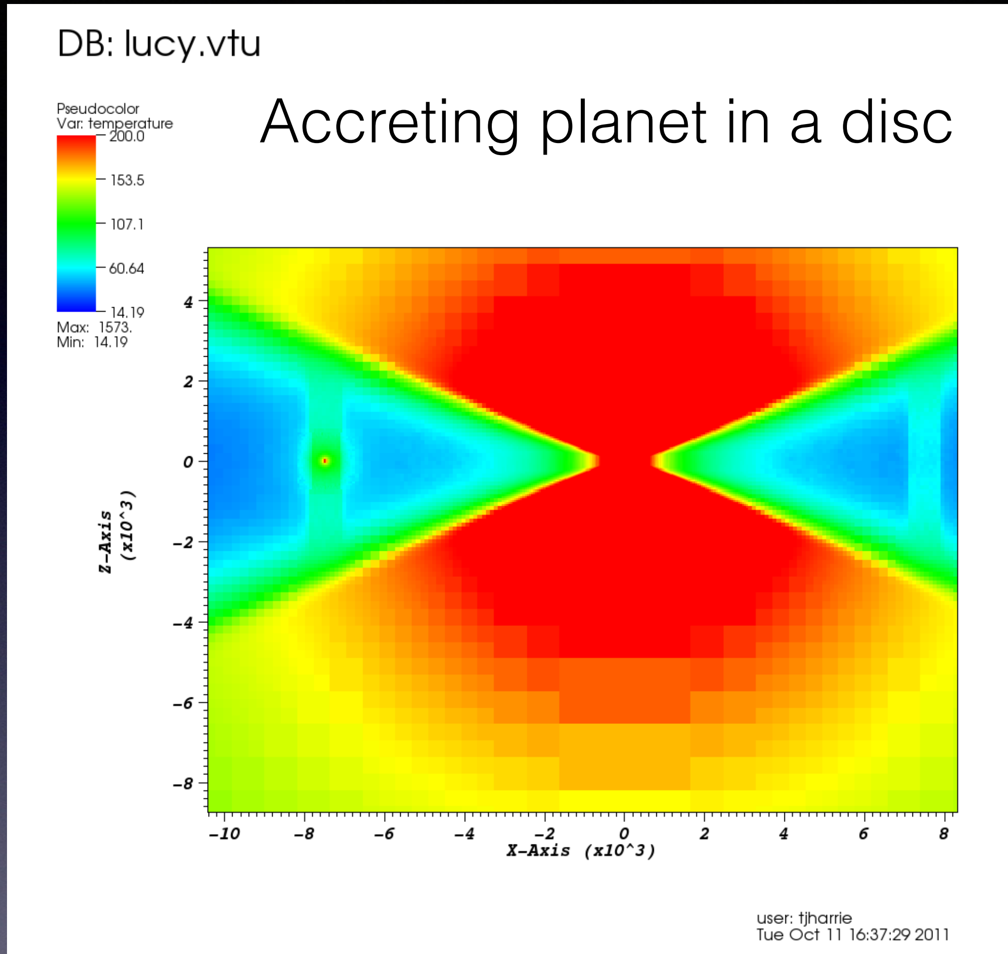




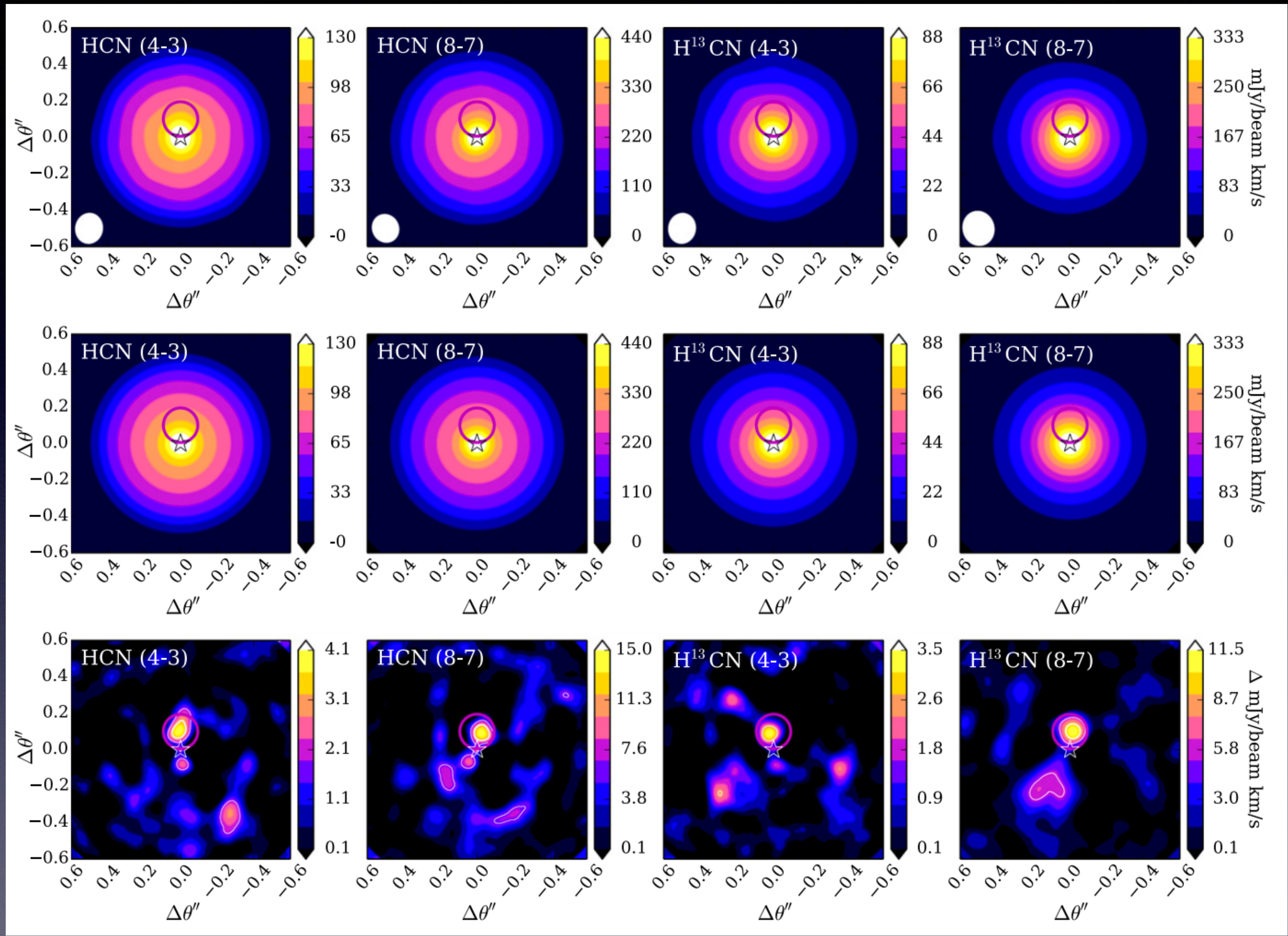




Rundle, Harries, Acreman & Bate, 2010, MNRAS, 407, 986



Cleeves, Bergin & Harries, 2015, ApJ, 807, 2



Cleeves, Bergin & Harries, 2015, ApJ, 807, 2

Using TORUS

- The publicly available version is on bitbucket

- There will be an exercise sheet on cloning, compiling, and running the code available on the summer school website this afternoon!

Installing TORUS

- You will need
  - The source code and data files (grain optical constants) from the TORUS pages
  - The cfitsio library (if you want write or read FITS images) http://heasarc.gsfc.nasa.gov/fitsio/
  - The VISIT visualisation code (to view the AMR mesh. Binaries are available from https://wci.llnl.gov/codes/visit/

Running TORUS

- There is a user manual on the TORUS website
- TORUS models are set up using a parameters file that is text file containing keywords and values

Part of a TORUS parameters file

```
! Torus parameter file for 2D benchmark disc
! See Pascucci et al, 2004, A&A, 417, 793

dustphysics T   ! use dust microphysics

radeq T  ! perform a radiative equilibrium calculation

! AMR grid parameters

readgrid F   ! we aren't reading a grid, we will set one up from scratch
writegrid F  ! we don't need to write out the AMR file - we just need SEDs
amrgridsize 4.e6 ! the linear size of the top-level AMR mesh in units of 10^10 cm
amr2d T  ! this is a 2d (cylindrical) model

! grid smoothing switches

smoothgridtau T  ! smooths the grid for optical depth, in order to resolve disc photosphere
dosmoothgrid T   ! smooth the grid for jumps in cell refinement
smoothfactor 3.0 ! make sure that neighbouring cells are not only one AMR depth apart


! Source parameters

nsource 1  ! there is just one source
radius1 1.  ! it has a radius of 1 solar radius
teff1 5800.  ! the source effective temperature
contflux1 blackbody ! the continuum flux is assumed to be a blackbody
mass1 1.   ! the source has a mass of one solar mass
sourcepos1 0. 0. 0  ! it is located at the grid cntre

! Geometry specific parameters

geometry benchmark  ! this is the Pascucci (2004) benchmark
rinner 1. ! inner disc radius (AU)
router 1000. ! outer disc radius
height 125. ! disc scaleheight at 100 AU (in AU)
rho 8.16136e-18  ! density at inner edge midplane (g/cc)

! Dust grain properties

iso_scatter T      ! Assume isotropic scattering (assumed by benchmark)
graintype1 draine_sil  ! Draine silicates
grainfrac1 0.01    ! grain mass fraction (in terms of gas)
amin1 0.12         ! minimum grain size (microns)
amax1 0.1201       ! maximum grain size (microns)
qdist1 0.01        ! power law index (flat)
```

What to do next

- If you want to install and run a test model using TORUS then please feel free
- Try running the sample parameters file from the web (a dusty disc) and calculate some SEDs and images...

Have fun!