FLASH Code Tutorial

part IV
radiation modules

Robi Banerjee
Hamburger Sternwarte
banerjee@hs.uni-hamburg.de

FLASH Code: RT modules

- The Radiation transfer unit

$\Longrightarrow$ idea: get solution of the radiation transfer equation

$$\frac{1}{c}\frac{\partial I}{\partial t} + \hat{\mathbf{\Omega}} \cdot \nabla I + \rho\kappa I = \eta$$

- $I(\boldsymbol{x},\boldsymbol{\Omega},\nu,t)$: radiation intensity
- $\kappa(\boldsymbol{x},\nu,t)$: opacity [cm^2/g]
- $\eta(\boldsymbol{x},\nu,t)$: emissivity

$\Longrightarrow$ so far: no generic coupling to the hydrodynamics (or MHD)

$\Longrightarrow$ but: possible via `3T` module (`multiTemp`)
for hydro only

FLASH Code: RT modules

- The Radiation transfer unit

$\Longrightarrow$ solution via

Multigroup **Diffusion** (MGD) solver
`physics/RadTrans/RadTransMain/MGD`

$\Longrightarrow$ coupling to electron internal energy

$$\frac{\partial u_e}{\partial t} = \int_0^\infty \mathrm{d}\nu \int_{4\pi} \mathrm{d}\hat{\mathbf{\Omega}}(\rho\kappa I - \eta)$$

- u_e : electron internal energy

$\Longrightarrow$ useful for HEDP simulations (LASER beam)
$\Longrightarrow$ but good starting point for own RT development

FLASH Code: RT modules

- The Radiation transfer unit
  $\Longrightarrow$ multigroup **diffusion** limit
  with frequency groups: ν_g to ν_{g+1} :

$$\frac{1}{c}\frac{\partial u_g}{\partial t} - \nabla \cdot \left(\frac{1}{3\sigma_{t,g}}\nabla u_g\right) + \sigma_{a,g} u_g = \sigma_{e,g} a T_e^4 \frac{15}{\pi^4}\left[P(x_{g+1}) - P(x_g)\right]$$

$$\frac{\partial u_e}{\partial t} = \sum_g \left\{\sigma_{a,g} u_g - \sigma_{e,g} a T_e^4 \frac{15}{\pi^4}\left[P(x_{g+1}) - P(x_g)\right]\right\}$$

- $\sigma_{t,g}$: transport opacity
- $\sigma_{a,g}$: absorption opacity
- $\sigma_{e,g}$: emission opacity
- T_e : electron temperature
- $P(x)$: Planck integral $P(x) = \int_0^x \mathrm{d}x' \frac{(x')^3}{\exp(x') - 1}$

$$x = h\nu / k_B T_e$$

FLASH Code: RT modules

- The Radiation transfer unit

$\Longrightarrow$ discretisation leads to implicit equations

$$\frac{1}{c}\frac{u_g^{n+1}-u_g^n}{\Delta t}-\nabla\cdot\left(D_g^n\nabla u_g^{n+1}\right)+\sigma_{a,g}^n u_g^{n+1}=\sigma_{e,g}^n a(T_e^n)^4\frac{15}{\pi^4}\left[P(x_{g+1}^n)-P(x_g^n)\right]$$

$$\frac{u_e^{n+1}-u_e^n}{\Delta t}=\sum_g\left\{\sigma_{a,g}^n u_g^{n+1}-\sigma_{e,g}^n a(T_e^n)^4\frac{15}{\pi^4}\left[P(x_{g+1}^n)-P(x_g^n)\right]\right\}$$

- $D_g = 1/3\sigma_{t,g}$: diffusion coefficient in the case without flux-limiter
  → some flux-limiter are available (e.g. min-max)

$\Longrightarrow$ solved for each frequency group g using the **diffusion unit**
`physics/Diffuse/DiffuseMain`
$\Longrightarrow$ uses **HYPRE** library to solve set of linear equations

FLASH Code: RT modules

- The Radiation transfer unit

  - dividing up the multigroup problem:
    - `./setup ... -mgd_meshgroups=`$N_{\rm mg}$
      $\Longrightarrow$ maximum number of groups per **mesh**

    - number of meshes:

      runtime parameter `meshCopyCount`=$N_{\rm mesh}$ (default 1)

  - at runtime: `rt_mgdNumGroups` = $N_{\rm g} \leq N_{\rm mg} \times N_{\rm mesh}$

$\Longrightarrow$ domain and frequency decomposition

FLASH Code: RT modules

- The Radiation transfer unit

$\Longrightarrow$ domain and frequency decomposition

- example: $N_{\rm proc} = 6, \quad N_{\rm mesh} = 2, \; N_{\rm mg} = 100$

$\Longrightarrow$ number of domain decompositions = $N_{\rm proc}/N_{\rm mesh} = 3$

$\Longrightarrow$ divide frequency space by $N_{\rm mg} = 2$

$\Longrightarrow$ each process solves multigroup diffusion equation for every other group (i.e. division in odd/even groups)

$\Longrightarrow$ each process solves for only 50 groups

$\Longrightarrow$ **note**: speed-up with mesh-replication must be tested

FLASH Code: RT modules

- The Radiation transfer unit

$\Longrightarrow$ initialise your setup:

- specific energy density per group: $e_{\rm g} = u_{\rm g}/\rho$
- specific total energy: $e_r = \sum_g e_g$

  or
- specific radiation temperature: $T_r = (u_r/a)^{1/4}$

$\Longrightarrow$ can be done with:

```
RadTrans_mgdEFromT(blockId, axis, trad, tradActual)
```

FLASH Code: RT modules

- The Radiation transfer unit $\Longrightarrow$ initialise your setup:

`RadTrans_mgdEFromT(blockId, axis, trad, tradActual)`

  - `trad` : desired radiation temperature (input)
  - `tradActual` : actual temperature from integration within group boundaries (output)

  - runtime parameters: `rt_mgdBounds_1` ... `rt_mgdBounds_`$N_{\rm g}$ group limit in **eV** !

$\Longrightarrow$ `tradActual` must be set to be used in the simulation

FLASH Code: RT modules

- The Radiation transfer unit ⟹ initialise your setup

```
do k = blkLimits(LOW,KAXIS),blkLimits(HIGH,KAXIS)
   do j = blkLimits(LOW,JAXIS),blkLimits(HIGH,JAXIS)
      do i = blkLimits(LOW,IAXIS),blkLimits(HIGH,IAXIS)

         axis(IAXIS) = i
         axis(JAXIS) = j
         axis(KAXIS) = k

         ...

         ! Set the secific energy in each radiation group using a
         ! radiation temperature of 1~eV (11604.55~K):
         call RadTrans_mgdEFromT(blockId, axis, 11604.55, tradActual)

         ! Set the radiation temperature:
         call Grid_putPointData(blockId, CENTER, TRAD_VAR, EXTERIOR, axis, tradActual)

         ! Alternatively, we could have set ERAD_VAR using a*(tradActual)**4

      enddo
   enddo
enddo
```

use actual radiation temperature

⟹ initial $u_{g(i)}$ set by $u_{g(i)} \propto T_{rad}{}^4 \times (P(x_{g(i)+1}) - P(x_{g(i)}))$

FLASH Code: RT modules

- The Radiation transfer unit $\Longrightarrow$ initialise your setup

  - set group energies manually

```
! Set the secific energy in each radiation group:
call RadTrans_mgdSetEnergy(blockId, axis, 1, a*sim_trad**4/sim_rho)
call RadTrans_mgdSetEnergy(blockId, axis, 2, 0.0)
call RadTrans_mgdSetEnergy(blockId, axis, 3, 0.0)
call RadTrans_mgdSetEnergy(blockId, axis, 4, 0.0)
```

$\Longrightarrow$ here: only group 1 is set: $e_1 = aT_r^4/\rho$

FLASH Code: RT modules

- The Radiation transfer unit

$\Longrightarrow$ opacities:

`physics/materialProperties/Opacity`

access via:

```
call Opacity(soln, ngrp, opacityAbsorption,
             opacityEmission, opacityTransport)
```

$\Longrightarrow$ returns $\sigma_{a,g}$, $\sigma_{e,g}$ and $\sigma_{t,g}$

FLASH Code: RT modules

- The Radiation transfer unit
  - possible opacities:
    - constant: `../OpacityMain/Constant`
    - tabulated: `../OpacityMain/Multispecies`

$\Longrightarrow$ user provides a table containing $\sigma_{\rm g}(T_{\rm i}, \rho_{\rm i})$
for each species

$\Longrightarrow$ module uses a bilinear interpolation to get $\sigma_{\rm g}(T, \rho)$

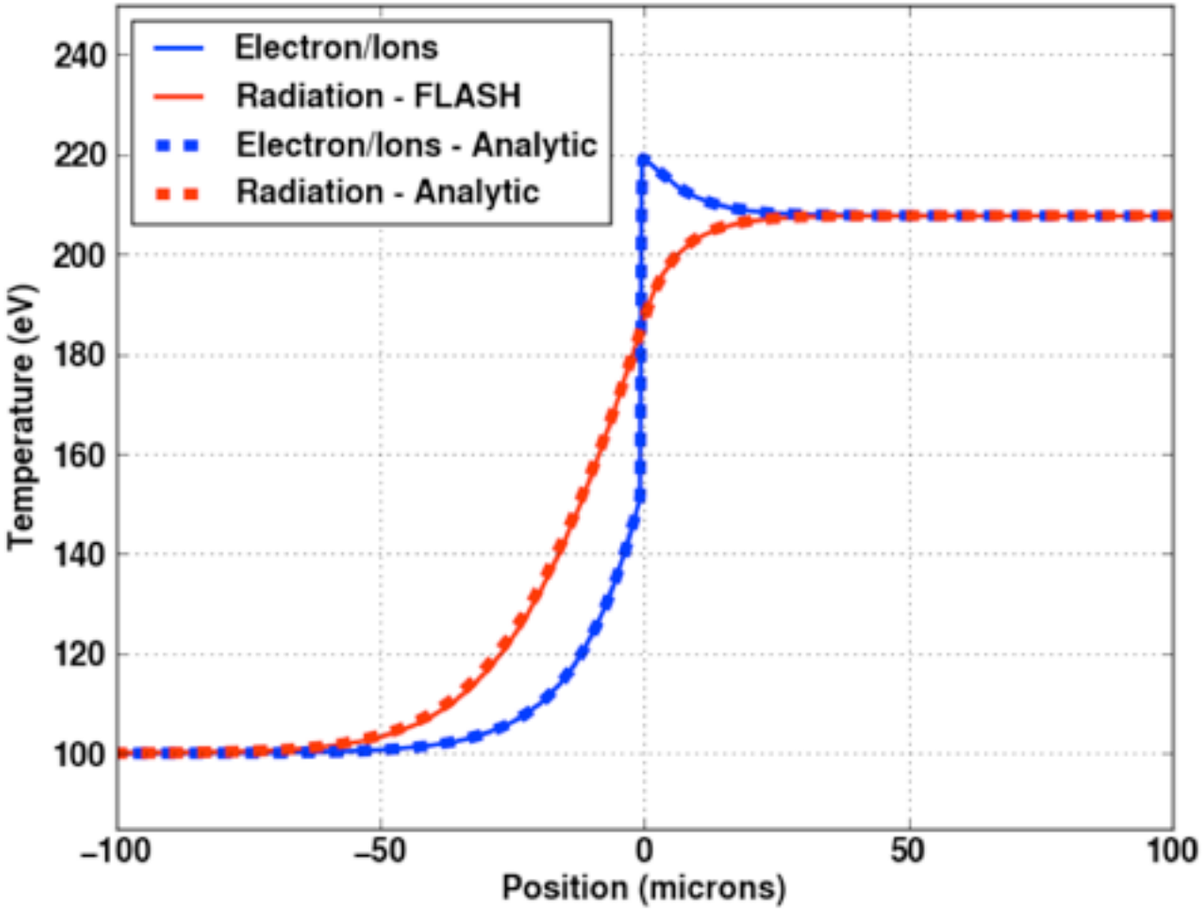
FLASH Code: RT modules

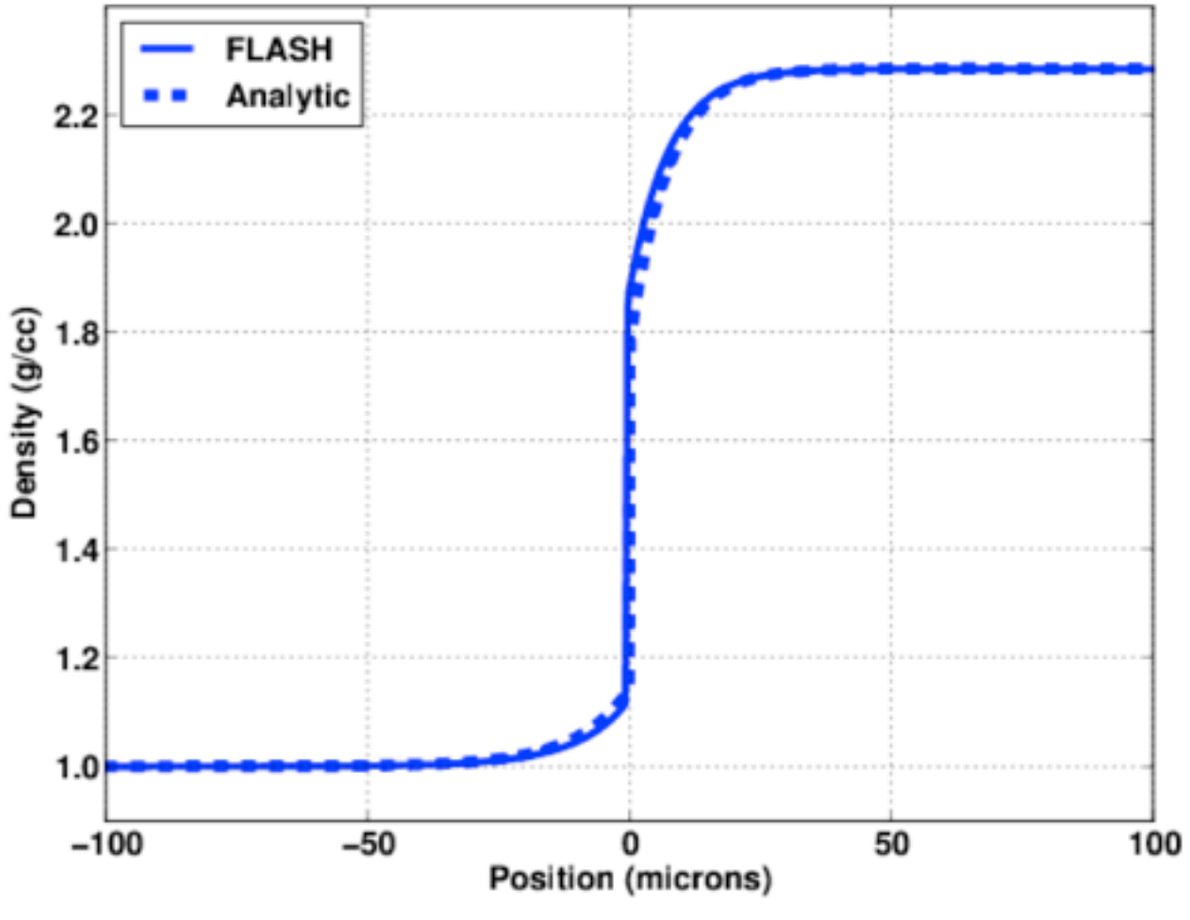
- The Radiation transfer unit: Examples:
  - `GrayDiffRadShock`:
    - 1D radiative shock problem (*Lowrie* 2008)

      $\Longrightarrow T_e = T_i$; $T_e \neq T_r$, one frequency group, constant opacity

      $\Longrightarrow$ density step function develops to steady state shock

      $\Longrightarrow$ "analytic" solution by solving an ODE



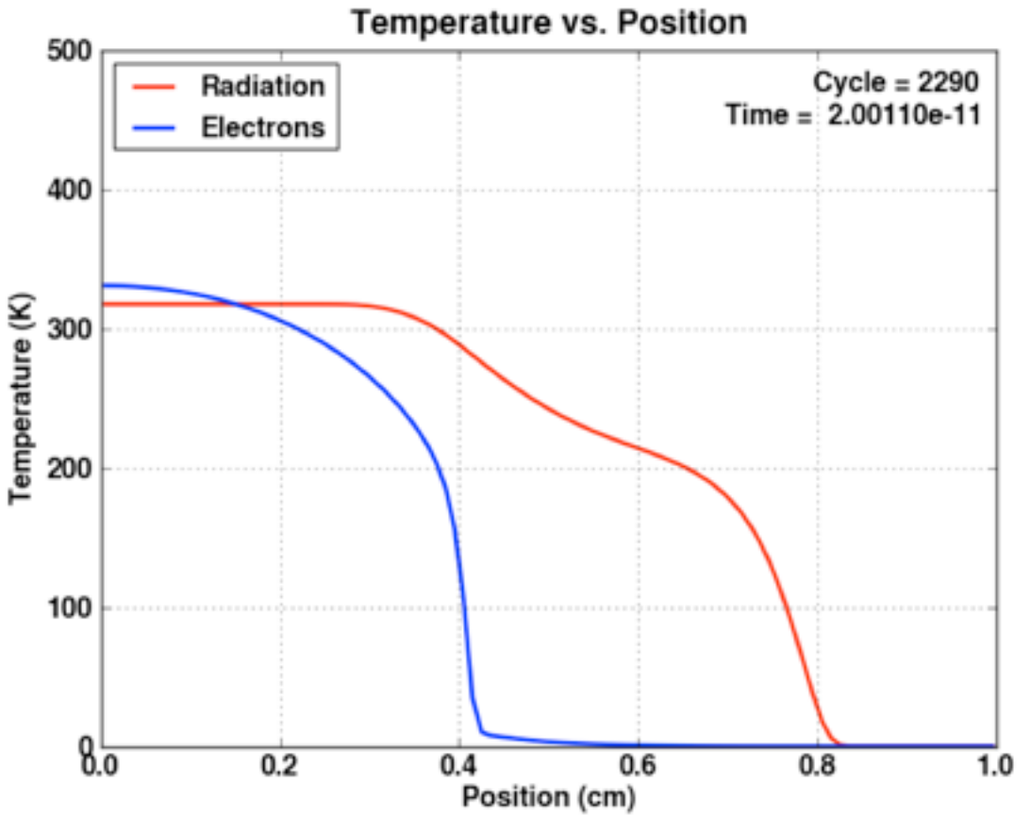


FLASH Code: RT modules

- The Radiation transfer unit:

further Examples

- `MGDStep` : 4 groups, constant opacity
  $\Longrightarrow$ initially discontinuous $T_{\rm e}$ and $T_{\rm r}$



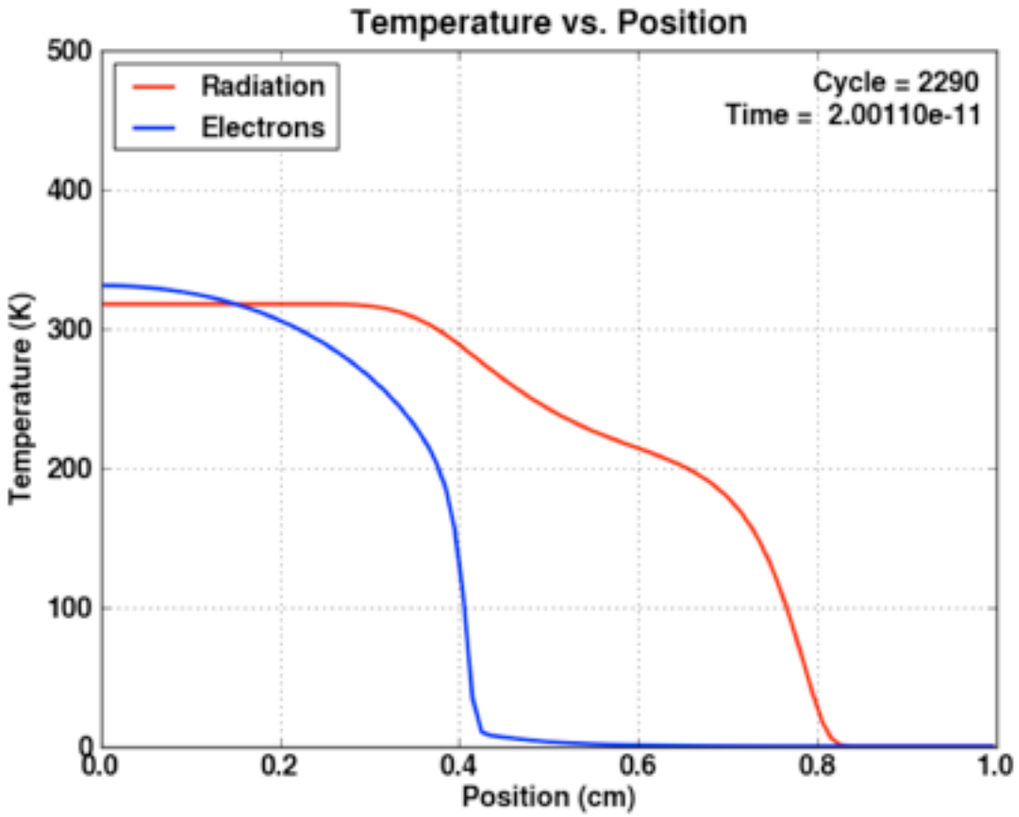
$\Longrightarrow$ heat is flowing from warm to cold regions

FLASH Code: RT modules

- The Radiation transfer unit:

further Examples

- `MGDStep` : 4 groups, constant opacity
  $\Longrightarrow$ initially discontinuous $T_{\rm e}$ and $T_{\rm r}$



$\Longrightarrow$ heat is flowing from warm to cold regions

$\Longrightarrow$ applicability of MGD for astrophysical problems ?

FLASH Code: RT modules

- The Radiation transfer unit: modules

- `Hydro`: Responsible for the 3T hydrodynamic update
- `Eos`: Computes 3T equation of state
- `Heatexchange`: Implements ion/electron equilibration
- `Diffuse`: Responsible for implementing implicit diffusion solvers and computes effect of electron conduction
- `RadTrans`: Implements multigroup radiation diffusion
- `Opacity`: Computes opacities for radiation diffusion
- `Conductivity`: Computes electron thermal conductivities
- `EnergyDeposition`: Computes the laser energy deposition

FLASH Code: RT modules

- **Ray trace** (*Rijkhorst et al.* 2006; *Peters et al.* 2010)

  - solves the radiation transfer equation along rays
  - here: without scattering / diffusion

$$\Longrightarrow \quad I(r) = I(0)\exp(-\tau(r))$$

with $\tau(r) = a_0 N(r)$

$N(r)$: column density → integrate $\rho(r)$ along r

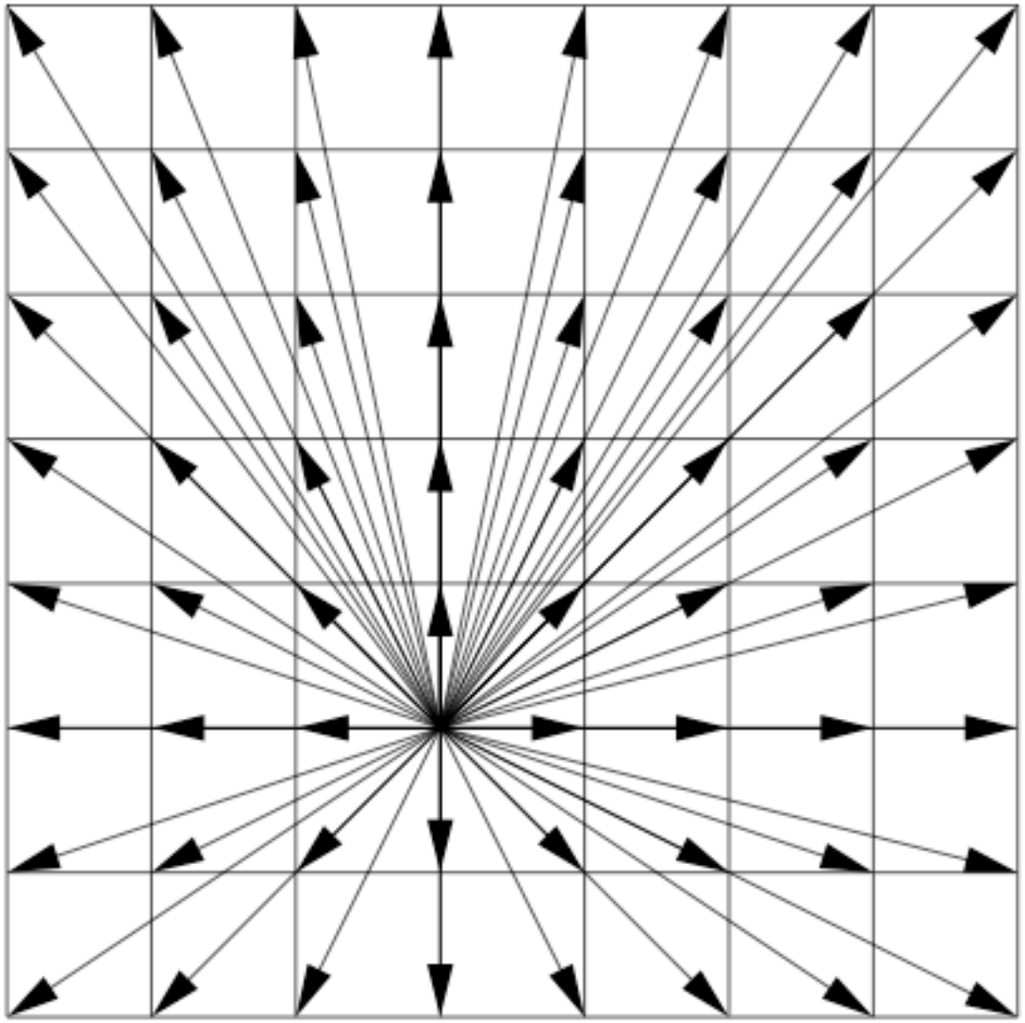
a_0 : absorption cross section

$\Longrightarrow$ calculate the column density $N(r)$:

→ integrate $\rho(r)$ along r

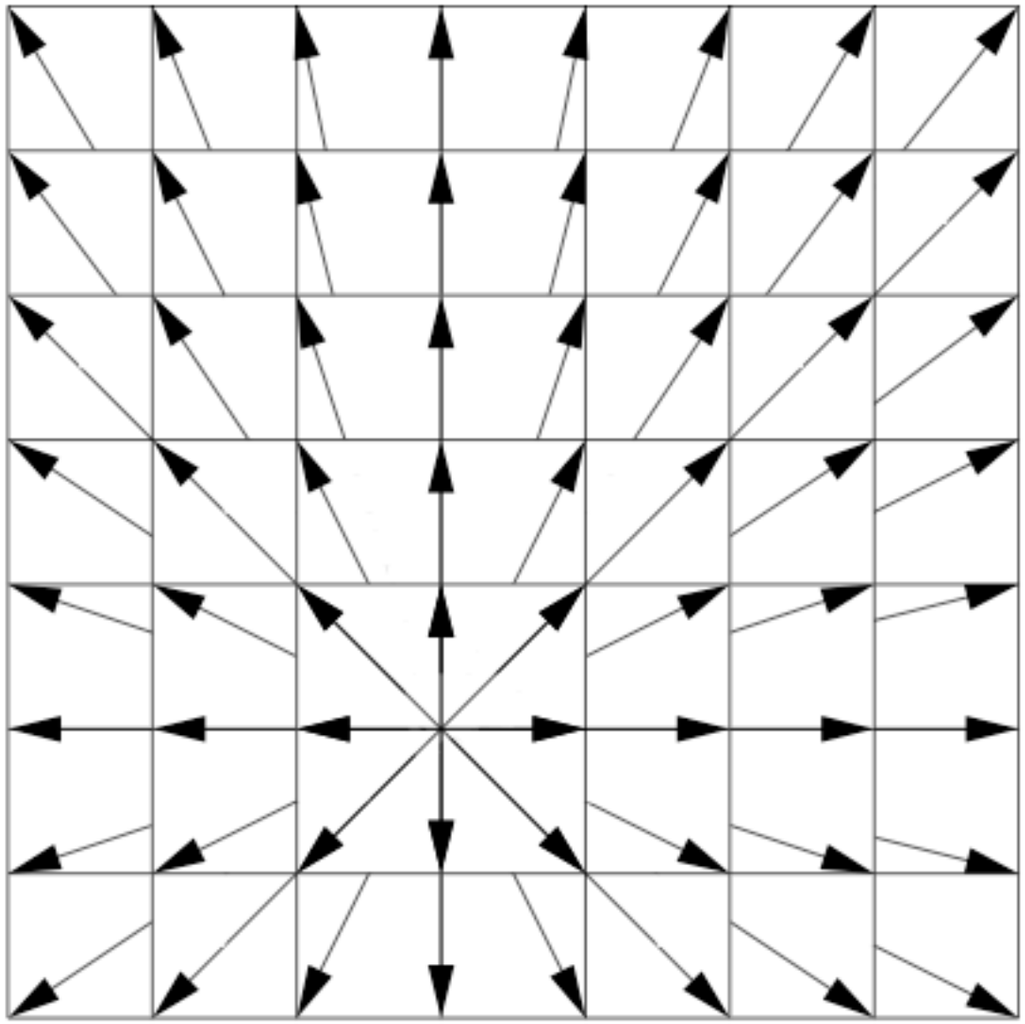
FLASH Code: RT modules

- Ray trace



long characteristics

vs.



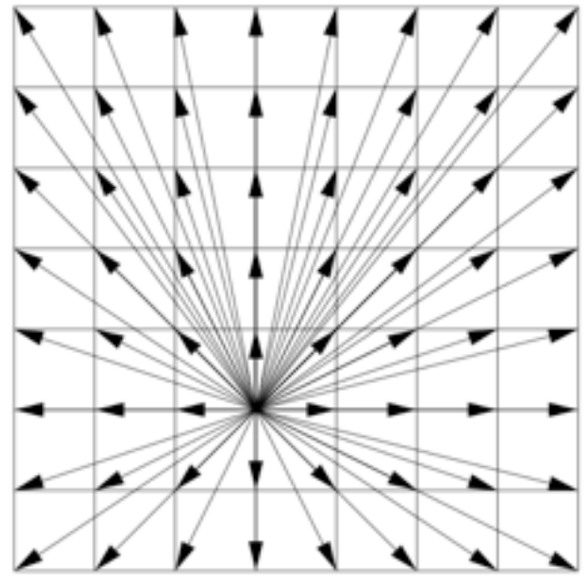
short characteristics

FLASH Code: RT modules

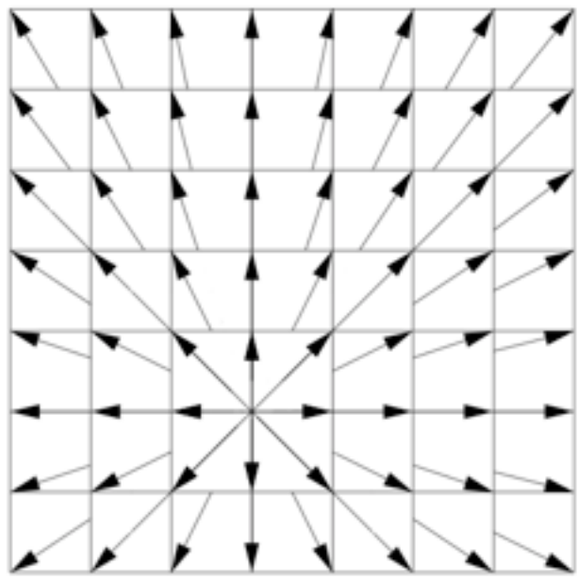
- Ray trace

- long characteristics

  - redundant calculations close to the source
    $\Longrightarrow$ slow

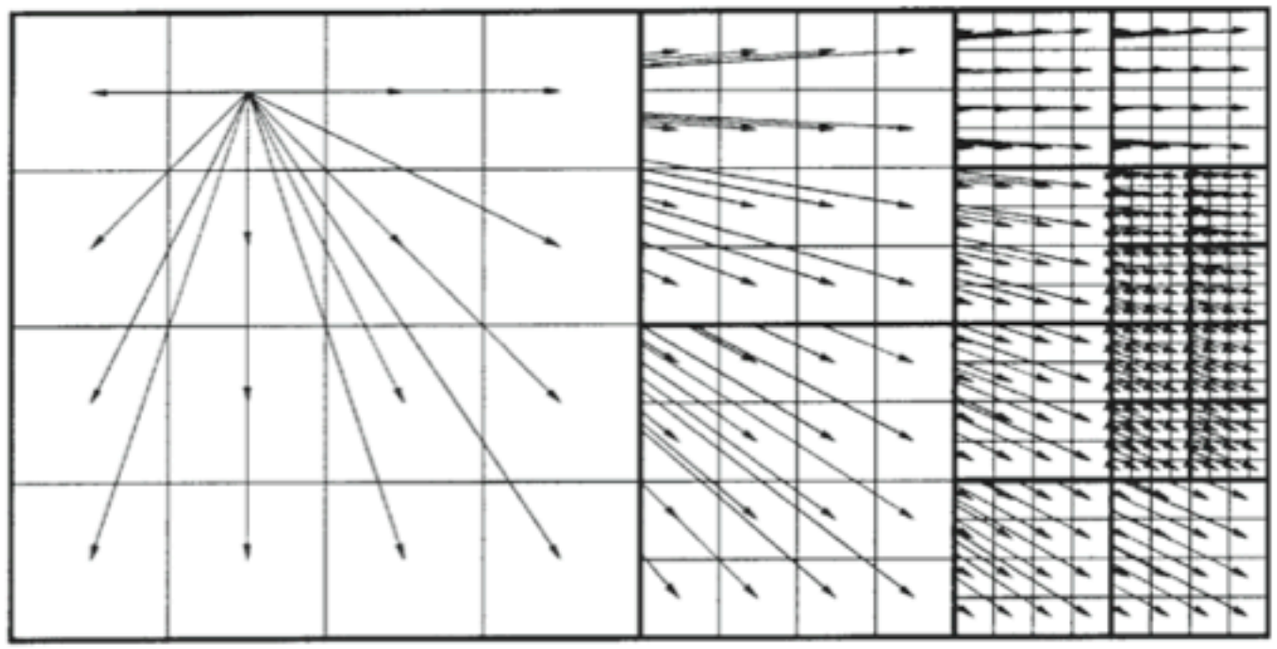


- short characteristics
  $\Longrightarrow$ faster
  $\Longrightarrow$ but has difficulties to handle point sources

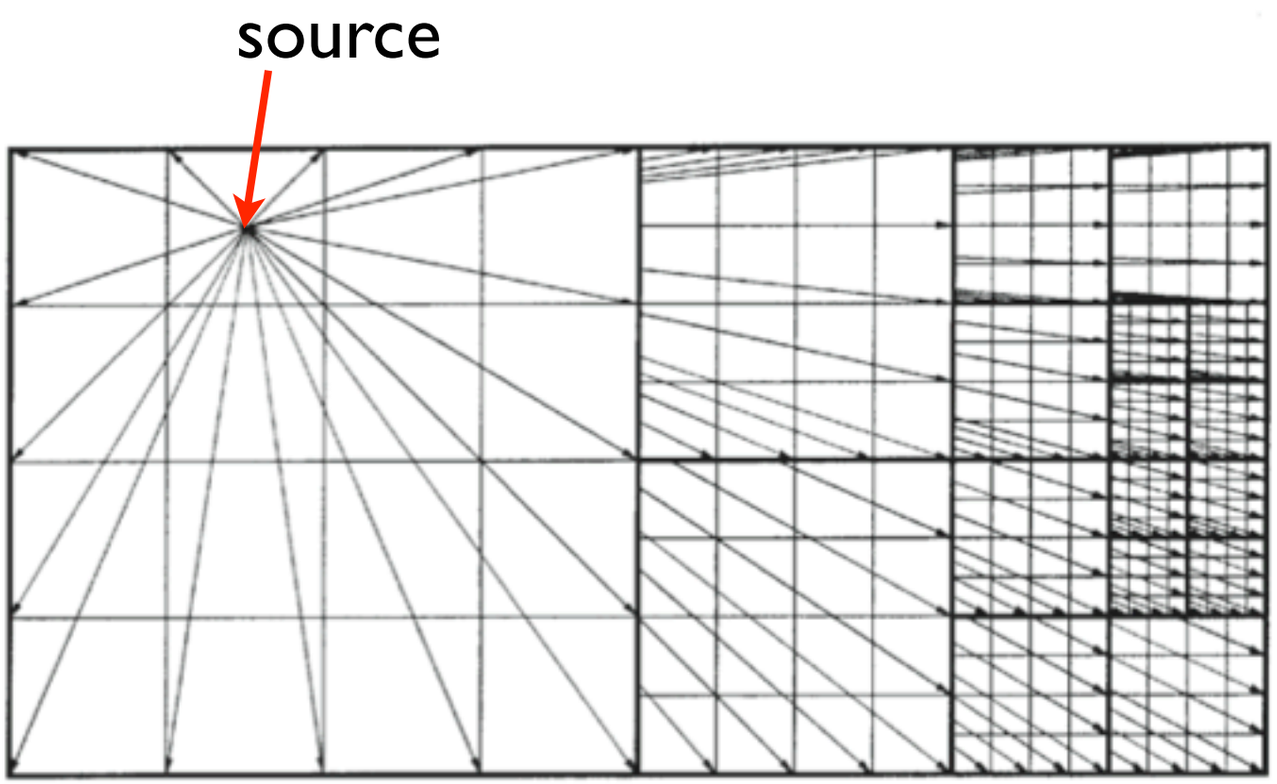


FLASH Code: RT modules

- ray trace: hybrid characteristics (*Rijkhorst et al.* 2006)



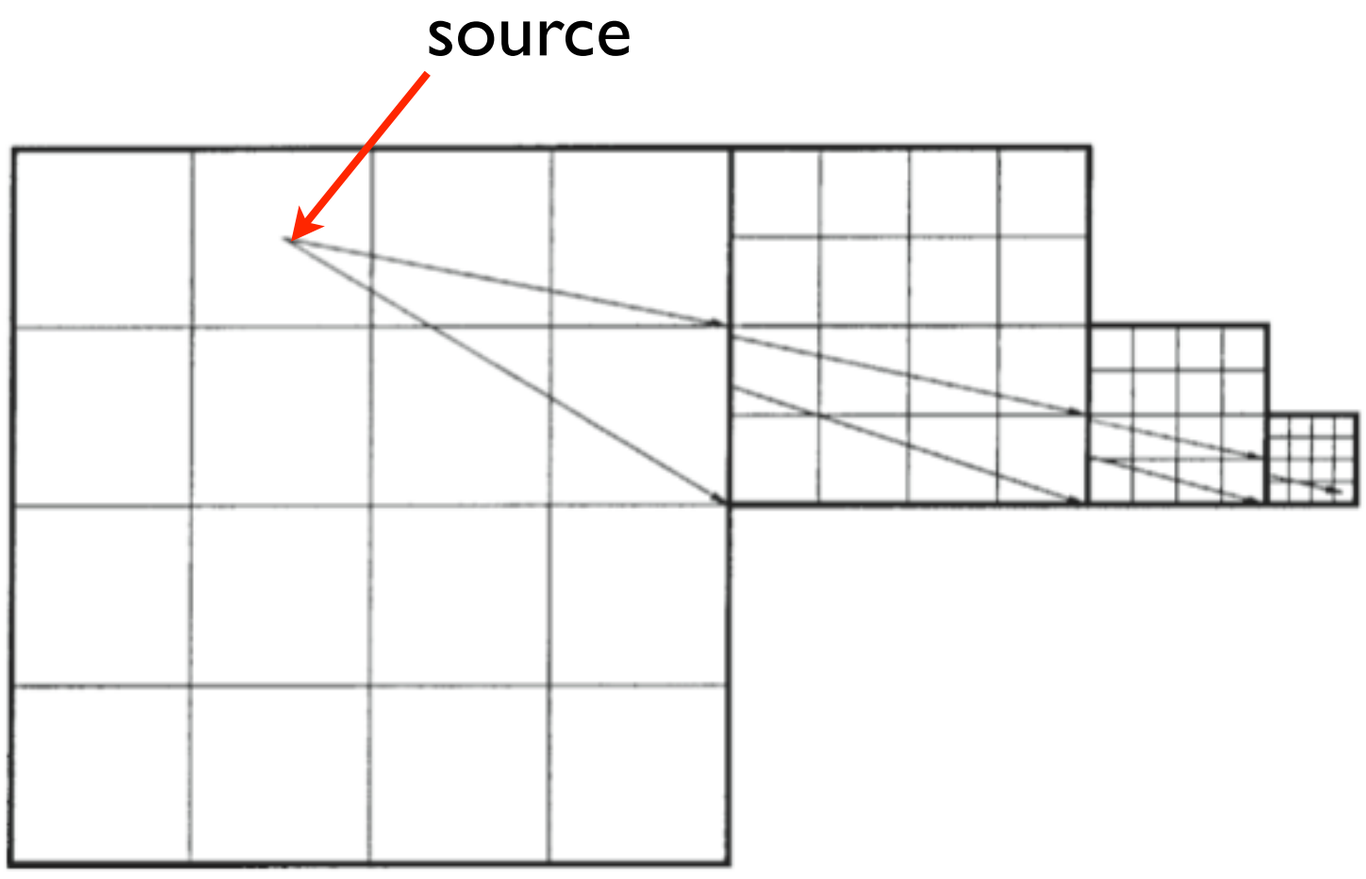
- local contribution to $N(r)$ using a fast-voxel transversal method based on cell-center values



- interpolated face values that need to be communicated

FLASH Code: RT modules

- ray trace: hybrid characteristics

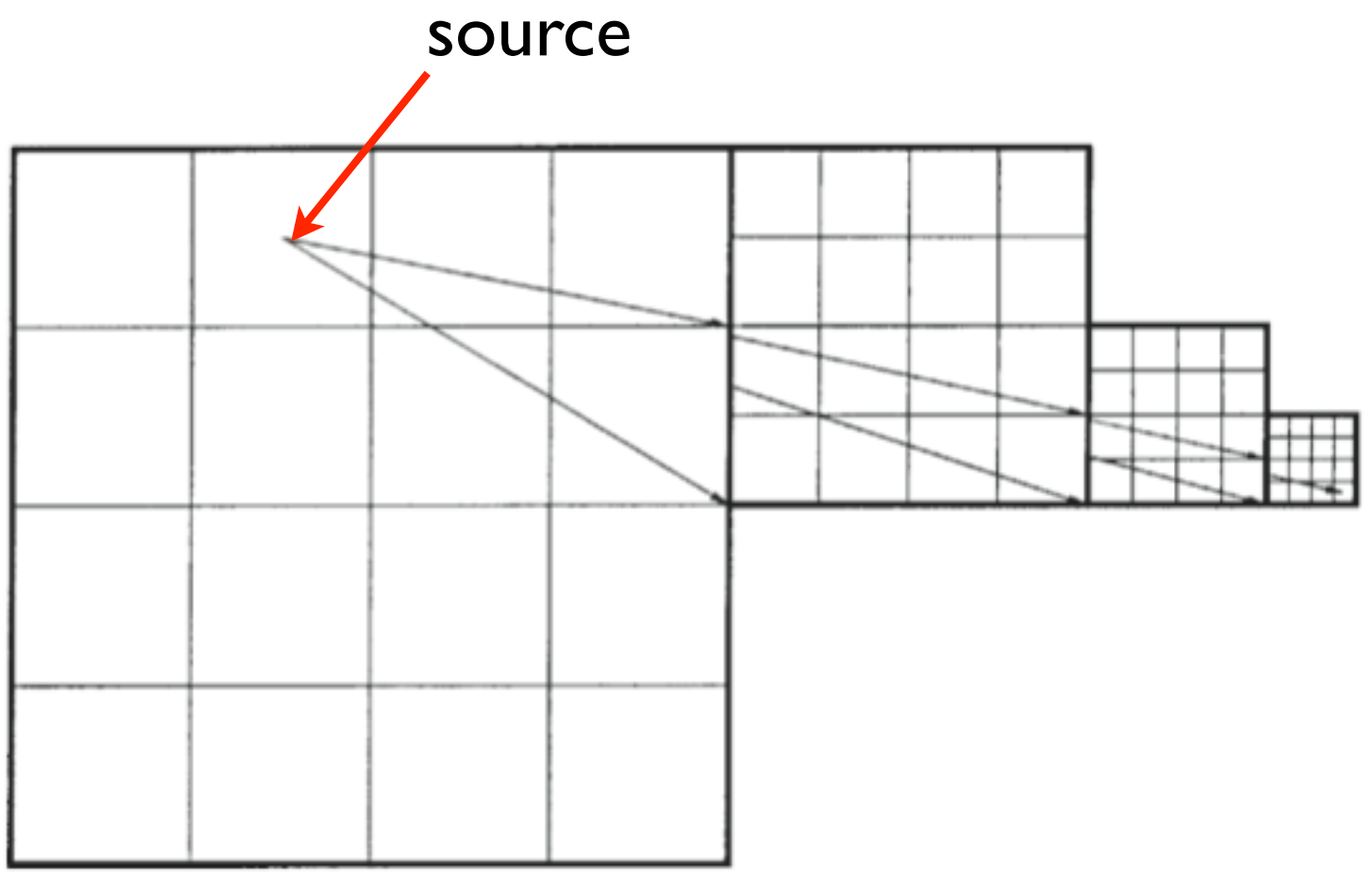


$\Longrightarrow$ build list of face values
for communication
for each process
$\Longrightarrow$ similar to
tree-algorithm

$\Longrightarrow$ list of *patches* (blocks)
which is traversed by
a ray must be known

FLASH Code: RT modules

- ray trace: hybrid characteristics



$\Longrightarrow$ build list of face values
for communication
for each process
$\Longrightarrow$ similar to
tree-algorithm

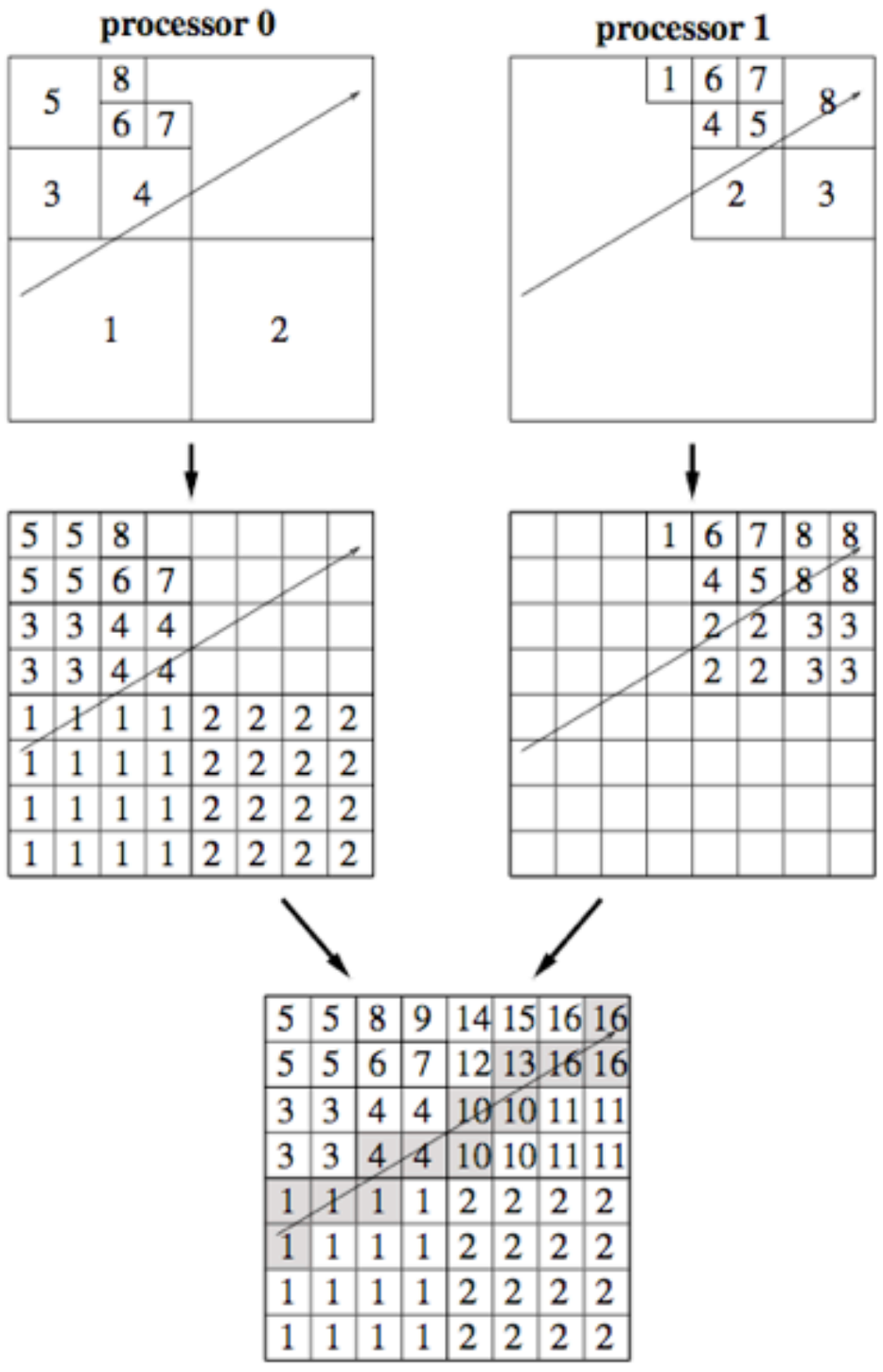
$\Longrightarrow$ list of *patches* (blocks)
which is traversed by
a ray must be known

$\Longrightarrow$ hybrid characteristics:

- no full ray-trace through the entire box
- only interpolated values of blocks are communicated

FLASH Code: RT modules

- Ray trace (*Rijkhorst et al.* 2006)
  $\Longrightarrow$ "patch mapping" of



- original version:
  entire grid on highest refinement level was communicated
  $\Longrightarrow$ communication of redundant information
  $\Longrightarrow$ strong limitations on max. refinement level ($l_{\max} < 7$)

- substantial improvement
  by T.Peters
  $\Longrightarrow$ only "block tree" is communicated

FLASH Code: RT modules

- Ray trace (*Rijkhorst et al.* 2006)
  $\Longrightarrow$ coupling to ionisation:

- rate equation for hydrogen

$$\frac{\mathrm{d}x(\mathrm{HII})}{\mathrm{d}t} = x(\mathrm{HI})(A_\mathrm{p} + A_\mathrm{c}) - x(\mathrm{HII})n_\mathrm{e}\alpha_\mathrm{R}$$

- photoionization rate

$$A_\mathrm{p} = \int_{\nu_0}^{\infty} \frac{4\pi J_\nu}{h\nu} a_\nu \,\mathrm{d}\nu \qquad 4\pi J_\nu(r) = \left(\frac{R_\mathrm{S}}{|r|}\right)^2 \frac{2\pi}{c^2} \frac{h\nu^3}{\exp(\frac{h\nu}{kT_\mathrm{S}}) - 1} \exp(-\tau(r))$$

- collisional ionization rate

$$A_\mathrm{c} = A_\mathrm{c}(\mathrm{HI}) n_\mathrm{e} \sqrt{T} \exp(-I(\mathrm{HI})/k_\mathrm{B}T)$$

- radiative recombination rate

$$\alpha_\mathrm{R} = \alpha_\mathrm{R}(10^4\,\mathrm{K}) \left(\frac{T}{10^4\,\mathrm{K}}\right)^{-0.7}$$

I(HI):
ionisation potential

FLASH Code: RT modules

- Ray trace (*Rijkhorst et al.* 2006)

$\Longrightarrow$ coupling to ionisation:

  - photoionisation heating

$$\Gamma_{\rm p} = n({\rm HII}) \int_{\nu_0}^{\infty} \frac{4\pi J_\nu}{h\nu} a_0 h(\nu - \nu_0)\, {\rm d}\nu,$$

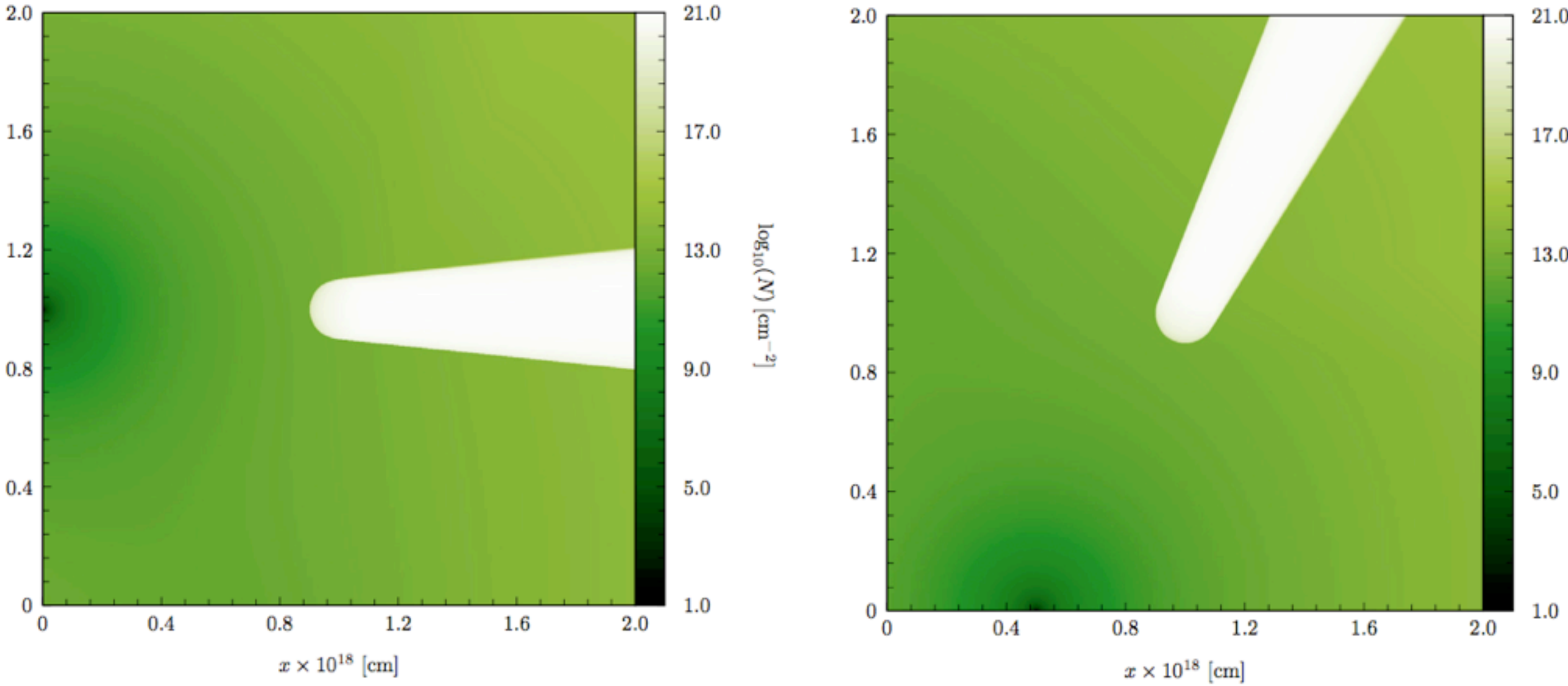
- include metal-line cooling

- sub-cycling on thermal timestep
  $\Longrightarrow$ find convergence of $x(T)$ and $T(x)$

FLASH Code: RT modules

- Ray trace (*Rijkhorst et al.* 2006)

Examples: "irradiated" clump

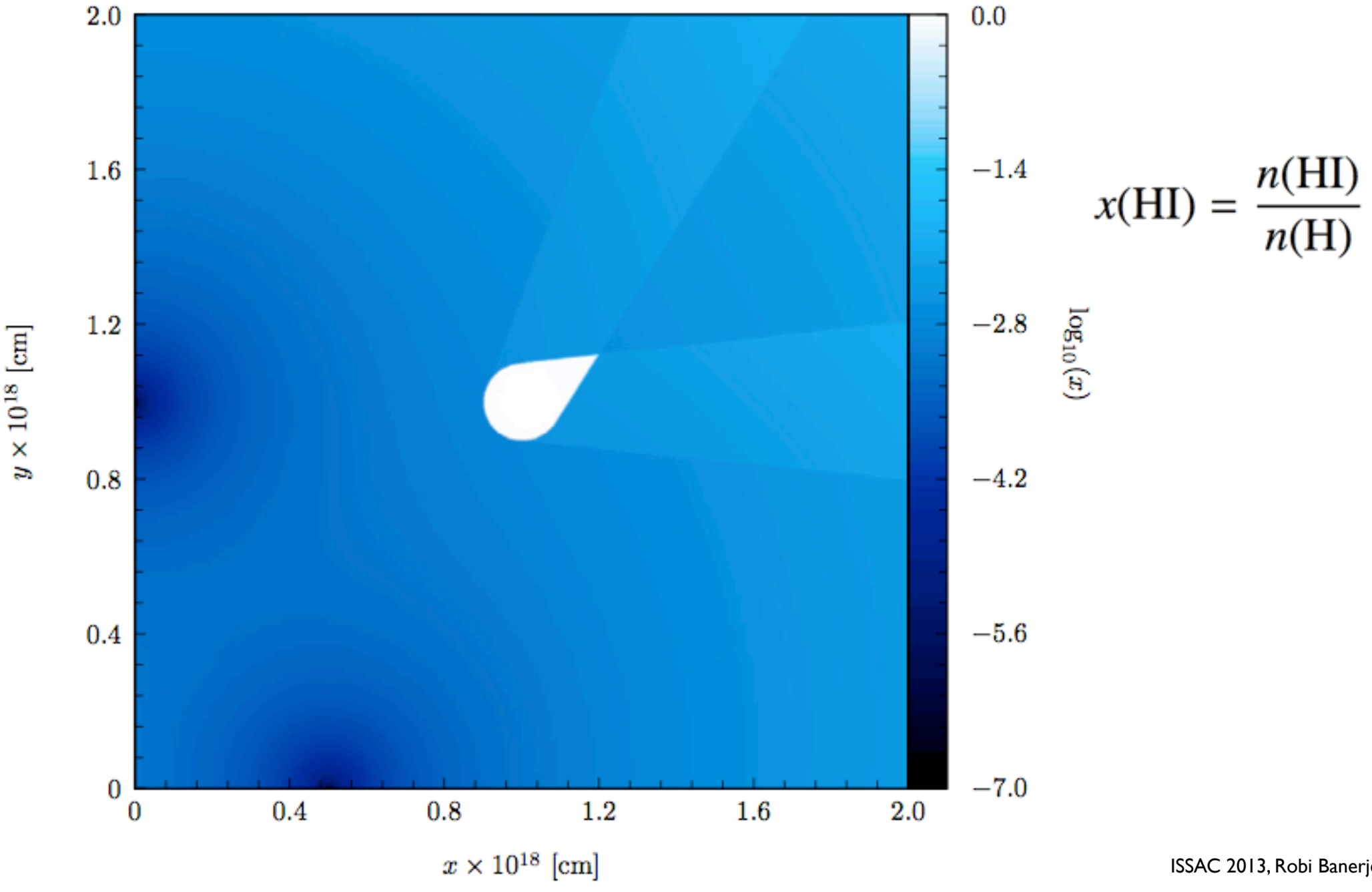


HI column density

FLASH Code: RT modules

- Ray trace (*Rijkhorst et al.* 2006)

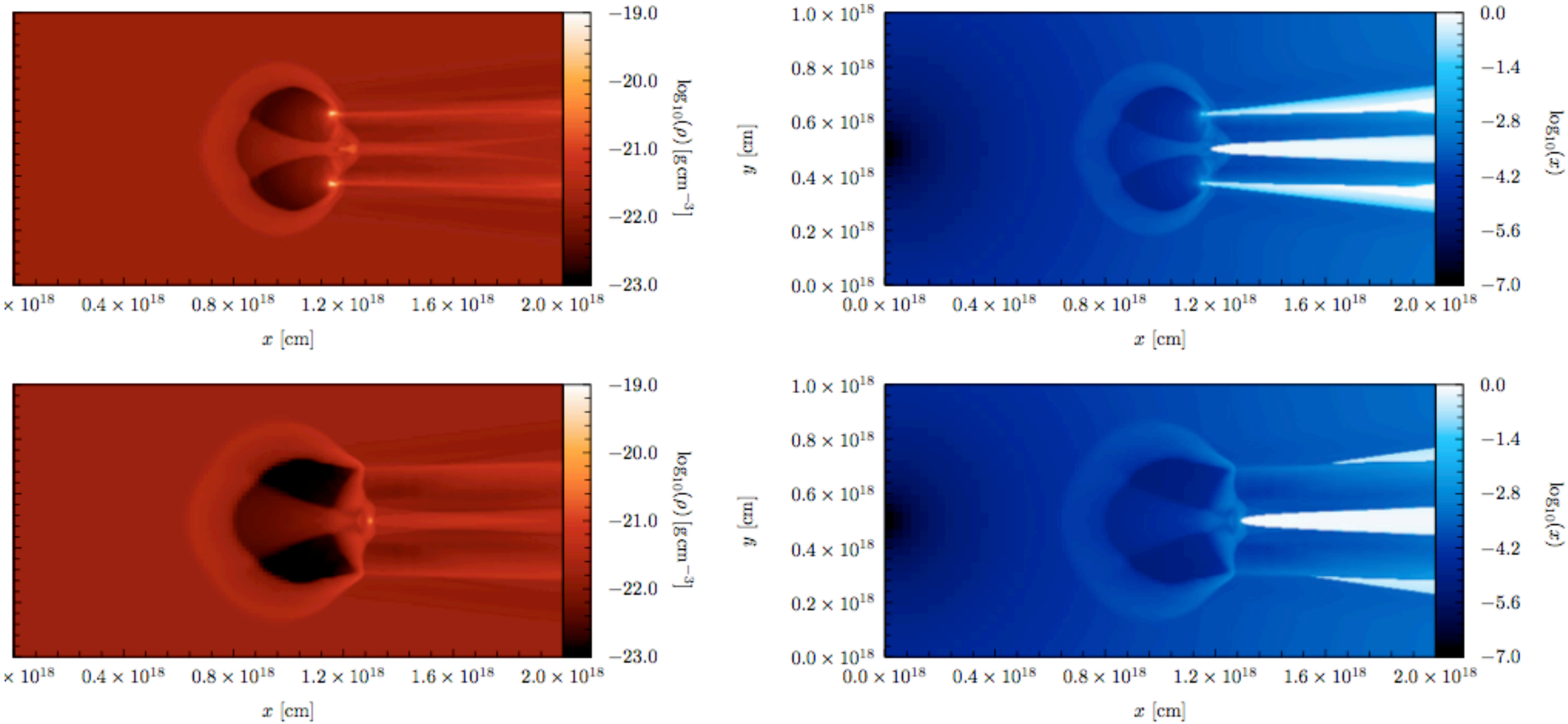
Examples: "irradiated" clumps with two sources



FLASH Code: RT modules

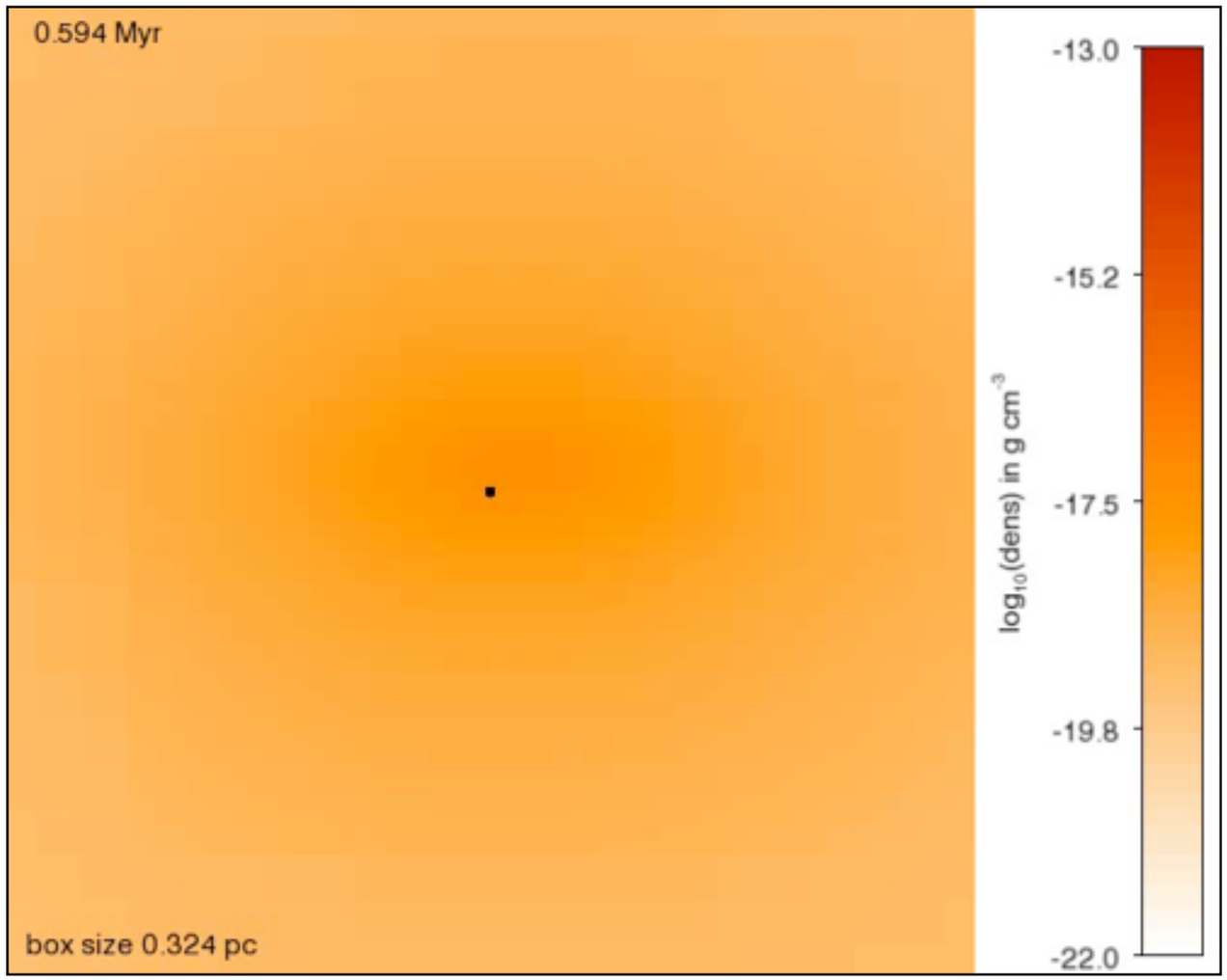
- Ray trace (*Rijkhorst et al.* 2006)

  Examples: photo-evaporation of two clumps
  with photo-ionisation heating

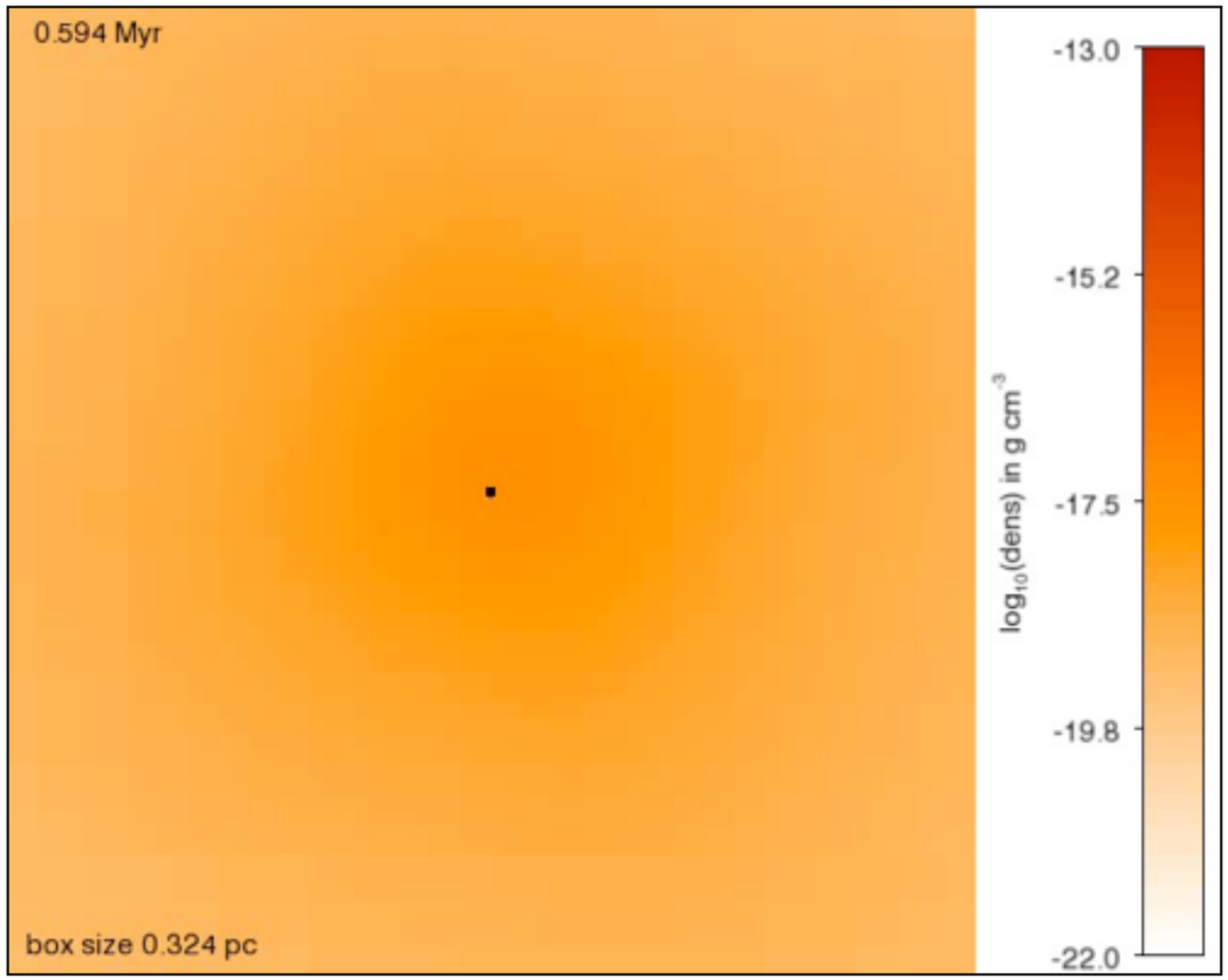


Massive Star Formation: Dynamics of HII Regions

Simulations by Thomas Peters (collapse of 1000 M_{sol} cloud core)
$\Longrightarrow$ use sink mass to get stellar luminosity and temperature
(*Paxton* 2004, ZAMS table)



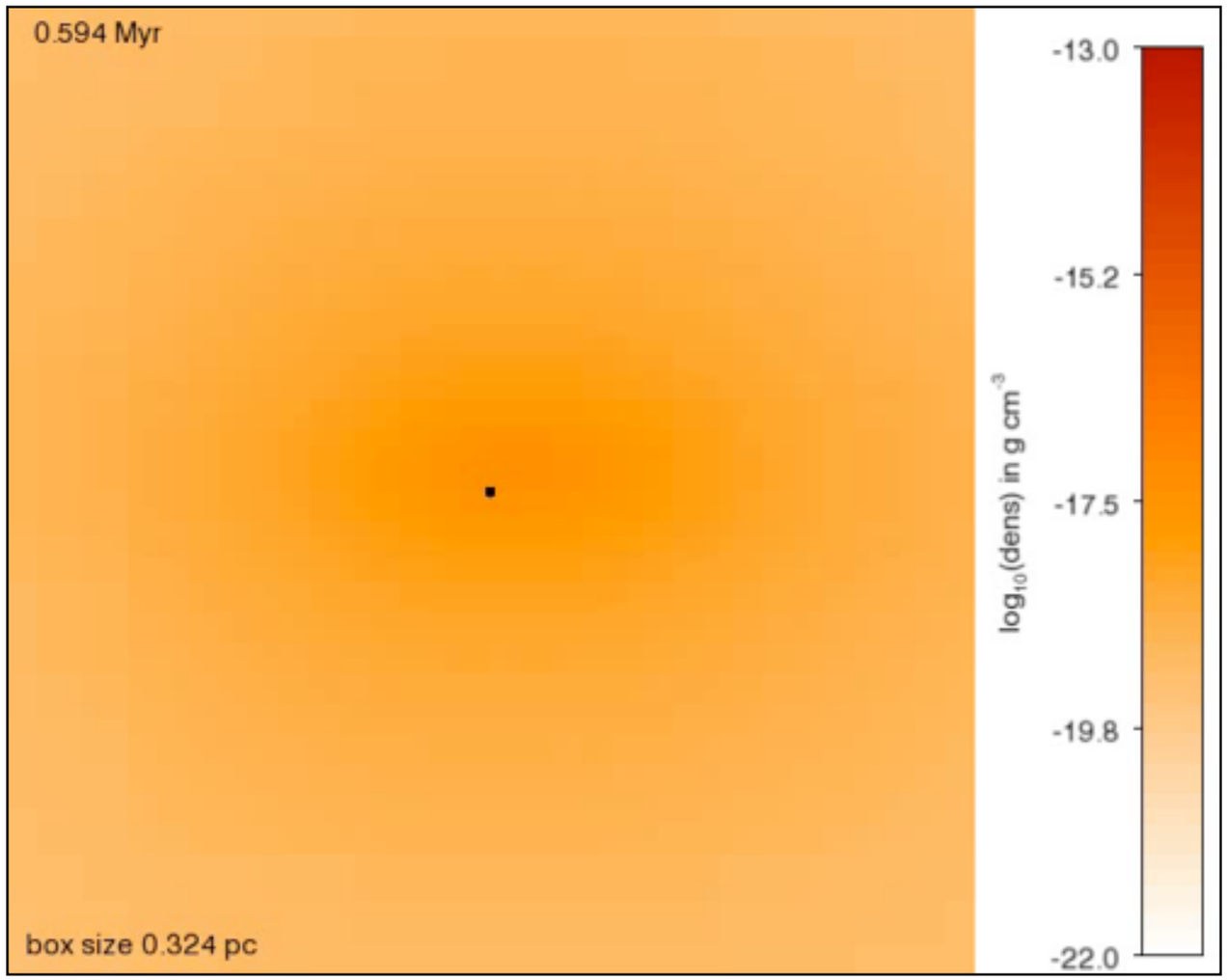
Disk edge on



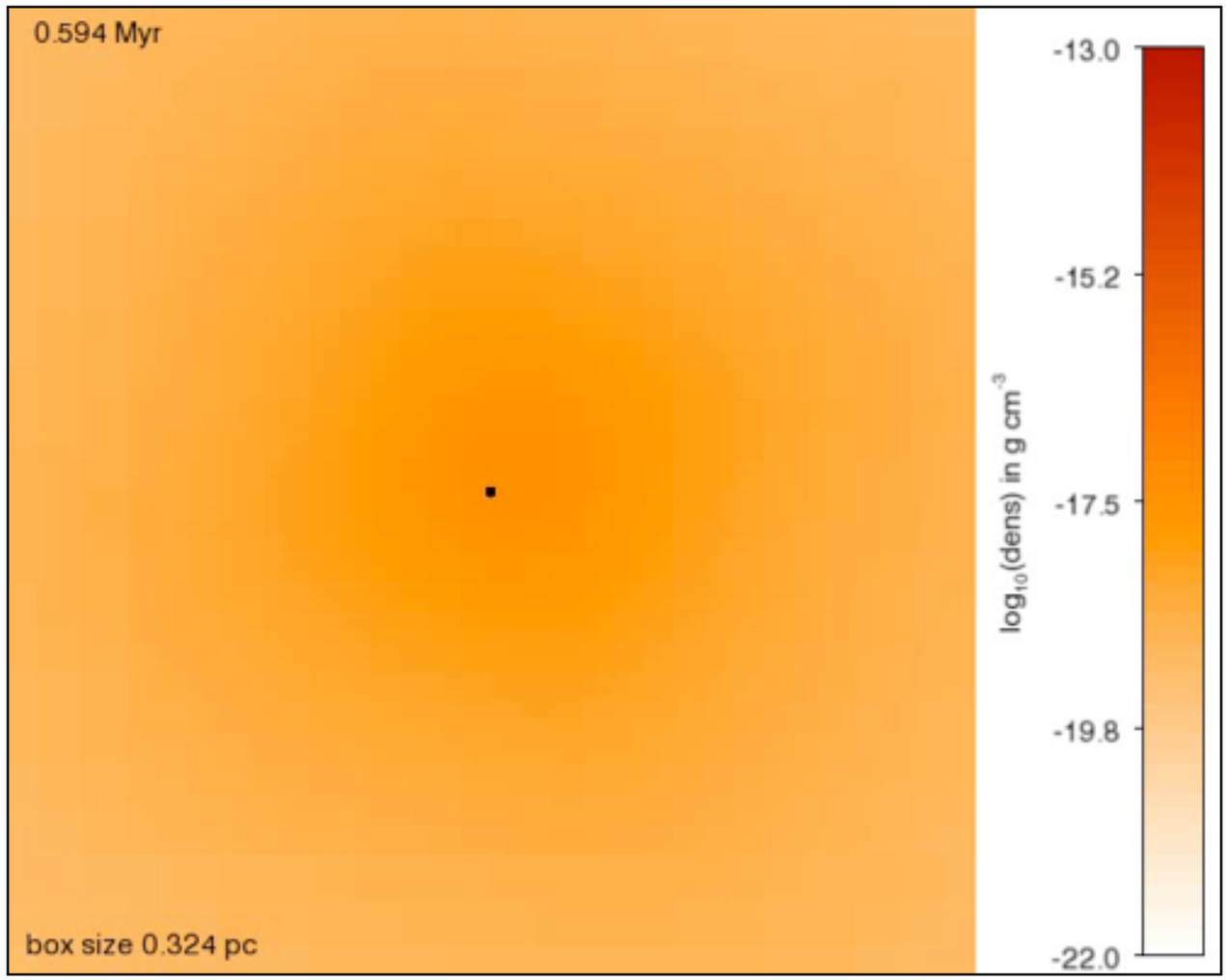
Disk plane

Massive Star Formation: Dynamics of HII Regions

Simulations by Thomas Peters (collapse of 1000 M_{sol} cloud core)
$\Longrightarrow$ use sink mass to get stellar luminosity and temperature
(*Paxton* 2004, ZAMS table)

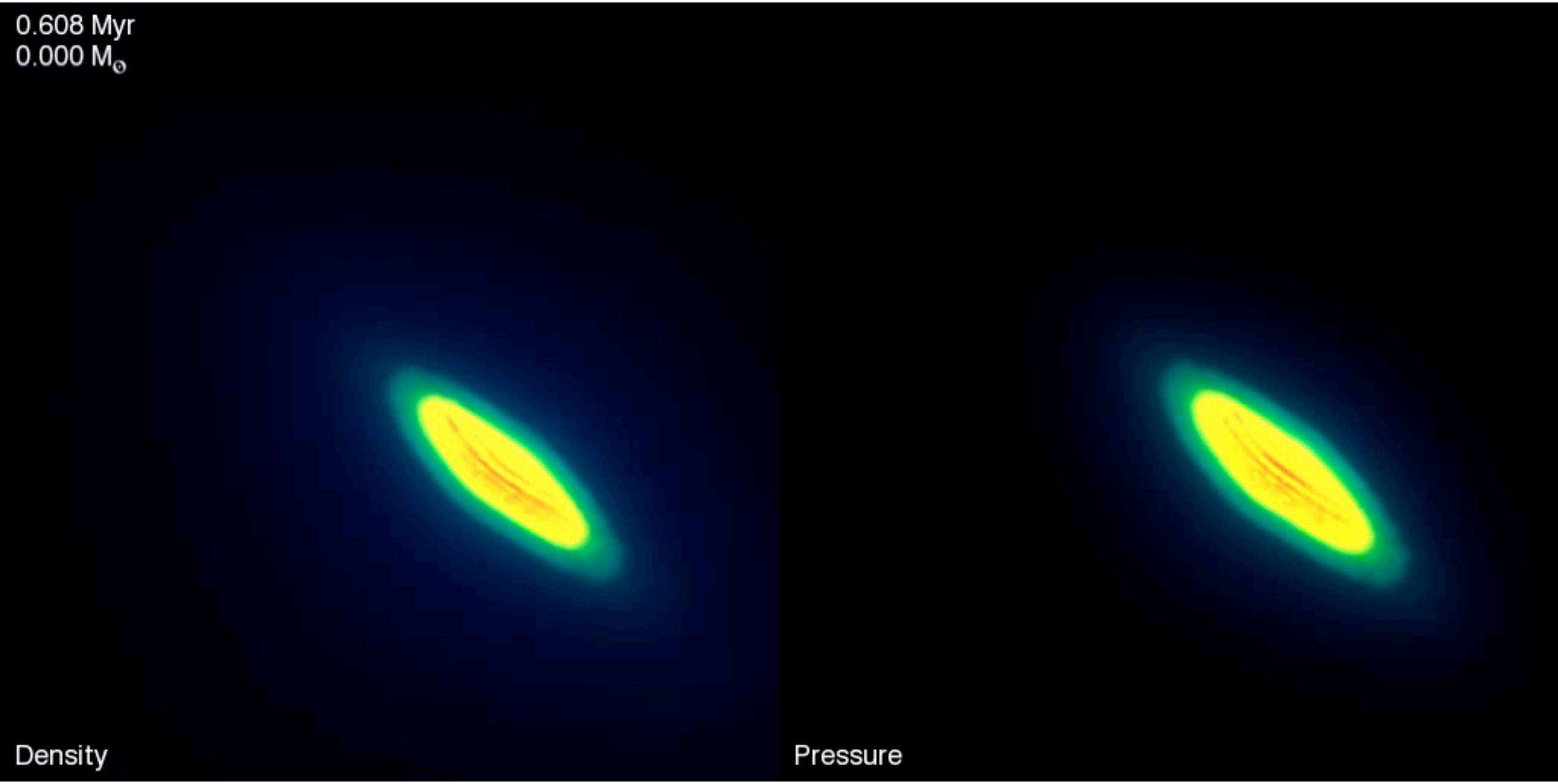


Disk edge on



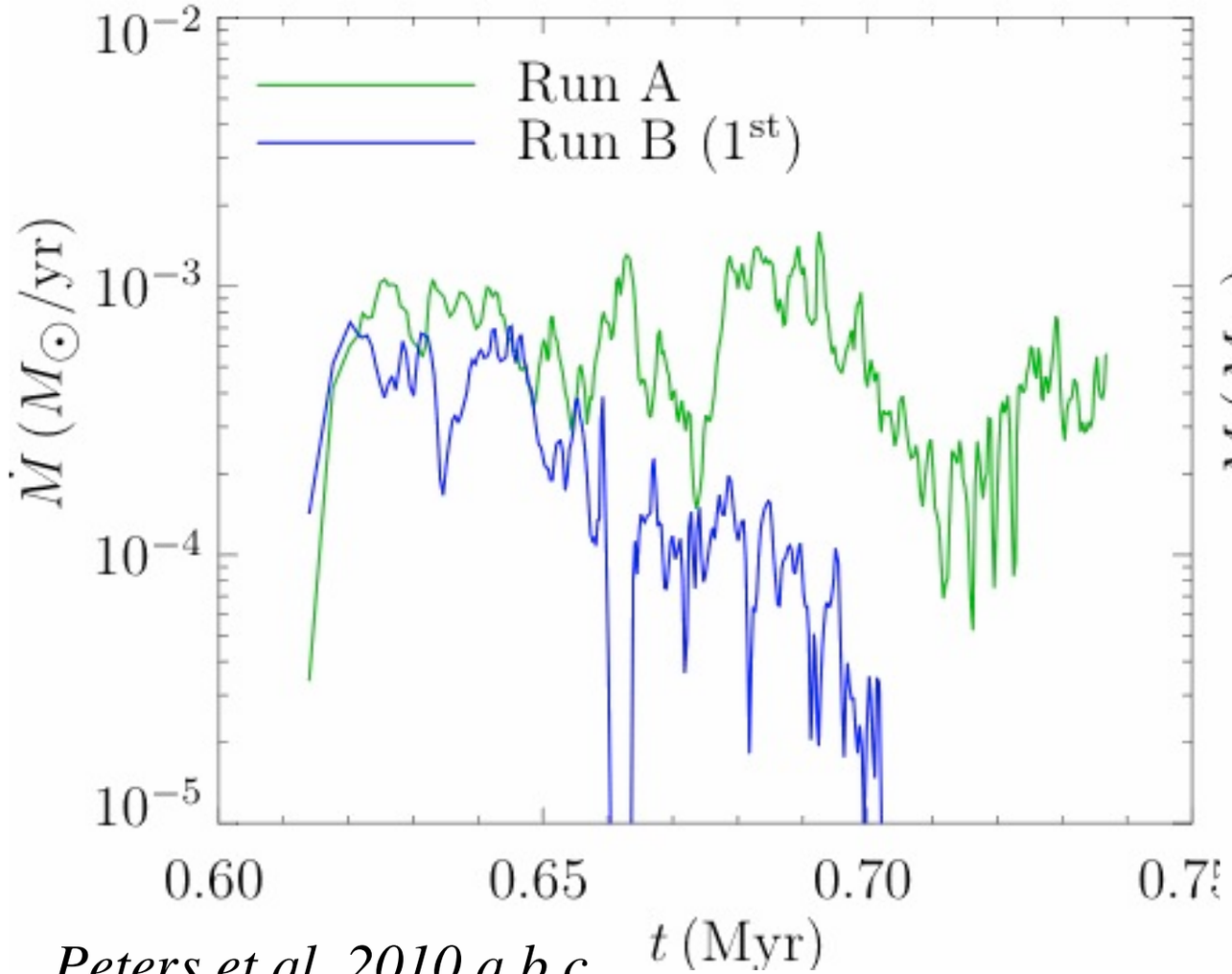
Disk plane

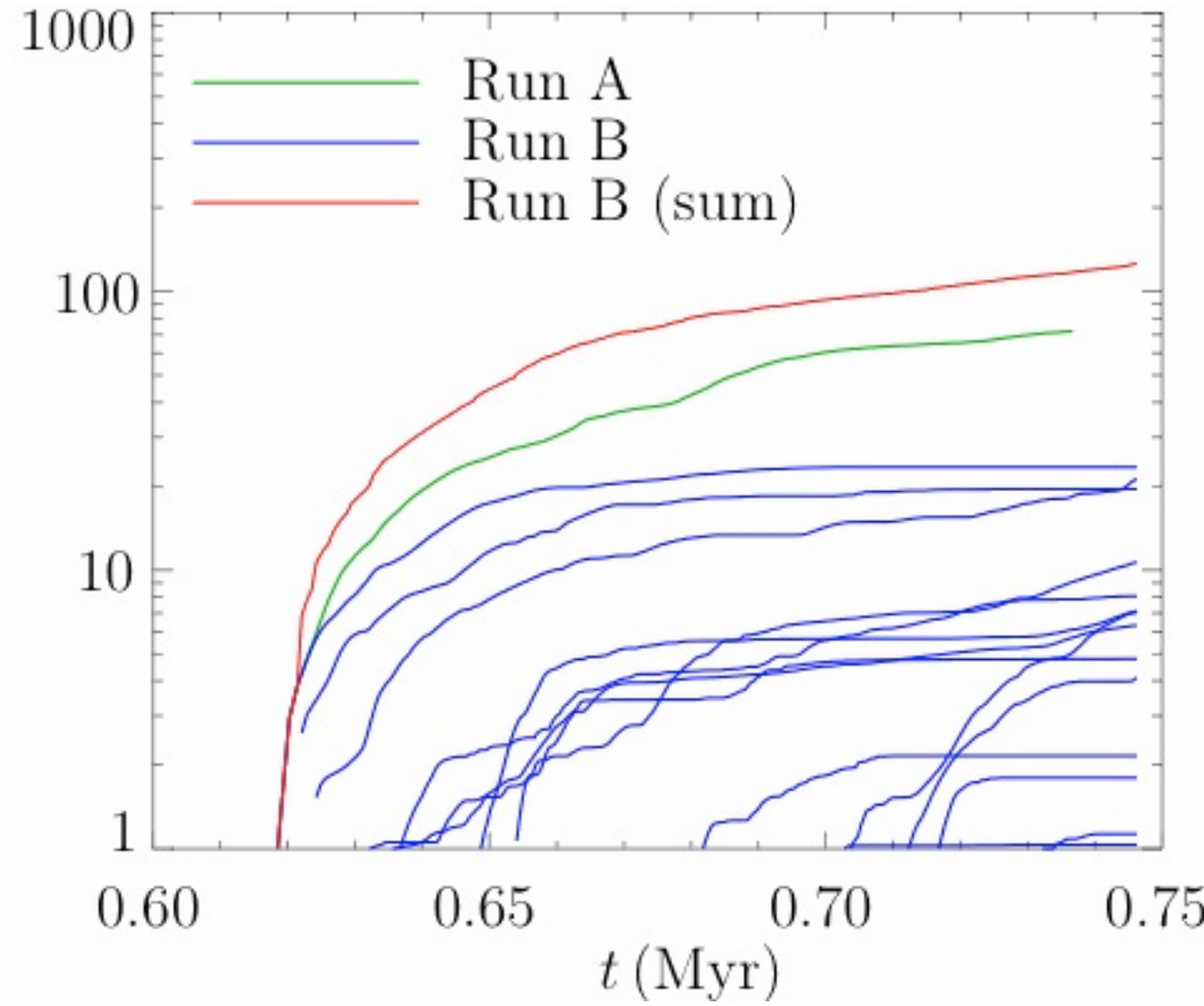
Massive Star Formation: Dynamics of HII Regions



courtesy: Zilken, NIC, Jülich

Multiple protostars: Dynamics of the H II Region

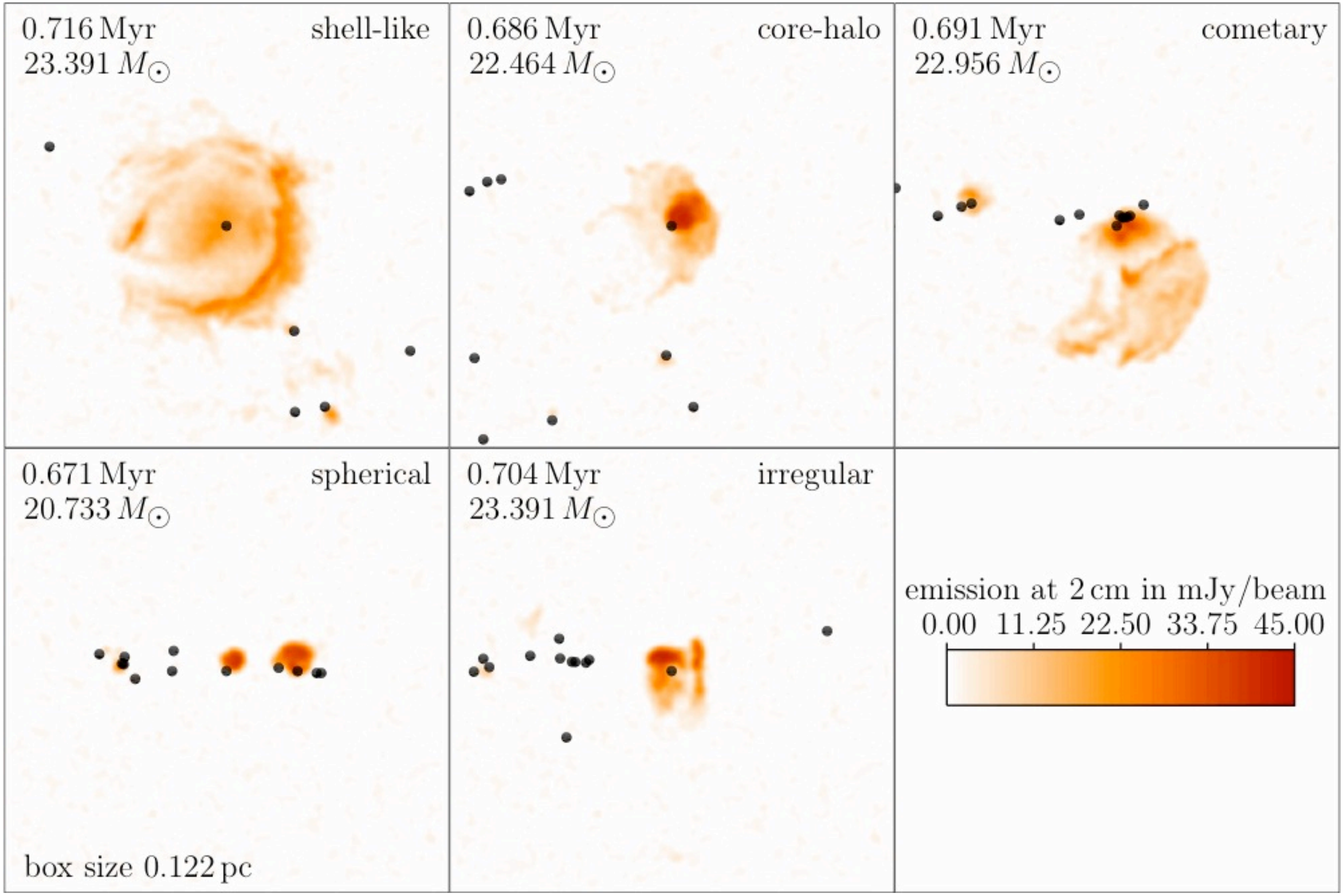




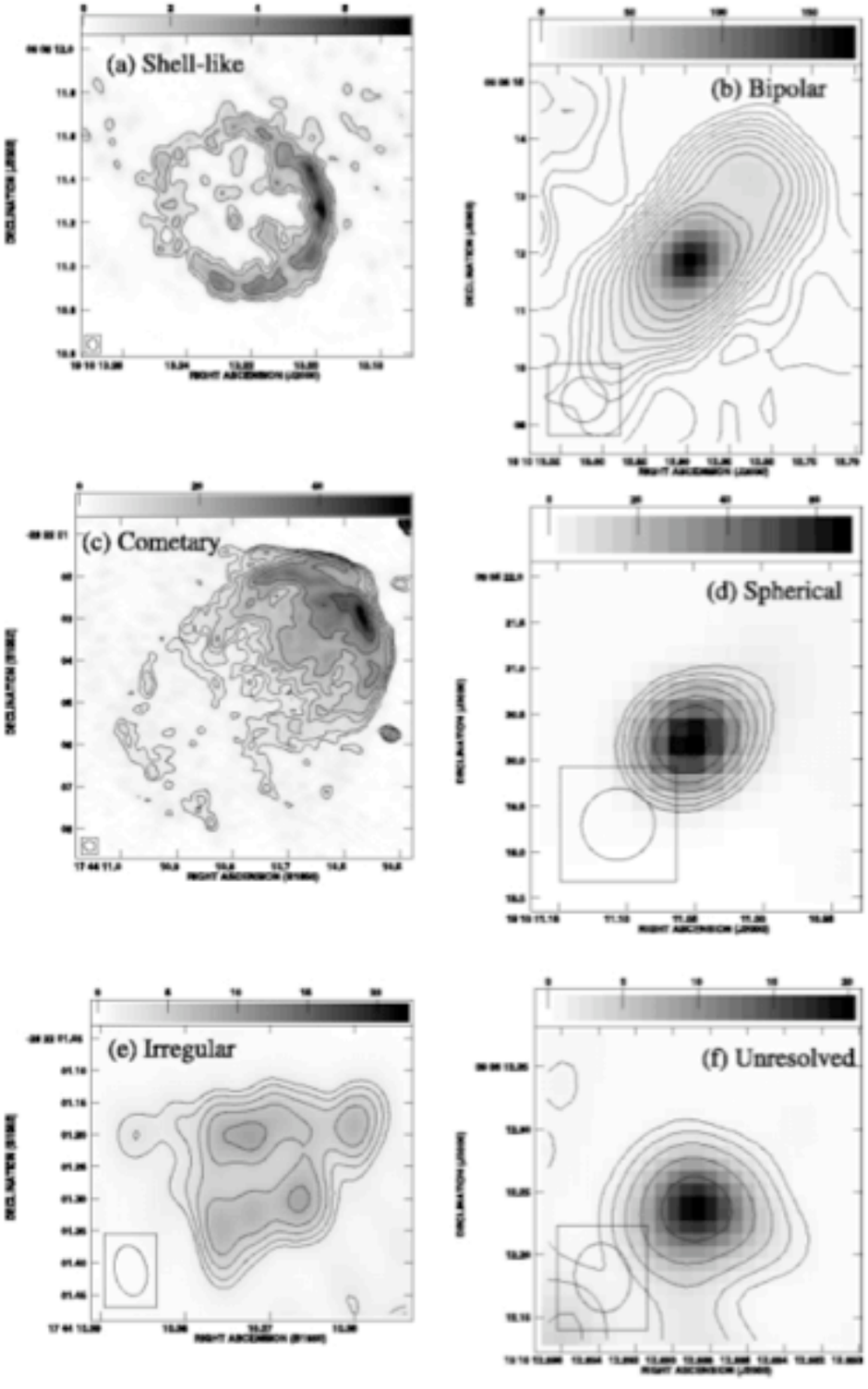
Peters et al. 2010 a,b,c

- ionization feedback does **not** shut off accretion
- **fragmentation**-induced starvation (FIS)
- massive stars form in cluster

H II Region Morphologies



H II Region Morphologies



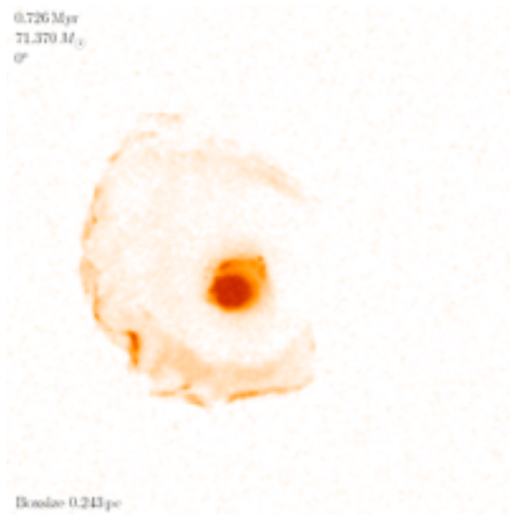
morphologies from *De Pree et al. 2005*

Table 3
Percentage Frequency Distribution of Morphologies

Type	WC89	K94	Run A	Run B
Spherical/Unresolved	43	55	19	60 ± 5
Cometary	20	16	7	10 ± 5
core–halo	16	9	15	4 ± 2
Shell-like	4	1	3	5 ± 1
Irregular	17	19	57	21 ± 5

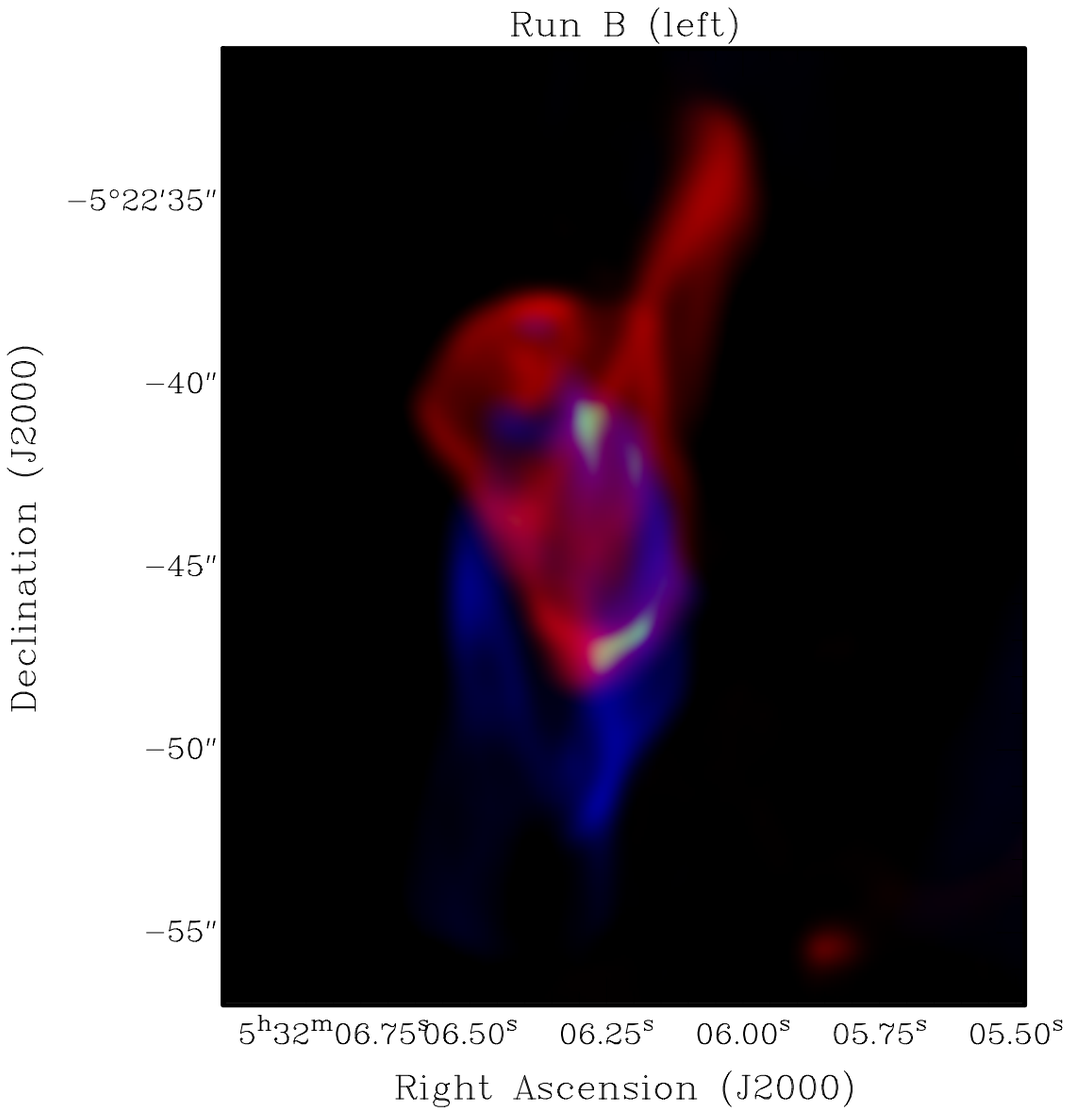
Peters et al. 2010b

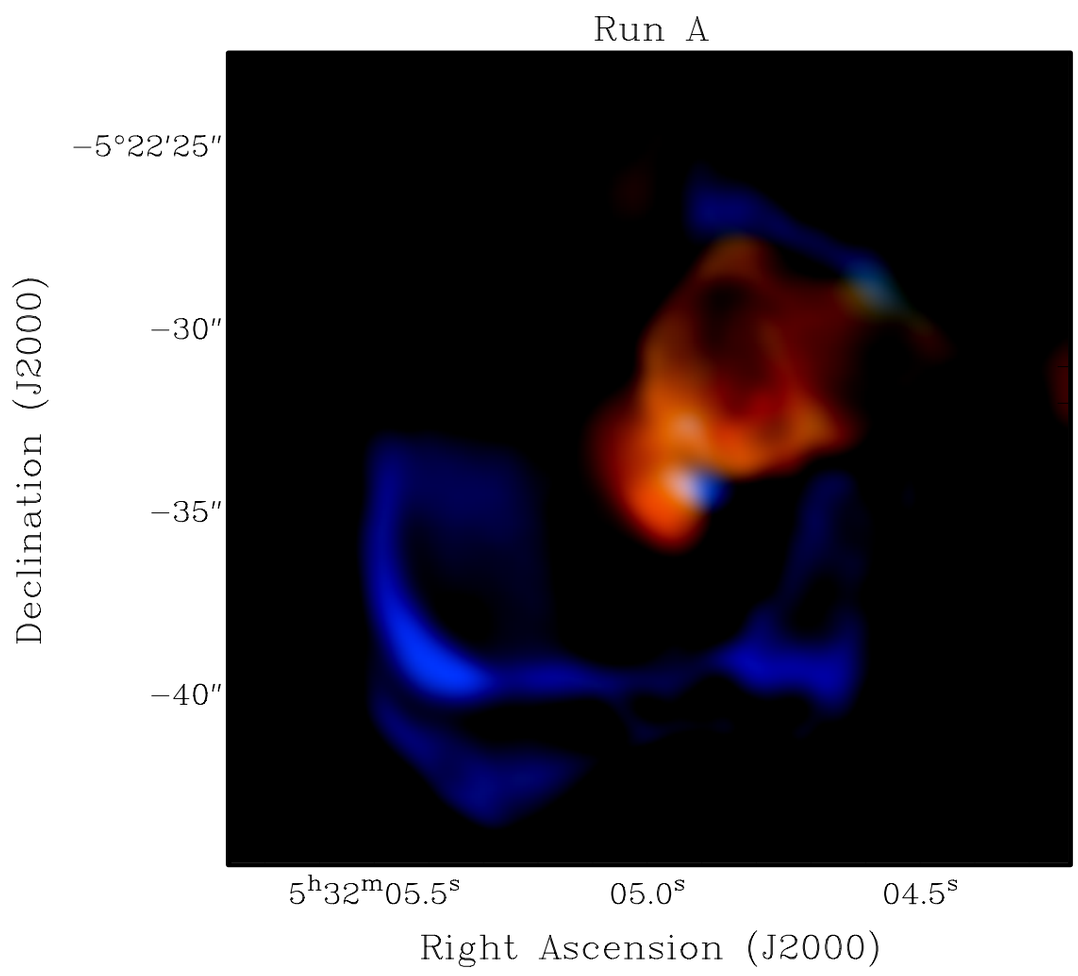
- only clustered SF match observed statistics



morphology at different viewing angles

Comparison with Observations: Outflows



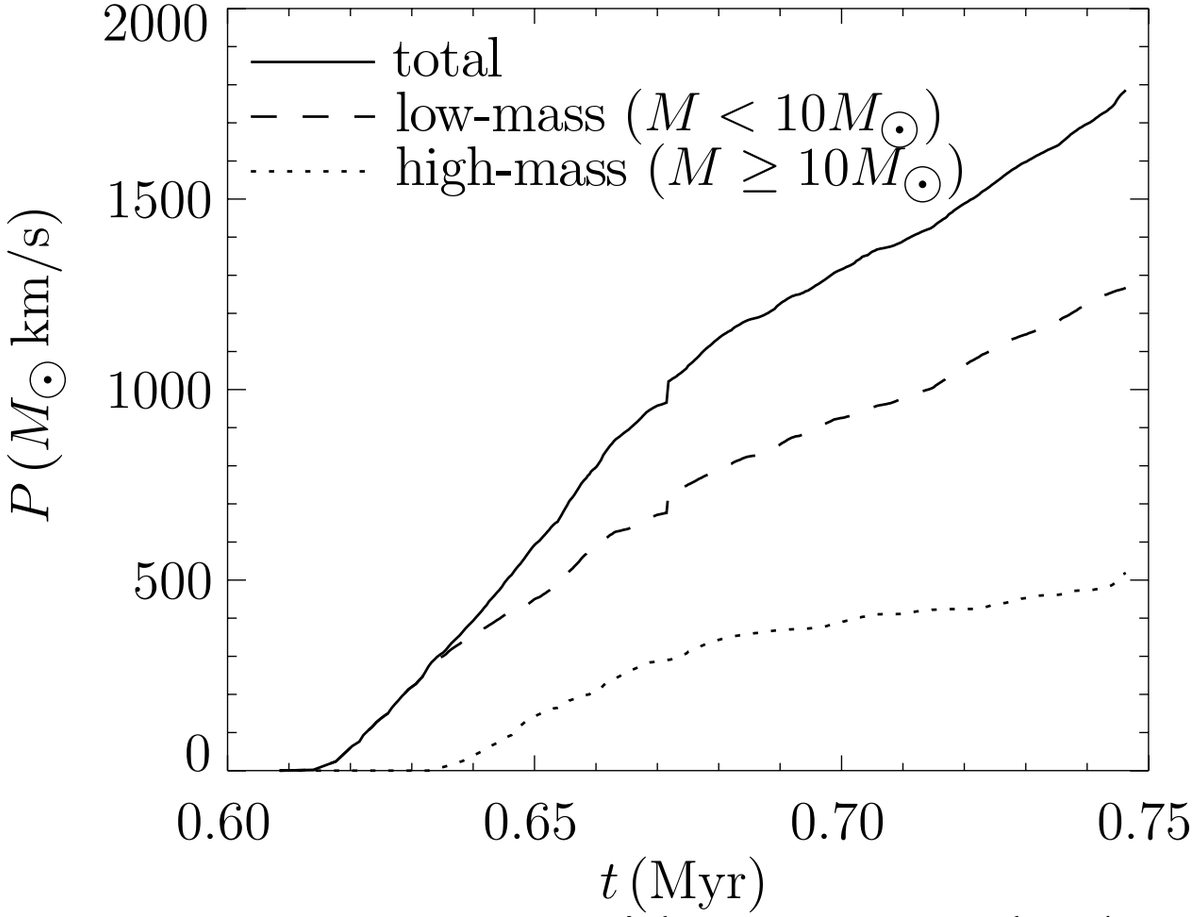


- Synthetic CO maps with the ALMA simulator CASA @ G5.89−0.39 distance: 1.3 kpc

Comparison with Observations: Outflows

ALMA		M ($M_\odot$)	V (km s^{-1})	P ($M_\odot$ km s^{-1})	E (10^{44} erg)	L ($L_\odot$)	$\dot{M}$ (10^{-3} $M_\odot$ yr^{-1})	R (AU)
Run A	Blue	0.082 ± 0.000	3.888 ± 0.099	0.320 ± 0.009	0.124 ± 0.007	0.255 ± 0.065	0.206 ± 0.042	4100
	Red	0.101 ± 0.000	3.297 ± 0.154	0.333 ± 0.016	0.109 ± 0.010	0.225 ± 0.066	0.252 ± 0.051	4100
Run B (left)	Blue	0.050 ± 0.000	3.446 ± 0.000	0.171 ± 0.001	0.059 ± 0.000	0.121 ± 0.025	0.124 ± 0.025	3300
	Red	0.141 ± 0.000	2.757 ± 0.184	0.388 ± 0.026	0.106 ± 0.014	0.219 ± 0.073	0.352 ± 0.071	2100
Run B (right)	Blue	0.060 ± 0.000	3.550 ± 0.143	0.213 ± 0.009	0.075 ± 0.006	0.155 ± 0.044	0.150 ± 0.030	5000
	Red	0.044 ± 0.000	2.370 ± 0.000	0.104 ± 0.000	0.024 ± 0.000	0.050 ± 0.010	0.109 ± 0.022	4100

Peters, Klaassen et al. 2012



$P \propto 0.1\, M_{\mathrm{accr}}$ with $v_{\mathrm{wind}} \sim 150$ km/sec

→ derived outflow parameters are on the **low** end of observations

→ Ionisation feedback is **not** the main driver of molecular outflows

→ common **low mass** companions drive large scale molecular outflows?

FLASH Code: RT modules

- current development: Lars Buntemeyer, Hamburg

  $\Longrightarrow$ extending ray-trace from point source

  to multiple plane parallel rays to handle scattering