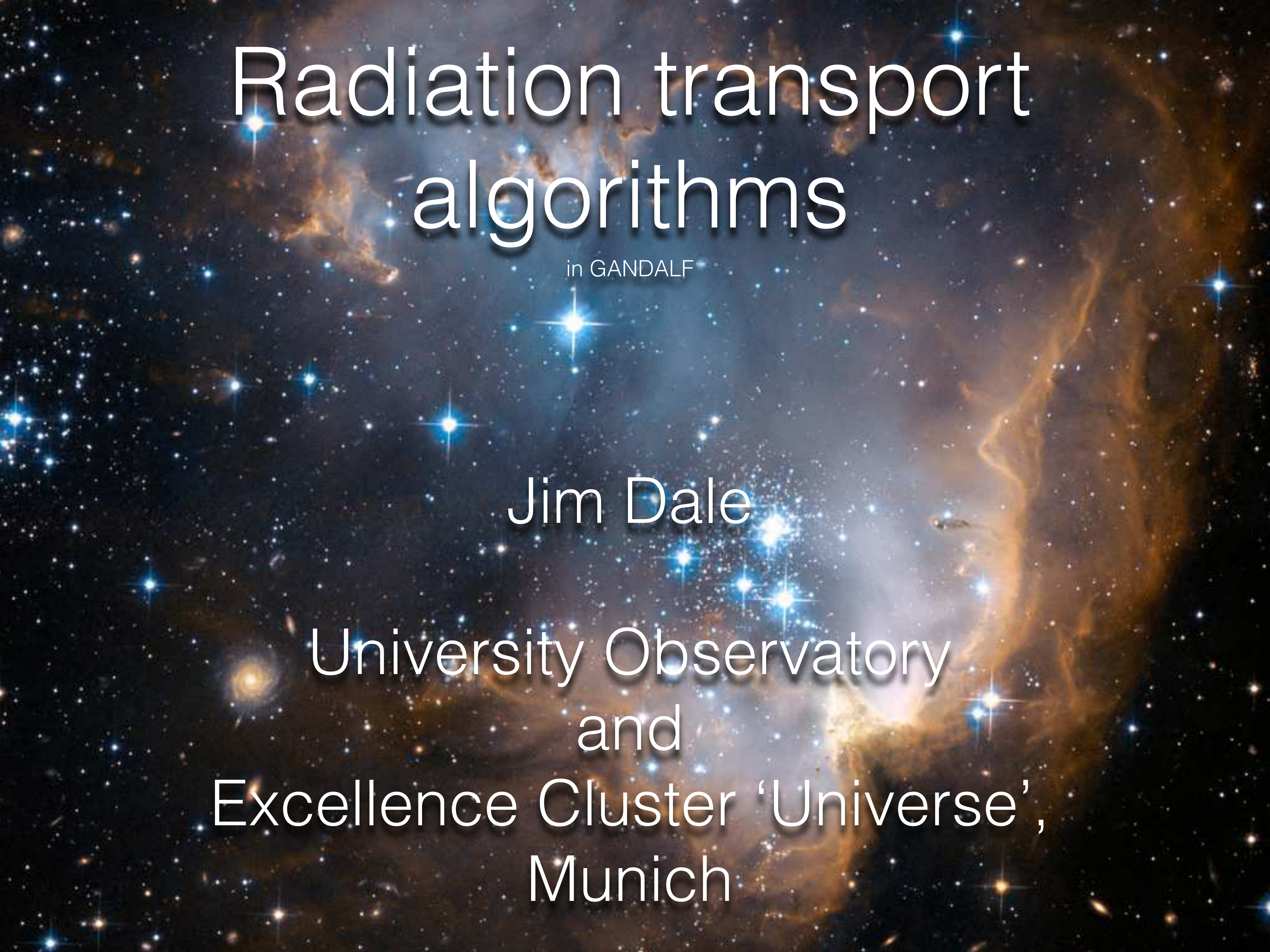


Radiation transport algorithms

in GANDALF

Jim Dale

University Observatory
and
Excellence Cluster ‘Universe’,
Munich

Introduction

This will be a discussion of radiation transport, biased towards SPH (GANDALF now has other hydro solvers…)

Most of what follows is covered in more detail in

'The modelling of feedback in star formation simulations'
Dale, J. E., 2015, *New Astronomy Reviews* vol. 68, pp. 1-33

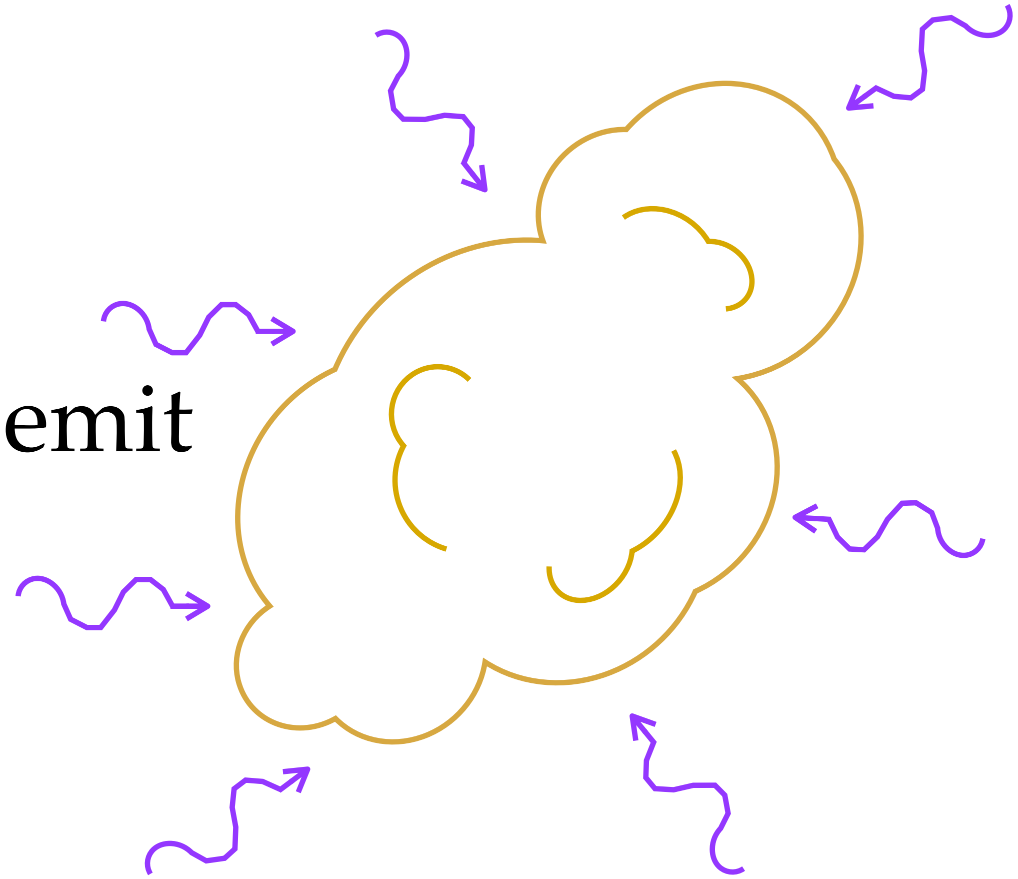
Other reviews on this topic:
'Three-dimensional dust radiative transfer'
Steinacker, J. et al, 2013 *ARA&A* vol. 51 pp. 63-104

'Computer simulations of cosmic reionisation'
Trac, H. Y. & Gnedin, N. Y., 2011 *Adv. Sci. Lett.*, vol. 4 p. 228

Introduction

What are your sources of radiation?

- one point source?
- many point sources?
- an extended source?
- inside/outside simulation domain?
- more than one of the above?



What kind of radiation do your sources emit and why?

- narrowband/continuum?
- steady (e.g. the UV background)?
- slowly-varying (e.g. an evolving O-star)?
- complicated (e.g. an accreting protostar)?

Transporting photons

Once you know what kind of radiation is entering your grid and where….what are you going to do with all the photons?

For the purposes of this talk, I am only going to mention methods that use some technique to *inject photons into, and follow photons through,* the computational domain.

I'm not going to talk about Stamatellos+ 2007 (estimates optical depths using polytropes, computes heating/cooling rates directly - *now in GANDALF!!*), Urban+ 2009, Henney+ 2009 (using tabulated results from CLOUDY code),…

$$\frac{1}{c}\frac{\partial I_\nu}{\partial t} + \mathbf{n}.\nabla I_\nu = \epsilon_\nu - \kappa_\nu \rho I_\nu$$

The radiation transport equation(s)

$$\frac{1}{c}\frac{\partial I_\nu}{\partial t} + \mathbf{n}.\nabla I_\nu = \epsilon_\nu - \kappa_\nu \rho I_\nu$$

where:

I_ν is the *specific intensity* at frequency ν, in erg s^{-1} sr^{-1} Hz^{-1}

$\mathbf{n}$ is a unit vector pointing in the direction the radiation is propagating in.

ϵ_ν is the specific *emissivity* of the medium, i.e. how good it is at *emitting* radiation itself at that frequency

κ_ν is the specific *absorption coefficient* of the medium, i.e. how good it is at *absorbing* radiation at that frequency

The radiation transport equation(s)

$$\frac{1}{c}\frac{\partial I_\nu}{\partial t} + \mathbf{n}.\nabla I_\nu = \epsilon_\nu - \kappa_\nu \rho I_\nu$$

Why is this equation such a pain to solve?

(i) ϵ_ν and κ_ν often functions of gas temperature…which is determined by solution of the radiation transport equation.

(ii) ϵ_ν and κ_ν can be very strong and non-smooth/non-monotonic functions of ν - may need to solve at many frequencies

(iii) radiation transport in principle allows every part of the computational domain to communicate with every other part, at the speed of light.

Simplifying the radiation transport equation

$$\frac{1}{c}\frac{\partial I_\nu}{\partial t} + \mathbf{n}.\nabla I_\nu = \epsilon_\nu - \kappa_\nu \rho I_\nu$$

There are ways of simplifying the RT equation:

(i) Ignore the time-dependent part entirely, ~equivalent to assuming that the radiation field finds an equilibrium on a timescale short compared with any of the hydrodynamic timescales.

$$\mathbf{n}.\nabla I_\nu = \epsilon_\nu - \kappa_\nu \rho I_\nu$$

Example: Monte Carlo methods

Emit large numbers N of photon packets from sources, carrying an energy

$$\epsilon = L\Delta t/N$$

Follow them through the simulation domain and allow resolution elements to absorb energy from packets, or consider the contribution to the local energy density of all packets passing through a given location.

Packets are followed until some stopping criterion is met:
(i) all packets have been absorbed or have left the domain
(ii) radiative equilibrium has been achieved

(e.g. Ercolano+ 2003, 2008, Pawlik & Schaye 2008, Haworth & Harries 2012)

Example: flux-limited diffusion

Monte Carlo simulations are slow. An alternative is flux-limited diffusion (FLD) - treat the radiation as a *fluid* inserted at the locations of the sources which diffuses away.

This is a *moment method,* where one integrates out the angular dependence of the intensity and integrates over frequency:

Radiative energy: $$E = \frac{1}{c}\int_{\nu=0}^{\infty}\int I_\nu \mathrm{d}\Omega \mathrm{d}\nu$$

Radiative flux: $$\mathbf{F}_i = \int_{\nu=0}^{\infty}\int I_\nu \hat{\mathbf{n}}.\hat{\mathbf{x}}_\mathbf{i} \mathrm{d}\Omega \mathrm{d}\nu$$

`Radiation pressure tensor’: $$\mathbf{P}_{ij} = \int_{\nu=0}^{\infty}\int I_\nu (\hat{\mathbf{n}}.\hat{\mathbf{x}}_\mathbf{i})(\hat{\mathbf{n}}.\hat{\mathbf{x}}_\mathbf{j}) \mathrm{d}\Omega \mathrm{d}\nu.$$

Example: flux-limited diffusion

Can now write down equations of *radiation-fluid dynamics*: (Turner & Stone 2001)

Mass:

$$\frac{\mathrm{D}\rho}{\mathrm{D}t} + \rho\nabla.\mathbf{v} = 0$$

Momentum of gas:

$$\rho\frac{\mathrm{D}\mathbf{v}}{\mathrm{D}t} = -\nabla p + \frac{1}{c}\chi_F F$$

Radiation energy:

$$\rho\frac{\mathrm{D}}{\mathrm{D}t}\left(\frac{E}{\rho}\right) = -\nabla.F - \nabla\mathbf{v} : \mathbf{P} + 4\pi\kappa_P B - c\kappa_E E$$

Gas energy:

$$\rho\frac{\mathrm{D}}{\mathrm{D}t}\left(\frac{e}{\rho}\right) = -p\nabla.\mathbf{v} - 4\pi\kappa_P B + c\kappa_E E$$

Radiation flux:

$$\frac{\rho}{c^2}\frac{\mathrm{D}}{\mathrm{D}t}\left(\frac{\mathbf{F}}{\rho}\right) = -\nabla\mathbf{P} - \frac{1}{c}\chi_F\mathbf{F}$$

This last one is the problem - how do you solve it?

Example: flux-limited diffusion

$$\frac{\rho}{c^2}\frac{\mathrm{D}}{\mathrm{D}t}\left(\frac{\mathbf{F}}{\rho}\right) = -\nabla\mathbf{P} - \frac{1}{c}\chi_F\mathbf{F}$$

First, assume radiative equilibrium:

$$\nabla\mathbf{P} = -\frac{1}{c}\chi_F\mathbf{F}$$

Now assume radiation field is locally isotropic, so $\mathbf{P} \approx \dfrac{E}{3}$

$$\mathbf{F} = -\frac{c}{3\chi}\nabla E$$

Looks simple, but disaster is near.

What if χ becomes very small (e.g. in regions of low density)?

The radiation flux can then become *arbitrarily large*.

This is bad.

Example: flux-limited diffusion

Possible solutions:

(i) give up and try something else

(ii) only use the algorithm for flows which are guaranteed to be optically thick

(iii) try and come up with a hack which prevents the radiation flux becoming ridiculously large

Option (iii) is surprisingly popular (e.g. Whitehouse & Bate 2006).

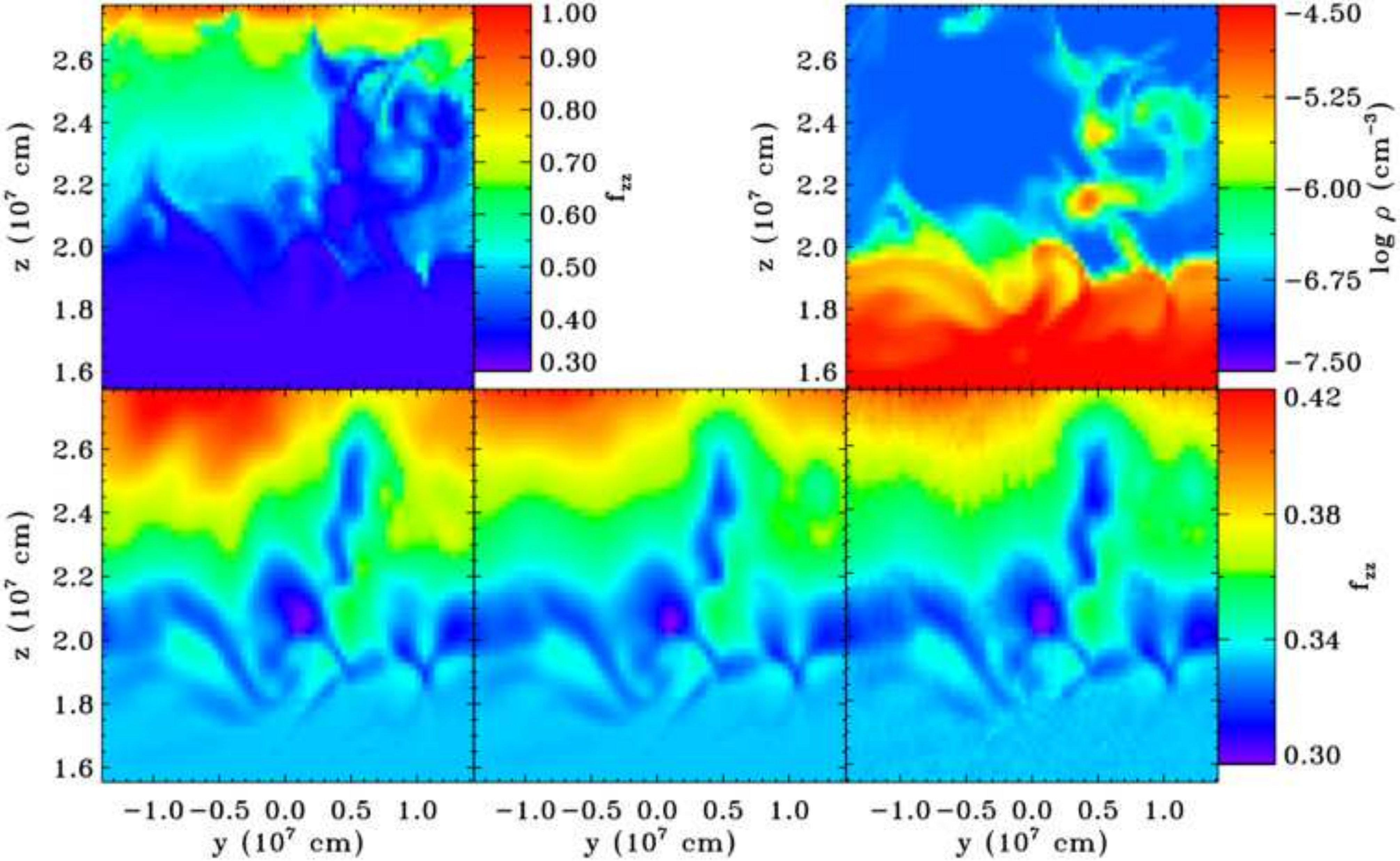
Insert a new parameter: $\mathbf{F} = -\frac{c\lambda(E)}{3\chi}\nabla E$

where $\lambda(E)$ is the flux limiter, a function which ensures that $\mathbf{F}$ is prevented from becoming arbitrarily large.

WARNING

FLD codes have been known to produce quite different results from other algorithms (e.g. Davis+ 2012):

FLD



density field

ray-trace

ray-trace

Monte Carlo

Boundary conditions (I)

Sources of radiation feed energy into the simulation.
The emissivity and the absorption coefficient tell you how good the gas is at moving radiative energy around.

But unless there is some way for radiative energy to leave the simulation domain, the mean gas temperature will rise inexorably.

In a grid code, the simulation can lose energy via the cells on the edge of the domain.

What about an SPH simulation that doesn’t have a boundary?

Boundary conditions (I)

Bate (2009), using an FLD code, force all particles with densities below a threshold to have a fixed gas and radiation temperature, effectively creating an *energy-absorbing halo* around the cloud.

Monte Carlo codes (e.g. TRAPHIC, Pawlik & Schaye 2008) allow photon packets to naturally leave the grid.

Otherwise, one has to do something clever (e.g. construct a *convex hull* around the simulation and treat it like a photosphere.

Other ways of avoiding overheating

Alternatively, one can use an *optically thin cooling curve*.

You use fundamental atomic physics to compute how fast a gas of a given density, temperature and composition loses energy, and assume that energy *magically disappears* - this gives you a volumetric or specific cooling rate.

This allows any location of the simulation to cool (as opposed to only allowing cooling across boundaries).

Simplifying the radiation transport equation

$$\frac{1}{c}\frac{\partial I_\nu}{\partial t} + \mathbf{n}.\nabla I_\nu = \epsilon_\nu - \kappa_\nu \rho I_\nu$$

There are ways of simplifying the RT equation:

(ii) Ignore emissivity of the gas (assume the radiation field is due to a few bright sources). This improves the scaling from $N^{7/3}$ to $N_{\rm source} N^{4/3}$

$$\mathbf{n}.\nabla I_\nu = -\kappa_\nu \rho I_\nu$$

Example: photoionisation

A classic example of this idea is the use of the 'On The Spot' (OTS) approximation to solve the problem of following *photoionising radiation from O-stars*.

Suppose you have a source of Q_{H} ionising photons in a uniform gas. The photons ionise the gas, but ionised gas recombines - reionising this gas uses up photons, until eventually, *all* the photons are used up by recombinations and no more gas can be ionised.

If the neutral gas has number density *n*, ionised gas is very nearly completely ionised, and charge is conserved:

$$n = n_{\mathrm{i}} = n_{\mathrm{e}}$$

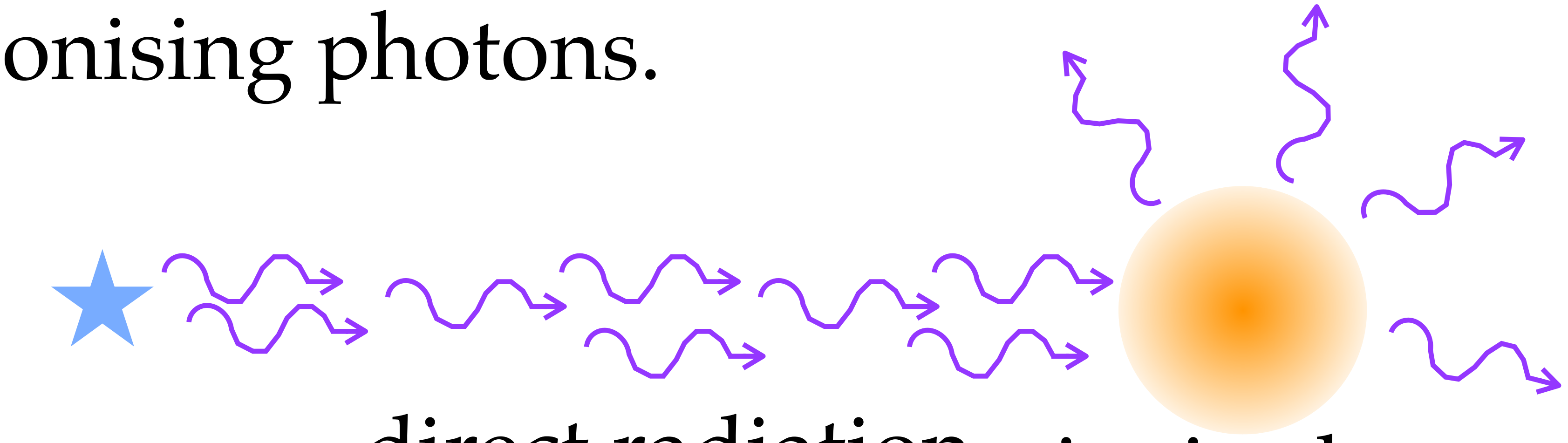
Example: photoionisation

$$n = n_{\mathrm{i}} = n_{\mathrm{e}}$$

Rate of recombinations (which acts like an opacity) in a unit volume of gas depends on ion-electron collision rate

$$\mathcal{R} = \alpha n_{\mathrm{i}} n_{\mathrm{e}} = \alpha n_{\mathrm{i}}^2 = \alpha n^2$$

But: when an ion and an electron recombine, a photon is released. If they recombine directly to the ground state, the photon is an *ionising photon*. So the ionised gas is a source of ionising photons.



re-emitted radiation field

O-star

direct radiation field

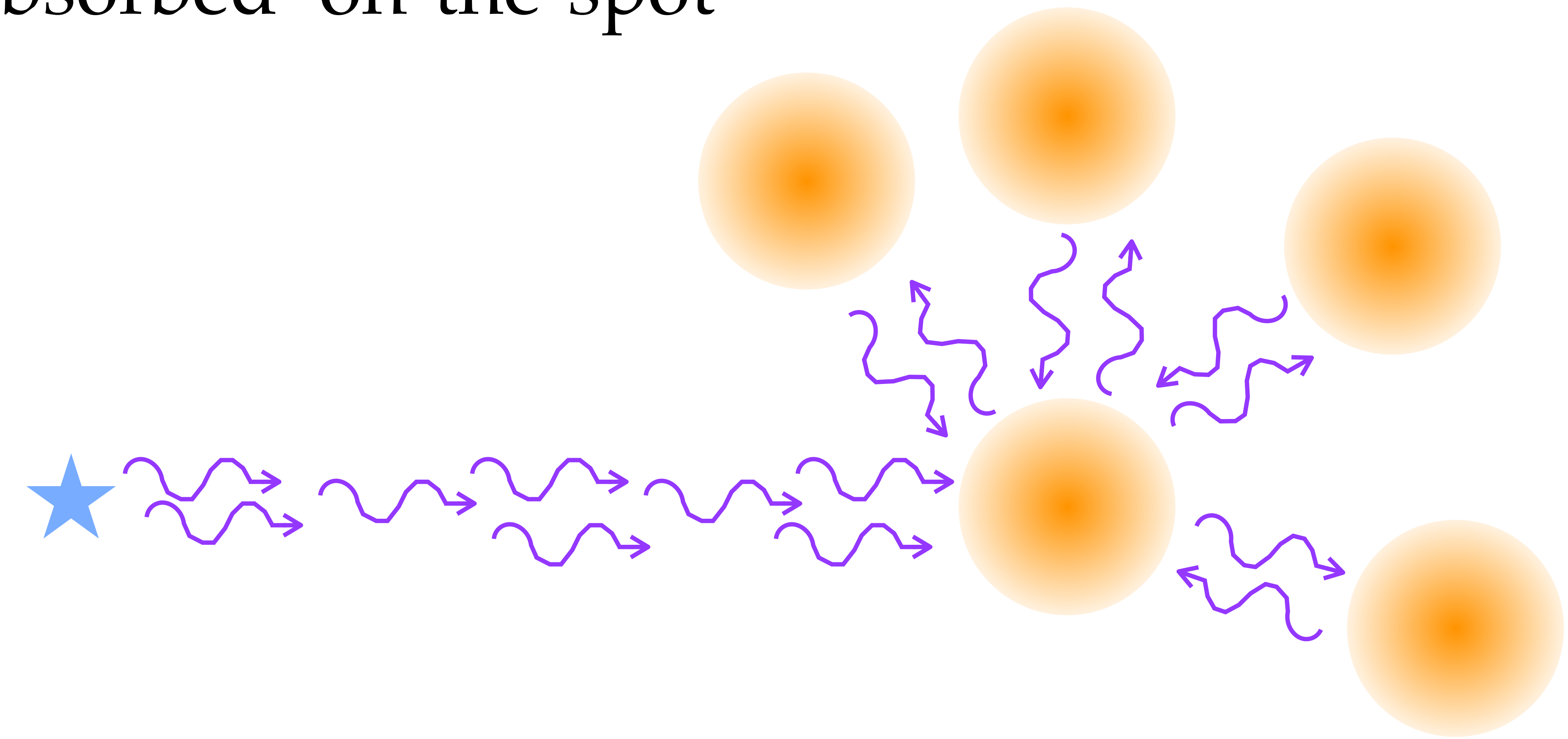
ionised gas particle

Example: photoionisation

How do we deal with this?

Most ionised gas particles will be surrounded by other ionised gas particles.

One could assume that for every ionising photon re-emitted by a particle, it absorbs an ionising photon re-emitted by a neighbouring particle, so that all re-emitted photons are absorbed 'on-the-spot'



Example: photoionisation

This means that re-emitted photons from neighbouring particles cancel each other out, and *can be ignored*. This is equivalent to saying that the *emissivity* of the gas is zero.

Since we are now claiming that recombinations to the ground state are unimportant, we can just define a new recombination coefficient which leaves them out.

$$\mathcal{R}' = \alpha_{\mathrm{B}} n^2$$

To find out how much gas is ionised, we just have to integrate over a large enough volume that the total recombination rate equals the ionising photon flux

Example: photoionisation

$$Q_{\mathrm{H}} = \int_{r'=0}^{r'=R_{\mathrm{IF}}} 4\pi r'^2 n^2 \alpha_{\mathrm{B}} \mathrm{d}r'.$$

Leading to the definition of the Stromgren radius

$$R_{\mathrm{s}} = \left(\frac{3Q_{\mathrm{H}}}{4\pi\alpha_{\mathrm{B}} n_0^2} \right)^{\frac{1}{3}} .$$

But what if the gas isn't uniform?

More on that later….

Example: photoionisation

What about heating and cooling?

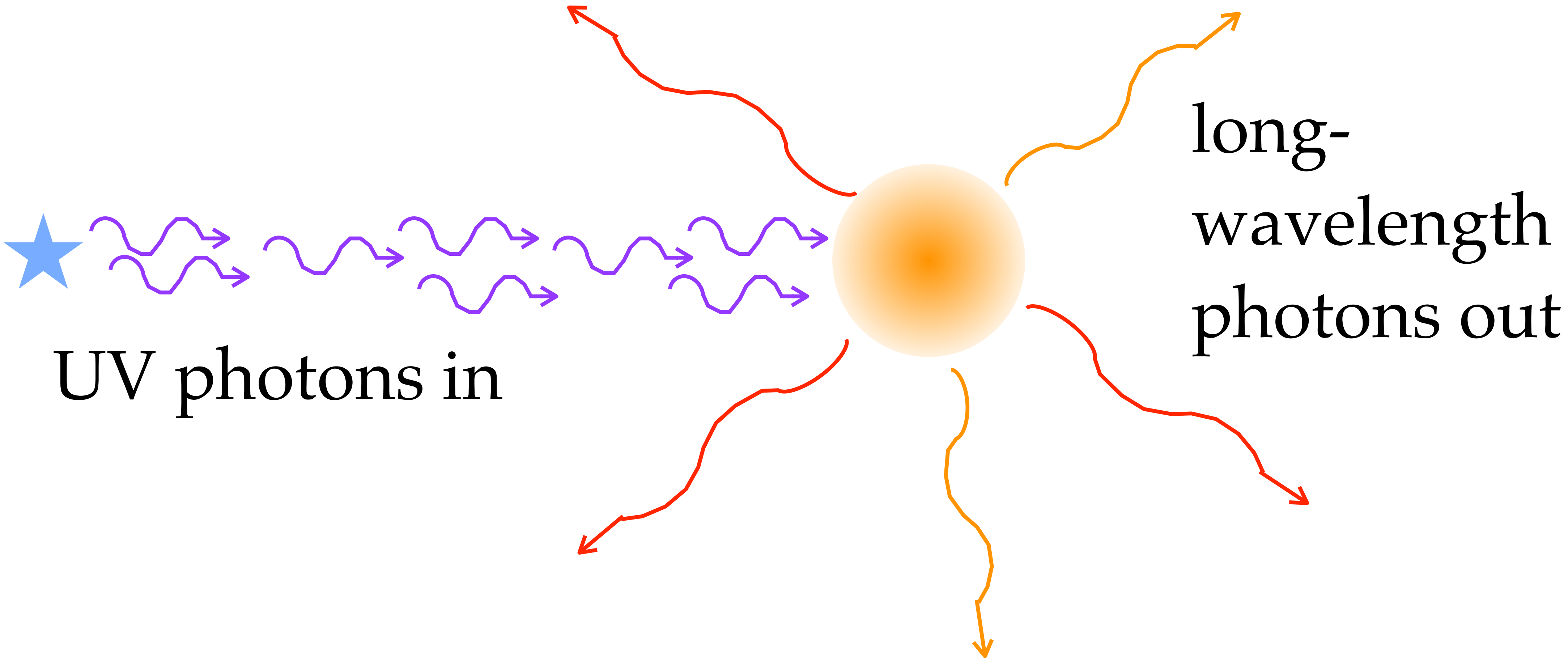
We know which gas particles are absorbing ionising photons, so we know what the heating rate is for each particle.
But if we just keep feeding in heat, the temperature of the ionised gas will keep rising.

This is similar to the problem with boundary conditions mentioned earlier.

How do we solve this?

Example: photoionisation

We cheat.



It turns out that, *at solar metallicity*, the heating and cooling processes equilibrate at ~10 000K, *if we can assume that all the long-wavelength photons escape*.

(At *low metallicity*, stars are hotter, the stellar radiation field is hotter, and cooling is less efficient - ionised gas also hotter.)

Simplifying the radiation transport equation

$$\frac{1}{c}\frac{\partial I_\nu}{\partial t} + \mathbf{n}.\nabla I_\nu = \epsilon_\nu - \kappa_\nu \rho I_\nu$$

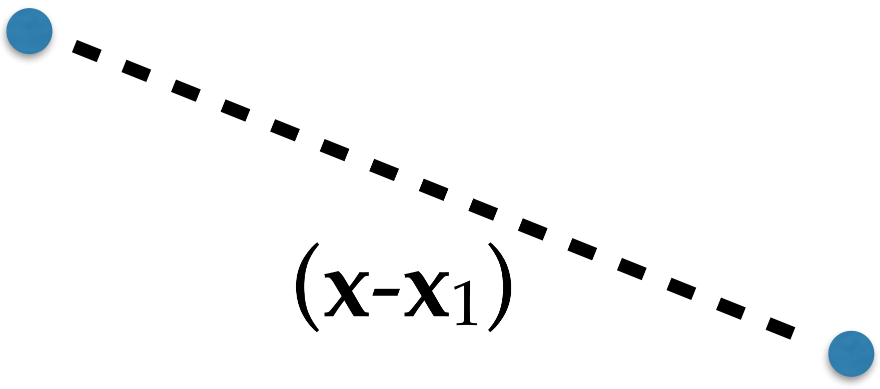
There are ways of simplifying the RT equation:

(iii) Treat the radiation field like the gravitational field - ignore emissivity and absorption so that there is *no optical depth* and the radiation field is just *geometrically diluted* (usually also that the speed of light is reduced)

$$\frac{1}{c}\frac{\partial I_\nu}{\partial t} + \mathbf{n}.\nabla I_\nu = 0$$

Example: Optically-Thin Eddington Tensors

(e.g Gnedin & Abel 2001)



x$_1$ (arbitrary point)

(**x**-**x**$_1$)

x (place where you want to calculate the radiation field)

luminosity at location **x**$_1$

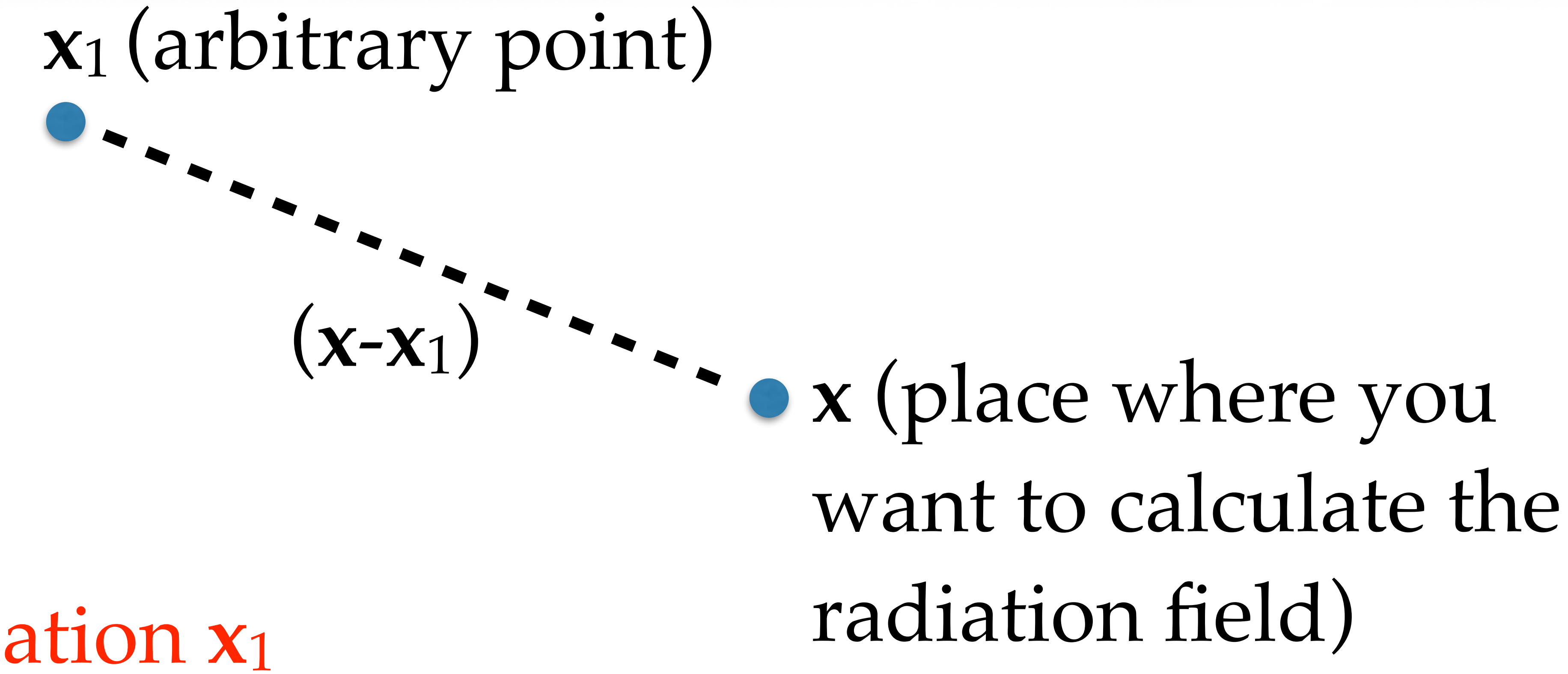
optical depth along (**x**-**x**$_1$)

$$\mathbf{P}_{ij} = \int_V \mathrm{d}^3 x_1 \frac{L(\mathbf{x}_1)\mathrm{e}^{-\tau(\mathbf{x},\mathbf{x}_1)}}{|\mathbf{x}-\mathbf{x}_1|^2} \underbrace{\frac{\mathbf{x}^i-\mathbf{x}_1^i}{|\mathbf{x}-\mathbf{x}_1|} \frac{\mathbf{x}^j-\mathbf{x}_1^j}{|\mathbf{x}-\mathbf{x}_1|}}_{\text{tensor encoding directionality of radiation}}$$

geometric dilution

Example: Optically-Thin Eddington Tensors

(e.g Gnedin & Abel 2001)



$\mathbf{x}_1$ (arbitrary point)

($\mathbf{x}$-$\mathbf{x}_1$)

$\mathbf{x}$ (place where you want to calculate the radiation field)

luminosity at location $\mathbf{x}_1$

$$\mathbf{P}_{ij} = \int_V \mathrm{d}^3 x_1 \frac{L(\mathbf{x}_1)}{|\mathbf{x}-\mathbf{x}_1|^2} \underbrace{\frac{\mathbf{x}^i - \mathbf{x}_1^i}{|\mathbf{x}-\mathbf{x}_1|} \frac{\mathbf{x}^j - \mathbf{x}_1^j}{|\mathbf{x}-\mathbf{x}_1|}}_{\text{tensor encoding directionality of radiation}}$$

geometric dilution

Ray-tracing in SPH

In general, assuming clouds or discs are *so optically thin that only geometric dilution is important, or so optically thick that you just have to solve a diffusion problem* is very brave.

Most of the time, you will be in the intermediate regime and you are going to need to do *ray-tracing* of some kind.

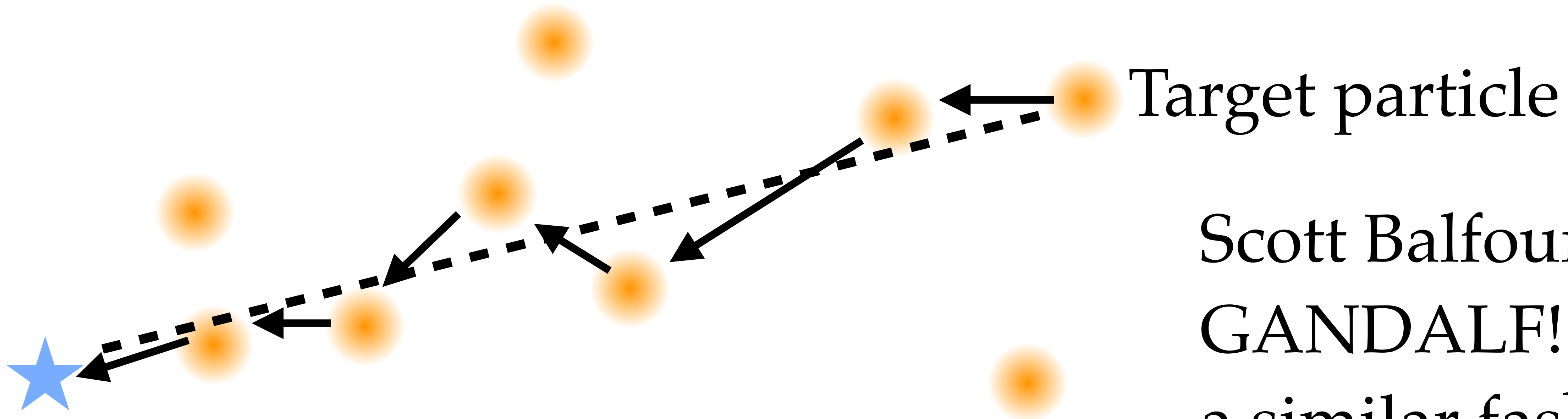
Ray-tracing methods are accurate and can in principle be made faster than Monte Carlo methods

Two problems to be solved:
(i) where to draw rays
(ii) how to compute the required hydrodynamic quantities along rays

Where to draw rays

One needs to strike a balance between speed (draw as few rays as possible) and ensuring that the domain is well sampled (draw lots of rays).

Some authors modelling photoionisation (e.g Dale+ 2007, 2012) draw rays from *all* particles to radiation sources, but grow HII regions outwards from the sources - most ray-traces can be terminated when they reach a particle to which the optical depth has already been computed.

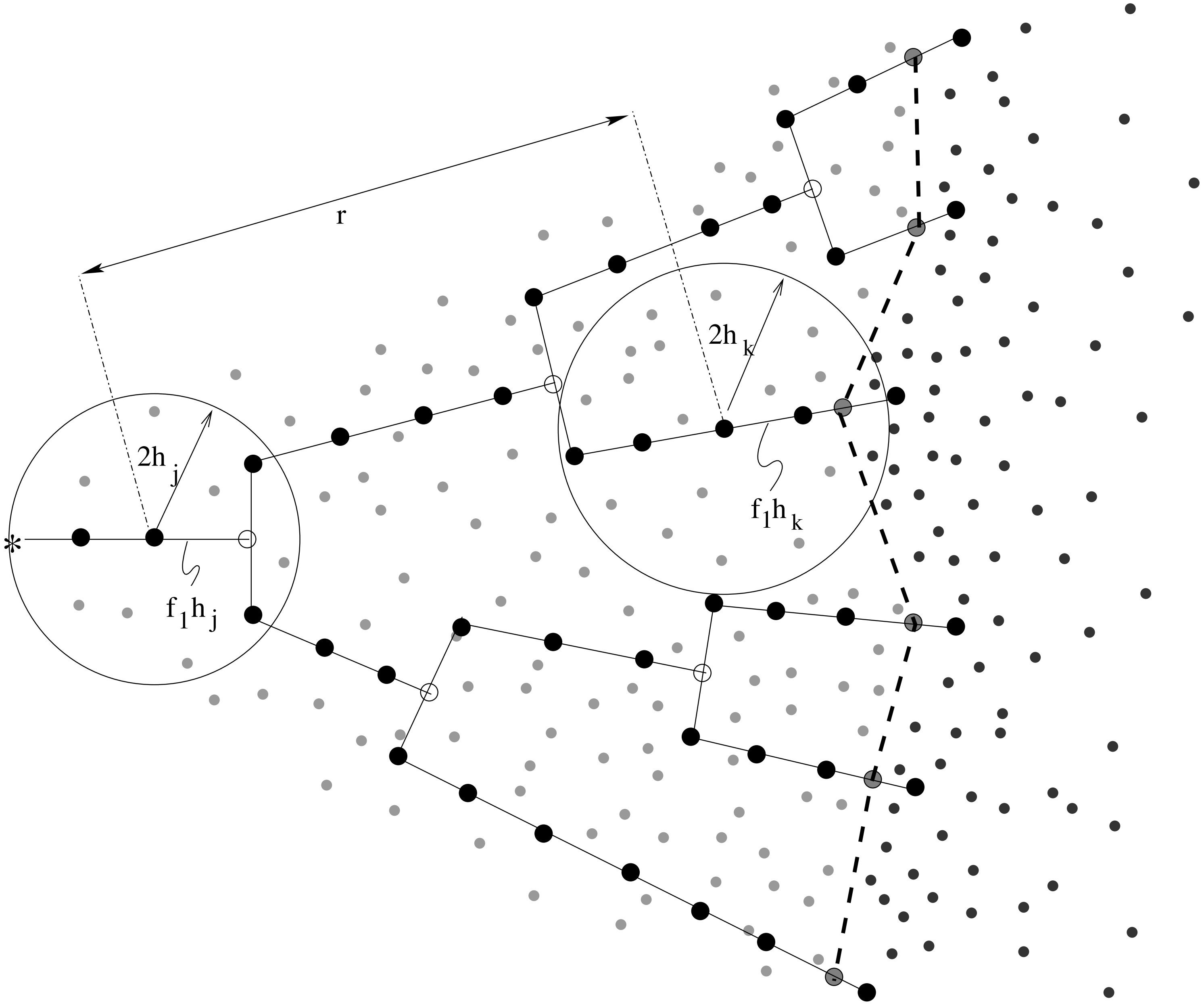


Scott Balfour's code (in GANDALF!!) works in a similar fashion

Where to draw rays

Other authors (e.g. Gritschneder+ 2009, Bisbas+ 2009) use a form of adaptive ray-splitting based on the local number density of SPH particles.

Bisbas+ 2009 use the HealPix method for exactly tessellating the surface of a sphere to *recursively split rays* into 4 child rays as many times as needed.



Evaluating quantities along rays

Once rays have been drawn, integrals on the rays need to be be performed to construct quantities resembling optical depths.

To do this, one has to locate the particles on/near the ray.

This is tricky, because SPH particle distributions are unstructured.

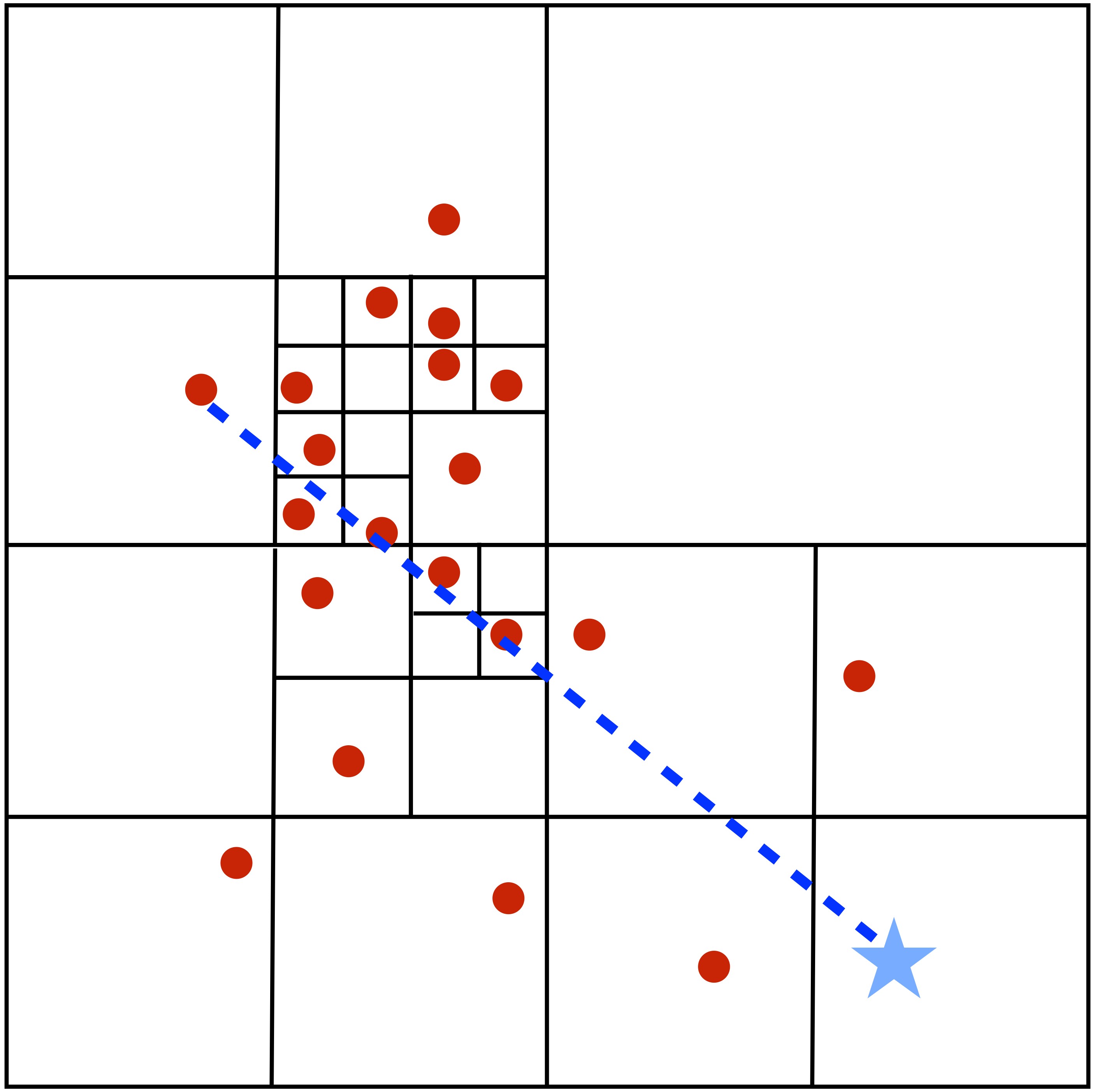
Most authors use the neighbour lists. Finding neighbours is fast, but this produces a lot of sample points along each ray, and has problems with crossing empty regions.

Evaluating quantities along rays

A more intelligent solution would be to use the SPH gravity tree - a data structure which does store the particles in a fashion that relates to their real geometrical distribution.

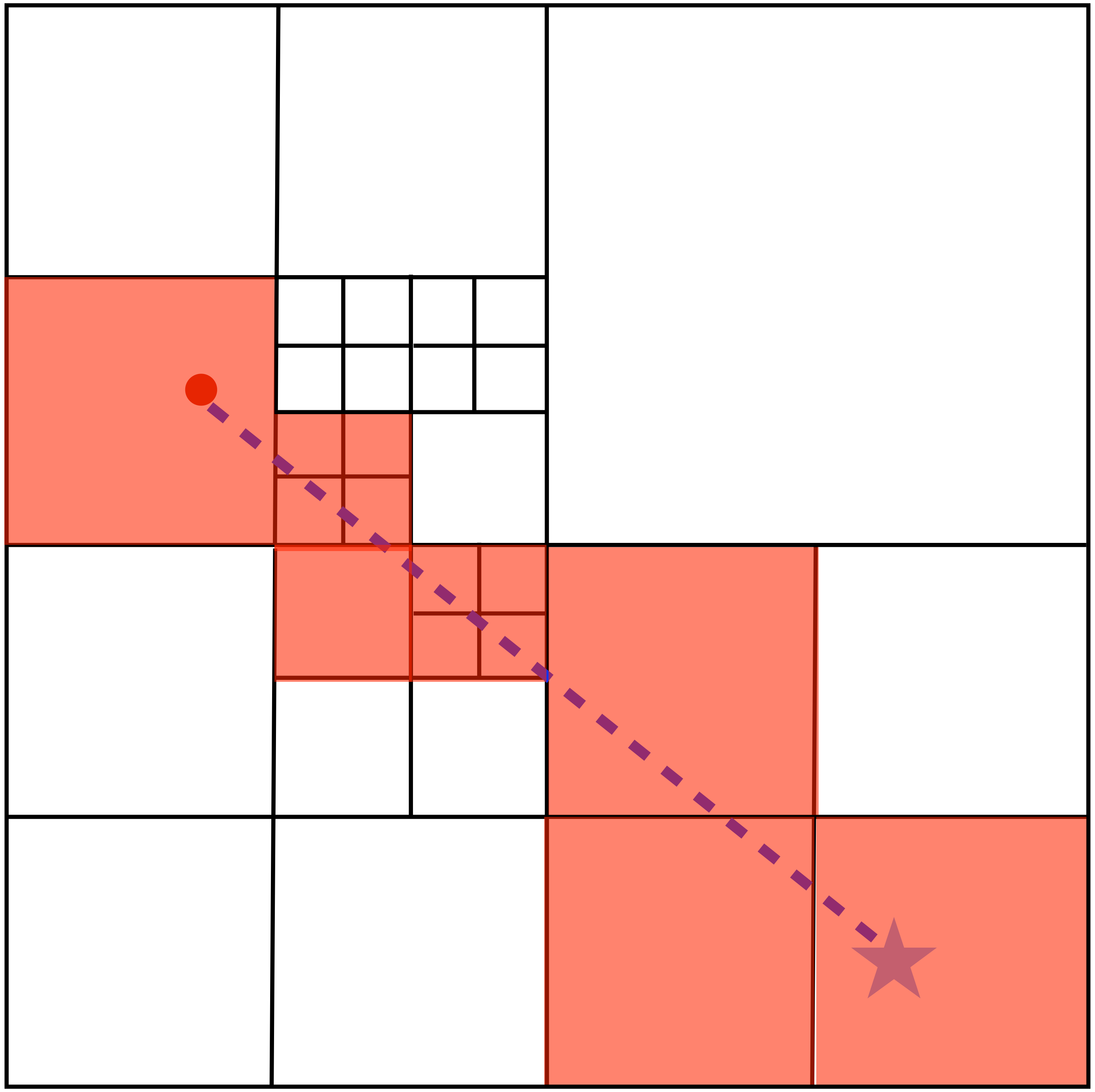
Every cell contains one or zero particles, has one larger parent and (possibly) four (in 2D) children.

If no suitable particles are found in a cell, the parent can be searched, and its parent, and so on.



Evaluating quantities along rays

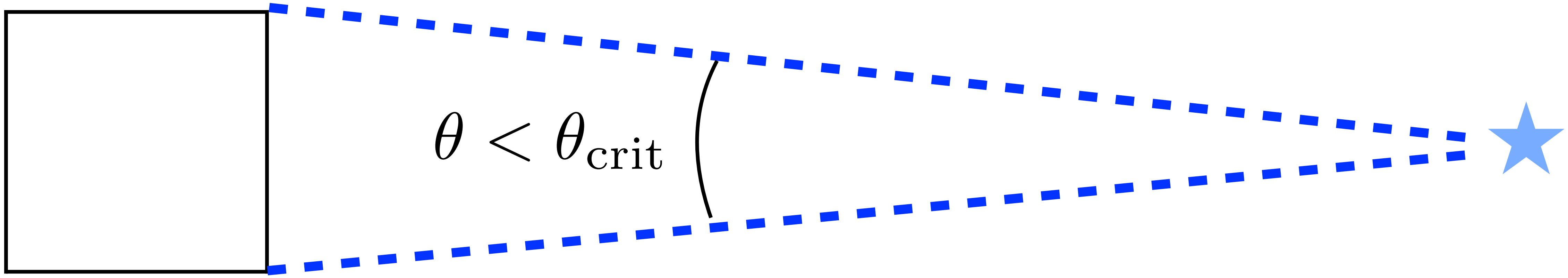
Instead of using the tree to help find particles along rays, one can use the tree nodes directly - they all have well-defined masses and radii.



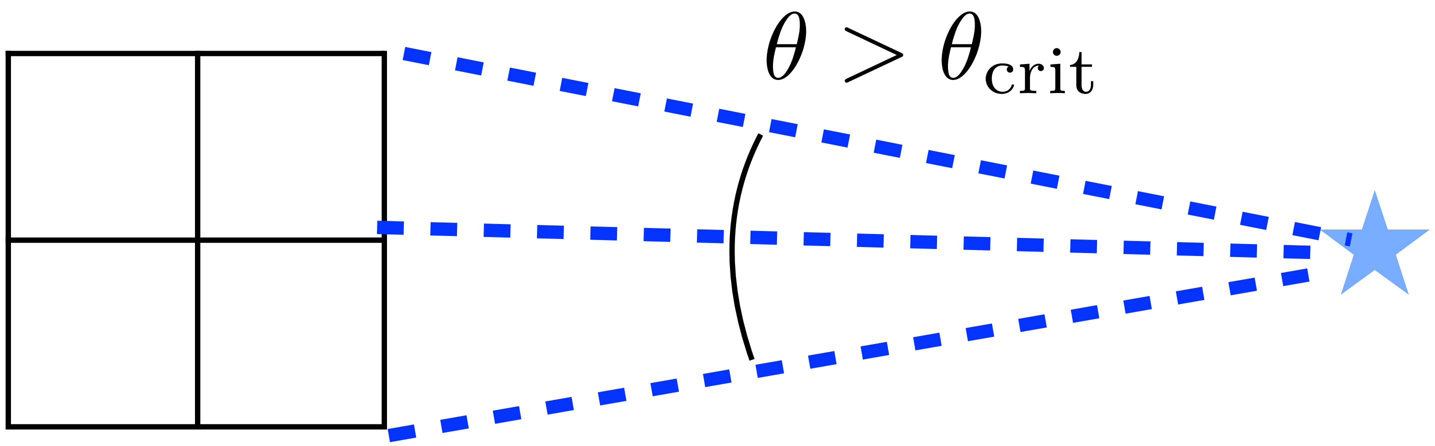
One can also use the hierarchical nature of the tree to save time - it may not be necessary to find *every particle* or every *lowest-level tree node* along a ray.

Evaluating quantities along rays

To decide whether to open a tree node and look at its daughters, you can use an angular criterion:



Open the node, and daughters…, until the angular criterion is satisfied.

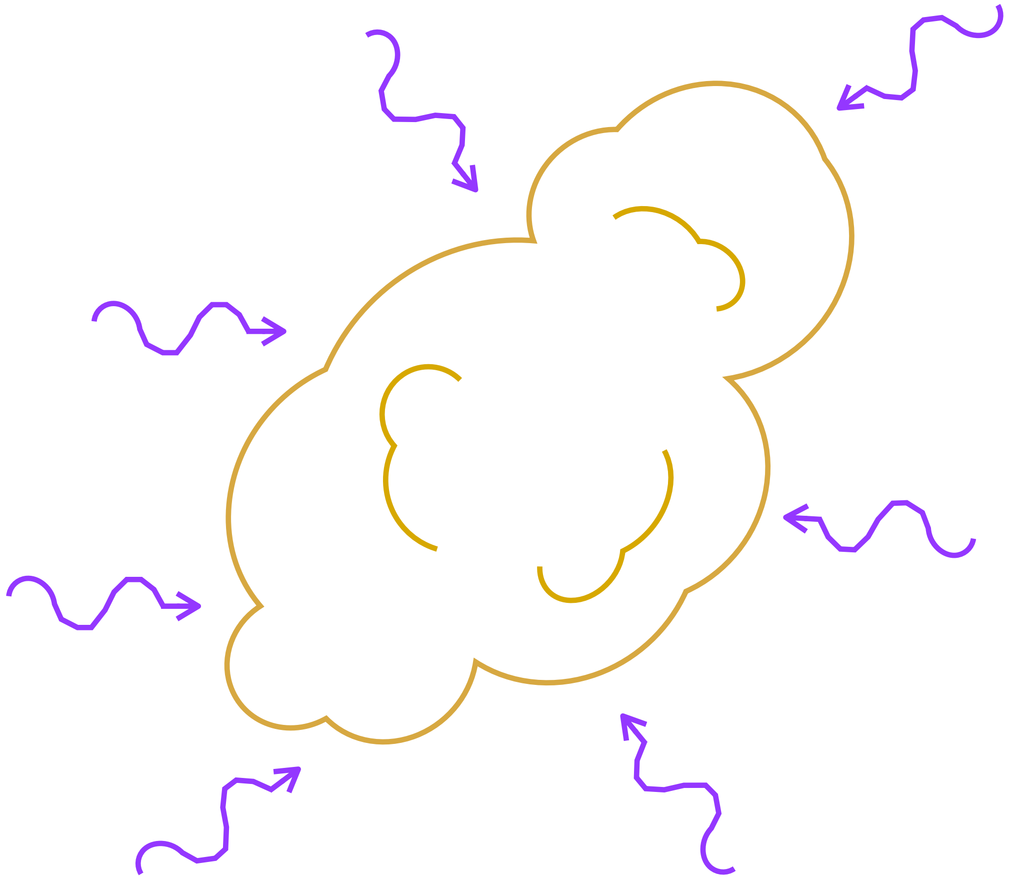


Boundary conditions (II)

That, roughly speaking, is how two new(ish) radiation transport schemes work - TREECOL (Clarke & Glover 2012) and TREERAY (Wünsch+, 20??)

I already mentioned *outflow boundaries* as an essential way of preventing a simulation overheating.

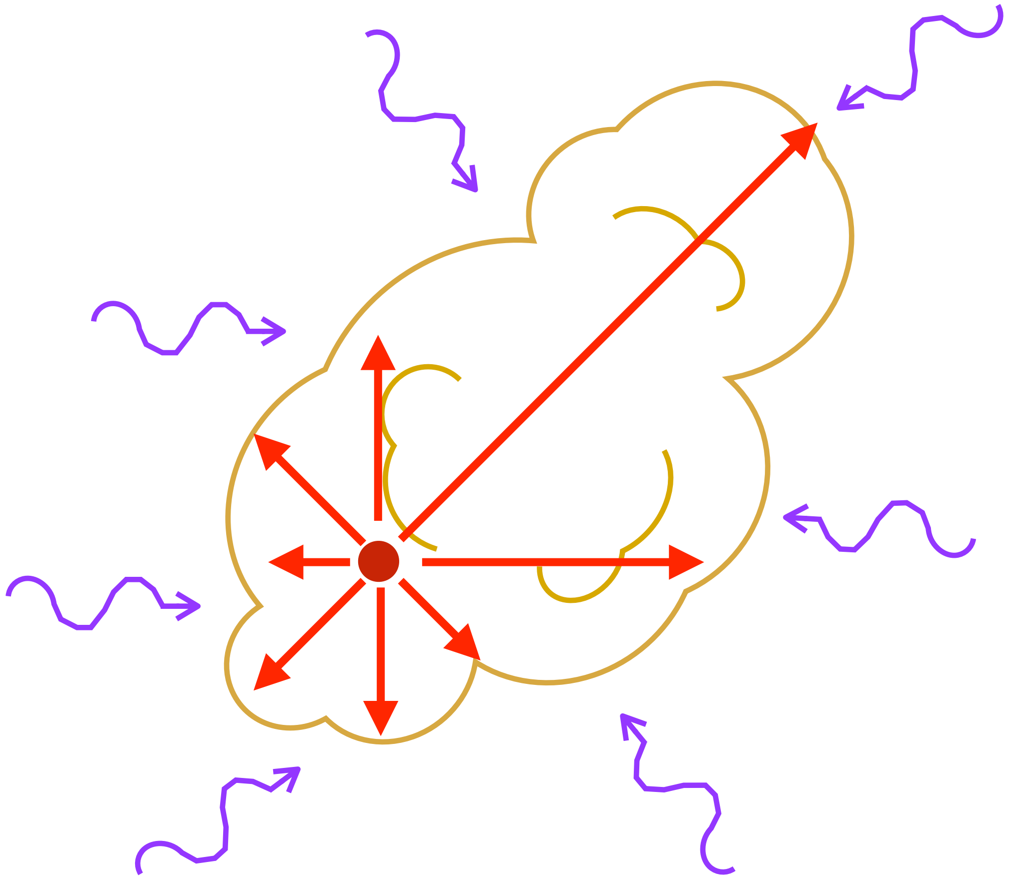
What if instead you want *inflow boundaries,* where the whole simulation domain is bathed in an external radiation field?



Boundary conditions (II)

This the problem that TREECOL was designed to solve. From any point in the simulation domain, it calculates in 48 (or 192) directions, the column-density to a given kind of radiation between the point and the edge of the cloud.

One can then model the effect of an external radiation bath on a cloud, such as the interstellar UV field in which all molecular clouds sit.



TREERAY

(Soon in GANDALF!!)

TREERAY uses this idea to compute the *internal radiation field*. From every point in the code, HealPix wedges are projected. The contributions to the radiation field entering each wedge from tree nodes inside the wedge are computed, using the opening angle criterion to decide which nodes to open.

Contributions from point sources (e.g. O-stars) are also added.

The total amount of radiation reaching each point from all (usually 48 but can be 192,….) HealPix wedges is then computed, giving a model of the whole radiation field.

Summary

(i) Radiative transfer is a hard problem

(ii) There are lots of different ways of approaching it

(iii) They all have various advantages and drawbacks

(iv) As always, there is a tradeoff between speed and accuracy

(v) One has to think hard about which method is most appropriate for the *type of radiation source* and the *type of radiation field* you are interested in

(vi) One might be able to make use of parts of the code written for completely different purposes (e.g the gravity tree)

(vii) There are a few radiation transport schemes in GANDALF - Stamatellos & Whitworth 2007 polytropic-cooling, Balfour+ 20?? ray-tracing, Wünsch+, 20?? TREERAY (nearly)