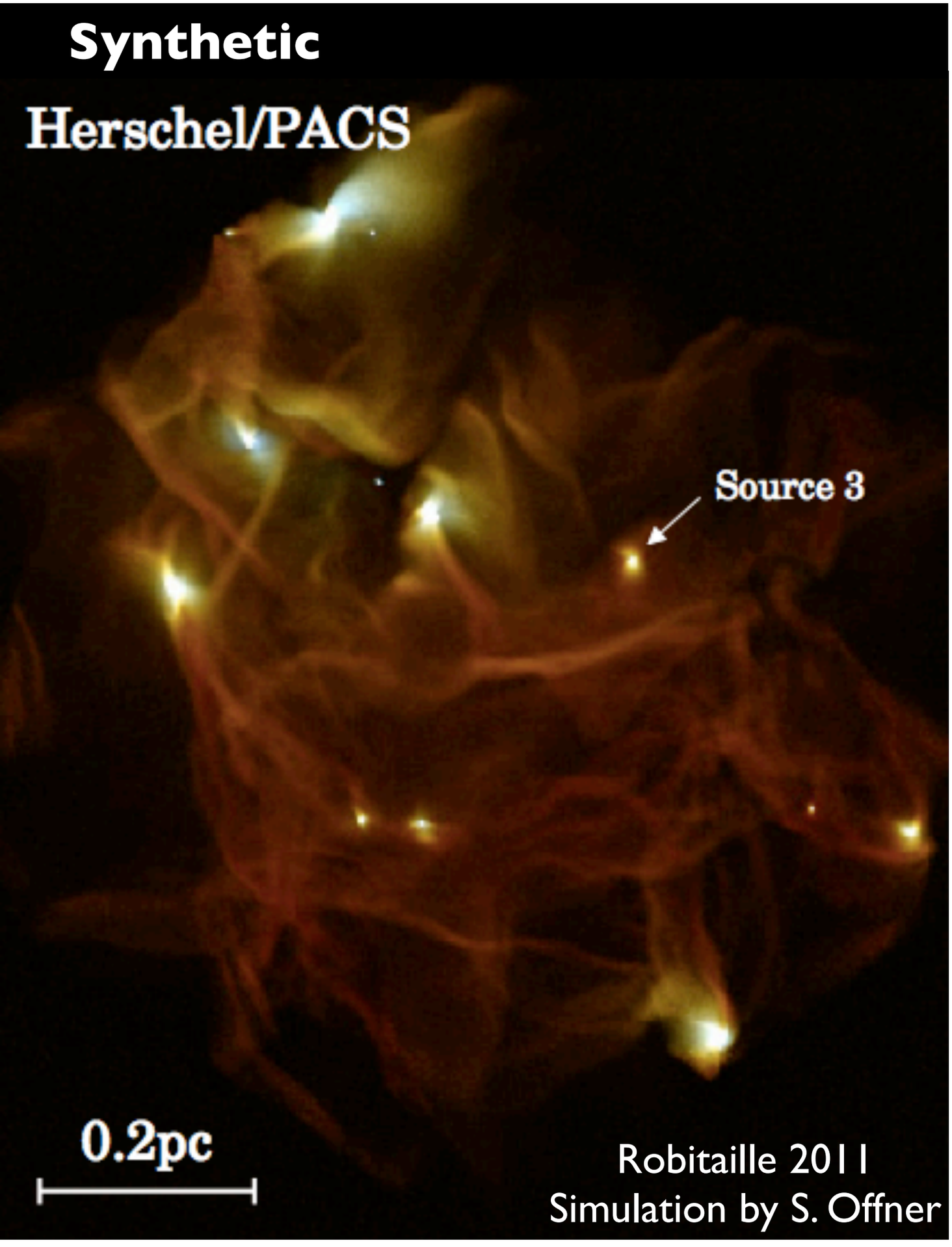


Dust Continuum Modeling: Monte Carlo Radiative Transfer with Hyperion

Stella Offner
HiPACC
Aug 6, 2013

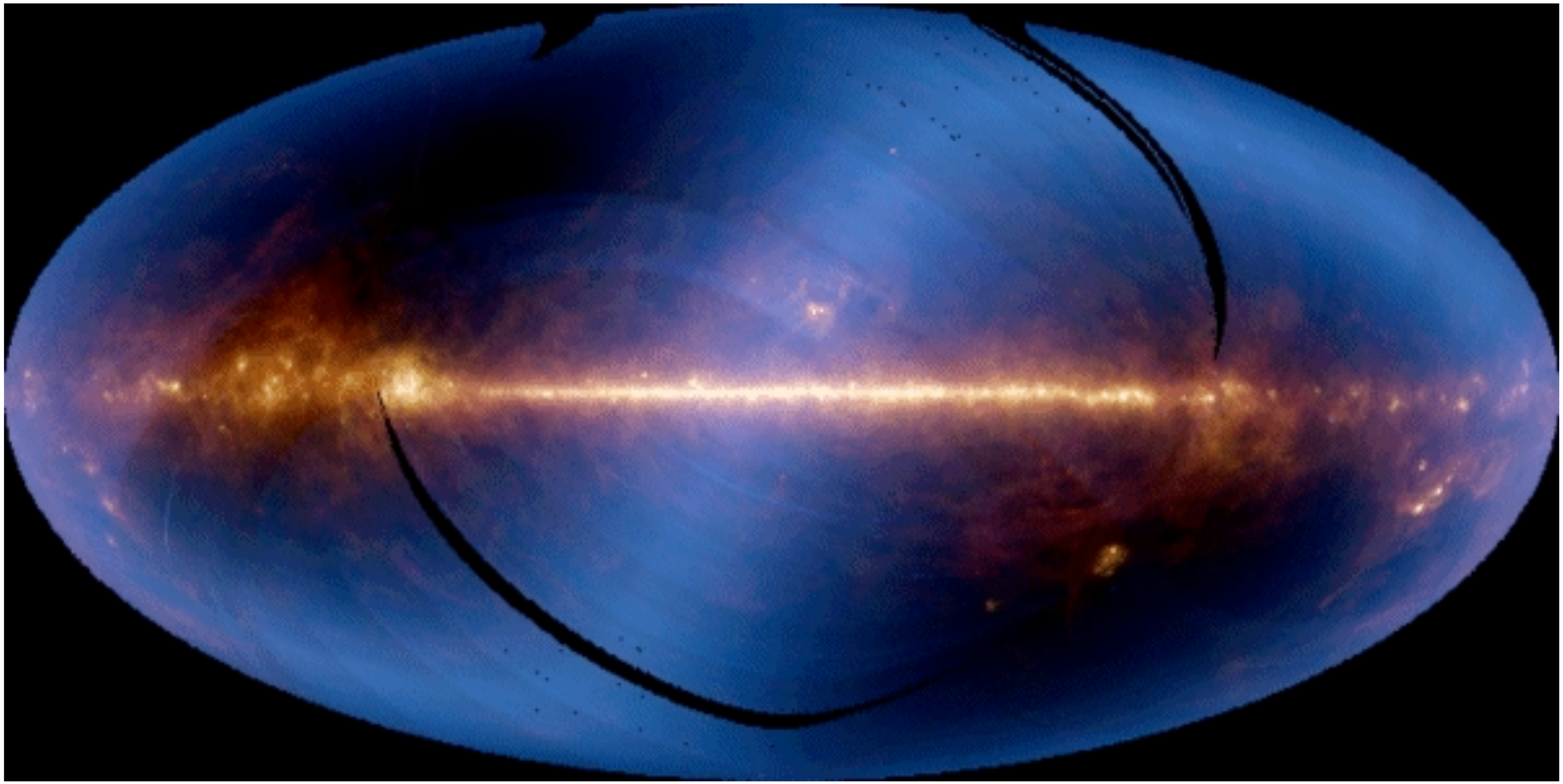
Outline

- Why model dust?
- The physics of dust: emission, absorption & scattering
- Monte Carlo methods
- Hyperion
- Project

Outline

- Why model dust?
- The physics of dust: emission, absorption & scattering
- Monte Carlo methods
- Hyperion
- Project

Why model dust?

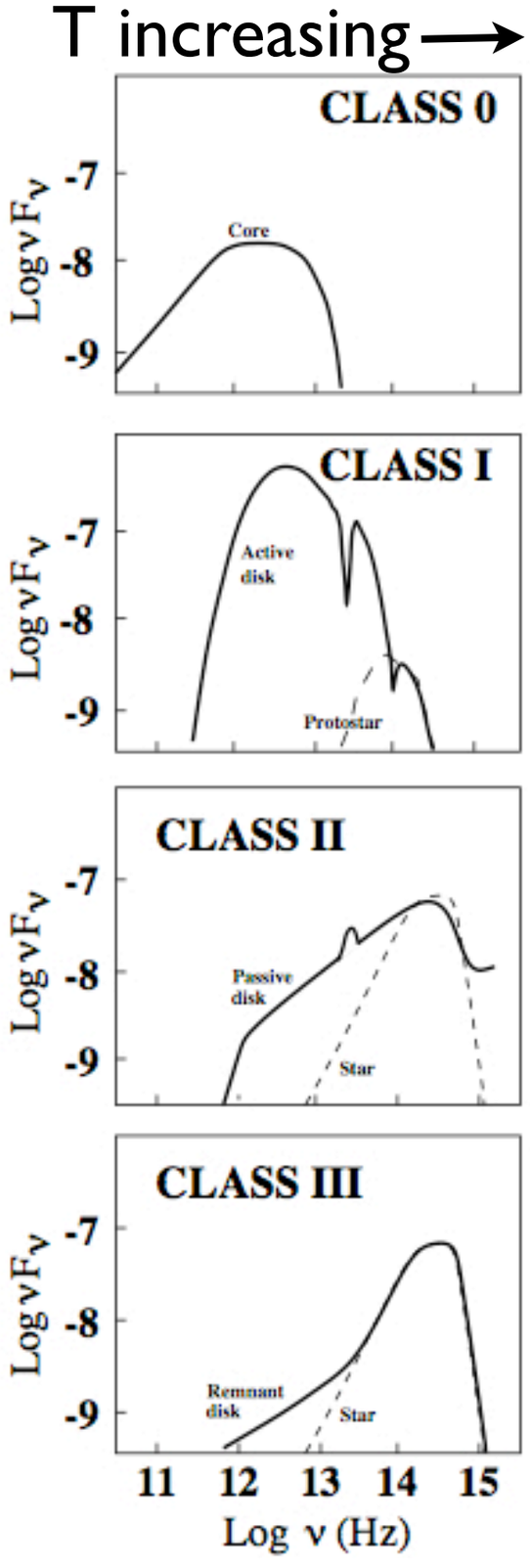


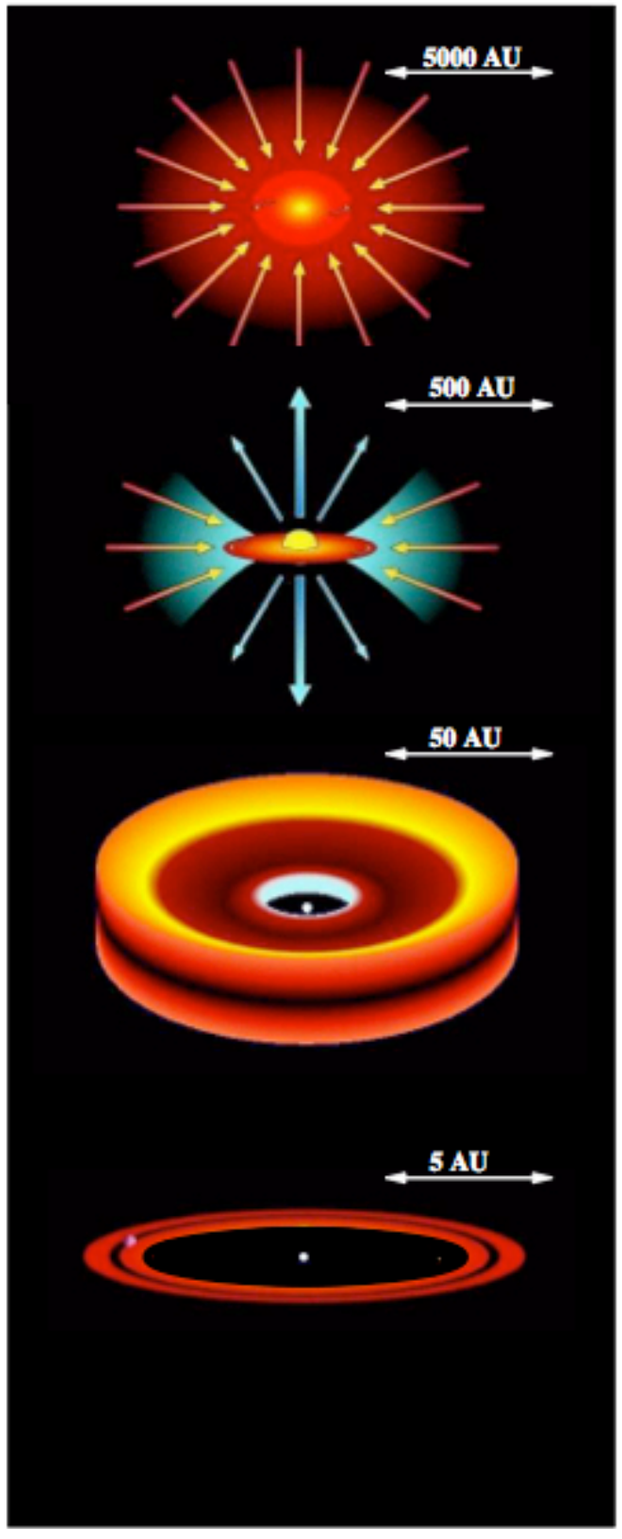
IRAS All Sky Survey

Why model dust?

- Radiation of forming stars is reprocessed by interstellar dust grains
- Reprocessing = extinction (=scattering+absorption), polarization
- Dust to gas ratio:

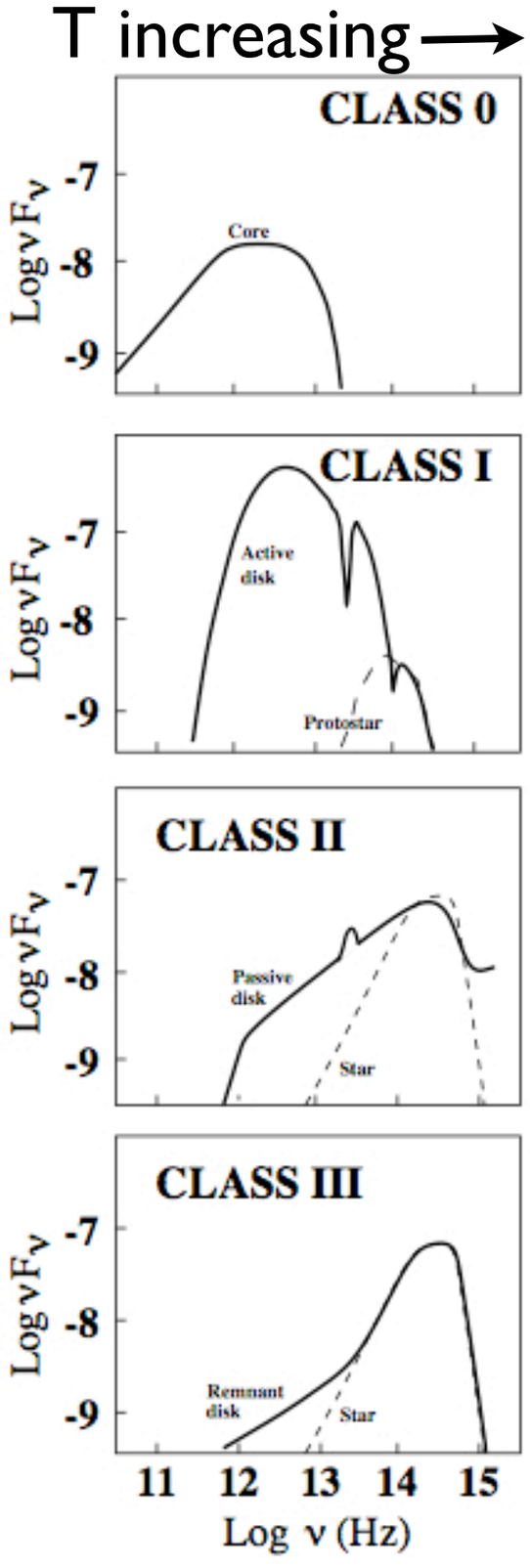
$$\langle \rho_{dust} / \rho_{gas} \rangle \approx 0.01$$

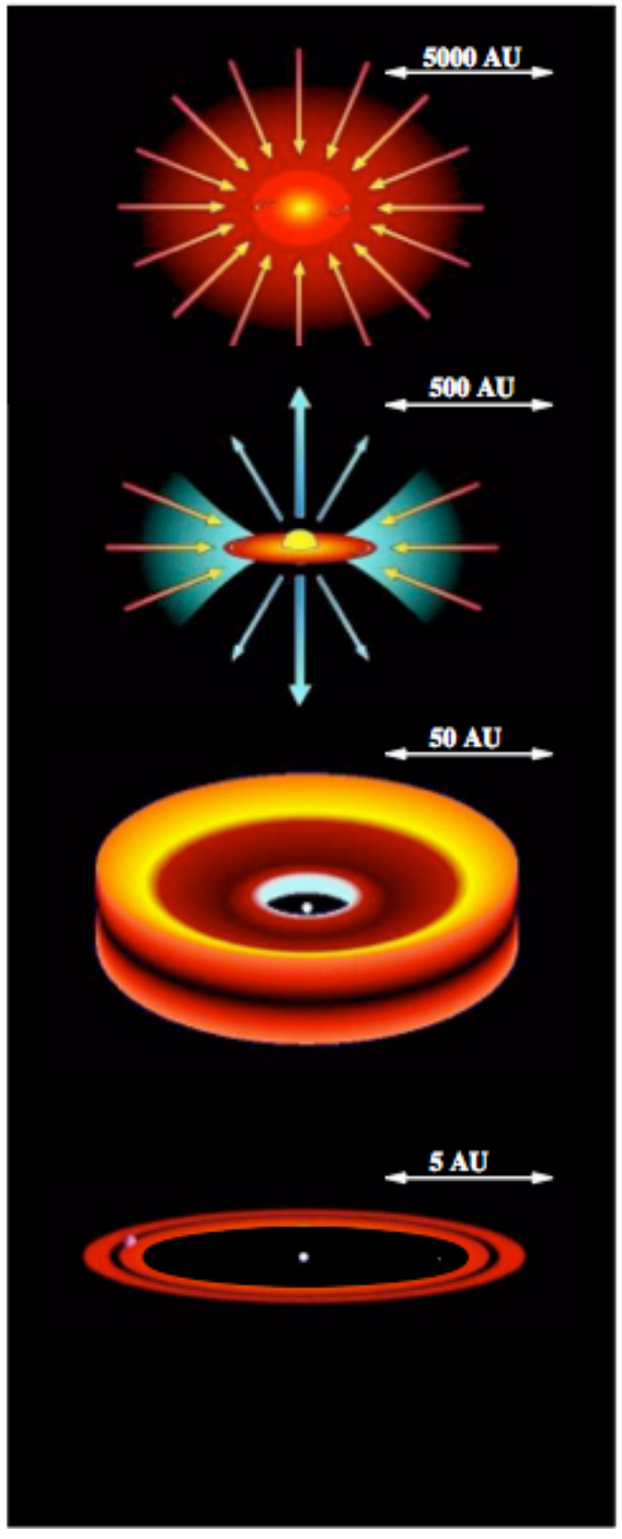




Why model dust?

- Perform modeling of radiation-dust interaction to reconstruct:
  - dust properties
  - dust/gas distribution
  - source properties
  - temperature distribution (esp. useful if dust+gas are thermally coupled, $n>10^4$ cm^{-3})



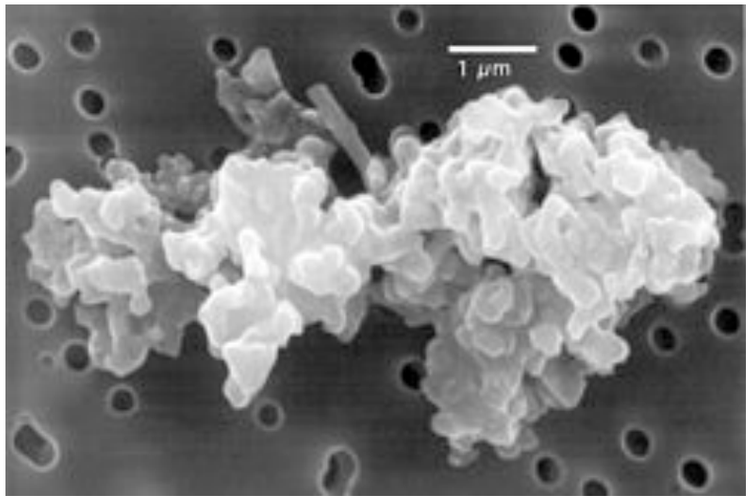


Outline

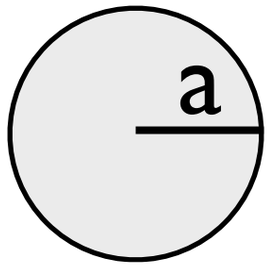
- Why model dust?
- The physics of dust: emission, absorption & scattering
- Monte Carlo methods
- Hyperion
- Project

Dust Basics

Dust grain



"Dust grain"



Extinction Cross Section:

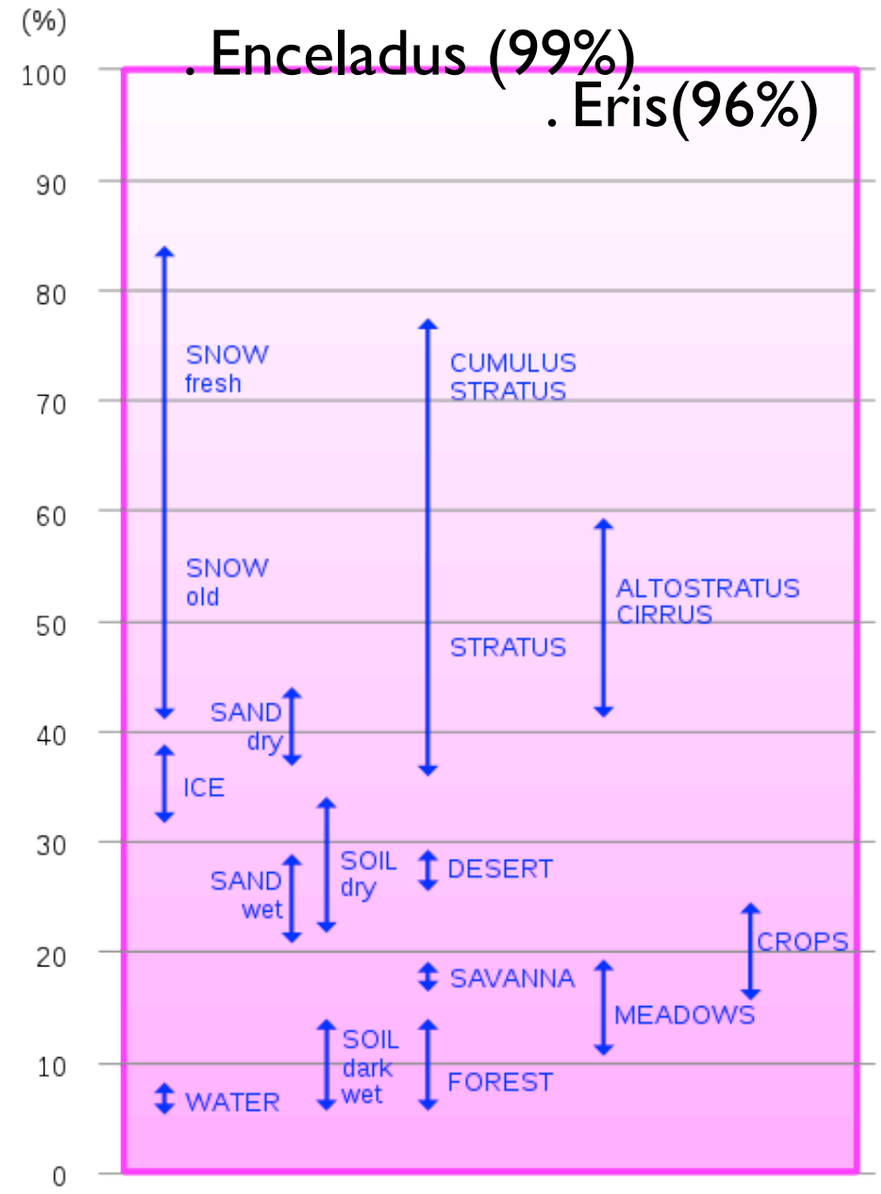
$$\sigma(\lambda) = \pi a^2 Q_{ext}(\lambda)$$

Amount of scattering:

$$\text{albedo} = \frac{\sigma_{sca}}{\sigma_{ext}} = \frac{Q_{sca}}{Q_{ext}} \leq 1$$

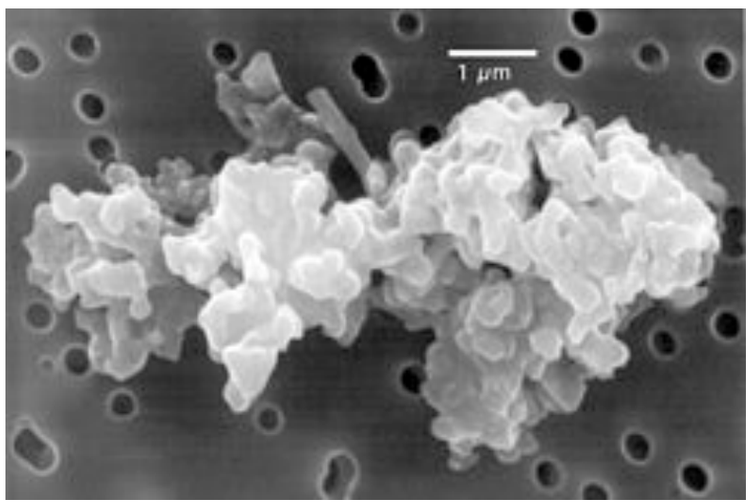
Absorption Opacity:

$$\kappa_{abs} = \pi a^2 Q_{abs} / m_{dust}$$

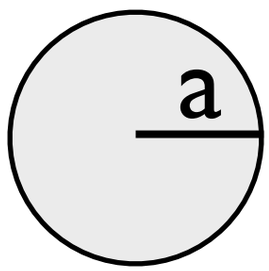


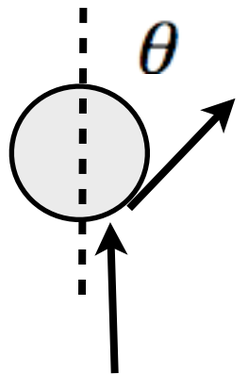
Dust Basics

Dust grain



"Dust grain"





Scattering Phase Function:

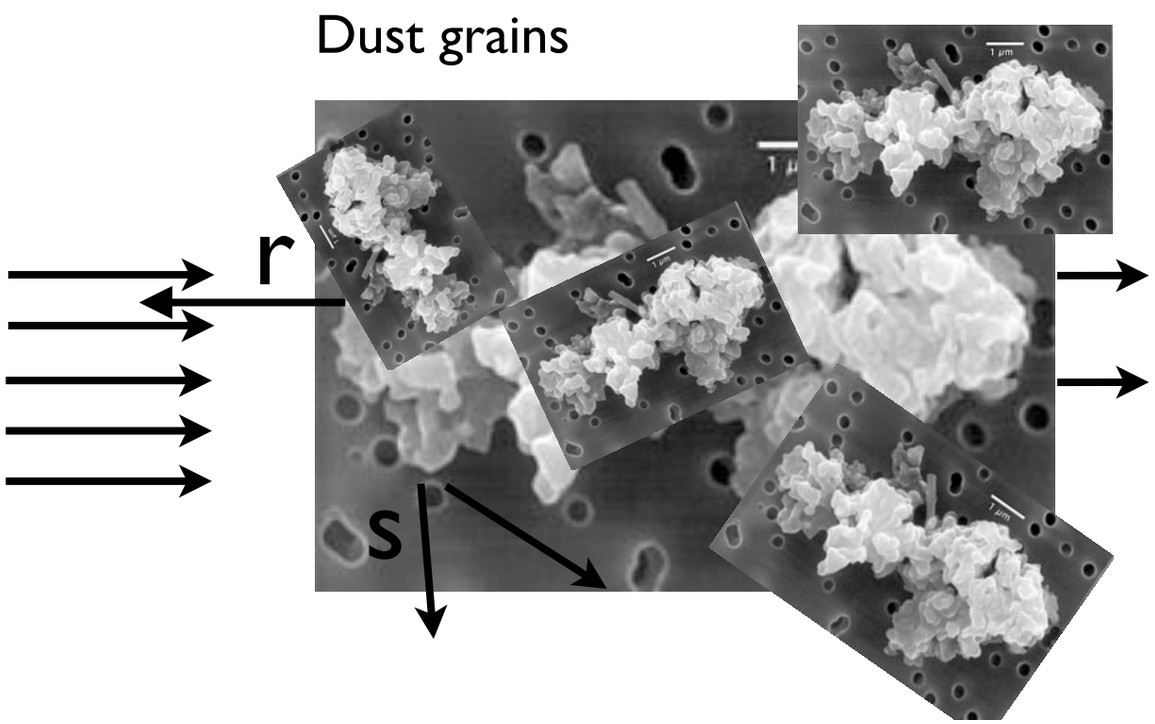
$$g = \langle \cos\theta \rangle = \frac{\int_0^\pi I(\theta)\cos\theta \, d\Omega}{\int_0^\pi I(\theta) \, d\Omega}$$

g = 1 Forward (diffraction)
g = 0 Isotropic
g = -1 Backscattered (reflection)

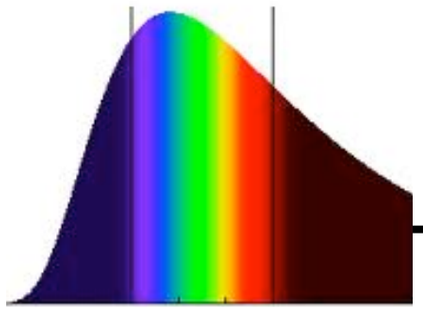
Dust Basics

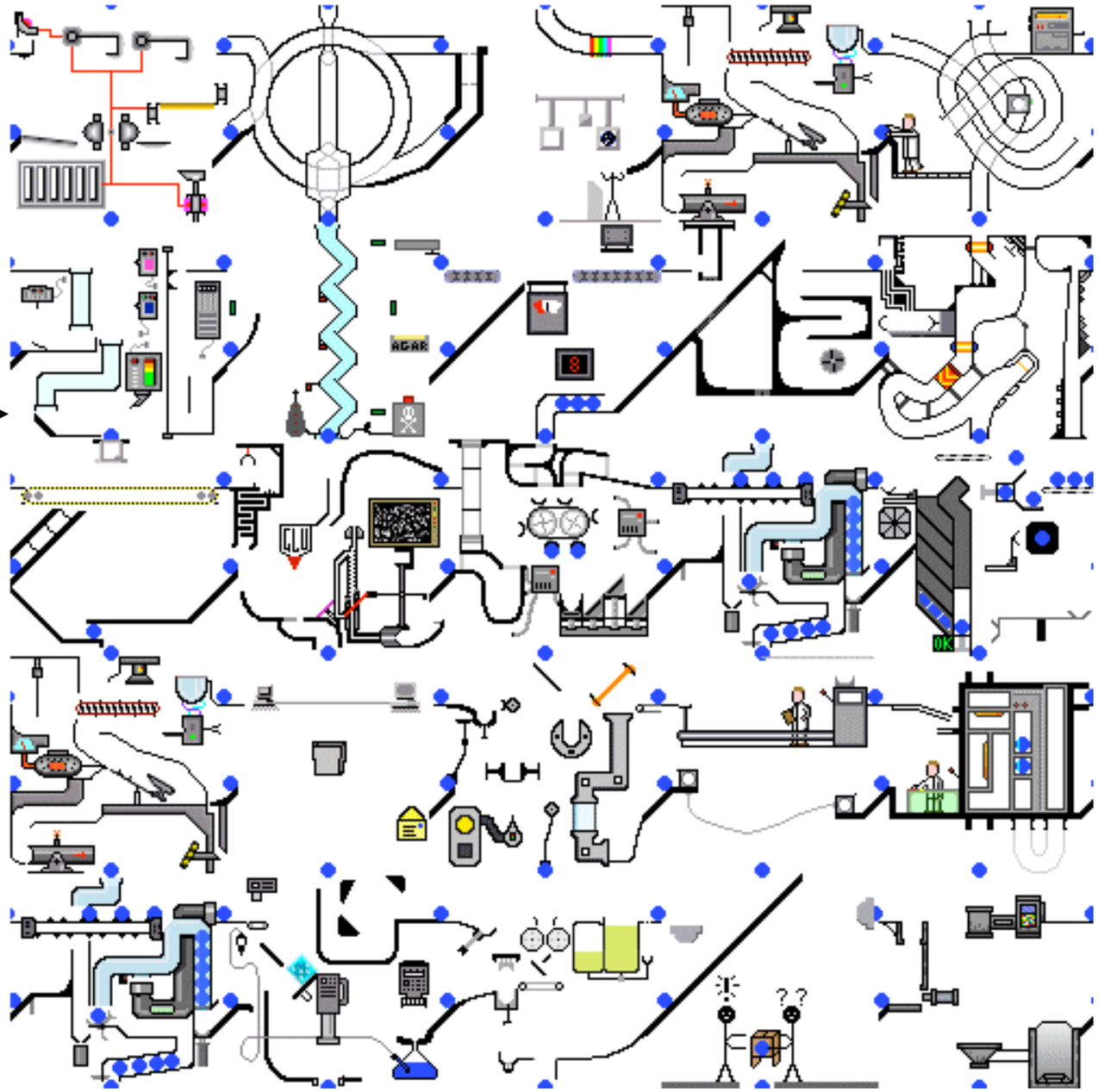
Find the dust:
http://www.astro.princeton.edu/~draine/dust/dustmix.html

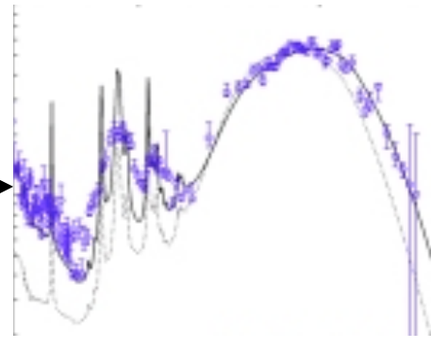
Draine (2003abc): extinction, absorption, albedo, $\langle\cos(\theta)\rangle$,$\langle\cos^2(\theta)\rangle$ for wavelengths 1cm - 1 Angstrom for a mixture of C and Silicate grains for ISM



The Problem







Outline

- Why model dust?
- The physics of dust: emission, absorption & scattering
- Monte Carlo methods
- Hyperion
- Project

Monte Carlo Method

- Repeated random sampling from a probability distribution
- Invented by Stanislaw Ulam, John Von Neumann, and Nicholas Metropolis in 1940s, who were working on the Manhattan Project at LANL
- Led to the field of random number generation

Monte Carlo Method

- Radiative Transfer: randomly sample frequencies of photons (or "photon packets (PPs)") from some source spectrum

- Don't solve radiation transfer directly; instead follow interaction of each PP with the dust

- After traveling some optical depth, PPs are either scattered, absorbed or re-emitted; actions also randomly sampled from some probability distribution function

Outline

- Why model dust?
- The physics of dust: emission, absorption & scattering
- Monte Carlo methods
- Hyperion
- Project

www.hyperion-rt.org

robitaille@mpia.de

"astrofrog"

hyperion

Read Docs | Download TAR Ball | View On GitHub

Getting started

Hyperion is a parallelized 3-d dust continuum radiative transfer code. The code is described in Robitaille (2011). Its main features include:

- Dust continuum radiative transfer
- Dust temperature calculation
- SEDs, images, and polarization maps
- Support for arbitrary 3-d geometry
- Support for multiple sources and dust populations
- Support for arbitrary dust properties
- Cartesian, Polar, and Adaptive grids (Octree and AMR)
- Easy-to-use Python library to set up, run, and post-process models
- High performance parallelized (MPI) Fortran 2003 core

The current stable version of Hyperion is 0.9.1 - download

The documentation and installation instructions are located at http://docs.hyperion-rt.org

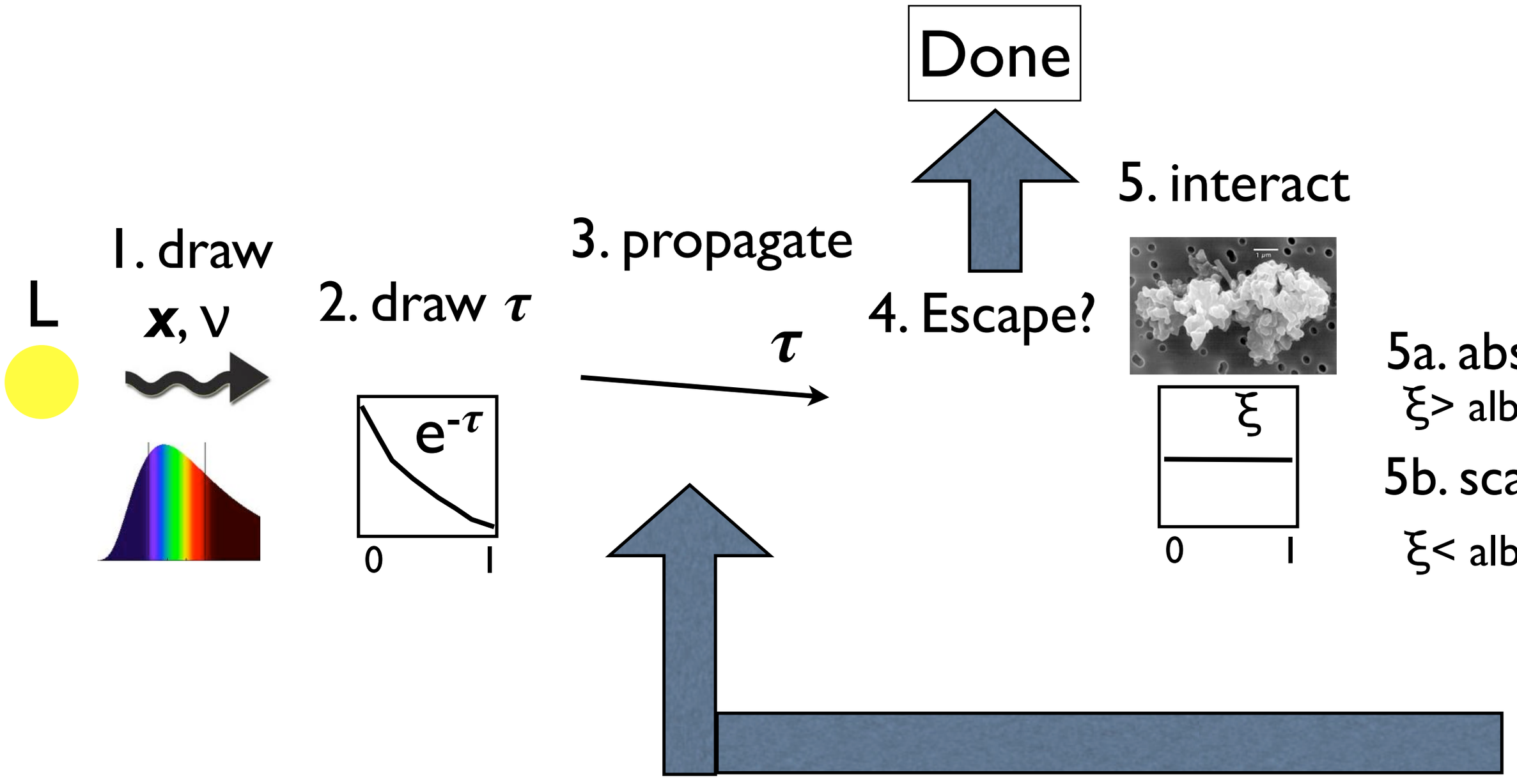
Inputs 1: Sources

- User specifies: (1) each source bolometric luminosity (2) source position (3) source spectrum
- Types of sources:
  - Isotropic point sources
    - Spherical sources (with or without limb darkening, hot/cool spots properties)
    - Diffuse sources: flux emitted from grid cells (e.g. unresolved sources like shocks or viscous energy dissipations)
    - External isotropic sources (e.g. interstellar radiation field)

Inputs 2: Dust

- User specifies:
  - (1) Dust density distribution*
  - (2) Dust properties (extinction coefficient, albedo, scattering properties)
  - (3) Dust sublimation (cap specific energy absorption; all dust removed from cells exceeding maximum specific energy absorption rate; dust density reduced but not zero)
- Mean dust opacities and emissivities are pre-computed (before MC), assuming LTE: $j_\nu = \kappa_\nu B_\nu(T)$
- *Types of grids: 3D cartesian, spherical, cylindrical, adaptive cartesian (octree and AMR)

Photon Packet Propagation



- For high optical depth, switch to diffusion approximation

Temperature/Energy absorption rate calculation I

- Iterative method to obtain temperature.
- In first iteration, compute the specific energy absorption rate of the dust:
- Then compute the dust temperature, T, assuming LTE:
- Dust quantities are tabulated as a functions of absorption rate not T (T is computed on-the-fly from A)

dt =time photon packets are emitted
V = cell volume
epsilon = energy of photon packet
l = path length traveled
kappa_nu = mass absorption coefficient

$$\dot{A} = \frac{1}{\Delta t}\frac{\epsilon}{V}\sum \ell\,\kappa_\nu$$

$$4\pi\,\kappa_P(T)\,B(T) = \dot{A}$$

kappa_P = Planck mean mass absorption coefficient
B(T) = nu integrated Planck function

$$B(T) = (\sigma/\pi)\,T^4$$

Temperature/Energy absorption rate calculation II

- Iterative method to obtain temperature.
- Because emissivity depends on A, which depends on photon packet propagation, which depends on the photon frequency, which depends on the emissivity, must iterate....
- Converged when 99.9% of cells have absorption rate differences by less than a factor of 2 (need to specify)
- Users can set this threshold

dt =time photon packets are emitted
V = cell volume
epsilon = energy of photon packet
l = path length traveled
kappa_nu = mass absorption coefficient

$$\dot{A} = \frac{1}{\Delta t}\frac{\epsilon}{V}\sum \ell\,\kappa_\nu$$

$$4\pi\,\kappa_P(T)\,B(T) = \dot{A}$$

kappa_P = Planck mean mass absorption coefficient
B(T) = nu integrated Planck function

$$B(T) = (\sigma/\pi)\,T^4$$

$$\kappa_P \equiv \frac{\int_0^\infty I_{b\lambda}\kappa_\lambda\,d\lambda}{\int_0^\infty I_{b\lambda}\,d\lambda}$$

SEDs and Images

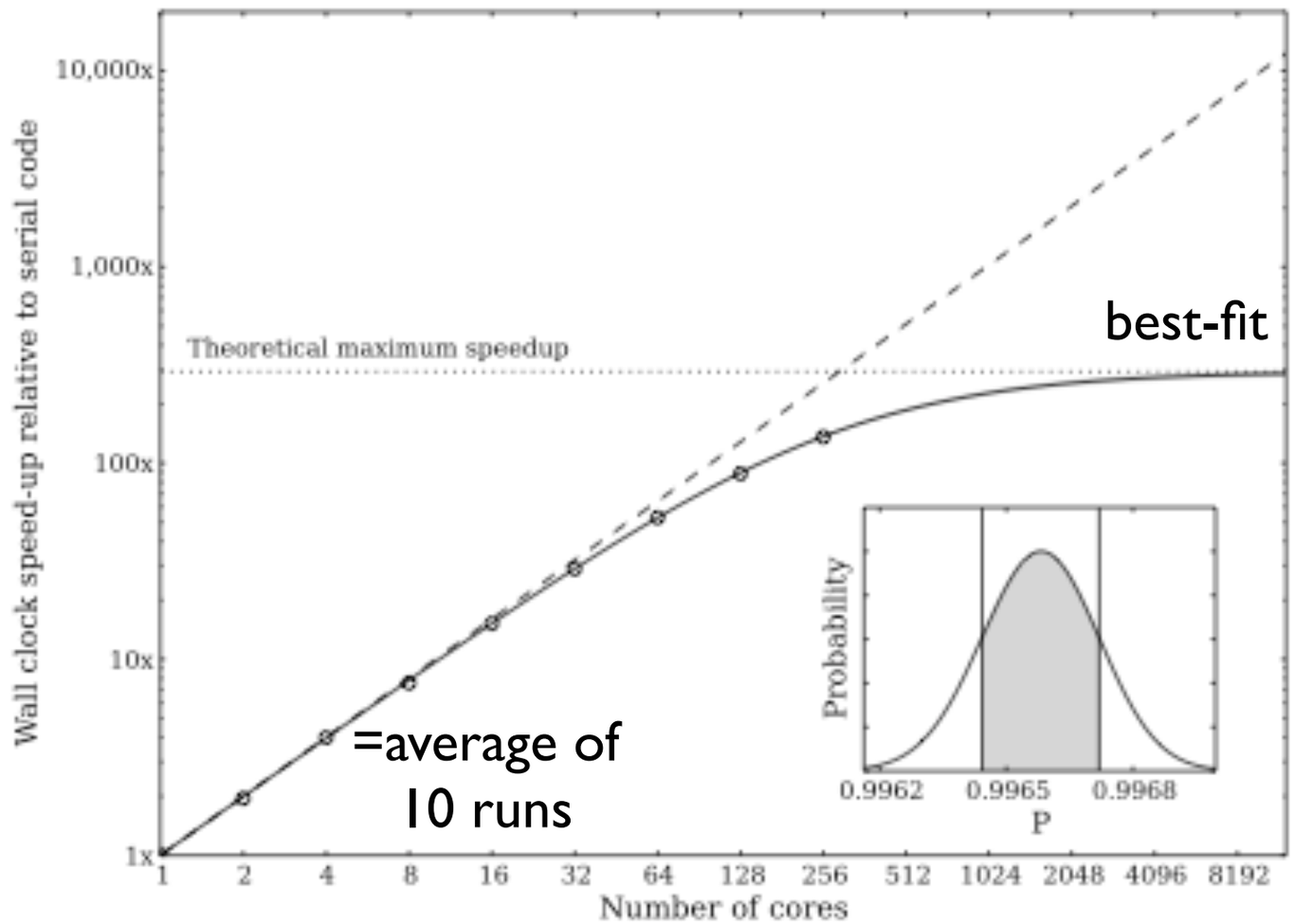
- Several options for computing these:
- "Photon binning" = bin PPs in viewing angles as they escape from the grid
  - Inefficient since PPs only contribute once and angles can't be arbitrarily small
- "Peeling-off" = after each scattering or re-emission, the probability, p, of the PP being scattered or re-emitted toward the observer is computed; this is added to the SED/images with weight $pe^{-\tau}$
  - better signal-to-noise
- "Raytracing"=determine the source function at each grid position and solving (in post-processing) the equation of radiative transfer along rays to observer
  - currently only works for thermal emission (long wavelengths); implemented for scattering in future

combine

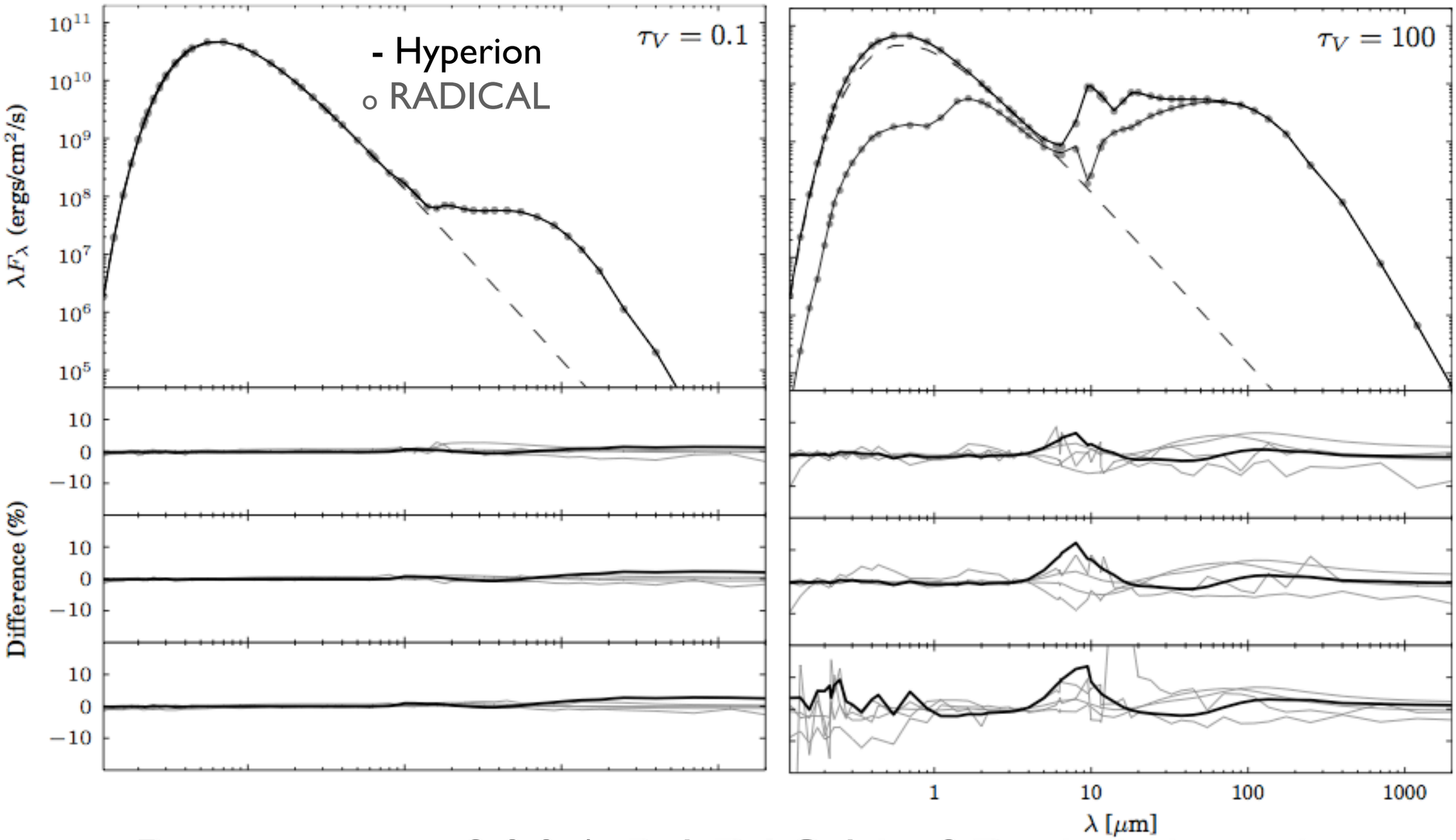
Hyperion is Parallel

- Scales great:
- Photon packet propagation is "embarrassingly parallel"
- Turnover is set by time of serial part (mostly I/O) of the code

Robitaille 2011



Test Problems

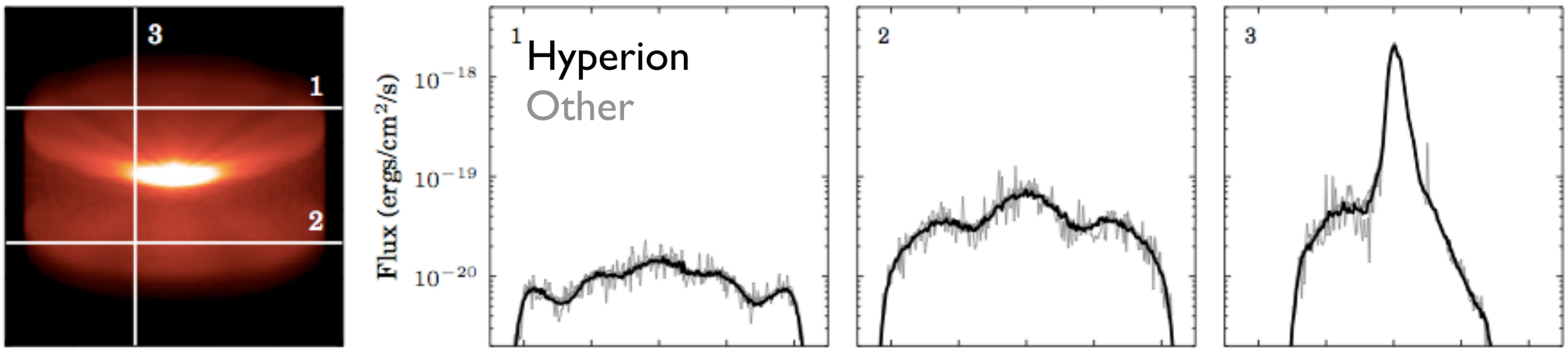


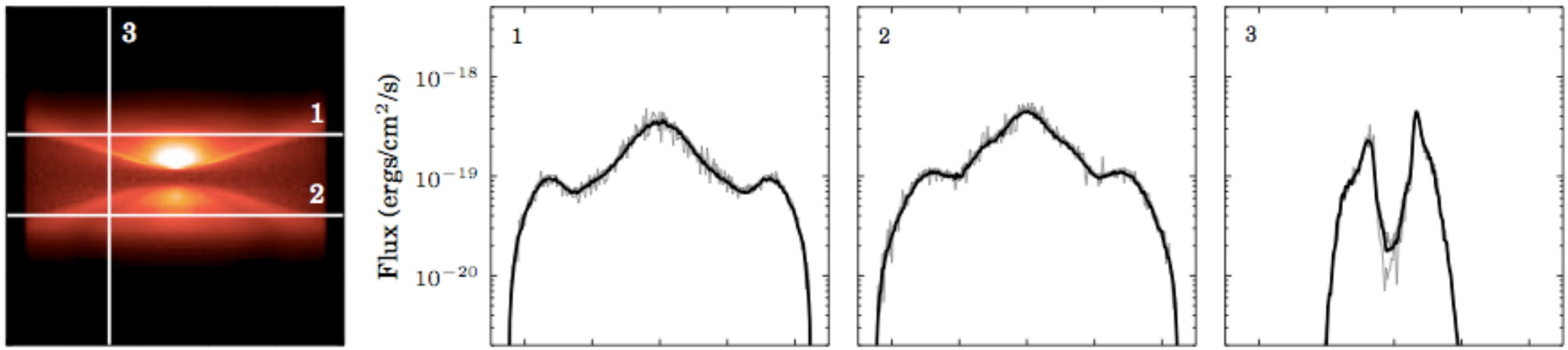
- Pascucci ea 2004, RADICAL; 2D disk bench mark
- Test opacity range with 4 disk masses

Robitaille 2011

Test Problems

Robitaille 2011



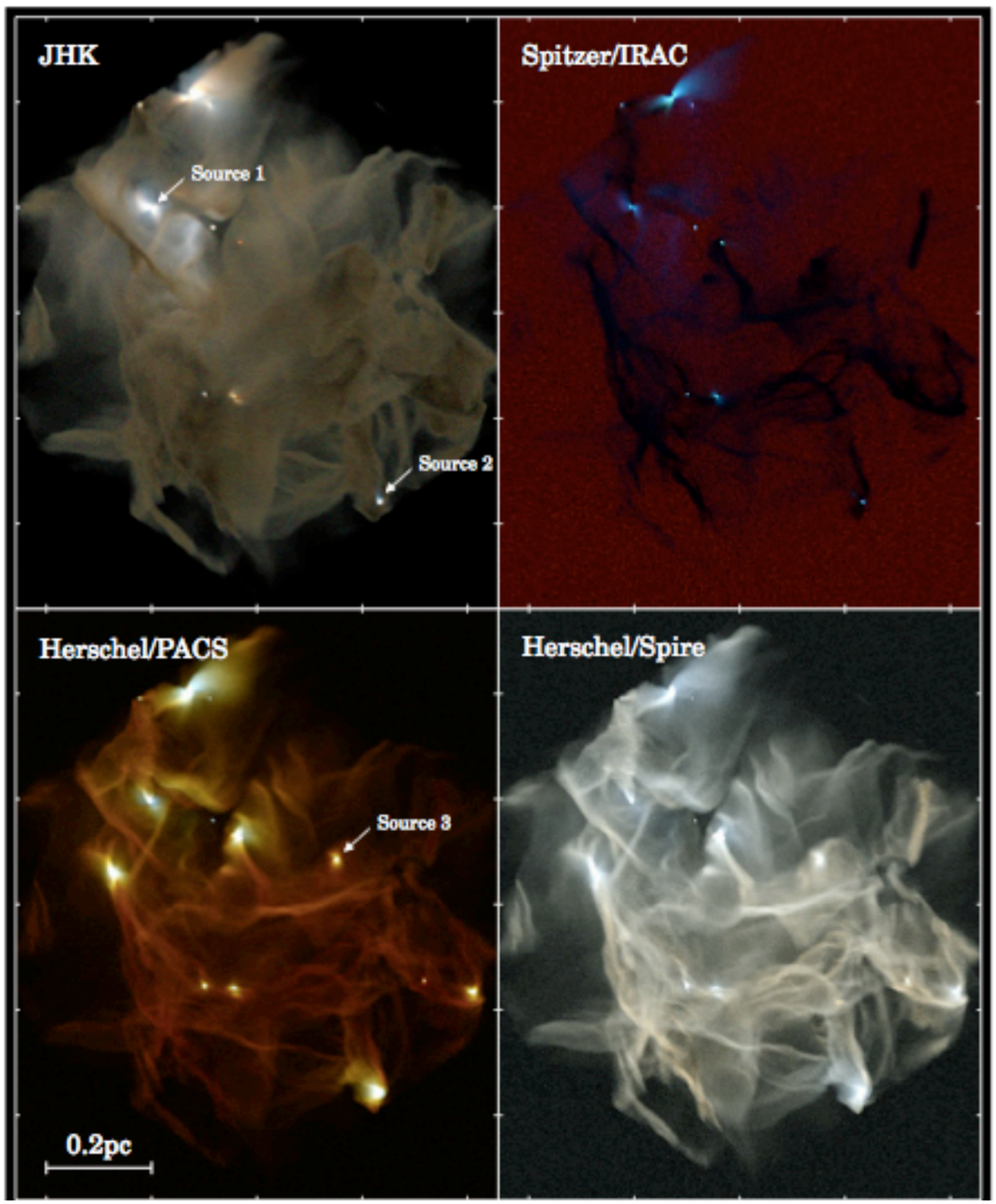


- Pinte et al 2009, MCFOST, MCMAX, TORUS and Pinball codes; disk bench mark
- Tests disk midplane $\tau v > 1e6$, anisotropic scattering, images and polarization

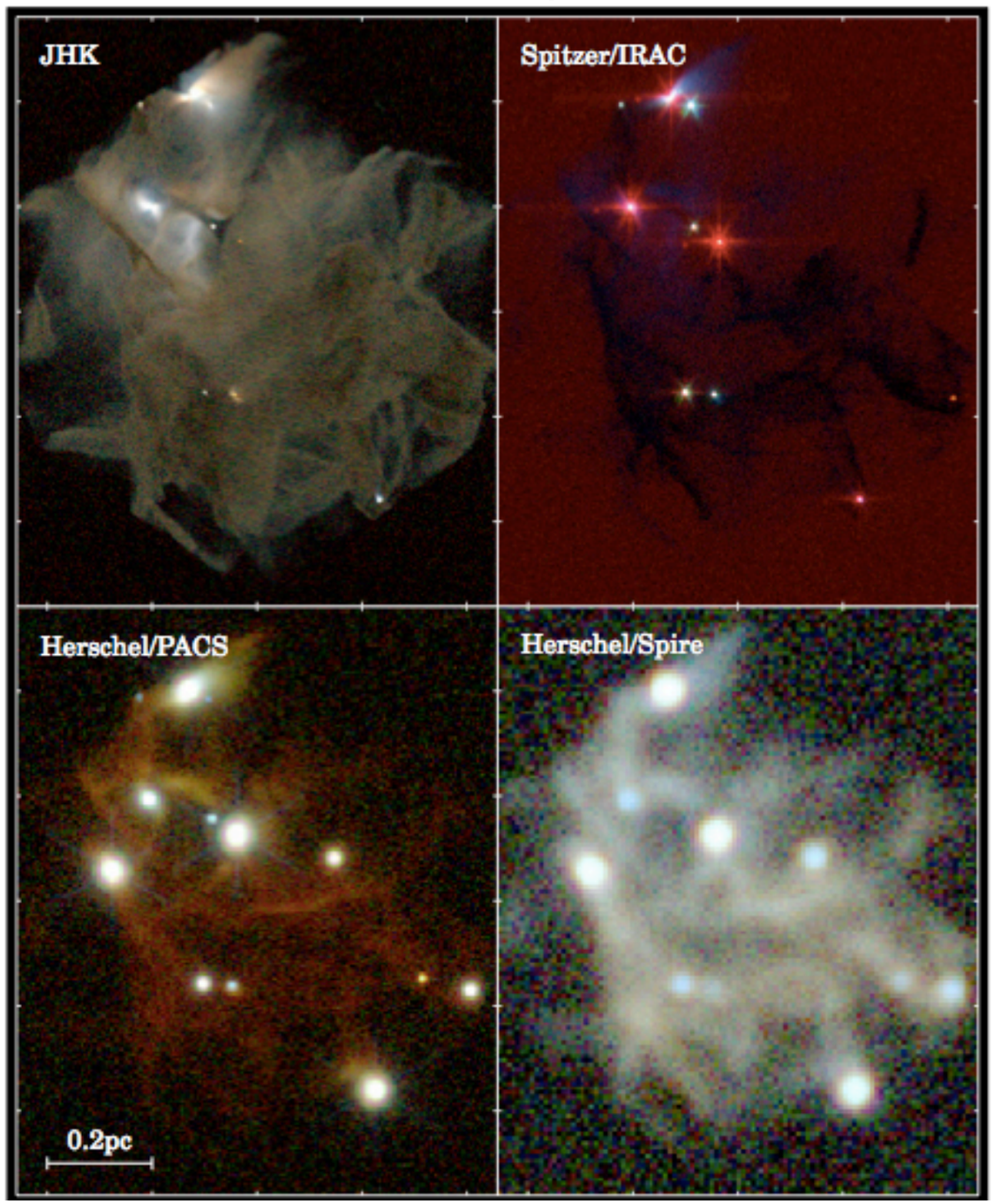
Application

- Synthetic observation of star formation simulation of forming cluster:

Robitaille 2011

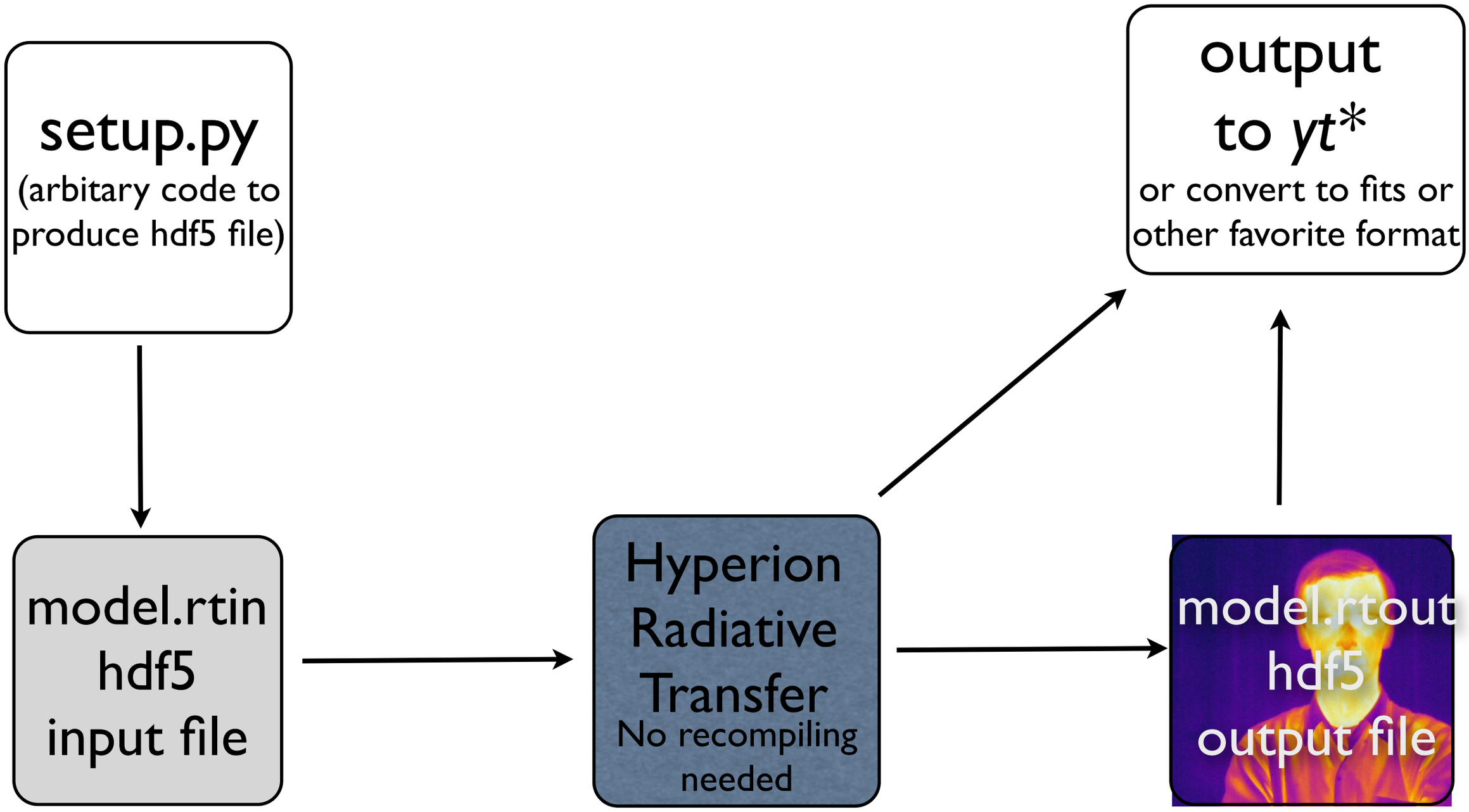


w/ noise, pixel resolution, PSF



Simulation from Offner ea 2009

Schematic



*Hyperion 0.9.2 to be released this week...

Example setup.py

The People Who Stare at Code II

Sample Python Setup Code

```
import numpy as np
from hyperion.model import Model
from hyperion.util.constants import pc, lsun

# Initialize model
m = Model()

# Set one-cell cartesian grid
w = np.linspace(-pc, pc, 32)
m.set_cartesian_grid(w, w, w)

# Add density grid with constant density
m.add_density_grid(np.ones(m.grid.shape) * 4.e-20, 'kmh_lite.hdf5')

# Add a point source in the center
s = m.add_point_source()
s.luminosity = 1000 * lsun
s.temperature = 6000.
```

Example setup.py

```
# Add 10 SEDs for different viewing angles
image = m.add_peeled_images(sed=True, image=False)
image.set_wavelength_range(250, 0.01, 5000.)
image.set_viewing_angles(np.linspace(0., 90., 10), np.repeat(20., 10))
image.set_track_origin('basic')

# Add multi-wavelength image for a single viewing angle
image = m.add_peeled_images(sed=False, image=True)
image.set_wavelength_range(30, 1., 1000.)
image.set_viewing_angles([30.], [20.])
image.set_image_size(200, 200)
image.set_image_limits(-1.5 * pc, 1.5 * pc, -1.5 * pc, 1.5 * pc)

# Add a fly-around at 500 microns
image = m.add_peeled_images(sed=False, image=True)
image.set_wavelength_range(1, 499., 501.)
image.set_viewing_angles(np.repeat(45., 36), np.linspace(5., 355., 36))
image.set_image_size(200, 200)
image.set_image_limits(-1.5 * pc, 1.5 * pc, -1.5 * pc, 1.5 * pc)
```

Example setup.py

Needed for calculation of dust temperature

```
# Set runtime parameters
    m.set_n_initial_iterations(5)
    m.set_raytracing(True)
    m.set_n_photons(initial=1e6, imaging=1e7,
                    raytracing_sources=1e6, raytracing_dust=1e6)

    # Write out input file
    m.write('tutorial_model.rtin')
```

Example setup.py

```
# Set runtime parameters
    m.set_n_initial_iterations(5)
    m.set_raytracing(True)
    m.set_n_photons(initial=1e6, imaging=1e7,
                    raytracing_sources=1e6, raytracing_dust=1e6)

    # Write out input file
    m.write('tutorial_model.rtin')
```

Photon # for the specific energy calculation

Rule of Thumb

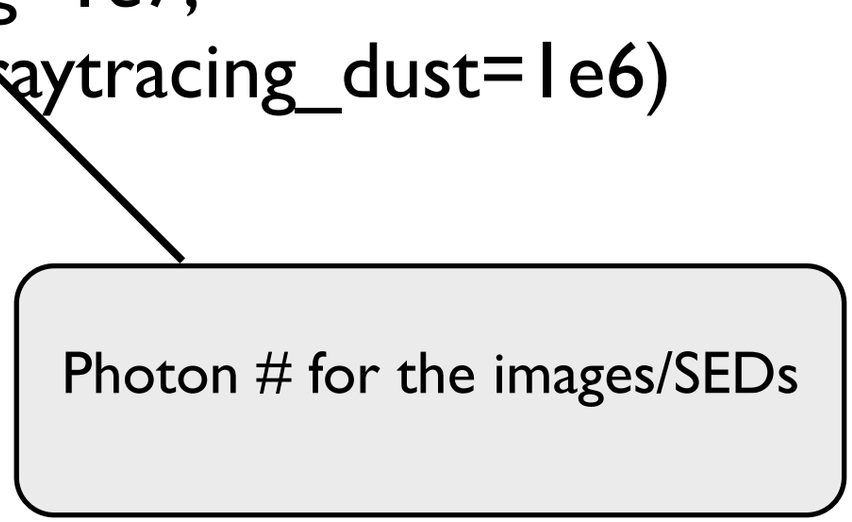
For optically thin: #PP ~ #cells

For optically thick: #PP ~ 10-100 #cells

Example setup.py

```
# Set runtime parameters
    m.set_n_initial_iterations(5)
    m.set_raytracing(True)
    m.set_n_photons(initial=1e6, imaging=1e7,
                    raytracing_sources=1e6, raytracing_dust=1e6)

    # Write out input file
    m.write('tutorial_model.rtin')
```



Example setup.py

```
# Set runtime parameters
    m.set_n_initial_iterations(5)
    m.set_raytracing(True)
    m.set_n_photons(initial=1e6, imaging=1e7,
                    raytracing_sources=1e6, raytracing_dust=1e6)

    # Write out input file
    m.write('tutorial_model.rtin')
```

Need to specific when raytracing (no scattering) is turned on

Example

setup.py

```
# Set runtime parameters
   m.set_n_initial_iterations(5)
   m.set_raytracing(True)
   m.set_n_photons(initial=1e6, imaging=1e7,
                   raytracing_sources=1e6, raytracing_dust=1e6)

   # Write out input file
   m.write('tutorial_model.rtin')
```

```
$ python setup.py
$ hyperion tutorial_model.rtin tutorial_model.rtout
        or
$mpirun -n 8 -env I_MPI_FABRICS shm:ofa
hyperion_car_mpi  tutorial_model.rtin tutorial_model.rtout
```

Example

setup.py

```
# Set runtime parameters
    m.set_n_initial_iterations(5)
    m.set_raytracing(True)
    m.set_n_photons(initial=1e6, imaging=1e7,
                    raytracing_sources=1e6, raytracing_dust=1e6)

    # Write out input file
    m.write('tutorial_model.rtin')
```

```
$ python setup.py
$ hyperion tutorial_model.rtin tutorial_model.rtout
        or
$mpirun -n 8 -env I_MPI_FABRICS shm:ofa
hyperion_car_mpi  tutorial_model.rtin tutorial_model.rtout
```

To view outputs:

1) hdfview tutorial_model.rtout

2) hyperion2fits --images tutorial_model.rtout

Outline

- Why model dust?
- The physics of dust: emission, absorption & scattering
- Monte Carlo methods
- Hyperion
- Project

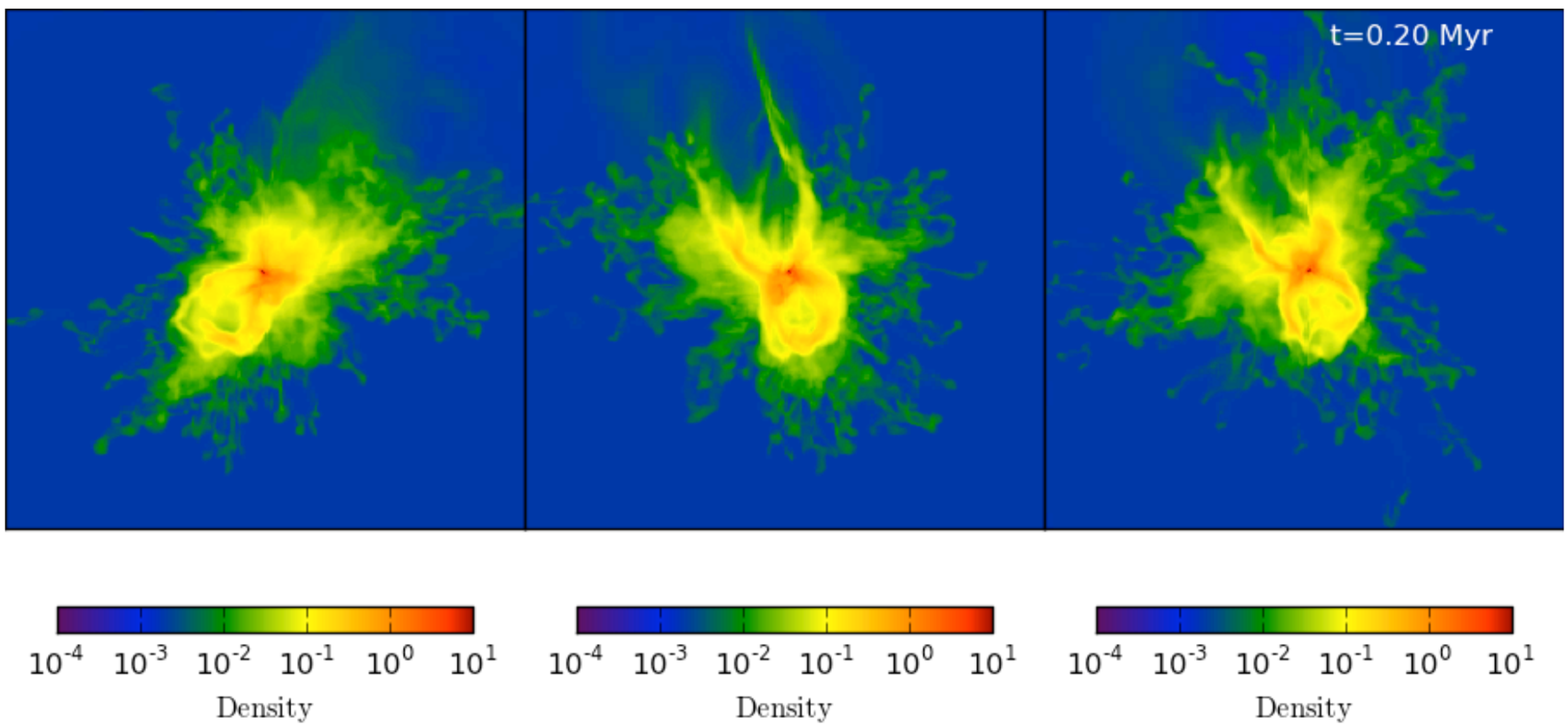
Project: Multi-wavelength images of a protostellar outflow

- Given: a fits file of the density distribution with a 1 Lsun protostellar source at the center
- Use Hyperion to compute an image of emission at several wavelengths along one (or more) sight-lines. Extract the estimated 3D dust temperature distribution.

Questions to answer:

(1) How many photons do I need for a converged image or SED?

2) What is the dust temperature distribution?



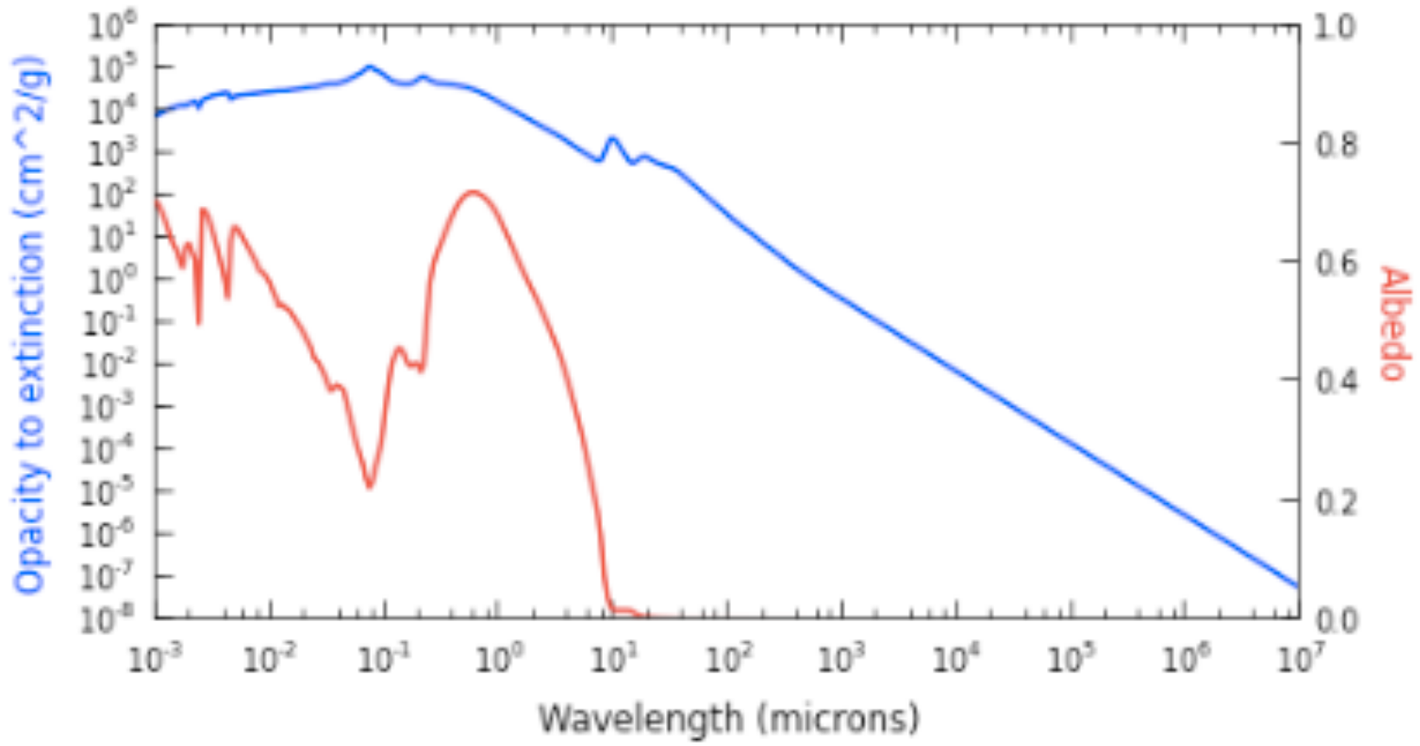
Project: Multi-wavelength images of a protostellar outflow

- Source code in: /home/soffner/hyperion-0.9.1/
- Project stuff: /home/soffner/hyperion-0.9.1/docs/tutorials/hipacc

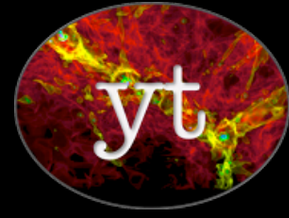
Contains:

Fits gas density file, hdf5 dust file, README

Rv=5.5, abundance and size distribution from Weingartner & Draine 2001



Outflow Temperature



-16
-17
-18
-19
-20
Log10(Density) [cgs]

160
80
40
20
Temperature [K]

T. Robitaille
& S. Offner

200 AU

Useful References

- Journal articles: B. Whitney et al. 2003ab, T. Robitaille 2011, Weingartner & Draine 2001, Draine 2003abc
- Books: *Physics of the Interstellar and Intergalactic Medium*, Bruce Draine; *The Physics and Chemistry of the Interstellar Medium*, A. Tielens
- help@hyperion-rt.org.