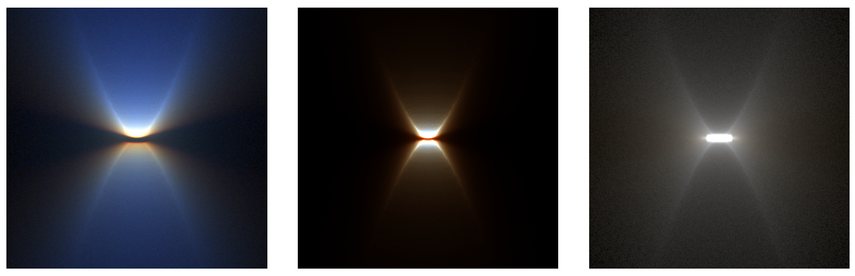
hyperion

Thomas P. Robitaille
Max Planck Institute for Astronomy

Introduction

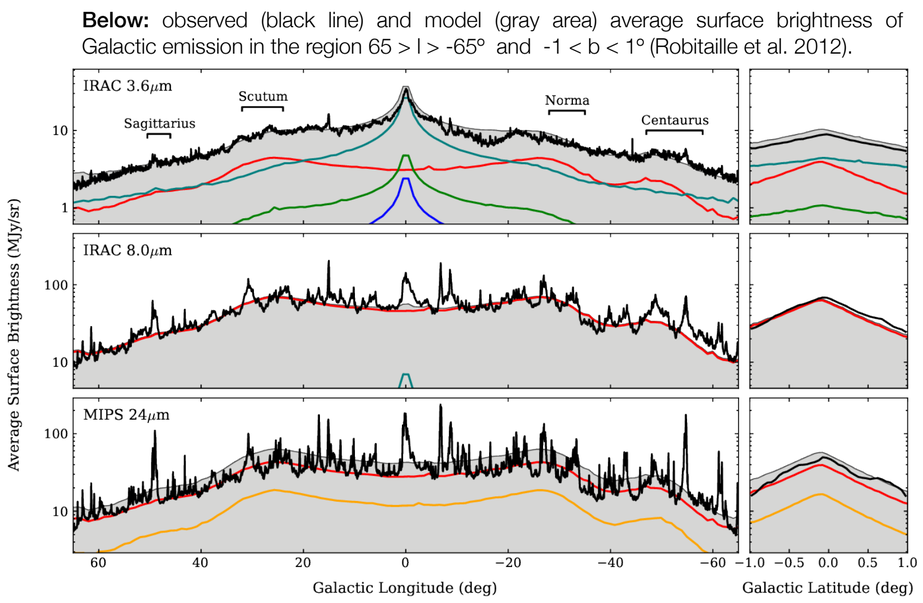
HYPERION is an open-source 3-d dust continuum Monte-Carlo radiative transfer code that is designed to be as generic as possible, allowing radiative transfer to be computed through a variety of three-dimensional grids (cartesian, spherical or cylindrical polar, or adaptive cartesian grids).

The main part of the code is problem-independent, and only requires an arbitrary three-dimensional density structure, dust properties, the position and properties of the illuminating sources, and parameters controlling the running and output of the code. HYPERION can compute temperatures, SEDs, images, and polarization maps.



Above: Near-infrared (left), mid-infrared (center), and far-infrared (right) synthetic images of an analytical protostellar model with a disk, envelope, and bipolar cavity.

Below: observed (black line) and model (gray area) average surface brightness of Galactic emission in the region 65 > l > -65° and -1 < b < 1° (Robitaille et al. 2012).



Star formation from local to global scales

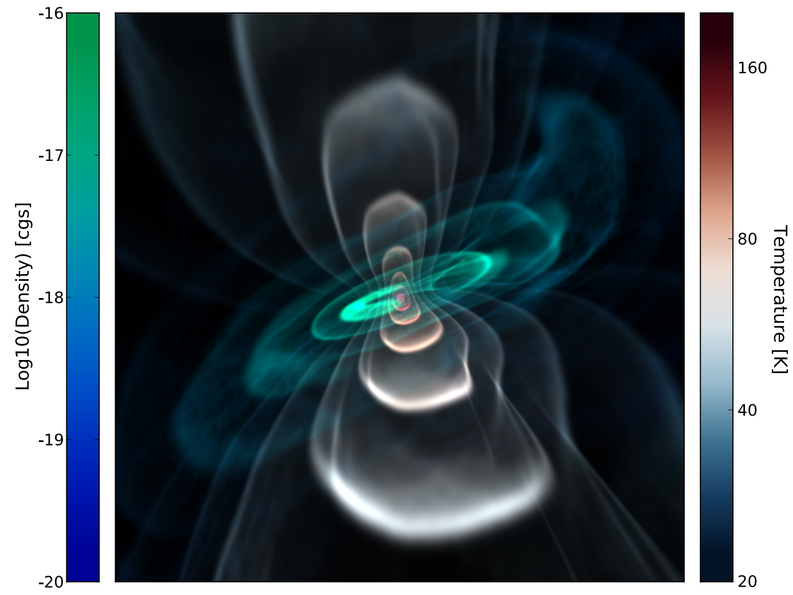
One of the fundamental aspects of HYPERION is the ability to compute radiative transfer on arbitrary dust densities, which makes it suitable for computing models relevant to star formation, from individual young stellar objects (YSOs) to whole star formation regions or galaxies.

For example, HYPERION is currently being used to compute a new generation of model YSO SEDs that will supersede the Robitaille et al. (2006) models, and has also been used to self-consistently compute the diffuse emission from the Milky-Way (Robitaille et al. 2012), shown on the left.

Interfacing with simulation tools

HYPERION is able to compute radiative transfer on adaptive mesh refinement (AMR) and octree grids, making it suitable to do (post-processing) radiative transfer on dynamical simulations. In addition, the output from Hyperion can be easily exported to the yt package (http://yt-project.org/), which can produce high quality volume rendering visualizations.

The plot on the right shows a simulation of a forming star by Offner et al. (in preparation). The density grid is taken from the simulation, the temperature was computed from Hyperion, and the visualization was produced by yt.



Above: volume rendering visualization of the density and temperature for a dynamical simulation of a forming star with a bipolar outflow (simulation from Offner et al. in preparation).

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

m = Model()
w = np.linspace(-pc, pc, 32)
m.set_cartesian_grid(w, w, w)
m.add_density_grid(np.ones(m.grid.shape) * 4.e-20, 'dust_file.hdf5')

s = m.add_point_source()
s.luminosity = 1000 * lsun
s.temperature = 6000.

image = m.add_peeled_images()
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)

m.set_raytracing(True)
m.set_n_photons(initial=1e6, imaging=1e7, raytracing_sources=1e6, raytracing_dust=1e6)

m.write('simple_model.rtin')
m.run('simple_model.rtout', mpi=True, n_processes=12)  # run in parallel on 12 processes
```

Implementation details

- Human-readable Python library for setting up, running, and analyzing models
- Core code written in Fortran 95/2003 for high performance
- Scripts to set up models allow much greater flexibility than parameter files
- Parallelized with MPI, tested on hundreds of cores
- Passes the Pascucci et al (2004) and Pinte et al (2009) disk benchmarks
- Extensive test suite to ensure stability
- Open source (source code available at https://github.com/hyperion-rt/hyperion)
- Extensive documentation at http://docs.hyperion-rt.org

Planned features

The following features are currently in development or planned in the long term:

- Basic hydrostatic equilibrium in disks
- Temperature-dependent opacities
- Raytracing for scattered light
- Basic photoionization
- Molecular lines
- Non-spherical grains aligned along magnetic fields
- Unstructured (Voronoi) meshes

The code is open source, so contributions are welcome!

http://www.hyperion-rt.org

OPEN SOURCE!

@astrofrog

Source code available on GitHub