

SKIRT Workshop

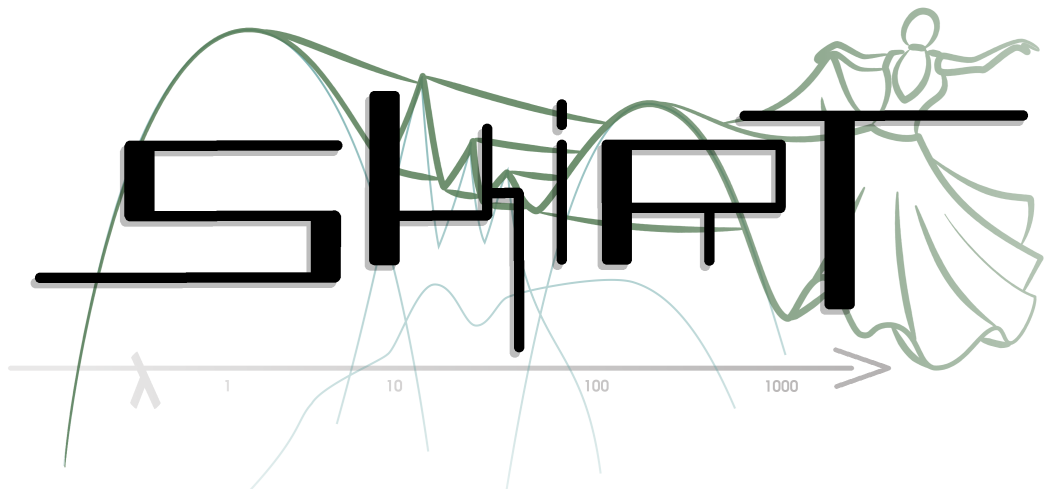
Saint Andrews Monte Carlo Summer School

August 2015

Peter Camps

peter.camps@ugent.be

Astronomical Observatory, Ghent University, Belgium



www.skirt.ugent.be

1 Agenda

The objective of this workshop is to practice radiative transfer with the SKIRT code. In addition to configuring and running a couple simulations, you will use the development environment to look at and extend the code (just a very basic introduction). So the agenda looks like:

- Using SKIRT: configuring, performing and interpreting simulations
- Extending SKIRT: a basic introduction

2 Installation

To complete all aspects of this workshop, you'll need a Mac OS X or Linux operating system, and you'll need to install:

- a recent C++11 compiler
- the Qt 5.3 development environment
- the SKIRT code
- a python distribution
- the Python Toolkit for SKIRT (PTS)
- a FITS file viewer such as DS9

2.1 Compiler, Qt and SKIRT

You can find an HTML version of the SKIRT documentation at www.skirt.ugent.be/skirt/ and a PDF version at www.skirt.ugent.be/downloads/SKIRT.pdf.

In the HTML version, go to the Installation Guide (first link on the Index page). In the PDF version, turn to chapter 12. Follow the instructions in the section "Installing and developing SKIRT on Mac OS X" or "Installing and developing SKIRT on Ubuntu Desktop" depending on your operating system.

Use the public SKIRT GitHub repository (<https://github.com/SKIRT/SKIRT>) to obtain the source code as described in the installation guide. You don't need to install the documentation streamliner or the documentation generator, so the last thing you'll do is verifying your installation.

2.2 Python and PTS

You can find an HTML version of the PTS documentation at www.skirt.ugent.be/pts/ and a PDF version at www.skirt.ugent.be/downloads/PTS.pdf.

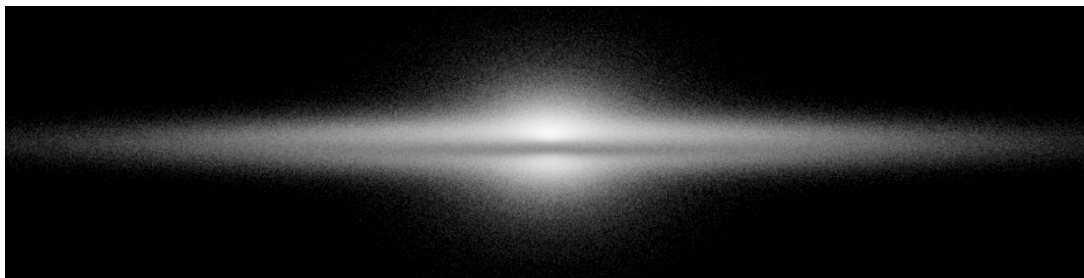
In the HTML version, go to the Installation Guide (first link on the Introduction page). In the PDF version, turn to chapter 4. Follow the instructions in the sections "Installing and developing PTS on Mac OS X" or "Installing and developing PTS on Ubuntu Desktop" depending on your operating system.

Use the public PTS GitHub repository (<https://github.com/SKIRT/PTS>) to obtain the source code as described in the installation guide. You don't need to install MEncoder nor configure Qt for PTS, so the last thing you'll do is setting PTS up for running.

2.3 FITS viewer

Visit the *SAOImage DS9* home page at <http://ds9.si.edu/site/Home.html> to install a FITS file viewer.

3 Monochromatic simulation of a dusty galaxy



In this tutorial you will use SKIRT to produce a nearly edge-on V-band image of a dusty galaxy model (see illustration). You will create a ski file containing the appropriate simulation parameters, and then you will actually perform the simulation. Finally you will adjust the ski file and perform a second simulation with slightly different parameters.

3.1 Creating the ski file

SKIRT stores all parameter values describing a particular simulation in a file with the ".ski" filename extension. To create a new ski file, you need to start SKIRT without any command line arguments. The ski file will be created in the current directory; for this tutorial you should use the run directory you created during SKIRT installation.

```
$ cd ~/SKIRT/run
$ skirt
Welcome to SKIRT v7 (git 475-389e7b7 built on ..)
Running on .. for ..
Interactively constructing a simulation...
? Enter the name of the ski file to be created: MonoDisc
```

Enter the displayed commands in a Terminal window. SKIRT responds with a welcome message and starts an interactive session in the terminal window, during which it will prompt you for all the information describing a particular simulation. The first question is for the filename of the ski file. For this tutorial, enter "MonoDisc".

3.2 Type of simulation

```
Possible choices for the simulation:
  1. An oligochromatic Monte Carlo simulation
  2. A panchromatic Monte Carlo simulation
? Enter one of these numbers [1,2] (1): 1
```

SKIRT offers two types of Monte Carlo simulation:

- An oligochromatic simulation operates at one or more distinct wavelengths rather than a discretized range. In such a simulation there can be absorption and scattering by dust grains, but by definition there is no thermal dust emission (calculating the dust temperature requires integrating over a range of wavelengths).
- A panchromatic simulation operates at a range of wavelengths (discretized on a grid) usually spanning UV, optical and infrared bands. In such a simulation there can be absorption, scattering and thermal emission by dust grains. This is the most powerful simulation type.

Your simulation will run at a single wavelength in the V-band, where thermal dust emission has no effect. An oligochromatic simulation is sufficient and most performance-effective for these purposes.

3.3 Random generator

Possible choices for the random number generator:

1. The default random generator

Automatically selected the only choice: 1

? Enter the seed for the random generator [1,2000000000] (4357):

SKIRT currently offers only a single random generator, so the default choice is automatically selected. If desired you can enter a custom seed for the random generator, but the default value is almost always fine. One reason to specify a custom seed is to repeat a simulation with a different realization of random model attributes, such as the clumps in a clumpy medium.

3.4 Units

Possible choices for the units system:

1. SI units
2. Stellar units (length in AU, distance in pc)
3. Extragalactic units (length in pc, distance in Mpc)

? Enter one of these numbers [1,3] (3):

SKIRT offers three sets of input/output units:

- SI units: standard international units. SKIRT always uses SI units for internal computations; the other unit systems are provided for user convenience: all values are converted to SI units during input and from SI units during output.
- Stellar units: use AU as the unit of length, parsec as the unit of distance, micron as the unit of wavelength, and M_{\odot} as the unit of mass.
- Extragalactic units: use parsec as the unit of length, Mpc as the unit of distance, micron as the unit of wavelength, and M_{\odot} as the unit of mass.

Since you will be simulating a galaxy, select extragalactic units (the default choice).

Possible choices for the output style for flux density and surface brightness:

1. Neutral: $\lambda F_{\lambda} = \nu F_{\nu}$
2. Wavelength: F_{λ}
3. Frequency: F_{ν}

? Enter one of these numbers [1,3] (1):

SKIRT can output fluxes in one of three different styles: per wavelength (F_{λ}), per frequency (F_{ν}), or neutral ($\lambda F_{\lambda} = \nu F_{\nu}$). You can obviously calculate one flux style from the other, but this way you can select your preference. For this simulation, just use the default.

3.5 Instrument system

Running a radiative transfer simulation is not particularly useful unless the results are registered by one or more instruments, which collect the photon packages that leave the system, in a process similar to the operation of real instruments at a telescope.

The SKIRT instruments essentially simulate imaging spectrometers, i.e. 3D spectrographs with one wavelength dimension (determined by the simulation's wavelength grid) and two spatial directions on the plane of the sky. Each instrument is characterized by its own instrumental characteristics (pixel size and field of view in the horizontal and vertical

directions) and by its position with respect to the system (for axisymmetric systems, this means the inclination and distance).

It should be noted that the number of instruments and the pixel size of these instruments affect both memory consumption (for storing the data cubes) and simulation run time (for peeling off photon packages in the direction of each instrument).

Possible choices for the instrument system:

1. An instrument system

Automatically selected the only choice: 1

There is only one type of instrument system (simply a list of instruments) so the default choice is automatically selected.

Possible choices for item #1 in the instruments list:

1. A basic instrument that outputs the total integrated flux as an SED
2. A basic instrument that outputs the total flux in every pixel as a data cube
3. A basic instrument that outputs the total flux as data cube and as SED
4. An advanced instrument that records individual contributions to the flux
- ...

? Enter one of these numbers or zero to terminate the list [0,6] (3): 2

? Enter the name for this instrument: i88

? Enter the distance to the system [0 Mpc,∞ Mpc]: 10

? Enter the inclination angle of the detector [0 deg,180 deg] (0 deg): 88

? Enter the azimuth angle of the detector [-360 deg,360 deg] (0 deg):

? Enter the position angle of the detector [-360 deg,360 deg] (0 deg):

? Enter the number of pixels in the horizontal direction [25,10000] (250): 800

? Enter the maximal horizontal extent [0 pc,∞ pc]: 20000

? Enter the number of pixels in the vertical direction [25,10000] (250): 200

? Enter the maximal vertical extent [0 pc,∞ pc]: 5000

For the first (and only) instrument in this tutorial simulation, select a basic instrument that outputs the total flux in every pixel. Since you will be shooting photon packages at a single wavelength only, the "data cube" will have only one frame. Also it doesn't make any sense to produce an SED.

Specify an appropriate name for the instrument. This name is used to identify the output files for the instrument. This is not so relevant here, but it becomes important when there are multiple instruments. Further specify a size of 800 times 200 pixels with a scale of 50 pc, a distance of 10 Mpc from the galaxy, and an inclination of 88 degrees. The remaining two position angles can be left to their default values of zero.

Possible choices for item #2 in the instruments list:

1. A basic instrument that outputs the total integrated flux as an SED
2. A basic instrument that outputs the total flux in every pixel as a data cube
3. A basic instrument that outputs the total flux as data cube and as SED
4. An advanced instrument that records individual contributions to the flux
- ...

? Enter one of these numbers or zero to terminate the list [0,6] (3): 0

When asked for the second instrument, enter a zero to terminate the list, since the tutorial simulation has only one instrument.

3.6 Photon packages

? Enter the number of photon packages per wavelength [0,1e15] (1e6):

The number of photon packages launched for each wavelength is a very important parameter, since it determines the accuracy of your simulation. On the other hand, the simulation run time essentially scales with the number of photon packages (multiplied by the number of wavelengths). Typical values range from 10^5 to 10^9 . An appropriate number is usually determined experimentally: first run your simulation with a modest number of photon packages, look at the results and then run the simulation again with more photon packages if needed.

For your first simulation, leave the number of photon package at the default value of 10^6 .

3.7 Wavelength grid

SKIRT works with a fixed wavelength grid, i.e. a vector of N_λ wavelengths, on which the entire simulation is run. For an oligochromatic simulation, the grid simply consists of one or more distinct wavelengths.

```
Possible choices for the wavelength grid:
  1. A list of one or more distinct wavelengths
Automatically selected the only choice: 1
? Enter the wavelengths [0.0001 micron,1e6 micron]: 0.55
```

For an oligochromatic simulation, the single applicable "wavelength grid" is automatically selected. To obtain a V-band image, enter a wavelength "list" containing the single value $\lambda = 0.55 \mu\text{m}$.

3.8 Stellar system

Each simulation must contain some luminous sources which emit radiation. In SKIRT terminology, these sources form a stellar system. A stellar system includes one or more stellar components. A stellar component in SKIRT can be based on a built-in geometry, or it can be created from the output of a hydrodynamical simulation (SPH or AMR-based). The stellar system in this tutorial simulation is composed of two components, a disk and a bulge, each based on a built-in geometry.

```
Possible choices for the stellar system:
  1. A stellar system
Automatically selected the only choice: 1
Possible choices for item #1 in the stellar components list:
  1. A stellar component with a built-in geometry (..)
  2. A stellar component derived from an SPH output file
  ...
? Enter one of these numbers [1,5] (1): 1
```

Select a first stellar component with a built-in geometry to represent the stellar disk.

```
Possible choices for the geometry of the spatial stellar distribution:
  1. A point source geometry
  2. A Plummer geometry
  3. A gamma geometry
  4. A Sersic geometry
  ...
  8. A shell geometry
  9. An exponential disk geometry
  10. A multi-gaussian expansion (MGE) geometry
  ...
```

- 30. A decorator that adds an offset to any geometry
- 31. A decorator that adds a rotation to any geometry
- 32. A decorator that constructs a spheroidal variant of any spherical geometry
- ...
- ? Enter one of these numbers [1,39] (1): 9
- ? Enter the radial scale length [0 pc,∞ pc]: 4000
- ? Enter the axial scale height [0 pc,∞ pc]: 350
- ? Enter the radial truncation length (zero means no truncation) [0 pc,∞ pc] (0 pc):
- ? Enter the axial truncation height (zero means no truncation) [0 pc,∞ pc] (0 pc):
- ? Enter the luminosities, one for each wavelength, in solar units [0,∞]: 5e9

For an oligochromatic simulation, each stellar component is characterized by a particular geometry and by the total component luminosity at each wavelength in the simulation's wavelength grid. For the geometry of the first stellar component, select an exponential disk with a scale length of 4 kpc and a scale height of 350 pc. Leave the truncation length and height at the default value of zero (meaning "no truncation"). Further specify a V-band luminosity of $L_V = 5 \times 10^9 L_{\odot,V}$.

- Possible choices for item #2 in the stellar components list:
 - 1. A stellar component with a built-in geometry (..)
 - ...
- ? Enter one of these numbers or zero to terminate the list [0,5] (0): 1
- Possible choices for the geometry of the spatial stellar distribution:
 - 1. A point source geometry
 - 2. A Plummer geometry
 - 3. A gamma geometry
 - 4. A Sersic geometry
 - ...
 - 30. A decorator that adds an offset to any geometry
 - 31. A decorator that adds a rotation to any geometry
 - 32. A decorator that constructs a spheroidal variant of any spherical geometry
 - ...
- ? Enter one of these numbers [1,39] (1): 32
- Possible choices for the spherical geometry to be made spheroidal:
 - 1. A Plummer geometry
 - 2. A gamma geometry
 - 3. A Sersic geometry
 - 4. A pseudo-Sersic geometry
 - 5. An Einasto geometry
 - 6. A shell geometry
- ? Enter one of these numbers [1,6] (1): 3
- ? Enter the Sersic index n [0.5,10] (1): 2
- ? Enter the effective radius [0 pc,∞ pc]: 1600
- ? Enter the flattening parameter q [0,1]: 0.7
- ? Enter the luminosities, one for each wavelength, in solar units [0,∞]: 3e9

Select a second stellar component with a built-in geometry to represent the stellar bulge. Use the spheroidal decorator to construct a flattened Sérsic model with Sérsic index $n = 2$, effective radius $R_e = 1.6$ kpc, and flattening parameter $q = 0.7$. Further specify a V-band luminosity of $L_V = 3 \times 10^9 L_{\odot,V}$.

- Possible choices for item #3 in the stellar components list:
 - 1. A stellar component with a built-in geometry (..)
 - ...
- ? Enter one of these numbers or zero to terminate the list [0,5] (0): 0

When asked for a third stellar component, terminate the list by entering a zero.

3.9 Dust system

The dust system contains all the information on the dust in a SKIRT Monte Carlo simulation. Broadly speaking, a dust system consists of two pieces of information. On the one hand, it includes a complete description of the spatial distribution and optical properties of the dust (the *dust distribution*). On the other hand, it contains all information on the grid structure on which this distribution is discretized (the *dust grid*).

Possible choices for the dust system:

1. A dust system for use with oligochromatic simulations
- ? Enter one of these numbers or zero to select none [0,1] (1): 1

For an oligochromatic simulation, the only applicable type of dust system is the oligochromatic dust system, which implements dust absorption and scattering but does not support thermal emission by the dust. However the dust system is optional: if you enter a zero in response to the above question, the simulated system will contain no dust (and hence there will be no dust absorption and scattering). For this tutorial simulation, do select a dust system (or the results would not be extremely interesting).

Possible choices for the dust distribution:

1. A dust distribution composed of various dust components
 2. A dust distribution derived from an SPH output file
 - ...
- ? Enter one of these numbers [1,5] (1): 1

Similar as for the stellar system, the dust distribution can be composed of a number of dust components, each with specific geometry and dust properties, or it can be created from the output of a hydrodynamical simulation (SPH or AMR-based). The dust distribution in this tutorial simulation is composed of a single disk component. So you need to select the "component" dust distribution type.

Possible choices for item #1 in the dust components list:

1. A dust component
- Automatically selected the only choice: 1

There is only one type of dust component, so the first dust component is automatically selected. Each dust component is characterized by a particular geometry (determining the dust density distribution), a dust mix (determining the optical properties of the dust grains), and normalization parameters (determining the total amount of dust in the component).

Possible choices for the geometry of the dust component:

1. A point source geometry
 2. A Plummer geometry
 3. A gamma geometry
 4. A Sersic geometry
 - ...
 8. A shell geometry
 9. An exponential disk geometry
 10. A multi-gaussian expansion (MGE) geometry
 - ...
- ? Enter one of these numbers [1,39] (9): 9
 ? Enter the radial scale length [0 pc,∞ pc]: 4000
 ? Enter the axial scale height [0 pc,∞ pc]: 140
 ? Enter the radial truncation length (zero means no truncation) [0 pc,∞ pc] (0 pc):
 ? Enter the axial truncation height (zero means no truncation) [0 pc,∞ pc] (0 pc):

For the geometry of your dust component, select an exponential disk with a scale length of 4 kpc and a scale height of 140 pc. Leave the truncation length and height at the default value of zero (meaning “no truncation”). Note that the dust scale height is much smaller than the stellar scale height, which should produce a nice dust lane when viewing the galaxy edge-on.

Possible choices for the dust mixture of the dust component:

1. A Draine & Li (2007) dust mix
2. A dust mix with average interstellar properties
3. A dust mix with mean Zubko properties
- ...

? Enter one of these numbers [1,12] (2):

? Do you want to output the optical properties of the dust mix? [yes/no] (yes):

? Do you want to output the mean optical properties of the dust mix? [yes/no] (yes):

SKIRT offers many different built-in dust mixes. For this tutorial simulation, select the default dust mix, which reflects the mean properties of the diffuse dust in the Milky Way. Leave the output options at their default values.

Possible choices for the type of normalization for the dust component:

1. Normalization by defining the total dust mass
2. Normalization by defining the radial optical depth at some wavelength
3. Normalization by defining the face-on optical depth at some wavelength
4. Normalization by defining the edge-on optical depth at some wavelength
- ...

? Enter one of these numbers [1,7] (1): 3

? Enter the wavelength for the optical depth [0.0001 micron,1e6 micron]: 0.55

? Enter the face-on optical depth at this wavelength [0,∞]: 1

The amount of dust is often determined by fixing the face-on or edge-on optical depth at a given wavelength. Specify a face-on optical depth of 1 at wavelength $\lambda = 0.55 \mu\text{m}$.

Possible choices for item #2 in the dust components list:

1. A dust component

? Enter one of these numbers or zero to terminate the list [0,1] (1): 0

When asked for the second dust component, terminate the list by entering a zero.

Possible choices for the dust grid structure:

1. A cylindrical grid with a linear distribution
2. A cylindrical grid with a power-law distribution
3. A cylindrical grid with a radial logarithmic and axial linear distribution
4. A cylindrical grid with a radial logarithmic and axial power-law distribution
- ...

10. An octtree dust grid structure

11. An k-d tree (binary tree) dust grid structure

...

? Enter one of these numbers [1,13] (10): 2

? Do you want output for plotting the structure of the grid? [yes/no] (yes):

? Enter the outer radius in the radial direction [0 pc,∞ pc]: 25000

? Enter ratio of the inner- and outermost bins in radial direction [0,1e4] (50): 30

? Enter number of radial grid points [5,100000] (250): 100

? Enter outer radius in the axial direction [0 pc,∞ pc]: 5000

? Enter ratio of the inner- and outermost bins in axial direction [0,1e4] (50): 50

? Enter number of axial grid points [5,100000] (250): 100

SKIRT discretizes the dust medium on a spatial grid structure, i.e. a collection of small dust cells covering three-dimensional space. Within each cell, the density and temperature of the dust are considered to be constant. SKIRT offers several types of dust grids, which can be selected depending on the requirements of the simulation.

Depending on the symmetries in the stellar and dust geometries, SKIRT supports 1D spherical grids (the grid cells are thin concentric shells), 2D axisymmetric grids (the grid cells have the form of a torus) and 3D cartesian grids (the grid cells are little cuboids). Naturally simulations in a 1D or 2D grid (when allowed by the system's geometry) are much more efficient than simulations in a full 3D grid.

Each grid type features parameters to set the detailed grid cell distribution in each direction. This is one of the most difficult tasks during model setup. The grid must be chosen in such a way that the entire configuration space is covered and with the resolution highest in those parts where the radiation field or the dust density changes most. In many cases, a linear distribution is not the best choice. A logarithmic grid is denser in the center than in the outer regions, which is often required. A logarithmic grid however, has the drawback that it does not go all the way down to zero. A useful alternative is the power-law grid, in which the dust cells are also gradually bigger when they are farther away from the center.

For this tutorial simulation, select a 2D cylindrical grid structure with a power-law distribution in both directions. Specify a radius of 25 kpc and height of 5 kpc, which should be sufficient to contain the vast majority of the dust, and 100 cells in each direction (so that there are 10000 cylindrical cells in total). Further specify a ratio of the width of the innermost and outermost bins of 30 in the radial and 50 in the vertical direction; the ratio is larger in the vertical direction because we want a higher resolution there in the inner regions.

```
? Enter random density samples for determining cell mass [10,1000] (100):
? Do you want to output convergence checks on the dust system? [yes/no] (yes):
. . .
```

Finally the dust system takes some additional parameters which can be left to their default value.

3.10 Writing the ski file

```
Successfully created ski file 'MonoDisc.ski'.
To run the simulation use the command: skirt MonoDisc
```

After all questions have been answered, SKIRT writes out the resulting ski file and terminates.

3.11 Running the simulation

To actually run the simulation, enter the following command (with the same current directory):

```
skirt MonoDisc
```

SKIRT responds with a welcome message and immediately starts performing the simulation. In this mode SKIRT never asks a question so it can be left to run unattended:

```
Welcome to SKIRT v7 (git 475-389e7b7 built on ..)
Running on .. for ..
Constructing a simulation from ski file 'MonoDisc.ski'...
Starting simulation MonoDisc...
```

```

Starting setup...
. . .
- Finished setup in 1.4 s.
Starting the simulation run...
Starting the stellar emission phase...
. . .
- Finished the stellar emission phase in 14.3 s.
Starting writing results...
. . .
- Finished writing results in 0.0 s.
- Finished the simulation run in 14.4 s.
- Finished simulation MonoDisc in 15.8 s.

```

SKIRT produces various log messages to report on specific activities and overall progress. The execution of a simulation progresses in three distinct stages:

- During *construction*, an internal data structure is created representing the simulation, based on the contents of the specified ski file. In addition to remembering all ski file parameters, the data structure provides room for all data used and/or produced during the simulation. Since the data is not yet initialized, construction completes in a fraction of a second (or fails if there is a problem with the ski file).
- During *setup*, all simulation data structures are initialized. For example, dust properties are read from data files, and the dust density is computed for all cells in the dust grid. Depending on the simulation model and parameters, setup may take seconds, minutes or even hours.
- During the *simulation run*, photon packages are launched through the dust grid and detected by instruments; at the end the simulation results are written down. For an oligochromatic simulation, there is just a stellar emission phase (for each wavelength). In panchromatic simulations, there can also be dust self-absorption and dust emission phases.

3.12 Output files

The output files for this tutorial simulation are written in the current directory. All filenames start with "MonoDisc", i.e. the name of the ski file. Three files are always generated for every simulation:

- MonoDisc_log.txt contains the progress log as it was written to the console.
- MonoDisc_parameters.xml contains a copy of the ski file, with a time stamp of when the simulation was run.
- MonoDisc_parameters.tex contains a \TeX "printout" of the contents of the ski file. Using a regular \TeX installation, this file can be easily processed into a nicely formatted document.

During setup SKIRT outputs several files describing the dust system (you can turn this off by answering "no" to the respective questions during the interactive session). The following files are written for this tutorial simulation (where "ds" stands for "dust system"):

- MonoDisc_ds_convergence.dat is a short text file providing a consistency check on the distribution of the grid cells. It contains the face-on and edge-on central surface density of the dust distribution as well as the total equivalent dust mass (1) obtained by integrating over the cells and (2) as expected without discretization. If these numbers match within a few percent, your dust grid samples the dust density well. If there are large deviations, it might be useful to consider a different grid cell distribution (e.g. with a larger extent, with more resolution, or with more or less contrast between inner and outer cells).

- MonoDisc_ds_0_opti.dat & MonoDisc_ds_0_mean.dat & MonoDisc_ds_0_mass.dat list properties of the dust used in the simulation. The first file contains extinction, absorption and scattering cross sections and the asymmetry parameter for each dust mix population used in the simulation. The second file provides the mean optical properties of the dust mixture, and the third file shows the mass fraction represented by each dust population.
- MonoDisc_ds_gridxy.dat & MonoDisc_ds_gridxz.dat are data files in a text format that can be easily plotted. Each file describes the intersection of the dust grid structure with one of the coordinate planes.
- MonoDisc_ds_trhoxy.fits & MonoDisc_ds_trhoxz.fits ("theoretical rho") are single-frame FITS data files, each providing a 1024 x 1024 pixel map of the theoretical dust density in a coordinate plane, across the total extension of the dust grid. These density values are determined from the dust distribution as if there is an infinitely fine dust grid.
- MonoDisc_ds_grhoxy.fits & MonoDisc_ds_grhoxz.fits ("gridded rho") are single-frame FITS data files, each providing a 1024 x 1024 pixel map of the grid-discretized dust density in a coordinate plane, across the total extension of the grid. These density values are read from the finite-resolution dust grid structure constructed for the simulation.

Finally the instrument specified for this tutorial simulation also produces an output file:

- MonoDisc_i88_total.fits (where "i88" is the specified instrument name) is a FITS file containing the data cube with the total flux as detected by the instrument. Since the simulation has only one wavelength, there is only a single frame in the output file.

3.13 Visualizing the results

You can open the ".dat" text files with any text editor. The ".fits" files can be opened with a FITS viewer such as DS9.

The Python Toolkit for SKIRT (PTS) offers a range of capabilities related to SKIRT, including basic visualization of SKIRT results.

```
$ pts
Available scripts:
. . .
  makergbimages
  plotgrids
  plotseds
. . .
  try
$
```

If you enter "pts" without any arguments, and PTS is properly installed, it responds with a list of available scripts.

```
$ pts tr arg1 arg2
Executing: try arg1 arg2
Starting try...
['/Users/pcamps/PTS/svn/src/do/try.py', 'arg1', 'arg2']
Finished try.
$
```

To execute a PTS script named “example”, enter “pts example” on the command line. You can include extra command line arguments as well, as in “pts example arg1 arg2”. You can shorten the script name in the pts command to the first few letters as long as there are no two scripts with matching names. For example “pts exa arg1 arg2” would still execute “example”, assuming that only one script has a name starting with the string “exa”. The above example invokes the trial script provided with PTS.

3.14 Plotting dust grids

When performing a simulation that includes a dust grid, SKIRT outputs some text files with information about the dust grid (you can turn this behavior off when configuring the ski file, but it’s on by default). Each of the generated files contains data for plotting the intersection of the grid with one of the coordinate planes. The number of data files written depends on the dimension of the simulation’s geometry: for spherical symmetry only the intersection with the xy plane is written, for axial symmetry the intersections with the xy and xz planes are written, and for general geometries all three intersections are written. The data files are called `prefix_ds_gridxy.dat`, `prefix_ds_gridxz.dat` and `prefix_ds_gridyz.dat`.

PTS offers a simple way to create PDF files containing these plots for the results of a SKIRT simulation. After the simulation has completed, make sure that the SKIRT output directory is your current directory, and simply enter the command “pts plotgrids”:

```
$ pts plotgrids
Executing: plotgrids
Starting plotgrids...
Created PDF grid plot file ../MonoDisc_ds_gridxy.pdf
Created PDF grid plot file ../MonoDisc_ds_gridxz.pdf
Finished plotgrids.
$
```

The PDF files are placed next to the original data files and have the same name except for the .pdf extension.

3.15 Making RGB images

When performing a simulation, SKIRT outputs a FITS file with surface brightness information for each instrument of the appropriate type (including Frame, Simple and Full instruments). These data files are called `prefix_instrument_total.fits`. Each file contains a data frame per wavelength in the simulation’s wavelength grid. The frames are sorted by increasing wave length.

PTS offers a simple way to create a regular RGB image (in PNG format) for each of these FITS files. By default PTS uses the first FITS frame for the Blue channel, the last FITS frame for the Red channel, and some frame in the middle for the Green channel. This works well for monochromatic simulations (producing a grayscale image) and for oligochromatic simulations with 3 wavelengths.

After the simulation has completed, make sure that the SKIRT output directory is your current directory, and simply enter the command “pts makergb”:

```
$ pts makergb
Executing: makergbimages
Starting makergbimages...
Created RGB image file ../MonoDisc_i88_total.png
Finished makergbimages.
$
```

The PNG files are placed next to the original data files and have the same name except for the .png extension.

3.16 Adjusting the ski file

There is no need to go through the lengthy interactive Q&A process again if you just want to re-run a simulation with slightly adjusted parameters. You can manually edit the ski file instead.

Duplicate the file MonoDisc.ski to a new file named MonoDisc2.ski and open this new file in Qt Creator (or in any decent text editor). It should look similar to the following snippet, where "..." indicates omitted information:

```

. . .
<skirt-simulation-hierarchy type="MonteCarloSimulation" ...>
  <OligoMonteCarloSimulation packages="1e6" ...>
    . . .
    <instrumentSystem type="InstrumentSystem">
      . . .
      <FrameInstrument instrumentName="i88" distance="10 Mpc"
        inclination="88 deg" azimuth="0 deg" positionAngle="0 deg"
        pixelsX="800" extentX="2e4 pc" pixelsY="200" extentY="5000 pc"/>
      . . .
    </instrumentSystem>
    <wavelengthGrid type="OligoWavelengthGrid">
      <OligoWavelengthGrid wavelengths="0.55 micron"/>
    </wavelengthGrid>
    <stellarSystem type="StellarSystem">
      . . .
    </stellarSystem>
    <dustSystem type="OligoDustSystem">
      . . .
    </dustSystem>
  </OligoMonteCarloSimulation>
</skirt-simulation-hierarchy>

```

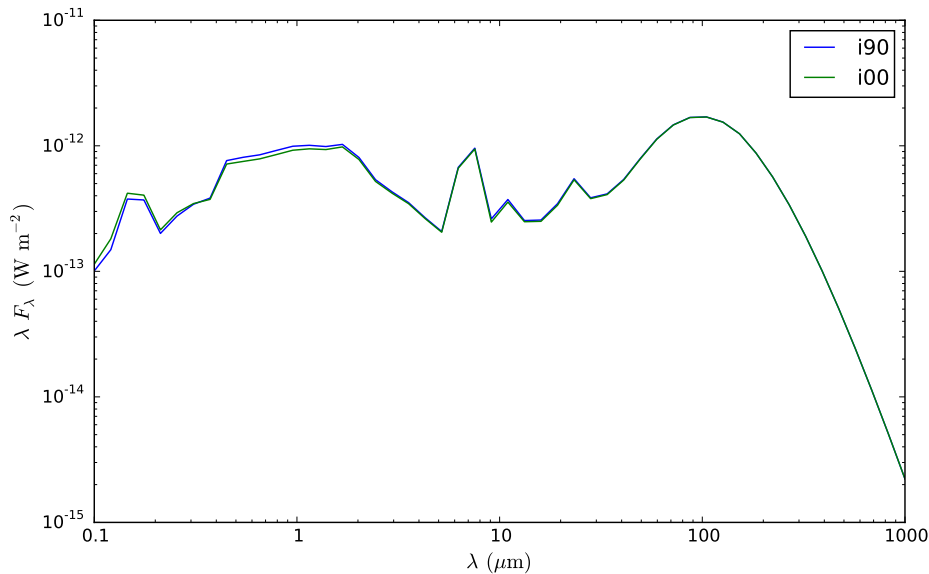
Make the following changes:

- increase the number of photon packages by a factor of five (replace "1e6" by "5e6");
- add a second instrument by duplicating the <FrameInstrument ... /> element;
- change the new instrument's name to i84 and adjust its inclination to 84 degrees.

Save your changes and execute the adjusted simulation using the command "skirt MonoDisc2". The simulation will run substantially slower because you launch more photons and because there are more instruments. Also run the "pts makergb" command again.

Look at the differences between output files (more photon packages; two different inclinations).

4 Producing an SED for SPH simulation results



In this tutorial you will use SKIRT to produce a spectral energy distribution (SED) spanning multiple wavelength bands for a model imported from the output of a Smoothed Particle Hydrodynamic (SPH) simulation.

4.1 Getting ready

To complete this tutorial, you need SPH output data that is converted to a format appropriate for importing in SKIRT. You need a file with stellar particle data and one with gas particle data. The required file formats are described in the documentation for the functions `SPHStellarComp::setFilename()` and `SPHDustDistribution::setFilename()`.

Download the files `eagle_gas.dat` and `eagle_stars.dat` from www.skirt.ugent.be/downloads/ and put them into your `~/SKIRT/run` directory.

4.2 Creating the ski file

```
$ skirt
Welcome to SKIRT v7 (git 475-389e7b7 built on ..)
Running on .. for ..
Interactively constructing a simulation...
? Enter the name of the ski file to be created: PanSPH
```

In a Terminal window, with an appropriate current directory, start SKIRT without any command line arguments. SKIRT responds with a welcome message and starts an interactive session in the terminal window, during which it will prompt you for all the information describing a particular simulation.

The first question is for the filename of the ski file. For this tutorial, enter "PanSPH".

4.3 Type of simulation

```
Possible choices for the simulation:
  1. An oligochromatic Monte Carlo simulation
  2. A panchromatic Monte Carlo simulation
? Enter one of these numbers [1,2] (1): 2
```

In this tutorial the goal is to produce a spectral energy distribution (*SED*) including infrared wavelengths and thus the effects of thermal emission by the dust. By definition this requires a panchromatic simulation.

...

Leave the parameters related to the random generator and the units system to their default values (not shown).

4.4 Instrument system

```
Possible choices for the instrument system:
  1. An instrument system
Automatically selected the only choice: 1
Possible choices for item #1 in the instruments list:
  1. A basic instrument that outputs the total integrated flux as an SED
  ...
? Enter one of these numbers or zero to terminate the list [0,5] (3): 1
? Enter the name for this instrument: i90
? Enter the distance to the system [0 Mpc,∞ Mpc]: 10
? Enter the inclination angle of the detector [0 deg,180 deg] (0 deg): 90
? Enter the azimuth angle of the detector [-360 deg,360 deg] (0 deg):
? Enter the position angle of the detector [-360 deg,360 deg] (0 deg):
Possible choices for item #2 in the instruments list:
  1. A basic instrument that outputs the total integrated flux as an SED
  ...
? Enter one of these numbers or zero to terminate the list [0,5] (3): 1
? Enter the name for this instrument: i00
? Enter the distance to the system [0 Mpc,∞ Mpc]: 10
? Enter the inclination angle of the detector [0 deg,180 deg] (0 deg): 0
? Enter the azimuth angle of the detector [-360 deg,360 deg] (0 deg):
? Enter the position angle of the detector [-360 deg,360 deg] (0 deg):
Possible choices for item #3 in the instruments list:
  1. A basic instrument that outputs the total integrated flux as an SED
  ...
? Enter one of these numbers or zero to terminate the list [0,5] (3): 0
```

For this tutorial, configure two instruments that output the total integrated flux as an SED. Position the first instrument at 0 degrees inclination and the second instrument at 90 degrees, and place both at a distance to the system of 10 Mpc.

```
? Enter the number of photon packages per wavelength [0,1e15] (1e6): 1e5
```

Set the number of photon packages to 10^5 , limiting the run time for this tutorial, while at the same time reducing the quality of the results.

4.5 Wavelength grid

```
Possible choices for the wavelength grid:
  1. A logarithmic wavelength grid
  2. A nested logarithmic wavelength grid
  3. A wavelength grid read from a file
? Enter one of these numbers [1,3] (1): 1
? Do you want to output the wavelength grid points? [yes/no] (no):
? Enter the shortest wavelength [0.0001 micron,1e6 micron]: 0.1
? Enter the longest wavelength [0.0001 micron,1e6 micron]: 1000
? Enter the number of wavelength grid points [3,2000000000] (25): 50
```


SKIRT offers three possibilities: a simple logarithmic grid, a logarithmic grid with a nested logarithmic subgrid and a wavelength grid read from a file. Here are some guidelines for an appropriate choice of the wavelength grid:

- The coverage of your wavelength grid should be appropriate to the radiative transfer problem. Typically radiation is absorbed at UV and optical wavelengths and re-emitted at far-infrared and submillimetre wavelengths. So your wavelength grid should typically range from the UV to the submm range, i.e. from about 0.1 to 1000 μm .
- On the other hand, don't make your wavelength coverage too large. In order to make a meaningful simulation, SKIRT must know the spectral properties of the emitters (the stars) and the optical properties of the dust at all wavelengths considered. Typically, you should not consider wavelengths shorter than 0.01 μm or longer than a few mm. If you choose a dust mixture (see later on) for which the optical properties are not defined over the entire specified wavelength range, SKIRT will not be able to perform the simulation.
- You should consider enough grid points to ensure that you have sufficient resolution and precision. On the other hand, don't choose a grid that is too fine: the simulation run time is approximately proportional to the number of grid points in the wavelength grid. A typical setup uses about 20 grid points for every order in wavelength. If you want much more resolution in a certain wavelength range, for example because you are studying a spectral region in detail, you can use the nested logarithmic grid.
- Whatever you decide to do, always make sure that the wavelength grid includes the V band, since the central wavelength of the V band is where the optical properties of the dust are normalized. If 550 nm is not covered by your wavelength grid, SKIRT will terminate.

For this tutorial, configure a wavelength grid with 50 grid points distributed logarithmically between 0.1 and 1000 μm .

4.6 Stellar system

```

Possible choices for the stellar system:
  1. A stellar system
Automatically selected the only choice: 1
Possible choices for item #1 in the stellar components list:
  1. A stellar component with a built-in geometry (..)
  2. A stellar component derived from an SPH output file
  ...
? Enter one of these numbers [1,5] (1): 2
? Enter the name of the file with the SPH star particles: eagle_stars.dat
? Do you want to output the luminosities per wavelength bin? [yes/no] (no):
Possible choices for item #2 in the stellar components list:
  1. A stellar component with a built-in geometry (..)
  2. A stellar component derived from an SPH output file
  ...
? Enter one of these numbers or zero to terminate the list [0,5] (0): 0

```

In this tutorial, the stellar system contains a single component that imports stellar particle data from the output of an SPH simulation. The file format is described in the documentation for the function `SPHStellarComp::setFilename()`.

For the first and only stellar component, select the appropriate stellar system type and enter the name of your stellar particle data file.

4.7 Dust system

```

Possible choices for the dust system:
  1. A dust system for use with panchromatic simulations
? Enter one of these numbers or zero to select none [0,1] (1):
Possible choices for the dust distribution:
  1. A dust distribution composed of various dust components
  2. A dust distribution derived from an SPH output file
  . . .
? Enter one of these numbers [1,5] (1): 2
? Enter the name of the file with the SPH gas particles: eagle_gas.dat
? Enter the fraction of the metal content locked up in dust grains [0,1] (0.3): 0.3
? Enter the max temperature for a gas particle to contain dust [0 K,1e6 K] (75000 K):

```

For a panchromatic simulation, the only applicable type of dust system is the panchromatic dust system, which supports thermal dust emission in addition to absorption and scattering. The dust system is optional, so you need to explicitly select it.

In this tutorial, the dust distribution is imported from the output of an SPH simulation. The SPH gas particle data format appropriate for importing in SKIRT is described in the documentation for the function `SPHDustDistribution::setFilename()`. Select the appropriate dust distribution type and enter the name of your gas particle data file.

SKIRT estimates the dust density distribution from the gas particle data through a very simple scheme: the dust density is assumed to be a fixed fraction of the density of the metallic gas that does not exceed a given temperature. For this tutorial, leave both options (dust fraction and temperature) at their default values.

```

Possible choices for the dust mix describing the attributes of the dust:
  1. A Draine & Li (2007) dust mix
  2. A dust mix with average interstellar properties
  . . .
 10. A Zubko et al. multi-component dust mix
  . . .
? Enter one of these numbers [1,12] (2): 10
? Do you want to output the optical properties of the dust mix? [yes/no] (yes):
? Do you want to output the mean optical properties of the dust mix? [yes/no] (yes):
? Do you want to output grain size information for the dust mix? [yes/no] (yes):
? Enter the number of graphite subpopulations [1,2000000000] (5):
? Enter the number of silicate subpopulations [1,2000000000] (5):
? Enter the number of PAH subpopulations [1,2000000000] (5):

```

SKIRT supports a set of built-in dust mixtures describing the optical properties of the dust. Several dust populations, each with their own chemical composition and size distribution, can live together in the model. All dust populations have the same spatial distribution, but they may have different optical properties and therefore they will reach different equilibrium temperatures.

For this tutorial, select the Zubko et al. multi-component dust mix. This dust mixture represents a mixture of graphite, silicate and PAH dust grains that reproduces the extinction curve of the Milky Way. For each grain type, the grain size distribution is discretized in a number of size bins, called subpopulations in SKIRT terminology. Leave the output options and the number of subpopulations for each grain type to their default values (although you would need about 15 subpopulation for each grain type to achieve quality results, at the cost of longer execution time).

```

Possible choices for the dust grid structure:
  1. A 3D cartesian grid structure with a linear distribution
  . . .

```

```

5. An octtree dust grid structure
. . .
? Enter one of these numbers [1,8] (5): 5
? Do you want to output for plotting the structure of the grid? [yes/no] (yes):
? Enter the outer radius of the root cell in the x direction [0 pc,∞ pc]: 20 kpc
? Enter the outer radius of the root cell in the y direction [0 pc,∞ pc]: 20 kpc
? Enter the outer radius of the root cell in the z direction [0 pc,∞ pc]: 20 kpc
? Enter the minimum level of grid refinement (typically 2 to 3) [0,50] (2): 2
? Enter the maximum level of grid refinement (typically 6 to 10) [2,50] (6): 7
Possible choices for the search method used for traversing the tree grid:
  1. Top-down (start at root and recursively find appropriate child node)
  2. Neighbor (construct and use neighbor list for each node wall)
  3. Bookkeeping (derive appropriate neighbor through node indices)
? Enter one of these numbers [1,3] (2):
? Enter random density samples for determining cell subdivision [10,1000] (100):
? Enter the maximum mean optical depth for each dust cell [0,1] (0):
? Enter the maximum fraction of dust mass in each dust cell [0,0.01] (1e-6): 5e-5
? Enter the maximum density dispersion fraction in each dust cell [0,0.01] (0):
? Do you want to use barycentric subdivision? [yes/no] (no):

```

Usually you have little a priori knowledge about a dust distribution imported from SPH simulation results. Due to the lack of spatial symmetries, the grid structure that discretizes the dust must be a full 3D grid. Moreover, since we have no a priori information on where the denser regions are located, it is best to choose an adaptive grid that automatically forms smaller cells in denser regions. The grid will take longer to construct, but the simulation itself will run faster and will be more accurate than if you would use a more uniform grid.

For this tutorial simulation, select an octtree grid. The grid must enclose most of the dust in the system so its size must be adjusted to the configuration space of the input data. For the `eagle_xxx.dat` particle data files, you should specify an extent of 20 kpc in each direction.

For a first simulation, typical values for the grid refinement level are a minimum of 2 and a maximum of 7. For high-quality simulations, you will need to raise the maximum level to 10 or more, so that the resulting octtree contains more and smaller cells. Of course this strongly increases execution time.

For this tutorial simulation, the maximum fraction of dust mass in each cell may be set to a value of 5×10^{-5} . For high-quality simulations, this parameter is usually left to the default value of 10^{-6} . Larger (and thus less stringent) values for this parameter result in an octtree with less cells, which lowers accuracy as well as execution time.

Leave the other parameters to their default values.

```

? Enter random density samples for determining cell mass [10,1000] (100):
. . .
? Do you want to include dust self-absorption? [yes/no] (yes): no
. . .

```

As soon as a simulation includes a dust system, dust absorption and scattering are always taken into account. For a panchromatic simulation, thermal emission by the dust is enabled by default. If the simulation's wavelength range includes MIR or FIR bands, dust emission is most likely relevant. If the simulation covers UV and optical bands only, dust emission is most likely irrelevant.

By default the dust emission is calculated under the assumption of local thermal equilibrium (LTE). As an option you can enable non-LTE (NLTE) dust emission which supports emission by stochastically heated dust grains. This is a lot more resource-intensive but produces more realistic results for the NIR and MIR wavelengths.

If dust emission is enabled, you can also enable dust self-absorption, implying a self-consistent calculation of the dust temperature through an iterative (and time-consuming) procedure. The relevance of dust self-absorption depends on the optical depth of the modeled system: if there is a modest amount of dust, the system is transparent to long-wavelength radiation, so self-absorption by dust is negligible. If the system is rather opaque, it will absorb also mid- and far-infrared radiation, so self-absorption might be important.

For this tutorial simulation, leave all options to their default values except for dust self-absorption, which you should disable.

4.8 Running the simulation

After all questions have been answered, SKIRT writes out the resulting ski file and terminates. Start SKIRT again, this time specifying the name of new ski file on the command line, to actually perform the simulation.

4.9 Output files

Most of the output files for this tutorial simulation have already been described for the previous tutorial. A notable exception is the output from the SED instruments:

- PanSPH_i00_sed.dat & PanSPH_i90_sed.dat are short text data files representing the spectral energy distribution of the photon packages detected by each of the instruments; there are two columns (wavelength, flux) that can easily be plotted.

The output files describing sections of the dust density along the coordinate planes now come in triples (xy, xz and yz) because there are no symmetries (i.e. the dust grid is fully 3D). There are also some additional files describing aspects of the dust system that make sense only for panchromatic simulations:

- PanSPH_ds_tempxy.fits, xz & yz ("temperature") are FITS data files, each providing a 1024 x 1024 pixel map of the dust temperature for each subpopulation in the dust mixture in a coordinate plane across the total extension of the dust grid.
- PanSPH_ds_isrfd.dat ("interstellar radiation field") is a text format data file describing the radiation field (flux for every wavelength) for each dust cell in the dust grid.

4.10 Visualizing the results

The visualization capabilities of the Python Toolkit for SKIRT (PTS) were introduced for the previous tutorial. To plot the grid used for the current tutorial without replotting the grid for the previous tutorial (residing in the same directory), add the name of the ski file as command line argument after the script name:

```
$ pts plotgrids PanSPH
Executing: plotgrids
Starting plotgrids...
Created PDF grid plot file ../PanSPH_ds_gridxy.pdf
Created PDF grid plot file ../PanSPH_ds_gridxz.pdf
Created PDF grid plot file ../PanSPH_ds_gridyz.pdf
Created PDF grid plot file ../PanSPH_ds_gridxyz.pdf
Finished plotgrids.
$
```

4.11 Plotting SEDs

When performing a panchromatic simulation, SKIRT outputs a text file with SED information for each instrument of the appropriate type (including SED, Simple and Full instruments). These data files are called `prefix_instrument_sed.dat`.

PTS offers a simple way to create a PDF file with a single plot showing these SEDs. After the simulation has completed, make sure that the SKIRT output directory is your current directory, and simply enter the command "pts plotseds":

```
$ pts plotseds
Executing: plotseds
Starting plotseds...
Created PDF SED plot file ../PanSPH_sed.pdf
Finished plotseds.
$
```

The PDF file is placed next to the original data files and has a similar name (the instrument name is removed and the filename extension is .pdf).

4.12 Enabling stochastic heating

You can easily adjust the configuration to enable stochastic heating of dust grains, and run the simulation again. Duplicate the file `PanSPH.ski` to a new file named `PanSPH2.ski` and open this new file in Qt Creator (or in any decent text editor). Look for the section near the end that describes the dust emissivity choice:

```
. . .
<dustEmissivity type="DustEmissivity">
  <GreyBodyDustEmissivity/>
</dustEmissivity>
<dustLib type="DustLib">
  <AllCellsDustLib/>
</dustLib>
. . .
```

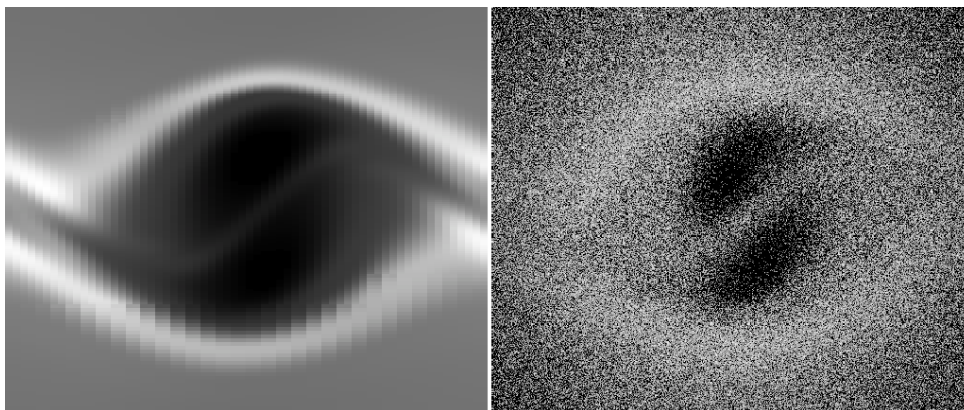
Make the following changes:

- replace the keyword "GreyBodyDustEmissivity" by "TransientDustEmissivity" to calculate dust emission with support for stochastically heated dust grains;
- replace the keyword "AllCellsDustLib" by "Dim1DustLib entries="20000"" to perform the calculation on a number of representative radiation fields rather than on each and every dust cell, saving a lot of processing time at the cost of lesser accuracy.

Save your changes and execute the adjusted simulation using the command "skirt PanSPH2". The simulation will now calculate the dust emission using a more time-consuming algorithm. After SKIRT finishes, also run the "pts plotseds" command again.

Evaluate the differences between the SEDs from the first simulation (which assumes local thermal equilibrium) and the second simulation (which supports stochastic heating).

5 Scattering from dust in a Kelvin-Helmholtz instability



In this tutorial you will use SKIRT to study the dust in a Kelvin-Helmholtz instability as it might occur in a molecular cloud. The dust distribution has been generated by a hydrodynamical simulation on an adaptive mesh (AMR = Adaptive Mesh Refinement) with the MPI-AMRVAC code. The simulation includes a gas component and two dust components with different characteristics. The simulation results are stored as a single data file in a binary format specific to MPI-AMRVAC. In addition to a more generic AMR import format using text columns, SKIRT can also directly import the MPI-AMRVAC binary format.

5.1 Getting ready

To complete this tutorial, you need the output file produced by the MPI-AMRVAC simulation. Download the file `kh_amr.dat` from www.skirt.ugent.be/downloads/ and put it into your `~/SKIRT/run` directory.

5.2 Creating the ski file

```
$ skirt
Welcome to SKIRT v7 (git 475-389e7b7 built on ..)
Running on .. for ..
Interactively constructing a simulation...
? Enter the name of the ski file to be created: ScatAMR
```

In a Terminal window, with an appropriate current directory, start SKIRT without any command line arguments. SKIRT responds with a welcome message and starts an interactive session in the terminal window, during which it will prompt you for all the information describing a particular simulation. The first question is for the filename of the ski file. For this tutorial, enter "ScatAMR".

5.3 Type of simulation

```
Possible choices for the simulation:
  1. An oligochromatic Monte Carlo simulation
  2. A panchromatic Monte Carlo simulation
? Enter one of these numbers [1,2] (1): 1
```

In an actual research setting you would probably run a panchromatic simulation to study the absorption, scattering and thermal emission by the dust over a range of wavelengths. In this tutorial you will simply study scattering by the dust at a single V-band wavelength. An oligochromatic simulation is sufficient and most performance-effective for these purposes.

Leave the option related to the random generator to its default value (not shown).

5.4 Units

Possible choices for the units system:

1. SI units
2. Stellar units (length in AU, distance in pc)
3. Extragalactic units (length in pc, distance in Mpc)

? Enter one of these numbers [1,3] (3): 2

The spatial scale of the dust distribution used in this tutorial is such that it seems best to select stellar units. While you can still use any of the supported units to enter parameter values, this means that all output values will be presented in stellar units. Leave the option related to flux output style to its default value (not shown).

5.5 Instrument system

Possible choices for the instrument system:

1. An instrument system

Automatically selected the only choice: 1

Possible choices for item #1 in the instruments list:

1. A basic instrument that outputs the total integrated flux as an SED
2. A basic instrument that outputs the total flux in every pixel as a data cube
3. A basic instrument that outputs the total flux as data cube and as SED
4. An advanced instrument that records individual contributions to the flux
- . . .

? Enter one of these numbers or zero to terminate the list [0,6] (3): 4

? Enter the name for this instrument: xy

? Enter the distance to the system [0 pc,∞ pc]: 1

? Enter the inclination angle of the detector [0 deg,180 deg] (0 deg):

? Enter the azimuth angle of the detector [-360 deg,360 deg] (0 deg):

? Enter the position angle of the detector [-360 deg,360 deg] (0 deg): 90

? Enter the number of pixels in the horizontal direction [25,10000] (250): 500

? Enter the maximal horizontal extent [0 AU,∞ AU]: 0.0315421e18 cm

? Enter the number of pixels in the vertical direction [25,10000] (250): 500

? Enter the maximal vertical extent [0 AU,∞ AU]: 0.03e18 cm

? Enter the number of scattering levels to be recorded individually [0,25] (0):

Possible choices for item #2 in the instruments list:

. . .

Enter one of these numbers or zero to terminate the list [0,6] (3): 0

Since you want to study the scattered light separately, you need to select the advanced instrument that records individual contributions to the flux. Position the instrument so that it shows a projection of the xy coordinate plane (angles: 0, 0, 90). This facilitates visual comparison of the recorded fluxes with the corresponding cut through the dust density automatically produced by SKIRT. The distance of the instrument to the model affects all fluxes with the same factor; use a value of your liking (for example 1 pc).

Setup a pixel resolution of your liking (for example 500×500 pixels) and provide a frame extent that matches the extent of the dust distribution to be imported. For the AMR file used in this tutorial, the dust distribution has extent (0.0315421, 0.03, 0.03) in units of 10^{18} cm. Since SKIRT understands centimeters, there is no need for manual conversion of these values.

Note: a SKIRT instrument frame is a rectangle centered on the origin and an imported AMR dust distribution is a cuboid centered on the origin. In both cases the extent specifies half the total size along one of the coordinate axes.

Leave the number of photon packages to the default value of 10^6 (not shown).

5.6 Wavelength grid

```

Possible choices for the wavelength grid:
  1. A list of one or more distinct wavelengths
Automatically selected the only choice: 1
? Enter the wavelengths [0.0001 micron,1e6 micron]: 0.55

```

For this tutorial enter a wavelength "list" containing the single value $\lambda = 0.55 \mu\text{m}$ (V-band).

5.7 Stellar system

```

Possible choices for the stellar system:
  1. A stellar system
Automatically selected the only choice: 1
Possible choices for item #1 in the stellar components list:
  1. A stellar component with a built-in geometry (..)
  ...
? Enter one of these numbers [1,5] (1): 1
Possible choices for the geometry of the spatial stellar distribution:
  1. A point source geometry
  2. A Plummer geometry
  3. A gamma geometry
  ...
? Enter one of these numbers [1,39] (1): 2
? Enter the scale length [0 AU,∞ AU]: 0.03e18 cm
? Enter the luminosities, one for each wavelength, in solar units [0,∞]: 1e-6
Possible choices for item #2 in the stellar components list:
  ...
? Enter one of these numbers or zero to terminate the list [0,5] (0): 0

```

Since the data file provided for this tutorial has no information on light sources, you need to provide an artificial "stellar system" to illuminate the dust. One option is to use a Plummer profile with a scale length comparable to the size of the dust distribution; this creates a fairly constant source of light across the configuration space. Specify a total luminosity consistent with an artificial background source, for example 10^{-6} solar luminosities.

5.8 Dust system

```

Possible choices for the dust system:
  1. A dust system for use with oligochromatic simulations
? Enter one of these numbers or zero to select none [0,1] (1): 1
Possible choices for the dust distribution:
  1. A dust distribution composed of various dust components
  2. A dust distribution derived from an SPH output file
  3. A dust distribution imported from an adaptive mesh data file
  ...
? Enter one of these numbers [1,5] (1): 3

```

For this tutorial, select an oligochromatic dust system (the only applicable type with an oligochromatic simulation) with a dust distribution imported from an adaptive mesh data file.

```

Possible choices for the adaptive mesh data file:
  1. An adaptive mesh data file in ASCII format
  2. An adaptive mesh data file in MPI-AMRVAC format

```



```

? Enter one of these numbers [1,2] (1): 2
? Enter the name of the adaptive mesh data file: kh_amr.dat
? Enter the mesh cells at the coarsest level, X direction [1,10000]: 32
? Enter the mesh cells at the coarsest level, Y direction [1,10000]: 128
? Enter the mesh cells at the coarsest level, Z direction [1,10000]: 32
? Enter the units for density values [0 Msun/AU3,∞ Msun/AU3]: 1e-21 g/cm3
? Enter the outer radius in the x direction [0 AU,∞ AU]: 0.0315421e18 cm
? Enter the outer radius in the y direction [0 AU,∞ AU]: 0.03e18 cm
? Enter the outer radius in the z direction [0 AU,∞ AU]: 0.03e18 cm

```

Select the MPI-AMRVAC format and enter the appropriate filename (kh_amr.dat). You must also provide some information that is not included in the AMR data file: the number of mesh cells at the coarsest mesh level in each direction; the spatial extent of the mesh in each direction; and the units of the density values stored in the file.

For the data file used in this tutorial, the number of mesh cells at the coarsest level is (32,128,32), the dust distribution has extent (0.0315421,0.03,0.03) in units of 10^{18} cm, and the density values in the file are given in units of 10^{-21} g cm $^{-3}$. Since SKIRT understands these units, there is no need for manual conversion.

```

Possible choices for item #1 in the dust components list:
  1. A mesh dust component
Automatically selected the only choice: 1
? Enter the column defining the density distribution [0,99] (0): 5
? Enter the column defining an extra multiplication factor, or -1 [-1,99] (-1):
? Enter the fraction of the density actually locked up in dust grains [0,1] (1):
Possible choices for the dust mixture for the dust component:
  1. A Draine & Li (2007) dust mix
  2. A dust mix with average interstellar properties
  . . .
? Enter one of these numbers [1,12] (2):
? Output the optical properties of the dust mix? [yes/no] (yes): no
? Output the mean optical properties of the dust mix? [yes/no] (yes): no

Possible choices for item #2 in the dust components list:
  1. An mesh dust component
? Enter one of these numbers or zero to terminate the list [0,1] (1): 1
? Enter the column defining the density distribution [0,99] (0): 6
? Enter the column defining an extra multiplication factor, or -1 [-1,99] (-1):
? Enter the fraction of the density actually locked up in dust grains [0,1] (1):
Possible choices for the dust mixture for the dust component:
  1. A Draine & Li (2007) dust mix
  2. A dust mix with average interstellar properties
  . . .
? Enter one of these numbers [1,12] (2):
? Output the optical properties of the dust mix? [yes/no] (yes): no
? Output the mean optical properties of the dust mix? [yes/no] (yes): no

Possible choices for item #3 in the dust components list:
  1. An adaptive mesh dust component
? Enter one of these numbers or zero to terminate the list [0,1] (1): 0

```

Configure the dust components for which the AMR data file defines a density distribution. The tutorial file contains two dust density distributions given respectively by the variables (or columns) with index 5 and 6. There is no extra multiplication factor or fraction, so you can leave these parameters to their default value. Also, select the default dust mix (with average interstellar properties) for both dust components. In a realistic research setting, you would probably specify different characteristics for each dust component.

Possible choices for the dust grid structure:

1. A 3D cartesian grid structure with a linear distribution
 - ...
 9. A dust grid structure based on the adaptive mesh dust geometry
- ? Enter one of these numbers [1,9] (5): 9

SKIRT discretizes the dust on a grid structure, i.e. a collection of small cells in which properties such as dust density and temperature are considered to be constant. You could select any of the 3D dust grids offered by SKIRT. The octree grid for example builds a grid adapted to the dust density distribution based on some given parameters. When importing a dust distribution from an adaptive mesh data file, SKIRT also offers the option to directly use the adaptive mesh (defined by the data file) as a dust grid. The dust grid then exactly mirrors the resolution structure of the hydrodynamical simulation, which seems a natural thing to do. On the other hand, in some cases building a new grid (perhaps with less cells) may prove meaningful to reduce the runtime of the SKIRT simulation.

For this tutorial select the option to use the imported adaptive mesh as a dust grid.

Finally the dust system takes a few additional parameters which can be left to their default value (not shown).

5.9 Running the simulation

After all questions have been answered, SKIRT writes out the resulting ski file and terminates. Start SKIRT again, this time specifying the name of new ski file on the command line, to actually perform the simulation.

5.10 Output files

Most of the output files have already been described for one of the previous tutorials. The instrument used in this tutorial does produce some additional files to record individual flux components:

- ScatAMR_xy_total.fits contains the total flux detected by the instrument (i.e. the sum of the direct and scattered flux).
- ScatAMR_xy_direct.fits contains the flux resulting from stellar photon packages that directly reach the instrument without being scattered.
- ScatAMR_xy_scattered.fits contains the flux resulting from stellar photon packages that were scattered by the dust before reaching the instrument.
- ScatAMR_xy_transparent.fits contains the total flux that would be detected by the instrument if there were no dust in the system.

5.11 Verifying results

The file ScatAMR_ds_trhoxy.fits shows a cut through the total dust density along the xy coordinate pane. It should resemble the left panel of the illustration at the beginning of this tutorial.

The file ScatAMR_xy_scattered.fits shows the flux resulting from photon packages that were scattered by the dust (i.e. excluding any photon packages that reach the instrument without being scattered). It should resemble the right panel of the illustration at the beginning of this tutorial.

6 Adding a new geometry

In this tutorial, you will extend the SKIRT code with a new geometry that can be used for stellar and dust density distributions.

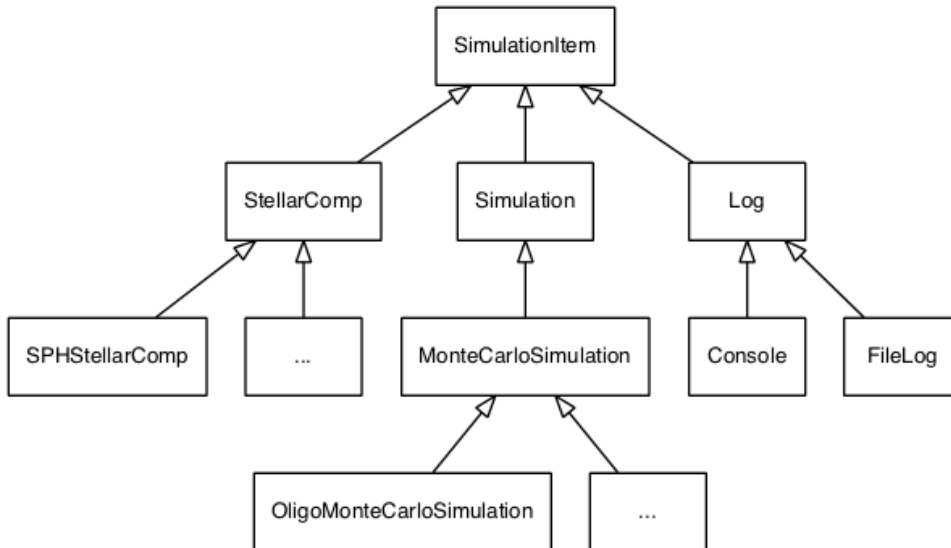
6.1 Structure of the SKIRT code

At run-time, a complete SKIRT simulation is represented in memory as a hierarchy of objects (instances of various C++ classes) called *simulation items*. The following diagram illustrates a simple run-time simulation hierarchy (a connection starting with a diamond loosely means "A owns B").



A run-time simulation hierarchy represents the configuration of the simulation, including all parameters and properties, mimicking the contents of the ski file from which it was constructed. The objects in the hierarchy also manage the data used or generated during the simulation.

Each simulation item is an instance of some C++ class that inherits the `SimulationItem` class. The diagram on the following page shows a small part of the inheritance tree for the `SimulationItem` class. The leaf nodes represent concrete simulation item classes. Instances of these classes can be part of a run-time simulation hierarchy. The non-leaf nodes represent abstract simulation item classes that can't be instantiated.



In conclusion, simulation items form a *compile-time* class hierarchy through inheritance (with the `SimulationItem` class at the top), and a *run-time* object hierarchy through ownership (with an instance of a `Simulation` subclass at the top).

The various classes in the SKIRT source code are distributed over several subprojects, each residing in its own directory. For our purposes, the SKIRTcore subproject is the most important one. It contains the core functionality for setting up and performing a simulation, including all simulation item classes.

6.2 Scoping out the work

The new geometry that you will implement is an isothermal disk with the axisymmetric density profile

$$\rho(\mathbf{r}) \equiv \rho(R, z) = \rho_0 \exp\left(-\frac{R}{h_R}\right) \operatorname{sech}^2\left(\frac{z}{h_z}\right)$$

with the scale length h_R and the scale height h_z as two free parameters.

To add a new geometry, you need to create a new class and tie it into the simulation item class hierarchy. Rather than reinventing the wheel, browse the SKIRT documentation and look for an existing class with properties similar to your needs. It so happens that the `ExpDiskGeometry` class also defines an axisymmetric dust disk. To examine it more closely, open the `ExpDiskGeometry.hpp` header file in the code editor.

It appears that `ExpDiskGeometry` inherits from `SepAxGeometry`, an abstract class representing axisymmetric dust geometries where the density is a separable function of R and z . `ExpDiskGeometry` offers two properties h_R and h_z corresponding to the free parameters you need, plus two properties related to truncation of the disk which you don't need but can easily be removed. There is also a data member for the central density ρ_0 .

Looking in some more detail, `ExpDiskGeometry` offers the following functions:

- a default constructor
- the `setupSelfBefore()` function, which calculates the central density
- setters and getters for the model parameters
- the functions `density()`, `randomR()`, `randomz()`, `SigmaR()`, and `SigmaZ()` that define various aspects of the geometry modeled by this class

Clearly you should use the same structure for your new class. The main work will consist of adjusting the key functions for the new geometry.

6.3 The mathematics

6.3.1 Density

The Geometry class documentation states that all geometries in SKIRT must be normalized so that the total mass is equal to one. Given this normalization, we can determine the central density for the new geometry as a function of h_R and h_z . We find

$$M = \iiint \rho(\mathbf{r}) \, d\mathbf{r} = 2\pi \rho_0 \int_0^\infty \exp\left(-\frac{R}{h_R}\right) R \, dR \int_{-\infty}^\infty \operatorname{sech}^2\left(\frac{z}{h_z}\right) \, dz.$$

Using the equalities

$$\int_0^\infty x e^{-x} \, dx = \int_0^\infty \operatorname{sech}^2 x \, dx = 1$$

we find that

$$M = 4\pi h_R^2 h_z \rho_0$$

and hence (using $M = 1$)

$$\rho_0 = \frac{1}{4\pi h_R^2 h_z}.$$

The density $\rho(R, z)$ is then easily calculated using its original definition above.

6.3.2 Surface densities

We also need to calculate the edge-on and face-on surface densities Σ_R and Σ_Z . The documentation for the `AxGeometry::SigmaR()` function defines the edge-on or radial surface density as the integration of the density along a line in the equatorial plane starting at the center of the coordinate system,

$$\Sigma_R = \int_0^\infty \rho(R, 0) \, dR.$$

The documentation for the `Geometry::SigmaZ()` function defines the face-on or Z-axis surface density as the integration of the density along the entire Z-axis,

$$\Sigma_Z = \int_{-\infty}^\infty \rho(0, z) \, dz.$$

In our present case, we find

$$\Sigma_R = \rho_0 \int_0^\infty \exp\left(-\frac{R}{h_R}\right) \, dR = \rho_0 h_R$$

and

$$\Sigma_Z = \rho_0 \int_{-\infty}^\infty \operatorname{sech}^2\left(\frac{z}{h_z}\right) \, dz = 2\rho_0 h_z.$$

6.3.3 Random position

Finally a geometry must be able to draw a random position from its three-dimensional probability density distribution. For an axisymmetric geometry with separable density function, the problem is reduced to drawing two values, each from a one-dimensional probability distribution, in the radial and vertical directions.

The documentation for the `SepAxGeometry::randomR()` function states that it returns a random R from the one-dimensional probability distribution

$$p(R) \, dR = 2\pi \rho_R(R) R \, dR.$$

In the present case we find

$$p(R) dR = \frac{\rho(R, z) R dR}{\int_0^\infty \rho(R', z) R' dR'} = \frac{1}{h_R^2} \exp\left(-\frac{R}{h_R}\right) R dR$$

with cumulative distribution

$$P(R) = \int_0^R p(R') dR' = 1 - \left(1 + \frac{R}{h_R}\right) \exp\left(-\frac{R}{h_R}\right).$$

Solving $\mathcal{X} = P(R)$ for R , where \mathcal{X} is a uniform random deviate, we obtain

$$R = h_R \left[-1 - W_{-1} \left(\frac{\mathcal{X} - 1}{e} \right) \right]$$

where W_{-1} is the Lambert function of order -1 , which can be calculated in SKIRT using the function `SpecialFunctions::LambertW1`.

The documentation for the `SepAxGeometry::randomz()` function states that it returns a random z from the one-dimensional probability distribution

$$p(z) dz = \rho_z(z) dz.$$

In the present case we find

$$p(z) dz = \frac{\rho(R, z) dz}{\int_{-\infty}^\infty \rho(R, z') dz'} = \frac{1}{2 h_z} \operatorname{sech}^2\left(\frac{z}{h_z}\right) dz$$

with cumulative distribution

$$P(z) = \int_{-\infty}^z p(z') dz' = \frac{1}{2} \left[1 + \tanh\left(\frac{z}{h_z}\right) \right].$$

Solving $\mathcal{X} = P(z)$ for z , where \mathcal{X} is a uniform random deviate, we obtain

$$z = h_z \operatorname{artanh}(2\mathcal{X} - 1) = \frac{1}{2} h_z \ln \frac{\mathcal{X}}{1 - \mathcal{X}},$$

where we used the identity

$$\operatorname{artanh} x = \frac{1}{2} \ln \frac{1 + x}{1 - x}.$$

6.4 Creating the new source and header files

It is quite possible to create a new class from within the Qt Creator development environment. However since your new class will so closely resemble an existing one, it is easier to copy the header and source files using a regular file browser (or if you prefer, command lines in a terminal window). Locate the files `ExpDiskGeometry.hpp` and `ExpDiskGeometry.cpp` in your local working copy of the SKIRT source code (in the `git/SKIRTcore` directory), and copy them to a set of new files named `IsoThermalDiskGeometry.hpp` and `IsoThermalDiskGeometry.cpp` (in the same directory). Do not remove the original files!

Go back to Qt Creator (in "Edit" mode) and right-click on the SKIRTcore project icon in the "Projects" pane. In the popup-menu, choose "Add Existing Files..." (sometimes you need to right-click a second time to make this menu item appear). In the dialog, locate and select the two new source files and press the "Open" button. In the next (smaller) dialog, confirm that you want to add the new files to source control. Now locate the new files in Qt Creator's project browser and open them in the editor.

In the new header file, replace the three occurrences of `"EXPDISKGEOMETRY_HPP"` by `"ISOTHERMALDISKGEOMETRY_HPP"`. This is the standard header guard to avoid multiple includes of the same class definition.

In both new files, replace all occurrences of "ExpDiskGeometry" by "IsoThermalDiskGeometry".

Press the "Hammer" icon in the left margin to save your changes and perform a build. There should be no errors.

6.5 The class declaration

Since the class declaration for IsoThermalDiskGeometry is very similar to that of ExpDiskGeometry, the header file IsoThermalDiskGeometry.hpp for your new class does not need a lot of work.

Most importantly, adjust the description of the class in the first occurrence of the Q_CLASSINFO() macro. For example, replace "an exponential disk geometry" by "an iso-thermal disk geometry". This text will be used for identifying your new geometry to a user, such as when constructing a new .ski file during an interactive Q&A session.

You should also remove the getters and setters for the truncation parameters, the corresponding Q_CLASSINFO() macros, and the corresponding data members, since these parameters are not needed in the new class.

While updating the documentation, take care to follow the rules for comments in "/ **"-style documentation blocks. These comments are processed by the Doxygen utility to produce the SKIRT manual in HTML and PDF formats. It is really important to write decent comments for every function.

6.6 The class definition

The source file IsoThermalDiskGeometry.cpp needs more extensive edits. The constructor should initialize all data members to some default value, without performing extensive calculations. Often the constructor simply sets all data members to zero. Since you removed the truncation-related data members declared for the class, you need to remove their initialization from the constructor as well:

```

IsoThermalDiskGeometry::IsoThermalDiskGeometry()
    : _hR(0), _hz(0), _rho0(0)
{
}

```

Note that there is no destructor. This is fine since the class does not allocate any resources that need deallocation.

The implementation of the function setupSelfBefore() must always start with a call to the same function in the base class (SepAxGeometry in this case). This call can remain unchanged. Remove the code that deals with the truncation-related properties; other than that everything can stay unchanged. The allowed range for the scale height and scale length is the same as for the exponential disk, and the central density happens to be the same as for the untruncated exponential disk:

```

void IsoThermalDiskGeometry::setupSelfBefore()
{
    SepAxGeometry::setupSelfBefore();
    // verify property values
    if (_hR <= 0) throw FATALERROR("Radial scale length hR should be positive");
    if (_hz <= 0) throw FATALERROR("Axial scale height hz should be positive");
    // calculate central density
    _rho0 = 1.0/(4.0*M_PI*_hz*_hR*_hR);
}

```

Update the implementation of the functions `density()`, `sigmaR()`, and `sigmaZ()` to reflect the (surface) density formulas derived before:

```
double IsoThermalDiskGeometry::density(double R, double z) const
{
    double sechz = 1.0 / cosh(z/_hz);
    return _rho0 * exp(-R/_hR) * sechz*sechz;
}
double IsoThermalDiskGeometry::sigmaR() const
{
    return _rho0*_hR;
}
double IsoThermalDiskGeometry::sigmaZ() const
{
    return 2.0*_rho0*_hz;
}
```

Update the implementation of the functions `randomR()` and `randomz()` to reflect the formulas derived before for generating the appropriate random values:

```
double IsoThermalDiskGeometry::randomR() const
{
    double X = _random->uniform();
    return _hR * (-1.0-SpecialFunctions::LambertW1((X-1.0)/M_E));
}
double IsoThermalDiskGeometry::randomz() const
{
    double X = _random->uniform();
    return 0.5 * _hz * log(X/(1.0-X));
}
```

Press the "Hammer" icon again to save your changes and perform a build. There should be no errors.

6.7 Registering the new class

Having declared and defined the new class, you must make its existence known to the SKIRT discovery mechanism so that it will be properly listed when constructing a new `.ski` file during an interactive Q&A session. Note that this registration requirement applies only simulation item classes, i.e. classes that inherit directly or indirectly from the `SimulationItem` class, as is the case for your new `IsoThermalDiskGeometry` class.

Locate the file `RegisterSimulationItems.cpp` (in the `git/Discover` directory) and open it in the editor. You need to add two lines in this file: an include directive for the new class header, and an `add<>` statement to register the new class.

Add the include directive to the already long list of include directives, in alphabetical order. For example:

```
...
#include "InterstellarDustMix.hpp"
#include "IsoThermalDiskGeometry.hpp"
...
```

Insert a new `add<>` statement to the existing list in the location you want it to appear in the corresponding choice list while creating a new `.ski` file in the interactive Q&A session. In this case, locate the list of `add<>` statements for geometry classes and insert a new statement (for example) right after the one for the `ExpDiskGeometry` class:


```

. . .
add<ExpDiskGeometry>();
add<IsoThermalDiskGeometry>();
. . .

```

Press the "Hammer" icon again to save your changes and perform a build. Assuming there are no errors, SKIRT is now ready for use with the new geometry.

6.8 Using the new class

Launch the newly built SKIRT version without any command line arguments. Provide appropriate responses to the interactive questions for a simulation with a stellar system or a dust distribution composed of various components. The list of choices will include a new item for the iso-thermal disk geometry:

```

. . .
8. A shell geometry
9. An exponential disk geometry
10. An iso-thermal disk geometry
11. A multi-gaussian expansion (MGE) geometry
. . .

```

Enter the appropriate choice (10 in the example) to select your new geometry. Complete the Q&A session to save the ski file. Alternatively, you can edit a copy of a ski file you made earlier to use your new geometry.

To verify the results for a dust density distribution, set the writeDensity property of the dust system to "true", so that SKIRT generates cuts through the dust along the coordinate axes (in the output files prefix_ds_trhoxy.fits and prefix_ds_trhoz.fits).

To verify the results for a stellar distribution, run a simulation without dust and with appropriate frame instruments to generate an image of the flux received in a few relevant directions.

The illustration below shows the exponential disk in the top row and the isothermal disk in the bottom row. The left columns shows a dust density cut along the xz-plane; the right column shows the edge-on stellar flux. All figures use a color scale to indicate intensity.

