

First run of SKIRT

Dr. Bert Vander Meulen
SKIRT workshop Kyoto 2026
14/05/2026

1. The **SKIRT** radiation transfer simulation **code** (hopefully installed):
 - **Command line program** (you run it from the terminal):
 - **Interactive mode**: Set up the simulation (**Q&A** to create a **.ski file**)
 - **Non-interactive mode**: Run the simulation (**run** the **.ski file**)
 - **Setting up a SKIRT simulation = creating a .ski parameter file**



.ski file: the parameter file describing the SKIRT simulation

- The **full description** of a single **SKIRT simulation** is stored in a **.ski parameter file**
- **SKIRT** offers an **interactive Q&A** to set up the simulation & create a **.ski file**
- To start the **Q&A session** :
 - Invoke **skirt** without arguments in a terminal window:

```
(base) bert.vandermeulen@EST01114502 run % skirt
13/05/2026 13:59:43.015 Welcome to SKIRT v9.0 (git d762d4b built on 13/05/2026 at 13:56:33)
13/05/2026 13:59:43.020 Running on EST01114502 for bert.vandermeulen
13/05/2026 13:59:43.199 Interactively constructing a simulation...
13/05/2026 13:59:43.200 ? Enter the name of the ski file to be created: example
```

- The **full description** of a single **SKIRT simulation** is stored in a **.ski parameter file**
- **SKIRT** offers an **interactive Q&A** to set up the simulation & create a **.ski file**
- To start the **Q&A session** :

- Invoke **skirt** without arguments in a terminal window:

```
(base) bert.vandermeulen@EST01114502 run % skirt
13/05/2026 13:59:43.015 Welcome to SKIRT v9.0 (git d762d4b built on 13/05/2026 at 13:56:33)
13/05/2026 13:59:43.020 Running on EST01114502 for bert.vandermeulen
13/05/2026 13:59:43.199 Interactively constructing a simulation...
13/05/2026 13:59:43.200 ? Enter the name of the ski file to be created: example
```

___More questions___

```
13/05/2026 14:06:22.865 Successfully created ski file 'example.ski'.
13/05/2026 14:06:22.865 To run the simulation use the command: skirt example
```

- If SKIRT is not in your system path, you must provide the path of the skirt executable, e.g.:

```
(base) bert.vandermeulen@EST01114502 run % ~/SKIRT/release/SKIRT/main/skirt
```

- The **full description** of a single **SKIRT simulation** is stored in a **.ski parameter file**
- **SKIRT** offers an **interactive Q&A** to set up the simulation & create a **.ski file**
- To start the **Q&A session** :
 - Invoke **skirt** without arguments in a terminal window
 - **Questions** and **errors** stand out:

```
13/05/2026 14:05:24.100 ? Enter the scale length ]0 pc,∞ pc[:  
13/05/2026 14:05:27.824 * Enter a valid floating point number  
13/05/2026 14:05:27.824 ? Enter the scale length ]0 pc,∞ pc[: 1
```

- The **full description** of a single **SKIRT simulation** is stored in a **.ski parameter file**
- **SKIRT** offers an **interactive Q&A** to set up the simulation & create a **.ski file**
- To start the **Q&A session** :
 - Invoke **skirt** without arguments in a terminal window
 - **Questions** and **errors** stand out
 - Accepted range of values **[minValue, maxValue]** and **(default)**:

```
13/05/2026 14:06:10.000 ? Enter the longest wavelength [1e-6 micron,1e6 micron]: 1 micron
13/05/2026 14:06:15.516 ? Enter one of these numbers or zero to terminate the list [0,6] (1):
```

- The **full description** of a single **SKIRT simulation** is stored in a **.ski parameter file**
- **SKIRT** offers an **interactive Q&A** to set up the simulation & create a **.ski file**
- To start the **Q&A session** :
 - Invoke **skirt** without arguments in a terminal window
 - **Questions** and **errors** stand out
 - Accepted range of values **[minValue, maxValue]** and **(default)**
 - For values with **units**: Default unit is mentioned in range. Or use custom unit (see [Supported Units](#)):

```
13/05/2026 14:06:10.000 ? Enter the longest wavelength [1e-6 micron,1e6 micron]: 1 micron
13/05/2026 14:06:10.000 ? Enter the longest wavelength [1e-6 micron,1e6 micron]: 1
13/05/2026 14:06:10.000 ? Enter the longest wavelength [1e-6 micron,1e6 micron]: 1e-6 m
```

- The **full description** of a single **SKIRT simulation** is stored in a **.ski parameter file**
- **SKIRT** offers an **interactive Q&A** to set up the simulation & create a **.ski file**
- To start the **Q&A session** :
 - Invoke **skirt** without arguments in a terminal window
 - **Questions** and **errors** stand out
 - Accepted range of values **[minValue, maxValue]** and **(default)**
 - For values with **units**: Default unit is mentioned in range. Or use custom unit (see [Supported Units](#))
 - If all went well: “Successfully created ski file” & and you can open the .ski file in a text editor

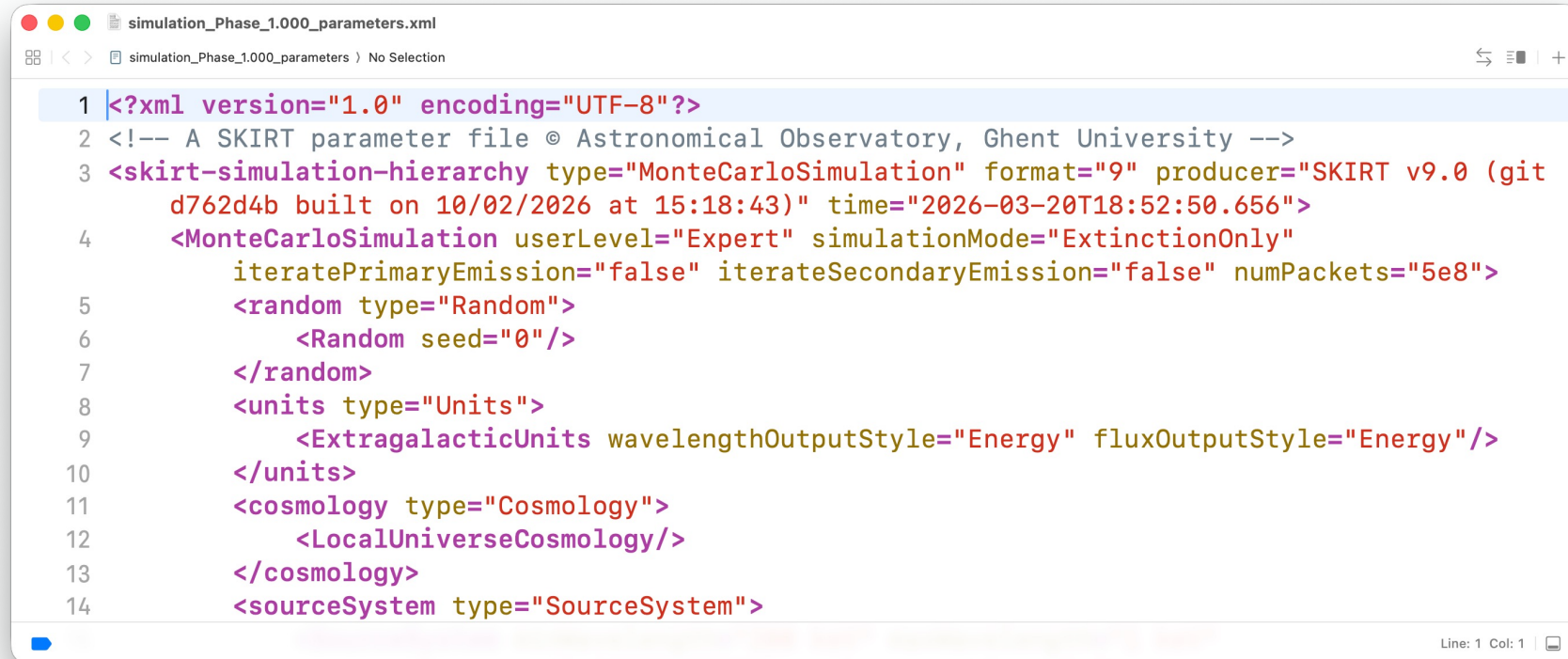
```
(base) bert.vandermeulen@EST01114502 run % skirt
13/05/2026 13:59:43.015 Welcome to SKIRT v9.0 (git d762d4b built on 13/05/2026 at 13:56:33)
13/05/2026 13:59:43.020 Running on EST01114502 for bert.vandermeulen
13/05/2026 13:59:43.199 Interactively constructing a simulation...
13/05/2026 13:59:43.200 ? Enter the name of the ski file to be created: example
```

___More questions___

```
13/05/2026 14:06:22.865 Successfully created ski file 'example.ski'.
13/05/2026 14:06:22.865 To run the simulation use the command: skirt example
```

The .ski parameter file

- We can have a look at the created .ski file
- The **full description** of a single **SKIRT** simulation is stored in a **.ski parameter file**
 - Follows the **XML** format (largely **self-documenting**)
 - Straightforward to **adjust a parameter value** or add component w/ a **text editor**:



```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!-- A SKIRT parameter file © Astronomical Observatory, Ghent University -->
3 <skirt-simulation-hierarchy type="MonteCarloSimulation" format="9" producer="SKIRT v9.0 (git
  d762d4b built on 10/02/2026 at 15:18:43)" time="2026-03-20T18:52:50.656">
4   <MonteCarloSimulation userLevel="Expert" simulationMode="ExtinctionOnly"
     iteratePrimaryEmission="false" iterateSecondaryEmission="false" numPackets="5e8">
5     <random type="Random">
6       <Random seed="0" />
7     </random>
8     <units type="Units">
9       <ExtragalacticUnits wavelengthOutputStyle="Energy" fluxOutputStyle="Energy" />
10    </units>
11    <cosmology type="Cosmology">
12      <LocalUniverseCosmology />
13    </cosmology>
14    <sourceSystem type="SourceSystem">
```

- We can have a look at the created .ski file
- **XML element** with a **capitalised name** = **SKIRT object**
 - I.e. a **SKIRT C++ class** (see [SKIRT Reference](#) on the SKIRT website)
 - Carries **attributes** reflecting **properties** of the object

```
<geometry type="Geometry">  
  <DonutGeometry largeRadius="1 pc" smallRadius="0.5 pc"/>  
</geometry>
```

- Can contain **nested elements** reflecting **compound properties**
- For **manual editing**: see the [Ski File Reference](#) on the SKIRT website
 - You can use all supported **units** for **quantities**

- The **full description** of a single **SKIRT simulation** is stored in a **.ski parameter file**
- **SKIRT** offers an **interactive Q&A** to set up the simulation & create a **.ski file**
- To **run the SKIRT simulation** (run one or more preconfigured **.ski files**):
 - Invoke **skirt** with the **name of a .ski file**:

```
(base) bert.vandermeulen@EST01114502 run % skirt example.ski
13/05/2026 14:15:57.879 Welcome to SKIRT v9.0 (git d762d4b built on 13/05/2026 at 13:56:33)
13/05/2026 14:15:57.880 Running on EST01114502 for bert.vandermeulen
13/05/2026 14:15:57.922 Constructing a simulation from ski file 'example.ski'...

__More progress messages__

13/05/2026 14:15:57.922 Finished simulation in __ s
```

- The **full description** of a single **SKIRT simulation** is stored in a **.ski parameter file**
- **SKIRT** offers an **interactive Q&A** to set up the simulation & create a **.ski file**
- To **run the SKIRT simulation** (run one or more preconfigured **.ski files**):
 - Invoke **skirt** with the **name of a .ski file**
 - Also possible to:
 - Perform multiple simulations in one go (provide multiple ski files, may contain ? and * wildcards)
 - Use some command line options (see [Command Line Options](#)):

```
13/05/2026 14:27:51.687 ! skirt [-t <threads>] [-s <simulations>] [-d]
13/05/2026 14:27:51.687 ! [-b] [-v] [-m] [-e]
13/05/2026 14:27:51.687 ! [-k] [-i <dirpath>] [-o <dirpath>]
13/05/2026 14:27:51.687 ! [-r] {<filepath>}*
```

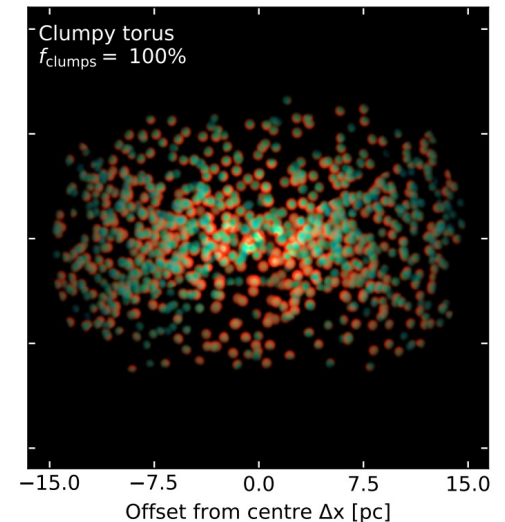
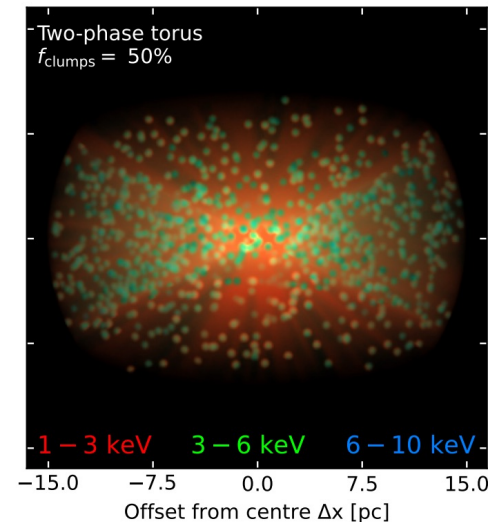
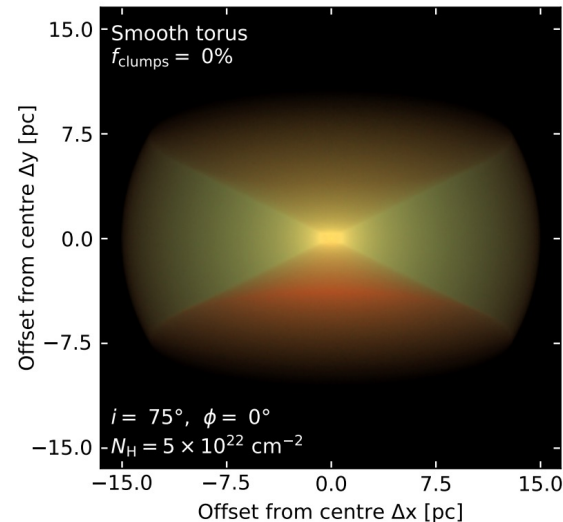
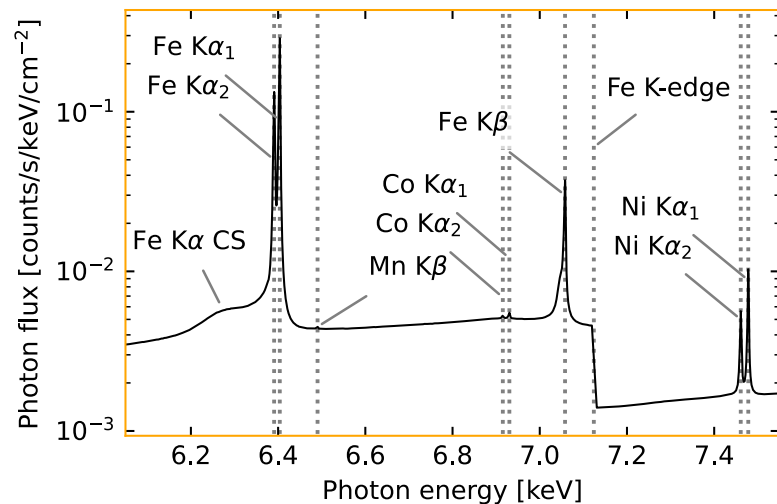
skirt.ugent.be → Documentation → User Guide → SKIRT Q&A and command line options

- The **full description** of a single **SKIRT simulation** is stored in a **.ski parameter file**
- **SKIRT** offers an **interactive Q&A** to set up the simulation & create a **.ski file**
- To **run the SKIRT simulation** (run one or more preconfigured **.ski files**):
 - Invoke **skirt** with the **name of a .ski file**
 - Also possible to:
 - Perform multiple simulations in one go (provide multiple ski files, may contain ? and * wildcards)
 - Use some command line options (see [Command Line Options](#)):
 - An example of the recursive directory descent (batch runs):

```
(base) bert.vandermeulen@EST01114502 run % skirt-r -k "*.ski"
```

The SKIRT simulation output

- A SKIRT run produces various files:
 - example_parameters.xml (**parameter file**: copy of the .ski file)
 - example_log.txt (**log file**: console output)
 - **output from probes**, e.g.: example_cnv_convergence.dat, example_dns_gas_N.fits, ...
 - **output from instruments**:
 - SEDInstrument produces an ascii file (a spectrum) → Plot with Python
 - FrameInstrument produces a .fits file (an image) → Open with ds9 (or Python)



Setting up a .ski file

- Following the Q&A and running the simulation is relatively straightforward
- Most thoughts must go into “setting up the simulation” = “setting up the .ski file”
 - **.ski file** mimics the **structure** of the simulation **runtime hierarchy**:

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <!-- A SKIRT parameter file © Astronomical Observatory, Ghent University -->
3  <skirt-simulation-hierarchy type="MonteCarloSimulation" format="9">
4      <MonteCarloSimulation userLevel="Expert" simulationMode="ExtinctionOnly" numPackets="1e7">
5          <sourceSystem type="SourceSystem">
6              <SourceSystem/>
7          </sourceSystem>
8          <mediumSystem type="MediumSystem">
9              <MediumSystem/>
10         </mediumSystem>
11         <instrumentSystem type="InstrumentSystem">
12             <InstrumentSystem/>
13         </instrumentSystem>
14         <probeSystem type="ProbeSystem">
15             <ProbeSystem/>
16         </probeSystem>
17     </MonteCarloSimulation>
18 </skirt-simulation-hierarchy>
```

Setting up a .ski file

- Following the Q&A and running the simulation is relatively straightforward
- Most thoughts must go into “setting up the simulation” = “setting up the .ski file”
 - **.ski file** mimics the **structure** of the simulation **runtime hierarchy**:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!-- A SKIRT parameter file © Astronomical Observatory, Ghent University -->
3 <skirt-simulation-hierarchy type="MonteCarloSimulation" format="9">
4     <MonteCarloSimulation userLevel="Expert" simulationMode="ExtinctionOnly" numPackets="1e7">
5         <sourceSystem type="SourceSystem">
6             <SourceSystem/>
7         </sourceSystem>
8         <mediumSystem type="MediumSystem">
9             <MediumSystem/>
10        </mediumSystem>
11        <instrumentSystem type="InstrumentSystem">
12            <InstrumentSystem/>
13        </instrumentSystem>
14        <probeSystem type="ProbeSystem">
15            <ProbeSystem/>
16        </probeSystem>
17    </MonteCarloSimulation>
18 </skirt-simulation-hierarchy>
```

→ here you define your sources:

- Geometry
- Velocity
- Spectrum
- Luminosity
- Angular dependence
- Polarisation profile

Setting up a .ski file

- Following the Q&A and running the simulation is relatively straightforward
- Most thoughts must go into “setting up the simulation” = “setting up the .ski file”
 - **.ski file** mimics the **structure** of the simulation **runtime hierarchy**:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!-- A SKIRT parameter file © Astronomical Observatory, Ghent University -->
3 <skirt-simulation-hierarchy type="MonteCarloSimulation" format="9">
4     <MonteCarloSimulation userLevel="Expert" simulationMode="ExtinctionOnly" numPackets="1e7">
5         <sourceSystem type="SourceSystem">
6             <SourceSystem/>
7         </sourceSystem>
8         <mediumSystem type="MediumSystem">
9             <MediumSystem/>
10        </mediumSystem>
11        <instrumentSystem type="InstrumentSystem">
12            <InstrumentSystem/>
13        </instrumentSystem>
14        <probeSystem type="ProbeSystem">
15            <ProbeSystem/>
16        </probeSystem>
17    </MonteCarloSimulation>
18 </skirt-simulation-hierarchy>
```

→ here you define your media:

- Geometry
- Velocity
- Medium type & physics

Setting up a .ski file

- Following the Q&A and running the simulation is relatively straightforward
- Most thoughts must go into “setting up the simulation” = “setting up the .ski file”
 - **.ski file** mimics the **structure** of the simulation **runtime hierarchy**:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!-- A SKIRT parameter file © Astronomical Observatory, Ghent University -->
3 <skirt-simulation-hierarchy type="MonteCarloSimulation" format="9">
4     <MonteCarloSimulation userLevel="Expert" simulationMode="ExtinctionOnly" numPackets="1e7">
5         <sourceSystem type="SourceSystem">
6             <SourceSystem/>
7         </sourceSystem>
8         <mediumSystem type="MediumSystem">
9             <MediumSystem/>
10        </mediumSystem>
11        <instrumentSystem type="InstrumentSystem">
12            <InstrumentSystem/>
13        </instrumentSystem>
14        <probeSystem type="ProbeSystem">
15            <ProbeSystem/>
16        </probeSystem>
17    </MonteCarloSimulation>
18 </skirt-simulation-hierarchy>
```

→ here you define your media:

- Geometry
- Velocity
- Medium type & physics
- **Spatial grid (!)**

Setting up a .ski file

- Following the Q&A and running the simulation is relatively straightforward
- Most thoughts must go into “setting up the simulation” = “setting up the .ski file”
 - **.ski file** mimics the **structure** of the simulation **runtime hierarchy**:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!-- A SKIRT parameter file © Astronomical Observatory, Ghent University -->
3 <skirt-simulation-hierarchy type="MonteCarloSimulation" format="9">
4     <MonteCarloSimulation userLevel="Expert" simulationMode="ExtinctionOnly" numPackets="1e7">
5         <sourceSystem type="SourceSystem">
6             <SourceSystem/>
7         </sourceSystem>
8         <mediumSystem type="MediumSystem">
9             <MediumSystem/>
10        </mediumSystem>
11        <instrumentSystem type="InstrumentSystem">
12            <InstrumentSystem/>
13        </instrumentSystem>
14        <probeSystem type="ProbeSystem">
15            <ProbeSystem/>
16        </probeSystem>
17    </MonteCarloSimulation>
18 </skirt-simulation-hierarchy>
```

- here you define your instruments
- Type (SED- or FrameInstrument)
 - Wavelength grid to record photons

Setting up a .ski file

- Following the Q&A and running the simulation is relatively straightforward
- Most thoughts must go into “setting up the simulation” = “setting up the .ski file”
 - **.ski file** mimics the **structure** of the simulation **runtime hierarchy**:

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <!-- A SKIRT parameter file © Astronomical Observatory, Ghent University -->
3  <skirt-simulation-hierarchy type="MonteCarloSimulation" format="9">
4      <MonteCarloSimulation userLevel="Expert" simulationMode="ExtinctionOnly" numPackets="1e7">
5          <sourceSystem type="SourceSystem">
6              <SourceSystem/>
7          </sourceSystem>
8          <mediumSystem type="MediumSystem">
9              <MediumSystem/>
10         </mediumSystem>
11         <instrumentSystem type="InstrumentSystem">
12             <InstrumentSystem/>
13         </instrumentSystem>
14         <probeSystem type="ProbeSystem">
15             <ProbeSystem/>
16         </probeSystem>
17     </MonteCarloSimulation>
18 </skirt-simulation-hierarchy>
```

→ Probes

Getting started with SKIRT

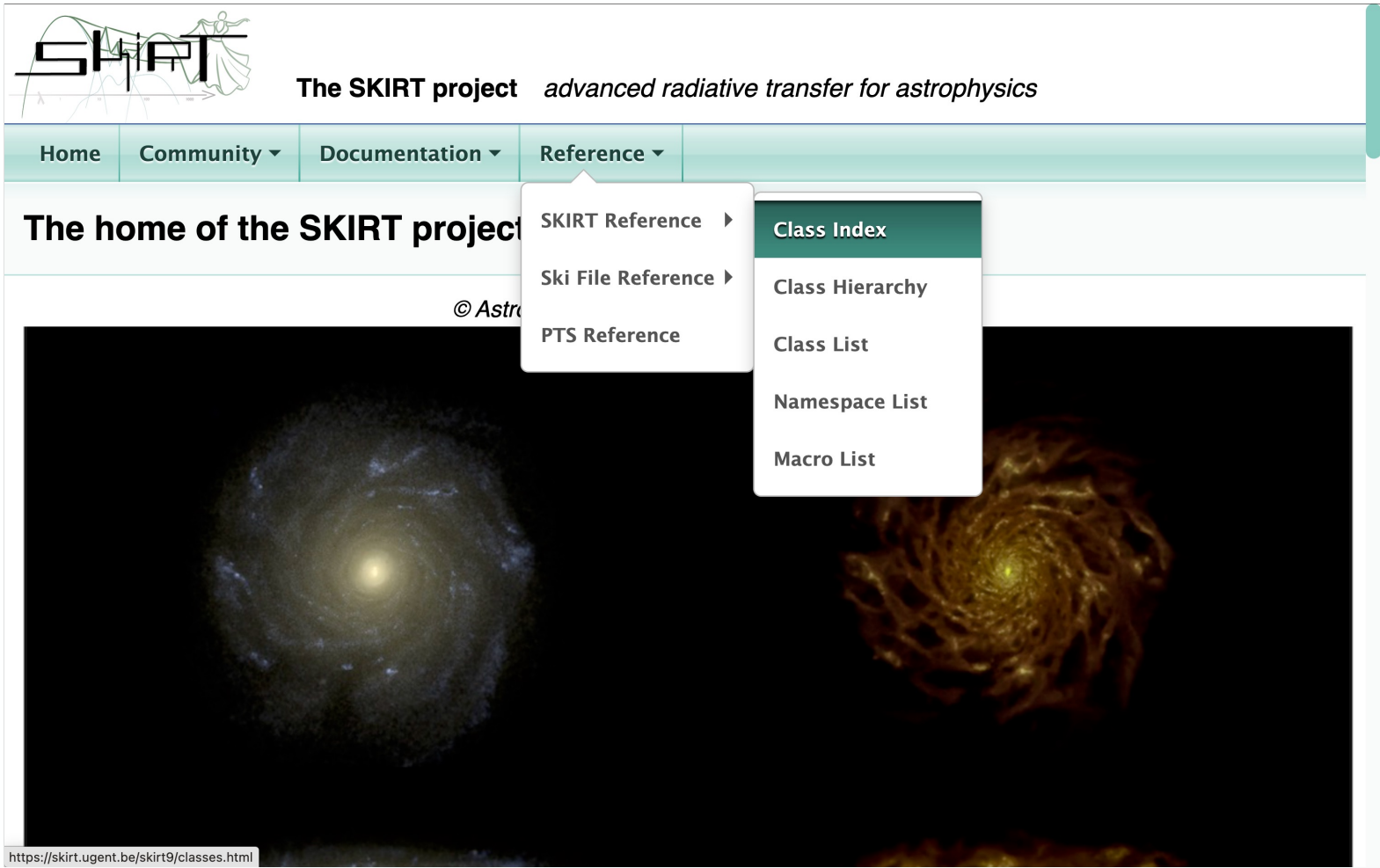


- Remainder of this session: **Start with the first five basic tutorials:**
 1. *Monochromatic simulation of a dusty disk galaxy*
 2. *Panchromatic simulation of a dust torus*
 3. Stochastic heating in an SPH-simulated galaxy
 4. *Scattering by dust in a Kelvin-Helmholtz instability*
 5. *Building a proper spatial grid for a spiral galaxy*
- As an introduction to the SKIRT simulation elements
- Every simulation element is documented on the website
 - We will walk around to answer questions
- We will move to the X-ray tutorial at 14:00

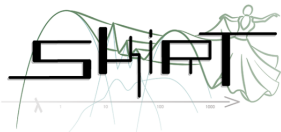


Level	Type	Topic
		Components of the SKIRT project
		Monochromatic simulation of a dusty disk galaxy
		Panchromatic simulation of a dust torus
		SKIRT Q&A and command line options
		Using PTS on the command line
		MakeUp, the graphical interface for SKIRT parameter files
		The SKIRT parameter file format (ski file)
		Upgrading SKIRT parameter files
		The SKIRT units system & supported units
		Stochastic heating in an SPH-simulated galaxy
		Visualizing SKIRT results with PTS
		X-ray reprocessing in a clumpy AGN torus
		Scattering by dust in a Kelvin-Helmholtz instability
		Building a proper spatial grid for a spiral galaxy
		Spatial grids and meshes overview
		Wavelength treatment during a simulation
		Wavelength grids overview
		Configuring wavelength grids
		Inspecting the built-in broadband filters
		Material mixes (dust, electrons, gas)

skirt.ugent.be → Documentation → User Guide



skirt.ugent.be



The SKIRT project *advanced radiative transfer for astrophysics*

Home Community ▾ Documentation ▾ Reference ▾

Class Index

A | B | C | D | E | F | G | H | I | L | M | N | O | P | Q | R | S | T | U | V | W | X | Z

A

- AbsorptionOnlyMaterialMixDecorator
- AbstractDoublePropertyHandler
- AbstractItemPropertyHandler
- AdaptiveMeshGeometry
- AdaptiveMeshInterface
- AdaptiveMeshMedium
- AdaptiveMeshSnapshot
- AdaptiveMeshSource
- AdaptiveMeshSpatialGrid
- AllCellsLibrary
- AllSkyInstrument
- AllSkyProjection
- AllSkyProjectionForm
- AngularDistribution
- AngularDistributionInterface
- AnnulusGeometry
- ApertureInstrument

skirt.ugent.be

Detailed Description

The [TorusGeometry](#) class is a subclass of the [AxGeometry](#) class and describes the geometry of an axisymmetric torus as assumed to be present in the centre of active galactic nuclei (AGN). This geometry is described by a radial power-law density with a finite opening angle; see Stalevski et al. (2012, MNRAS, 420, 2756–2772) and Granato & Danese (1994, MNRAS, 268, 235). In formula, it is most easily expressed in spherical coordinates as

$$\rho(r, \theta) = A r^{-p} e^{-q|\cos \theta|} \quad \text{for } r_{\min} < r < r_{\max} \text{ and } \frac{\pi}{2} - \Delta < \theta < \frac{\pi}{2} + \Delta.$$

There are five free parameters describing this dust geometry: the inner and outer radii $r_{\min}$ and $r_{\max}$ of the torus, the radial power law index p , the polar index q and the angle Δ describing the opening angle of the torus.

If the dusty system under consideration is in the vicinity of an AGN central engine or another source which is luminous enough to heat the dust up to sublimation temperature, the inner radius should correspond to sublimation radius and scale as $r_{\min} \propto L(\theta)^{0.5}$ (Barvainis, 1987, ApJ, 320, 537, eq (5)). If the primary source assumes anisotropic emission, the inner radius must follow the same dependence as the distribution of the primary source luminosity. Otherwise, dust temperature on the inner boundary of geometry is very likely to be under- or over-estimated. Thus, if the [NetzerAccretionDiskGeometry](#) distribution is chosen to describe primary source emission, it is recommended to turn on the anisotropic inner radius option for the torus geometry. The inner radius will then be set by the following formula:

$$r_{\min} \propto (\cos \theta (2 \cos \theta + 1))^{0.5}.$$

This should allow dust to approach all the way to the primary central source in the equatorial plane. However, due to the finite resolution of dust cells, it may happen that some of the innermost cells end up with unphysically high temperatures. For this reason,

skirt.ugent.be