



Full length article

SKIRT: The design of a suite of input models for Monte Carlo radiative transfer simulations[☆]M. Baes^{*}, P. Camps

Sterrenkundig Observatorium, Universiteit Gent, Krijgslaan 281 S9, B-9000 Gent, Belgium

ARTICLE INFO

Article history:

Received 27 February 2015

Accepted 27 May 2015

Available online 9 June 2015

Keywords:

Radiative transfer
Methods: numerical
Designing software
Design patterns

ABSTRACT

The Monte Carlo method is the most popular technique to perform radiative transfer simulations in a general 3D geometry. The algorithms behind and acceleration techniques for Monte Carlo radiative transfer are discussed extensively in the literature, and many different Monte Carlo codes are publicly available. On the contrary, the design of a suite of components that can be used for the distribution of sources and sinks in radiative transfer codes has received very little attention. The availability of such models, with different degrees of complexity, has many benefits. For example, they can serve as toy models to test new physical ingredients, or as parameterised models for inverse radiative transfer fitting. For 3D Monte Carlo codes, this requires algorithms to efficiently generate random positions from 3D density distributions.

We describe the design of a flexible suite of components for the Monte Carlo radiative transfer code SKIRT. The design is based on a combination of basic building blocks (which can be either analytical toy models or numerical models defined on grids or a set of particles) and the extensive use of decorators that combine and alter these building blocks to more complex structures. For a number of decorators, e.g. those that add spiral structure or clumpiness, we provide a detailed description of the algorithms that can be used to generate random positions. Advantages of this decorator-based design include code transparency, the avoidance of code duplication, and an increase in code maintainability. Moreover, since decorators can be chained without problems, very complex models can easily be constructed out of simple building blocks. Finally, based on a number of test simulations, we demonstrate that our design using customised random position generators is superior to a simpler design based on a generic black-box random position generator.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

Radiative transfer, the process that describes radiation propagating through and interacting with matter, is a general problem that is encountered in all areas of astronomy (and far beyond). Due to its high dimensionality and nonlocal and nonlinear behaviour, it is generally considered as one of the most challenging problems in numerical astrophysics.

Recent years have seen an impressive advancement of three-dimensional (3D) radiative transfer studies, thanks to the increase in computational power, the availability of a wealth of new observational constraints and the development of new algorithms. Among the different approaches available to solve the radiative transfer problem, the Monte Carlo method is generally the most

popular one. The first Monte Carlo radiative transfer codes were developed more than four decades ago (e.g., [Mattila, 1970](#); [Roark et al., 1974](#); [Witt, 1977](#)), and since then the method has steadily increased its market share in all fields where 3D radiative transfer is important. Many powerful Monte Carlo codes are available in various fields of computational astrophysics, including dust radiative transfer ([Steinacker et al., 2013](#), and references therein), Ly α line transfer ([Verhamme et al., 2006](#); [Dijkstra et al., 2006](#); [Laursen et al., 2009](#)), ionising radiation transport ([Ciardi et al., 2001](#); [Maselli et al., 2003](#); [Wood et al., 2004](#)), neutron transport ([Romano and Forget, 2013](#)) and neutrino radiation transport ([Abdikamalov et al., 2012](#)). These and many other papers and monographs discuss at length the main algorithms behind Monte Carlo radiative transfer and the various proposed improvements and acceleration techniques.

A very different aspect of Monte Carlo radiative transfer codes that has received very little attention in the literature, is the setup of a suite of components that describe the distribution of the sources and sinks in the radiative transfer code (i.e., the objects that add radiation to or remove radiation from the radiation field).

[☆] This code is registered at the ASCL with the code entry ascl:1109.003.

^{*} Corresponding author.

E-mail address: maarten.baes@ugent.be (M. Baes).

This is an aspect outside the real core of Monte Carlo radiative transfer problem. Virtually all of the available codes require the user to hard-code the model makeup for each distinct problem (although the results of hydrodynamical simulations can generally be processed out of the box, given the appropriate import module). We argue, however, that it is very useful for a radiative transfer code to provide a suite of input models, or *geometries*, as built-in components. Such toy models can provide a low-threshold introduction to new users. They also have a crucial role in a more scientific way: toy input models are used to benchmark different codes (Ivezić et al., 1997; Pascucci et al., 2004; Pinte et al., 2009), to investigate the effects of dust attenuation on the apparent structural parameters of galaxies (Möllenhoff et al., 2006; Gadotti et al., 2010; Pastrav et al., 2013a,b), or to investigate the physical impact of e.g. clumpiness or spiral arms (Witt and Gordon, 1996; Bianchi et al., 2000; Misiriotis et al., 2000; Semionov et al., 2006). Finally, they are essential for so-called inverse radiative transfer, i.e. when a parameterised radiative transfer model is fitted to observational data (Xilouris et al., 1999; Bianchi, 2007; Schechtman-Rook et al., 2012; De Geyter et al., 2013, 2014).

In order to be useful for these goals, the suite of input models should be diverse enough and contain models with different degrees of complexity, ranging from the metaphorical spherical cows to more realistic toy models that consider, for example, spheroidal and triaxial geometries, including bars, spiral arms and clumpy distributions. Setting up such a suite is more complex than it might appear at first sight. While each model is in principle completely determined by the 3D density distribution $\rho(\mathbf{x})$, the setup requires more than just an implementation for this density function.

A crucial aspect of Monte Carlo radiative transfer codes is the emission of a multitude of simulated photon packets from random locations sampled from the source density distribution. So each model in the suite should contain a routine that generates random numbers according to $\rho(\mathbf{x})$. Moreover, this random position generation needs to be very efficient, since the random positions are sampled extremely often. In this aspect, Monte Carlo radiative transfer simulations differ from other codes where random positions need to be sampled from arbitrary density distributions, e.g. to generate the initial conditions for N -body or hydrodynamical simulations. The efficient generation of random positions from complex 3D density distribution is not a trivial task.

Another aspect that needs consideration is the organisation of such a suite of input models. One could provide a parameterised model that can cover every possible option, but this would quickly lead to an explosion of options that is hard to overview and maintain, and would inevitably contain substantial code duplication.

In this paper, we describe how these issues are dealt with in the publicly available 3D Monte Carlo dust radiative transfer code SKIRT¹ (Baes et al., 2003, 2011; Camps and Baes, 2015). While we summarise the relevant information for the discussion here, an in-depth description of the architecture and overall design of the SKIRT code can be found in Camps and Baes (2015). It should also be noted that the presented design issues, while discussed in the context of SKIRT and thus dust radiative transfer, are fully applicable to other Monte Carlo transport problems.

This paper is laid out as follows. In Section 2 we review the general techniques to generate random vectors from multivariate distributions that are available in the specialised numerical analysis literature, but less so in the astrophysics community.

In Section 3 we present the general setup of the suite of models for the SKIRT code. In Section 4 we describe various building blocks currently present in the SKIRT code, and in Section 5 we present a number of decorators that can be used to combine and add complexity to simple building blocks. Some decorators are analysed in more detail, focusing on the methods used to efficiently generate random positions. In Section 6 we discuss the advantages of the decorator-style design of our suite of components, and we critically investigate an alternative option in which random positions are generated using a generic routine rather than customised generators. We show that, while such an approach is simpler to implement and might hence seem an attractive alternative, it cannot compete with our approach, in terms of accuracy and efficiency. Section 7 sums up.

2. Multivariate random number generation techniques

The generation of random numbers from univariate distributions is a well-known topic in numerical analysis. A number of standard methods are widely used and clearly described in standard textbooks (e.g., Press et al., 2007). Additional techniques that are less known in the (astro)physics community include the acceptance-complement method (Kronmal and Peterson, 1981, 1984) and the Forsythe–von Neumann method (von Neumann, 1951; Forsythe, 1972). For an extensive overview of random number generation from univariate distribution, see Devroye (1986). Unfortunately, the generation of random vectors from multivariate distributions is much more complex. The only multivariate distributions from which random vectors can be generated directly are those where the density can be written as a product of independent univariate density distributions. In general, however, more advanced techniques are necessary.

2.1. The inversion method

The inversion method, also known as inversion sampling, is the most popular method for univariate generation problems. The basis of this method is the following: if y is distributed according to a density² $g(y)$, then the variable x defined as the solution of the equation $y = y(x)$ is distributed according to the density

$$f(x) = g(y(x)) \left| \frac{dy}{dx}(x) \right|. \quad (1)$$

If we now want to generate a number from a given density $f(x)$, we can take a uniform distribution

$$g(y) = \begin{cases} 1 & \text{if } 0 \leq y \leq 1, \\ 0 & \text{else,} \end{cases} \quad (2)$$

and set $y(x) = F(x)$ where F is the cumulative distribution corresponding to the density $f(x)$. The inversion method can hence be used to generate random variates with an arbitrary density, provided that the inverse function $F^{-1}(y)$ to the cumulative distribution $F(x)$ is explicitly known. Classical examples include the exponential distribution, the Cauchy distribution, the Rayleigh distribution and the logistic distribution.

Formula (1) can directly be expanded to multiple dimensions: if \mathbf{y} is distributed according to the joint probability distribution $g(\mathbf{y})$, then the vector \mathbf{x} defined as the solution of the vector equation $\mathbf{y} = \mathbf{y}(\mathbf{x})$ is distributed according to the joint distribution

$$f(\mathbf{x}) = g(\mathbf{y}(\mathbf{x})) \left| \frac{\partial \mathbf{y}}{\partial \mathbf{x}}(\mathbf{x}) \right| \quad (3)$$

¹ SKIRT documentation: <http://www.skirt.ugent.be>.
SKIRT code repository: <https://github.com/skirt/skirt>.

² Throughout this paper, all densities, univariate or multivariate, are assumed to be properly normalised.

where $|\partial\mathbf{y}/\partial\mathbf{x}|$ is the Jacobian determinant. Probably the most famous application of this formula is the Box–Muller method to generate random normally distributed deviates (Box and Muller, 1958; Bell, 1968). Another interesting special application is the case of a linear transformation. In this case, the transformation $\mathbf{y} = \mathbf{y}(\mathbf{x})$ can be written as a matrix multiplication $\mathbf{y} = \mathbf{H}\mathbf{x}$, and the distribution of \mathbf{x} is

$$f(\mathbf{x}) = |\mathbf{H}|g(\mathbf{H}\mathbf{x}) \quad (4)$$

with $|\mathbf{H}|$ the absolute value of the determinant of \mathbf{H} . This kind of transformation is particularly useful for the generation of random vectors with a given dependence structure, as measured by the covariance matrix (Scheuer and Stoller, 1962; Barr and Slezak, 1972).

In general, however, it is not straightforward to use this identity (3) to construct a method that can be used to generate random positions from an arbitrary multidimensional distribution.

2.2. The rejection method

Apart from the inversion method, the rejection method, also known as the acceptance–rejection method, is the most popular method to sample nonuniform random numbers from univariate distributions (von Neumann, 1951; Devroye, 1986). The basic idea behind the method is that, if one wants to sample a random number from a function $f(x)$, one can sample uniformly from the 2D region under the graph $f(x)$. More concretely, assume that $f(x) \leq c f_{\text{ref}}(x)$, where $f_{\text{ref}}(x)$ is another distribution from which random numbers are easily generated, and c is the so-called rejection constant (it obviously satisfies the condition $c \geq 1$). One then generates a uniform deviate ξ and a random number x from the reference distribution $f_{\text{ref}}(x)$, and calculates the quantity $t = \xi c f_{\text{ref}}(x)/f(x)$. This procedure is repeated until $t \leq 1$, in which case x is the desired random number.

One of the advantages of the rejection method is that it does not require that the cumulative distribution function be analytically known, let alone be invertible. However, its effectiveness depends on how accurate f is approximated from above by $c f_{\text{ref}}$. Less accurate approximation leads to a greater chance of rejection; on average c iterations of the loop are required before one successful random number is generated. Moreover, the reference distribution should be such that random numbers can be easily generated from it and that the computation of $c f_{\text{ref}}(x)/f(x)$ is simple. For a range of standard distributions, such as the gamma distribution and the Poisson distribution, efficient reference functions can be constructed.

The rejection method is by no means limited to univariate distributions, and can immediately be applied to multivariate generation problems. However, the rejection rate typically increases rapidly when going from one to more dimensions, which decreases the efficiency of the method. Moreover, the design of a suitable reference distribution becomes much more complicated.

2.3. The composition method

The composition method or probability mixing method is another important technique in both univariate and multivariate random number generation (Marsaglia and Bray, 1964; Hörmann et al., 2004). Rather than a method on itself to generate non-uniform random numbers, it is a principle to facilitate and speed up other random number generating methods. The simple idea behind composition is to decompose the distribution $f(\mathbf{x})$ as a weighted sum,

$$f(\mathbf{x}) = \sum_{i=1}^K w_i f_i(\mathbf{x}) \quad (5)$$

with all f_i normalised densities, and the weights w_i a probability vector (i.e., all $w_i \geq 0$ and $\sum_i w_i = 1$). To generate a random \mathbf{x} from the distribution $f(\mathbf{x})$, we first generate a random integer number k from the discrete probability vector w_i , and subsequently generate a random \mathbf{x} from the density $f_k(\mathbf{x})$. A prerequisite for this method to work is that, obviously, the decomposition can be done efficiently, and that the complexity of the problem is reduced by the decomposition.

2.4. The conditional distribution method

Finally, a powerful technique that applies only to multivariate random number generation is the so-called conditional distribution method (Devroye, 1986; Hörmann et al., 2004). It is based on the Bayesian identity

$$f(x_1, x_2) = f_1(x_1) f_2(x_2 | x_1) \quad (6)$$

which expresses that a joint distribution of two variables can be calculated as the product of the marginal distribution of the former variable and the conditional distribution of the latter. Expanding it to multiple dimensions,

$$f(\mathbf{x}) = f_1(x_1) f_2(x_2 | x_1) \cdots f_N(x_N | x_1, \dots, x_{N-1}). \quad (7)$$

The beauty of this technique is that it reduces the multidimensional generation problem to a sequence of independent univariate generation problems. The main drawback is that it can only be used when much detailed information is known about the distribution $f(\mathbf{x})$. In particular, it is by far not always the case that marginal and conditional distributions are easily calculated in closed form. The standard textbook example of this technique is the multivariate Cauchy distribution.

3. General setup of the SKIRT Geometry suite

SKIRT is a multi-purpose 3D Monte Carlo dust radiative transfer code that is mainly used to simulate dusty galaxies and active galactic nuclei (e.g., Baes and Dejonghe, 2002; Baes et al., 2010; Stalevski et al., 2012; De Looze et al., 2012, 2014). The code was designed as a highly modular software, with a particular consideration for the development of a flexible and easy user interface and the use of proven software design principles as described by Camps and Baes (2015).

The SKIRT code offers a wealth of configurable features that are ready to use without any programming. In particular, the SKIRT code is equipped with a suite of input model components, the so-called Geometry classes, that can be used to represent distributions of either sources or sinks. Essentially, the suite consists of a number of classes that all inherit from an abstract Geometry class, for which the C++ class interface looks like

```
class Geometry {
public:
    Geometry();
    virtual ~Geometry();
    virtual double density(Position x) const = 0;
    virtual Position generatePosition() const = 0;
};
```

Both the density and generatePosition functions are pure virtual functions, which means that each model in the suite should provide a routine that implements the (normalised) density $\rho(\mathbf{x})$, and a routine that generates random positions according to this density. For example, the interface of a concrete ConcreteGeometry class that contains only a single parameter p would look as follows:

```

class ConcreteGeometry : public Geometry {
public:
    ConcreteGeometry(double p) {_p=p}
    double density(Position x) const;
    Position generatePosition() const;
private:
    double _p;
}

```

For the design of this suite of input models, a naive option would be to provide a set of parameterised models that contain free parameters to control every possible option. A typical standard component of such a suite would be a very generalised Plummer model, with free parameters setting the location of the centre, the orientation with respect to the coordinate system, the scales, the flattening describing potential triaxiality, possible degrees of clumpiness, etc. This approach has a number of strong disadvantages. Clearly, it would quickly lead to an explosion of different options that is hard to overview. It would inevitably lead to substantial code duplication (nearly all the code would need to be copied if we would consider a Hernquist profile instead of a Plummer profile) and code that is virtually impossible to maintain (the code for all models would need to be updated if a new feature is added or altered). Also concerning efficiency, such a design would not be optimal. Indeed, a random position generating routine for such a generalised Plummer model is hard to construct and is certainly much less efficient than the simple routine that is possible for a plain spherical Plummer model (using the inversion method).

To overcome these problems, we adopted a completely different approach that is much simpler but still provides the flexibility and functionality needed to set up complex models. This is achieved using a combination of simple base models on the one hand, and so-called decorator geometries on the other hand. Decorator geometries are not real models on their own, but they apply modifications upon other models in interesting ways, following the *Decorator* design pattern. In object-oriented programming, a decorator attaches additional responsibilities to an object dynamically and provides a flexible and powerful alternative to subclassing for extending functionality (Gammma et al., 1994).

In our present context, a decorator geometry is a special kind of geometrical model (i.e., it is also a C++ class in the general *Geometry* suite) that takes one or more other components and adds a layer of complexity upon them. Simple examples of decorators that can easily be implemented are the relocation of the centre of a given component, or the rotation of a given model with respect to the coordinate system. More complex decorators deform a spherical model to a triaxial one, or add a spiral perturbation to an axially symmetric model. The advantage of the decorator approach is clear: each decorator needs to be implemented only once, and can then be applied to any possible model.

In the following two sections we describe the building blocks in the SKIRT *Geometry* suite, and a number of decorator geometries that can alter these building blocks to more complex structures.

4. The SKIRT Geometry building blocks

The SKIRT *Geometry* suite contains a limited number of elementary input models, which can be used either as elementary toy models, or as building blocks for more complex geometries. For each of them, the density can be expressed as a simple analytical function, and the generation of random positions reduces to three independent univariate generation problems.

Apart from these analytical components, SKIRT also offers the possibility to set up components in which the geometry of sources and/or sinks is defined by means of particles or on a grid. In particular, SKIRT can import a snapshot from a (magneto)hydrodynamical

simulation. One obvious goal would be to post-process a hydrodynamical simulation in order to calculate the observable multi-wavelength properties of the simulated objects (e.g., Jonsson et al., 2010; Schartmann et al., 2014). In this case, it would be sufficient to just read in the geometry of the snapshot as it is, and start the radiative transfer simulation. More generally, however, it would be useful if these numerical components would be at a similar level as the analytical components. This would open up the possibility to decorate them and combine them with other analytical and/or numerical components to more complex models.

4.1. Analytical components

A first group consists of spherically symmetric models. In spherical symmetry, the generation of a random position from the 3D density $\rho(\mathbf{x}) = \rho(r)$ simplifies to the generation of a random azimuth from a uniform distribution, a random polar angle from a sine distribution, and a random radius from the univariate density $f(r) = 4\pi r^2 \rho(r)$. The SKIRT suite includes the most popular models used to represent shells, star clusters, early-type galaxies, or galaxy bulges, such as a power-law shell model (Ivezić et al., 1997), the Plummer model (Plummer, 1911), the γ -model (Dehnen, 1993; Tremaine et al., 1994), the Sérsic model (Sérsic, 1963; Ciotti and Bertin, 1999; Baes and Gentile, 2011), and the Einasto model (Einasto, 1965; Retana-Montenegro et al., 2012). Within this family of models, many other famous profiles are contained, including the Hernquist model (Hernquist, 1990), the Jaffe model (Jaffe, 1983) and the $R^{1/4}$ -model (de Vaucouleurs, 1948).

A second group of elementary input models consists of axisymmetric models in which the density is a separable function. The standard example with a density distribution separable in cylindrical coordinates is the double-exponential disc that is the de facto standard to represent disc galaxies in radiative transfer studies (Xilouris et al., 1997; Baes et al., 2003; Bianchi, 2007). How random positions can be drawn from this distribution is discussed in Appendix A of Baes et al. (2003). An example of an axisymmetric model where the density is separable in spherical coordinates is the torus model that is often used to represent the dusty tori around stars or active galactic nuclei (Collison and Fix, 1991; Granato and Danese, 1994; Stalevski et al., 2012).

4.2. Components based on smoothed particles

The first group of numerical components in SKIRT are defined as a set of smoothed particles. This is mainly useful when we want to use the output of a smoothed particle hydrodynamics (Lucy, 1977; Monaghan, 1992; Springel, 2010b) simulation. In spite of claims that the technique suffers from fundamental problems (Agertz et al., 2007; Bauer and Springel, 2012), it is still the most popular hydrodynamics technique, especially for cosmological simulations of galaxy formation (e.g., Guedes et al., 2011; Feldmann and Mayer, 2015; Schaye et al., 2015). The output of an SPH snapshot consists of a set of “particles” (or rather anchor points in a co-moving grid), each characterised by a large suite of physical quantities.

As far as SKIRT is concerned, most of these physical quantities are irrelevant. An SPHGeometry component in SKIRT is essentially defined by a list of N smoothed particles and the assumed smoothing kernel $W(r, h)$, with each smoothed particle characterised by a position \mathbf{x}_j , a fractional mass m_j and a smoothing length h_j . The total density at an arbitrary position \mathbf{x} is then given by

$$\rho(\mathbf{x}) = \sum_{j=1}^N m_j W(|\mathbf{x} - \mathbf{x}_j|, h_j). \quad (8)$$

In practice, the kernels used in SPH simulations almost always have a finite support (e.g., Monaghan and Lattanzio, 1985; Desbrun and

Gascuel, 1996; Müller et al., 2003), and in this case only a relatively small number of terms in the sum have a non-zero contribution.

The SPHGeometry class in SKIRT employs smoothing kernel implementations optimised for each specific task. The geometry's density at a given location is calculated according to Eq. (8) using a finite-support cubic spline kernel, so that the operation can be limited to particles that potentially overlap the location of interest. To facilitate this process, the setup phase of the simulation places a rough grid over the spatial domain and constructs a list of overlapping particles for each grid cell. As described in section 3.3 of Camps and Baes (2015), a further optimisation is provided to calculate the mass within a given box (a cuboid lined up with the coordinate axes), as an alternative to sampling the density in various locations across the volume of the box. In this case, the calculation uses the analytical properties of a scaled and clipped Gaussian kernel, designed to approximate the cubic spline kernel, to directly determine the mass in the box. This optimisation accelerates the density calculation for typical cartesian grids by an order of magnitude.

On the other hand, the generation of random positions sampled from the geometry's density distribution is rather straightforward, thanks to the composition method. The first step is the choice of a random smoothed particle, based on a discrete distribution where each particle is weighted by its relative mass contribution. The second step is generating a random position according to the distribution of the chosen particle. The current implementation samples a random number from a Gaussian smoothing kernel with infinite support using the inversion method.

In the future we may provide a suite of smoothing kernels, allowing the user to select the kernel that best fits the SPH snapshot being imported. The methods used for calculating the density and generating a random radius would obviously depend on the actual shape of the kernel.

In addition to what is described above, SKIRT has facilities to associate a spectral energy distribution with each SPH star particle based on properties such as age and metallicity, or to associate a dust mass with each SPH gas particle based on properties such as temperature and metallicity. In the former case, the geometry will weigh the particles by luminosity, for each wavelength separately, and in the latter case, the geometry will weigh the particles by dust mass. A detailed description of these features is beyond the scope of this paper.

4.3. Components based on hierarchical grids

Apart from smoothed particle hydrodynamics, the main other technique that is used to perform hydrodynamics simulations is Eulerian mesh-based hydrodynamics (Stone and Norman, 1992; Fryxell et al., 2000). A fundamental ingredient of this technique is the use of grids with adaptive mesh refinement. Eulerian AMR simulations are used in virtually all fields of astrophysics, and Monte Carlo codes have been adapted to work directly on these hierarchical grids (Kurosawa and Hillier, 2001; Niccolini and Alcolea, 2006; Saftly et al., 2013, 2014).

The HierarchicalGridGeometry in SKIRT reads in snapshots defined by density fields discretised on hierarchical cartesian grids and converts them to the format of the other components in the SKIRT Geometry suite. Calculating the density at an arbitrary position comes down to identifying the cell that contains this position and returning the density associated with that cell. As hierarchical grids have the structure of a tree, isolating the correct cell is a straightforward and computationally cheap operation. Generating random positions from such a component is similar to the case of the smoothed particles, and is based on the composition method. We first generate a random cell from a discrete distribution where each cell is weighted by its relative mass. Secondly, we determine

a random position within the chosen cell; as the cells in a hierarchical cartesian grids are cuboids, this is a trivial task.

4.4. Components based on Voronoi grids

Recently, a new Lagrangian technique that solves the hydrodynamics equations on a moving, unstructured Voronoi grid is gaining popularity. It is claimed to avoid some of the difficulties of smoothed particle hydrodynamics on the one hand and Eulerian grid-based hydrodynamics on the other hand. This technique has been used for many years in the computational fluid dynamics community (Mavriplis, 1997), and a number of novel codes based on this principle have recently been developed in the astrophysics community (Springel, 2010a; Duffell and MacFadyen, 2011). Moving mesh hydrodynamics is mainly applied to simulations of galaxy formation and evolution (e.g., Marinacci et al., 2014; Vogelsberger et al., 2014).

SKIRT contains a VoronoiGridGeometry class that converts a snapshot from a Voronoi hydrodynamical simulation (or any other density field defined on a Voronoi grid) to a regular SKIRT Geometry component. Due to the nature of a Voronoi grid, the only necessary input is the list of all the generating sites and the associated densities; it is hence not necessary to store all the vertices and sides of each of the cells. Based on the generating sites, SKIRT constructs the corresponding unique Voronoi grid using the public Voro++ library (Rycroft, 2009).

The density routine essentially works in the same way as for the case of the hierarchical grids: it comes down to identify the cell that contains the given position and returning the density associated to this cell. In the case of a Voronoi grid, however, the cell identification is not as simple as in a hierarchical tree structure of cartesian grid cells. Due to the nature of Voronoi grids, this is essentially a nearest neighbour search. Rather than looping over all possible sites, SKIRT implements an approach using cuboidal blocks, as explained in detail in Camps et al. (2013). This task could be optimised even further using more advanced techniques based on space partitioning structures such as *k*-d trees or R-trees (Friedman et al., 1977; Guttman, 1984; Liaw et al., 2010).

Also the generation of random positions works essentially in the same way as for hierarchical grids. The first step is identical: we generate a random cell from a discrete distribution where each cell is weighted by its relative mass contribution. The second step, generating a random position from within the chosen cell, is significantly more complex than in the case of a cuboidal cell. To the best of our knowledge, there are no dedicated techniques to generate a random point from a Voronoi cell. There are two possible options.

The first option is to partition the cell into a set of tetrahedra, subsequently select a random tetrahedron from a discrete distribution where every tetrahedron is weighted by its relative volume, and finally generate a random position from the selected tetrahedron. Specific algorithms are available for both the tetrahedrisation of convex polyhedra (Edelsbrunner et al., 1990; Max, 2002) and the generation of random positions from a tetrahedron (Rocchini and Cignoni, 2000).

The second option, which is more simple and which we have adopted in SKIRT, is to use the rejection technique. As the reference distribution we use a uniform density in a cuboidal volume, defined as the 3D bounding box of the cell. As Voronoi cells are convex polyhedra, this bounding box is directly obtained when the vertices of the cell are known (these are calculated using the Voronoi grid setup). Extensive testing has shown that, depending on the distribution of the generating Voronoi sites, the average ratio of the volume of the bounding box of a Voronoi cell over the actual cell volume is about 3–4. This ratio immediately represents the average rejection rate for the random position generation.

Table 1
An overview of the geometry decorators currently implemented in SKIRT, referencing the section in which they are presented. Geometry decorators can be applied to basic geometry building blocks, and can be chained and combined to create complex structures.

5.1	Offset	applies an arbitrary offset to any geometry
5.2	Rotate	applies an arbitrary rotation to any geometry
5.3	SpheCavity	carves out a spherical cavity from any geometry
5.4	Crop	crops any geometry to a given box; a variation on the Cavity decorator
5.5	Combine	combines two geometries into a single geometry
5.5	Triaxial	transforms any spherical geometry to a triaxial form
5.6	Spheroidal	transforms any spherical geometry to a spheroidal form; a special case of the Triaxial decorator
5.7	Spiral	applies spiral arm structure to any axisymmetric geometry
5.7	Clumpy	replaces a portion of the mass in any geometry by randomly placed clumps
6.2	Foam	replaces the model-specific random position generator by a generic routine based on the Foam library

5. The SKIRT Geometry decorators

In this section we describe a number of decorator geometries that can be applied on the building blocks described in the previous section in order to convert them to more complex structures; see Table 1 for an overview. The implementation of the density of a decorator geometry is usually not a major problem; the main challenge is to implement the routine that generates random positions from a decorator geometry, so this is what we focus on.

5.1. Offsets

The `OffsetGeometryDecorator` decorator in SKIRT applies an arbitrary offset \mathbf{a} to any density distribution. If the original density is $\rho_s(\mathbf{y})$, the new density is simply $\rho(\mathbf{x}) = \rho_s(\mathbf{x} - \mathbf{a})$. Generating random positions is equally simple: we generate a random \mathbf{y} from the original density distribution and return $\mathbf{x} = \mathbf{y} + \mathbf{a}$. The C++ implementation of the class in SKIRT looks like:

```
class OffsetGeometryDecorator : public Geometry {
public:
    OffsetGeometryDecorator(Geometry* g, Position a)
        {_g = g; _a = a;}
    double density(Position x) const
        {return _g->density(x-a);}
    Position generatePosition() const
        {return _g->generatePosition()+a;}
private:
    Geometry* _g;
    Position _a;
}
```

5.2. Rotation

Similarly, the `RotateGeometryDecorator` decorator rotates any density distribution. If the rotation is characterised by the orthonormal matrix \mathbf{H} , the new density is $\rho(\mathbf{x}) = \rho_s(\mathbf{H}\mathbf{x})$. To generate a random position from this new density, we generate a random \mathbf{y} from the original density and rotate it over the inverse rotation matrix, i.e. $\mathbf{x} = \mathbf{H}^T\mathbf{y}$.

5.3. Cavities

A third simple decorator, the `CavityGeometryDecorator`, carves out a cavity from another density $\rho_s(\mathbf{y})$. In formula form, we have

$$\rho(\mathbf{x}) = \begin{cases} 0 & \mathbf{x} \text{ in cavity} \\ \frac{\rho_s(\mathbf{x})}{1 - \chi} & \text{else} \end{cases} \quad (9)$$

with χ the fraction of the mass of the original density located in the cavity. This decorator is useful to represent density distributions of dust close to a star or active galactic nucleus, where the dust has been cleared due to sublimation. To generate random positions

from this new density distribution, we just generate a random position from the original density distribution and reject it when it is located in the cavity. This is, in fact, an almost trivial application of the rejection technique, where the original density assumes the role of the reference function, and the rejection constant is $c = 1/(1 - \chi)$.

5.4. Composition

The `CombineGeometryDecorator` combines two density distributions into a single distribution according to

$$\rho(\mathbf{x}) = \frac{w_1 \rho_1(\mathbf{x}) + w_2 \rho_2(\mathbf{x})}{w_1 + w_2} \quad (10)$$

with ρ_1, ρ_2 the original distributions and w_1, w_2 their respective weights in the composite distribution. Generating random positions for this new density distribution is a trivial application of the composition method.

5.5. Triaxial geometries

As a first more complex decorator, we consider the case of a triaxial decorator geometry, which converts a spherically symmetric density distribution into one that is stratified on concentric and confocal ellipsoids. More concretely, assume that we have a spherically symmetric density distribution $\rho_s(\mathbf{y}) = \rho_s(y)$, we then consider its triaxial counterpart

$$\rho(\mathbf{x}) = \frac{1}{pq} \rho_s \left(\sqrt{x_1^2 + \frac{x_2^2}{p^2} + \frac{x_3^2}{q^2}} \right). \quad (11)$$

Such triaxial models are discussed and used extensively to describe the stellar distribution of elliptical galaxies and galactic nuclei and the shape of galactic haloes (e.g., Stark, 1977; Merritt and Fridman, 1996; Trujillo et al., 2002; van den Bosch et al., 2008). Oblate and prolate spheroidal distributions are just a special case of these triaxial models in which $p = 1$.

It is clear that the density (11) cannot be written as a product of independent univariate density distributions. We can use the conditional distribution method. In order to develop a general recipe for generating random positions, we start by rewriting the probability distribution according to (11) in spherical coordinates,

$$\begin{aligned} f(r, \theta, \phi) \\ = \frac{r^2 \sin \theta}{pq} \rho_s \left(r \sqrt{\frac{\sin^2 \theta (\sin^2 \phi + p^2 \cos^2 \phi)}{p^2} + \frac{\cos^2 \theta}{q^2}} \right). \end{aligned} \quad (12)$$

According to formula (7), we now rewrite this expression as

$$f(r, \theta, \phi) = f(\phi) f(\theta | \phi) f(r | \theta, \phi). \quad (13)$$

After some calculation, one finds for the marginal distribution for ϕ the simple expression

$$f(\phi) = \frac{1}{2\pi} \frac{p}{\sin^2\phi + p^2 \cos^2\phi}. \quad (14)$$

Note that this marginal distribution is independent of the specific shape of the density profile and only depends on the flattening parameter p (and for $p = 1$, it reduces to a simple uniform distribution, as expected for a spheroidal distribution). Generating a random azimuth from this density can be done using the standard inversion method; the corresponding cumulative distribution is

$$F(\phi) = \frac{1}{2\pi} \arctan\left(\frac{\tan\phi}{p}\right) \quad (15)$$

and is readily inverted. Once this random ϕ has been determined, we can generate a random polar angle θ from the conditional distribution

$$f(\theta | \phi) = \frac{\sin\theta}{2s} \left(\frac{\sin^2\theta}{s^2} + \cos^2\theta\right)^{-3/2} \quad (16)$$

where we have set $s^2 = q^2(\sin^2\phi + p^2 \cos^2\phi)/p^2$. Also this distribution is independent of the specific shape of the density profile and only depends on the flattening parameters p and q (and the already randomly determined azimuth). Again, generating a random polar angle from this distribution can be done using the standard inversion method: the corresponding cumulative distribution is

$$F(\theta | \phi) = \frac{1}{2} \left(1 - \frac{1}{\sqrt{1 + s^2 \tan^2\theta}}\right) \quad (17)$$

and is again readily inverted. Finally, we consider the conditional distribution for r ,

$$f(r | \theta, \phi) = \frac{4\pi r^2}{q^3} \left(\frac{\sin^2\theta}{s^2} + \cos^2\theta\right)^{3/2} \times \rho_s \left(\frac{r}{q} \sqrt{\frac{\sin^2\theta}{s^2} + \cos^2\theta}\right). \quad (18)$$

If we apply a linear transformation from r to

$$y = \frac{r}{q} \sqrt{\frac{\sin^2\theta}{s^2} + \cos^2\theta} \quad (19)$$

the corresponding density for m is simply

$$f(y | \theta, \phi) = 4\pi y^2 \rho_s(y) \quad (20)$$

which is nothing but the distribution for a random radius from the original density $\rho_s(\mathbf{y})$. The last step in the generation of a random position is to generate a random position \mathbf{y} from the original spherical density distribution $\rho_s(\mathbf{y})$, and transform the radius $y = |\mathbf{y}|$ of this position to a new radius r using the linear transformation (19), in which we use the previously determined values for θ and ϕ .

While this method based on conditional probabilities has a certain beauty, there is, actually, a simpler and more efficient method that is based on the inversion method. Indeed, consider the simple transformation

$$(y_1, y_2, y_3) = (x_1, x_2/p, x_3/q) \quad (21)$$

or in matrix format

$$\mathbf{y} = \mathbf{H} \mathbf{x} \quad \mathbf{H} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1/p & 0 \\ 0 & 0 & 1/q \end{pmatrix}. \quad (22)$$

According to the law of linear transformations (4), if \mathbf{y} is distributed according to the spherical density $\rho_s(\mathbf{y})$, \mathbf{x} will be distributed exactly according to the triaxial density (11). An easy way to generate random positions from a triaxial counterpart of a spherical symmetric density is thus simply generating a random position \mathbf{y} from the spherical density distribution, from which the desired position can easily be calculated as $\mathbf{x} = \mathbf{H}^T \mathbf{y}$.

5.6. Spiral arm structure

As a second complex decorator, we consider a logarithmic spiral arm perturbation that can be applied to any axisymmetric density distribution,

$$\rho(\mathbf{x}) = \rho_s(R, z) [(1 - w) + w \xi(R, \phi)] \quad (23a)$$

where (R, ϕ, z) are the usual cylindrical coordinates, and the function $\xi(R, \phi)$ is defined as

$$\xi(R, \phi) = C_N \sin^{2N} \left[\frac{n}{2} \left(\frac{\ln(R/R_0)}{\tan p} - \phi \right) + \frac{\pi}{4} \right]. \quad (23b)$$

The factor C_N is a normalisation constant that guarantees that the density is normalised, and is given by

$$C_N = \frac{1}{2\pi} \int_0^{2\pi} \sin^{2N} \phi \, d\phi = \frac{\sqrt{\pi} \Gamma(N + 1)}{\Gamma(N + \frac{1}{2})}. \quad (24)$$

This general spiral arm perturbation is a generalisation of the models used by Misiriotis et al. (2000) and Schechtman-Rook et al. (2012), and allows for an arbitrary number of arms n , pitch angle p , spiral perturbation weight w , and arm–interarm size ratio (controlled by the parameter N). For $N = 1$ the density reduces to a more simple perturbation with equal arm–interarm size ratio,

$$\rho(\mathbf{x}) = \rho_s(R, z) \left\{ 1 + w \sin \left[n \left(\frac{\ln(R/R_0)}{\tan p} - \phi \right) \right] \right\}. \quad (25)$$

For the general 3D density distribution (23), we can again use the conditional distribution method, and write

$$f(R, \phi, z) = f(R, z) f(\phi | R, z). \quad (26)$$

Since the spiral perturbation is such that it averages out along every single circle around the z -axis, i.e.,

$$\int_0^{2\pi} \xi(R, \phi) \, d\phi = 1 \quad (27)$$

we trivially find

$$f(R, z) = 2\pi R \rho_s(R, z). \quad (28)$$

This is exactly the 2D probability density distribution that corresponds to the density of the axisymmetric density distribution. We hence simply generate a random position \mathbf{x} from our original density distribution and save the radius R and the height z . In order to generate a random azimuth ϕ , we consider the conditional distribution $f(\phi | R, z)$, for which we find directly

$$f(\phi | R, z) = \frac{(1 - w) + w \xi(R, \phi)}{2\pi}. \quad (29)$$

This univariate density is independent of the shape of the original axisymmetric density profile; it only depends on the parameters of the spiral perturbation and randomly determined radius R . For the general expression (23b), the standard inversion technique is not easily applicable; the cumulative distribution corresponding to the density (29) can be calculated analytically, but the resulting expression cannot be inverted analytically. One could resort to a numerical inversion, but a better approach is to use the rejection technique. We can use a simple uniform density as the reference

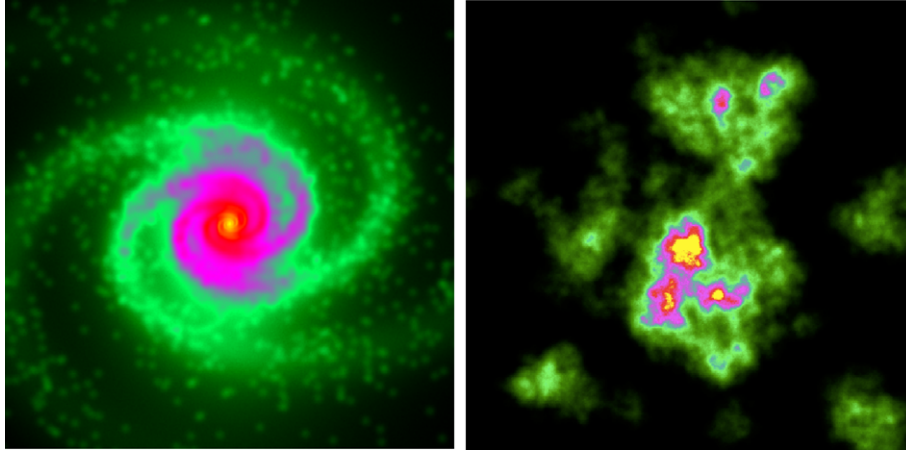


Fig. 1. Two examples of the chained application of decorators. The left panel is a simulated image of a two-component model consisting of a triaxial bulge, and a flattened disc model on which a three-armed logarithmic spiral perturbation and a clumping decorator were applied. For both the bulge and the disc component, the starting model was a simple spherical γ model. The model to the right is a triaxial Plummer model, with four recursive applications of the `ClumpyGeometryDecorator` decorator. The result is a model with a fractal structure. In both models 10^9 photon packages were released to create the images, and the noise in the images is negligible (i.e., all the features in the image are real).

distribution, $f_{\text{ref}}(\phi) = 1/2\pi$. The rejection constant c should ideally be chosen as the smallest value for which $f(\phi | R, z) \leq cf_{\text{ref}}(\phi)$ for all values of ϕ . Since the maximum value of the perturbation function $\xi(R, \phi)$ is C_N , we find

$$c = 1 + (C_N - 1)w. \quad (30)$$

Actually, as an alternative to the conditional probability technique, we could also simply have used the rejection technique from the beginning. The 3D density $\rho(\mathbf{x})$ satisfies the condition $\rho(\mathbf{x}) \leq c\rho_s(\mathbf{x})$, so if we simply use the original density $\rho_s(\mathbf{x})$ as our reference distribution, we have essentially the same algorithm with the same rejection constant. The difference between this 3D rejection technique and the previous version, where we used the rejection technique only for the conditional distribution for the azimuth, is a matter of efficiency. In the former version only one random position needs to be generated from the original density, and the rejection is only to determine the azimuth. In the latter version, an entirely new position is generated for every rejection, which is less efficient.

5.7. Clumpy models

As a last example, we consider the `ClumpyGeometryDecorator` class that turns any density distribution $\rho_s(\mathbf{y})$ into a clumpy analogue, i.e., a density distribution in which a fraction of the mass is distributed “smoothly” as in the original density distribution, and the remaining fraction is locked up into compact clumps. The density of this distribution can be written as

$$\rho(\mathbf{x}) = (1 - f_{\text{cl}}) \rho_s(\mathbf{x}) + \frac{f_{\text{cl}}}{N} \sum_{i=1}^N W(|\mathbf{x} - \mathbf{x}_i|, h) \quad (31)$$

where f_{cl} is the fraction of the mass in clumps, N is the number of clumps, and the positions \mathbf{x}_i are the central positions of the N clumps, determined randomly according to the original density function ρ_s . The function $W(r, h)$ is the smoothing function that sets the distribution of matter within every single clump, with h a characteristic length scale. This can in principle be any spherical 3D distribution; in practice we use either a uniform density sphere or one of the compact support kernels that are used in smoothed particle hydrodynamics simulations.

It is straightforward to generate random positions from this distribution. It is a direct application of the composition method, and the case is very similar to smoothed particle building blocks discussed in Section 4.2.

6. Discussion

6.1. Advantages of the decorator-style design

As indicated in Section 3, a major advantage of the decorator-style approach is that each decorator needs to be implemented only once, and can then be applied to different models. This avoids the need to implement complex geometries with many different possible features and heavy code duplication. That alone is already an argument strong enough to justify its use.

Another strong advantage is the flexibility offered by this approach. The decorators discussed in Section 5 transform an input model into a more complex model. Such a decorated model is often still relatively simple: it typically adds one layer of complexity onto the model that is being decorated (think of a rotated exponential disc or a triaxial Sérsic model). If we want to generate toy models that can be used as idealised representations of, for example, a spiral galaxy, various layers of complexity have to be added. The decorator-style approach is ideal for this purpose: decorators can be applied also to models that have already been decorated. In other words, decorators can easily be chained or nested.

The left-hand panel of Fig. 1 shows a projected image of a relatively complex toy model, constructed using the `Geometry` suite in `SKIRT`. The model consists of a bulge component and a clumpy spiral disc. Each of these two components has a very simple spherical γ -model as their starting point, on which a chain of decorators was applied. The former was first decorated into a triaxial model, the latter was turned into an axisymmetric disc by applying a very strong flattening, then a spiral perturbation was added, and a fraction of its mass was turned into clumps.

In decoration chains the successive decorators do not necessarily need to be different. It is also possible to chain the same decorator several times, i.e. to apply them recursively. A powerful example of this nesting is a repeated application of the `ClumpyGeometryDecorator`. The right-hand panel of Fig. 1 shows an application of this principle on a simple triaxial Plummer model. In this case, we have applied four nested applications of the decorator, in which the number of clumps increases in each level, whereas the smoothing length decreases. This algorithm results in a fractal density distribution that is self-similar over an order of magnitude in scale (Elmegreen, 1997; Mathis et al., 2002; Indebetouw et al., 2006).

The possibility to chain decorators, including a repeated application of the same decorator, facilitates the construction of

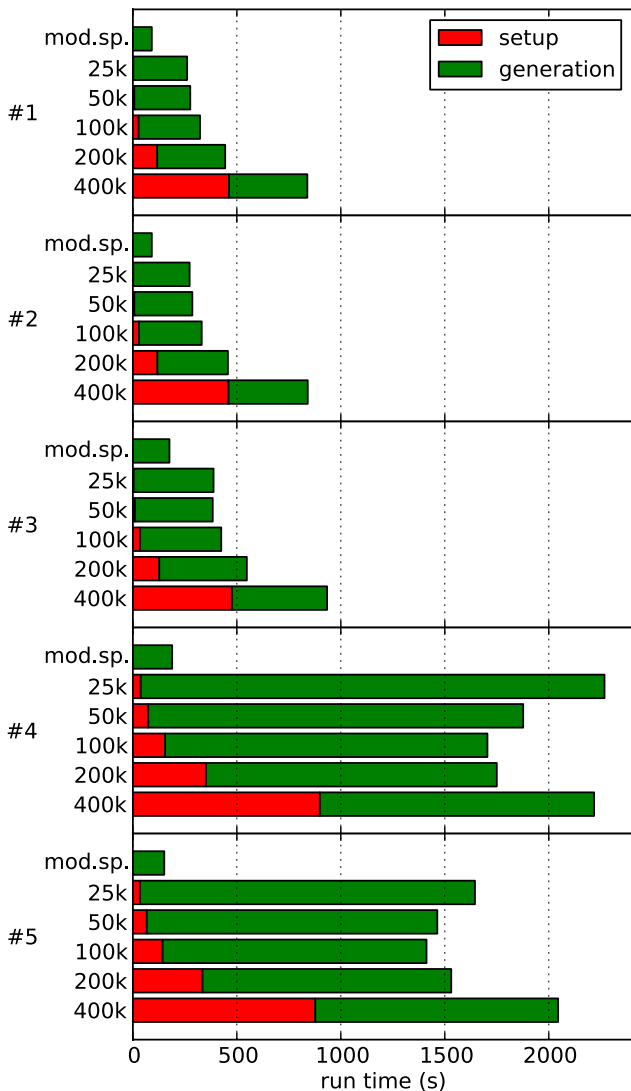


Fig. 2. Results of the timing tests for the test models #1 through #5, using the model-specific and the generic Foam-based random position generators with 25,000 through 400,000 cells. The horizontal bars show the simulation run-time in seconds, split into the fraction devoted to the setup of the grid (red) and the actual time spent on random position generation (green) (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

very complex models out of simple building blocks. Analytical components can easily be combined with numerical components based on smoothed particles or grids, and decorators can be applied regardless of the underlying component type. It is, for example, possible to add a smooth triaxial bulge to an irregular disc structure defined using SPH particles, or to carve out a cavity from a complex hydrodynamics system and locate a small nuclear structure there to simulate the effects of an AGN on the large-scale structure of a galaxy.

6.2. The use of a generic random position generator

One could argue that the setup that we have chosen is unnecessarily complex. An alternative approach would be a design where we only provide the density for each component in the suite, and where the generation of random positions is executed by a generic routine rather than a geometry-specific routine for each component. Such a suite could still use the decorator design pattern, but would be simpler to implement. The main challenge is

the design of a generic routine that can generate random positions corresponding to an arbitrary 3D density function.

This is in principle possible: there are a few multivariate black-box random number generation libraries available, including Foam (Jadach, 2000, 2003) and RanLip (Beliakov, 2005a,b), based on the so-called composition–rejection method, a combination of the composition and rejection methods. In a first phase, the exploration phase, the domain of a distribution is partitioned into different cells. In the generation phase, the rejection technique is used on each cell, where typically a constant density function is used as the reference distribution. The advantages of this approach are that, in principle, it can be applied for all distributions and in any dimension, and that only the density of each component is required. The main disadvantage is the complexity and the overhead, both in memory and run-time, that are linked to the exploration phase.

In order to test whether or not our design choice of providing a customised random position generator for each concrete Geometry class (and for each decorator, in particular) is justified, we have set up a comparison between our routines and a generic black-box routine. We have implemented a new decorator class, FoamGeometryDecorator, that replaces the model-specific random position generator by a generic routine based on the Foam library. Foam is a self-adapting cellular code that uses iterative binary splitting, using either simplexes or hyper-rectangles, to subdivide the configuration space.

We set up a simple suite of test models, for which we generate random positions using both the model-specific generator and the generic Foam generator. Our suite consists of a sequence of models with an increasing level of complexity. It essentially follows the different steps in building the model shown in the left panel of Fig. 1. The following models are considered

1. a spherical $\gamma = \frac{1}{2}$ model,
2. model #1 flattened to an axisymmetric disc,
3. model #2 with spiral perturbation,
4. model #3 with a fraction turned into clumps,
5. model #4 with a triaxial $\gamma = \frac{1}{2}$ bulge added.

For each model, we generate 10^8 random positions (we do not use any photon propagation or detection as would be used in Monte Carlo radiative transfer simulation in order to isolate the random position generation process). The timings were done using a single core and with averages over multiple test runs. One crucial aspect for the Foam-based generator is the choice of the Foam parameters, in particular of the number of cells in the grid. A larger number of cells imply a higher computational cost of the exploration phase on the one hand, but a better approximation of the density and smaller rejection rates. The ideal choice of this parameter is impossible to determine in a general way. We consider 5 different values for the number of cells in the Foam for every model, ranging from 25,000 to 400,000.

The results are illustrated in Fig. 2, where, for each model, we give the total run time and the contributions for the setup of the foam and the actual time spent on generating the random positions.

A first major conclusion that can be drawn from these results is the efficiency of the customised random position generation routines. From a simple spherical model, to a complex model composed of two components that both have been decorated multiple times, the computational cost increases only by a factor two. Moving from a spherical to a triaxial model does not affect the efficiency of the generation of random positions at all (not a surprise, as it just implies two simple multiplications, as shown in Section 5.5). The biggest jump in timing occurs when spiral structure is added to the model (i.e., from model #2 to #3), because the random position generation is based on the rejection

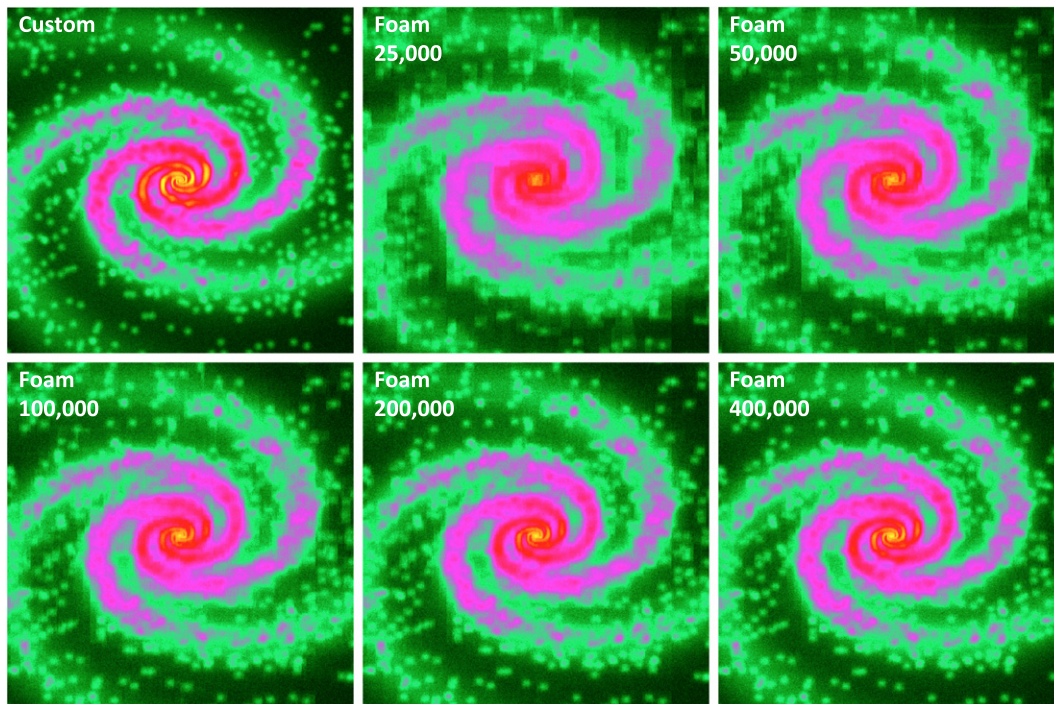


Fig. 3. A comparison of the Monte Carlo generated images for test model #4, based on 10^8 photon packages. For the top-left panel, we have used the customised random position generator based on the chain of decorators. For the other panels, we have used the generic Foam-based generator, with the number of cells in each panel indicated in the top-left corner.

method, which has more overhead compared to the inversion and composition methods. The addition of clumping (from model #3 to #4) hardly implies an increase in computation time. Note that the addition of a bulge even decreases the computation time. In model #5 half of the positions are generated according to the density of the relatively simple triaxial bulge, whereas in model #4 they all are generated according to the more complex clumpy spiral structure.

A second major conclusion is that the generic Foam-based random position generator is significantly less efficient than the customised generator. For the most simple models (#1 and #2), the difference in speed is a factor 3–8, depending on the number of cells used. The increase in total run time for the Foam-based models with increasing number of cells is mainly due to the time necessary to set up the foam. The time necessary to generate the random positions also increases, but not as strongly.

For the more complex models, the efficiency of the Foam-based generator decreases even more. The biggest jump in timing now occurs when we add clumping to the model, because the calculation of the density for the clumpy models is computationally very demanding (each density evaluation requires a sum over a large number of clumps). For models #4 and 5, the Foam-based generators are at least an order of magnitude slower than the customised routines. Interestingly, the run time of these simulations does not increase with the number of cells. For simulations with few cells, the setup phase is relatively quick, but the generation phase is very inefficient. The reason is that the cells are relatively large with a strong variation in density values, and hence the rejection rates are uncomfortably large. If the number of cells in the grid increases, the setup time naturally increases, but the actual position generation becomes more efficient because the average rejection rates are smaller.

Moreover, the inefficiency of the Foam-based generator is not the only problem, but also accuracy is an issue, in particular for complex models with multiple local maxima and strong gradients, such as models #4 and #5. If the number of cells is limited and

the cells hence rather large, both the total mass and the local maximum within a cell are very hard to guess (they are determined by random sampling the cell). As a result, both the relative weights of the different cells and the rejection constants within the individual cells are poorly determined. This will lead to an artificial smoothing and wrong results. Fig. 3 shows a sequence of images corresponding to model #4, created using the customised random position generator (top left) and using the Foam-based one with different values for the number of cells (remaining panels). The customised generator generates an image that reveals all the details that are expected, including the sharp density gradients and the contrast between arm and interarm regions. The only limitation in the image is the finite resolution due to the pixel size and the unavoidable Poisson noise. The Foam images on the other hand show a clear signature of degradation that gradually decreases when the number of cells grows. For the grids with 25,000 and 50,000 cells, and even for the one with 100,000 cells, the effects of the grid are clearly visible, and the individual clumps and the spiral structure are insufficiently resolved. For the grid with 400,000 cells, the individual clumps are well resolved, but the sharpness of the spiral arms is still under-resolved, especially in the central regions.

Since both the accuracy and the efficiency of the generic Foam-based random position generator cannot compete with the customised versions, we conclude that it is worth investing in the latter, and that our design choice is justified.

7. Conclusions

We have described the design of a suite of components that can be used to model the distribution of sources and sinks in the Monte Carlo radiative transfer code SKIRT. Our main conclusions are the following:

- The availability of a well-designed suite of input models, with enough variety and different degrees of complexity, in a publicly available Monte Carlo code has a strong added value.

Such models can serve as toy models to test new physical ingredients, or as parameterised models for inverse radiative transfer fitting. They also provide a low-threshold introduction to new users, since models with differing degrees of complexity can be run without any coding at all.

- Each model is in principle completely determined by the 3D density distribution $\rho(\mathbf{x})$. In order to be used in Monte Carlo radiative transfer, however, each model should contain a routine that generates random numbers according to this $\rho(\mathbf{x})$. Finding the most suitable algorithm to implement this is the most challenging aspect of the design of a suite of input models.
- The design of the SKIRT Geometry suite is based on a combination of basic building blocks and the extensive use of decorators. The building blocks can either be simple analytical components, or they can be numerical components defined as a set of smoothed particles or on a hierarchical or unstructured Voronoi grid. The Geometry decorators combine and alter these building blocks to more complex structures.
- The different multivariate random number generation techniques that exist in the specialised numerical analysis literature can be used to efficiently implement complex decorators, for example those that add triaxiality, spiral structure or clumpiness to other models. Decorators can be chained without problems and essentially without limitation. Different layers of complexity can hence be added one by one. The result is that very complex models can easily be constructed out of simple building blocks, without any coding at all.
- From the software design point of view, decorators have many advantages, including code transparency, the avoidance of code duplication, and an increase in code maintainability. This is a clear example that adhering to proven software design principles pays off, even for small and mid-sized projects.
- Finally, we demonstrate that our design using customised random position generators is superior to a simpler suite design based on a generic black-box random position generator. Using a suite of test simulations with increasing complexity we demonstrate that our customised random number generators are more accurate and more efficient.

Acknowledgements

This work fits in the CHARM framework (Contemporary physical challenges in Heliospheric and AstRophysical Models), a phase VII Interuniversity Attraction Pole (IAP) program organised by BELSPO, the BELgian federal Science Policy Office. The authors thank Ilse De Looze and Sébastien Viaene for their careful reading of a draft version of this paper, and all SKIRT users for their feedback, which has led to many improvements and additions to the code.

References

- Abdikamalov, E., Burrows, A., Ott, C.D., Löffler, F., O'Connor, E., Dolence, J.C., Schnetter, E., 2012. A new Monte Carlo method for time-dependent neutrino radiation transport. *Astrophys. J.* 755, 111.
- Agertz, O., Moore, B., Stadel, J., Potter, D., Miniati, F., Read, J., Mayer, L., Gawryszczak, A., Kravtsov, A., Nordlund, Å., Pearce, F., Quilis, V., Rudd, D., Springel, V., Stone, J., Tasker, E., Teyssier, R., Wadsley, J., Walder, R., 2007. Fundamental differences between SPH and grid methods. *Mon. Not. R. Astron. Soc.* 380, 963–978.
- Baes, M., Davies, J.L., Dejonghe, H., Sabatini, S., Roberts, S., Evans, R., Linder, S.M., Smith, R.M., de Blok, W.J.G., 2003. Radiative transfer in disc galaxies—III. The observed kinematics of dusty disc galaxies. *Mon. Not. R. Astron. Soc.* 343, 1081–1094.
- Baes, M., Dejonghe, H., 2002. Kinematics of elliptical galaxies with a diffuse dust component—III. A Monte Carlo approach to include the effects of scattering. *Mon. Not. R. Astron. Soc.* 335, 441–458.
- Baes, M., Fritz, J., Gadotti, D.A., Smith, D.J.B., Dunne, L., da Cunha, E., Amblard, A., Auld, R., Bendo, G.J., Bonfield, D., Burgarella, D., Buttiglione, S., Cava, A., Clements, D., Cooray, A., Dariush, A., de Zotti, G., Dye, S., Eales, S., Frayer, D., Gonzalez-Nuevo, J., Herranz, D., Ibar, E., Ivison, R., Lagache, G., Leeuw, L., Lopez-Caniego, M., Jarvis, M., Maddox, S., Negrello, M., Michałowski, M., Pascale, E., Pohlen, M., Rigby, E., Rodighiero, G., Samui, S., Serjeant, S., Temi, P., Thompson, M., van der Werf, P., Verma, A., Vlahakis, C., 2010. Herschel-ATLAS: The dust energy balance in the edge-on spiral galaxy UGC 4754. *Astron. Astrophys.* 518, L39.
- Baes, M., Gentile, G., 2011. Analytical expressions for the deprojected Sérsic model. *Astron. Astrophys.* 525, A136.
- Baes, M., Verstackpen, J., De Looze, I., Fritz, J., Saftly, W., Vidal Pérez, E., Stalevski, M., Valcke, S., 2011. Efficient three-dimensional NLTE dust radiative transfer with SKIRT. *Astrophys. J. Suppl.* 196, 22.
- Barr, D.R., Slezak, N.L., 1972. A comparison of multivariate normal generators. *Commun. ACM* 15, 1048–1049.
- Bauer, A., Springel, V., 2012. Subsonic turbulence in smoothed particle hydrodynamics and moving-mesh simulations. *Mon. Not. R. Astron. Soc.* 423, 2558–2578.
- Beliakov, G., 2005a. Class library ranlib for multivariate nonuniform random variate generation. *Comput. Phys. Comm.* 170, 93–108.
- Beliakov, G., 2005b. Universal nonuniform random vector generator based on acceptance-rejection. *ACM Trans. Model. Comput. Simul.* 15, 205–232.
- Bell, J.R., 1968. Algorithm 334: Normal random deviates. *Commun. ACM* 11, 498.
- Bianchi, S., 2007. The dust distribution in edge-on galaxies. Radiative transfer fits of V and K'-band images. *Astron. Astrophys.* 471, 765–773.
- Bianchi, S., Ferrara, A., Davies, J.L., Alton, P.B., 2000. Effects of clumping on the observed properties of dusty galaxies. *Mon. Not. R. Astron. Soc.* 311, 601–610.
- Box, G.E.P., Muller, M.E., 1958. A note on the generation of random normal deviates. *Ann. Math. Statist.* 29, 610–611.
- Camps, P., Baes, M., 2015. SKIRT: An advanced dust radiative transfer code with a user-friendly architecture. *Astron. Comput.* 9, 20–33.
- Camps, P., Baes, M., Saftly, W., 2013. Using 3D Voronoi grids in radiative transfer simulations. *Astron. Astrophys.* 560, A35.
- Ciardi, B., Ferrara, A., Marri, S., Raimondo, G., 2001. Cosmological reionization around the first stars: Monte Carlo radiative transfer. *Mon. Not. R. Astron. Soc.* 324, 381–388.
- Ciotti, L., Bertin, G., 1999. Analytical properties of the $R^{1/m}$ law. *Astron. Astrophys.* 352, 447–451.
- Collison, A.J., Fix, J.D., 1991. Axisymmetric models of circumstellar dust shells. *Astrophys. J.* 368, 545–557.
- De Geyter, G., Baes, M., Camps, P., Fritz, J., De Looze, I., Hughes, T.M., Viaene, S., Gentile, G., 2014. The distribution of interstellar dust in CALIFA edge-on galaxies via oligochromatic radiative transfer fitting. *Mon. Not. R. Astron. Soc.* 441, 869–885.
- De Geyter, G., Baes, M., Fritz, J., Camps, P., 2013. FitSKIRT: genetic algorithms to automatically fit dusty galaxies with a Monte Carlo radiative transfer code. *Astron. Astrophys.* 550, A74.
- De Looze, I., Baes, M., Fritz, J., Verstackpen, J., 2012. Panchromatic radiative transfer modelling of stars and dust in the Sombrero galaxy. *Mon. Not. R. Astron. Soc.* 419, 895–903.
- De Looze, I., Fritz, J., Baes, M., Bendo, G.J., Cortese, L., Boquien, M., Boselli, A., Camps, P., Cooray, A., Cormier, D., Davies, J.L., De Geyter, G., Hughes, T.M., Jones, A.P., Karczewski, O.L., Lebouteiller, V., Lu, N., Madden, S.C., Rémy-Ruyer, A., Spingoglio, L., Smith, M.W.L., Viaene, S., Wilson, C.D., 2014. High-resolution, 3D radiative transfer modeling. I. The grand-design spiral galaxy M51. *Astron. Astrophys.* 571, A69.
- de Vaucouleurs, G., 1948. Recherches sur les nébuleuses extragalactiques. *Ann. Astrophys.* 11, 247–287.
- Dehnen, W., 1993. A family of potential-density pairs for spherical galaxies and bulges. *Mon. Not. R. Astron. Soc.* 265, 250–256.
- Desbrun, M., Gascuel, M.-P., 1996. Smoothed particles: A new paradigm for animating highly deformable bodies. In: *Proceedings of the Eurographics Workshop on Computer Animation and Simulation'96*, pp. 61–76.
- Devroye, L., 1986. *Non-Uniform Random Variate Generation*. Springer-Verlag.
- Dijkstra, M., Haiman, Z., Spaans, M., 2006. Ly α radiation from collapsing protogalaxies. I. Characteristics of the emergent spectrum. *Astrophys. J.* 649, 14–36.
- Duffell, P.C., MacFadyen, A.I., 2011. TESS: A relativistic hydrodynamics code on a moving Voronoi mesh. *Astrophys. J. Suppl.* 197, 15.
- Edelsbrunner, H., Preparata, F., West, D., 1990. Tetrahedrizing point sets in three dimensions. *J. Symbolic. Comput.* 10, 335–347.
- Einasto, J., 1965. On the construction of a composite model for the galaxy and on the determination of the system of galactic parameters. *Tr. Astrofizicheskogo Inst. Alma-Ata* 5, 87–100.
- Elmegreen, B.G., 1997. Intercloud structure in a turbulent fractal interstellar medium. *Astrophys. J.* 477, 196–203.
- Feldmann, R., Mayer, L., 2015. The Argo simulation—I. Quenching of massive galaxies at high redshift as a result of cosmological starvation. *Mon. Not. R. Astron. Soc.* 446, 1939–1956.
- Forsythe, G.E., 1972. von Neumann's comparison method for random sampling from the normal and other distributions. *Math. Comp.* 26, 817–826.
- Friedman, J.H., Bentley, J.L., Finkel, R.A., 1977. An algorithm for finding best matches in logarithmic expected time. *ACM Trans. Math. Software* 3, 209–226.
- Fryxell, B., Olson, K., Ricker, P., Timmes, F.X., Zingale, M., Lamb, D.Q., MacNeice, P., Rosner, R., Truran, J.W., Tufo, H., 2000. FLASH: an adaptive mesh hydrodynamics code for modeling astrophysical thermonuclear flashes. *Astrophys. J. Suppl.* 131, 273–334.

- Gadotti, D.A., Baes, M., Falony, S., 2010. Radiative transfer in disc galaxies—IV. The effects of dust attenuation on bulge and disc structural parameters. *Mon. Not. R. Astron. Soc.* 403, 2053–2062.
- Gamma, E., Helm, R., Johnson, R., Vlissides, J., 1994. *Design Patterns: Elements of Reusable Object-Oriented Software*. Pearson Education.
- Granato, G.L., Danese, L., 1994. Thick tori around active galactic nuclei—a comparison of model predictions with observations of the infrared continuum and silicate features. *Mon. Not. R. Astron. Soc.* 268, 235–252.
- Guedes, J., Callegari, S., Madau, P., Mayer, L., 2011. Forming realistic late-type spirals in a Λ CDM universe: The eris simulation. *Astrophys. J.* 742, 76.
- Guttman, A., 1984. R-trees: A dynamic index structure for spatial searching. *SIGMOD Rec.* 14, 47–57.
- Hernquist, L., 1990. An analytical model for spherical galaxies and bulges. *Astrophys. J.* 356, 359–364.
- Hörmann, W., Leydold, J., Derflinger, G., 2004. *Automatic Nonuniform Random Variate Generation*. Springer-Verlag.
- Indebetouw, R., Whitney, B.A., Johnson, K.E., Wood, K., 2006. Three-dimensional models of embedded high-mass stars: Effects of a clumpy circumstellar medium. *Astrophys. J.* 636, 362–380.
- Ivezic, Z., Groenewegen, M.A.T., Men'shchikov, A., Szczerba, R., 1997. Benchmark problems for dust radiative transfer. *Mon. Not. R. Astron. Soc.* 291, 121–124.
- Jadach, S., 2000. Foam: Multi-dimensional general purpose Monte Carlo generator with self-adapting simplicial grid. *Comput. Phys. Comm.* 130, 244–259.
- Jadach, S., 2003. Foam: A general-purpose cellular Monte Carlo event generator. *Comput. Phys. Comm.* 152, 55–100.
- Jaffe, W., 1983. A simple model for the distribution of light in spherical galaxies. *Mon. Not. R. Astron. Soc.* 202, 995–999.
- Jonsson, P., Groves, B.A., Cox, T.J., 2010. High-resolution panchromatic spectral models of galaxies including photoionization and dust. *Mon. Not. R. Astron. Soc.* 403, 17–44.
- Kronmal, R.A., Peterson, A.V., 1981. A variant of the acceptance-rejection method for computer generation of random variables. *J. Amer. Statist. Assoc.* 76, 446–451.
- Kronmal, R.A., Peterson Jr., A.V., 1984. An acceptance-complement analogue of the mixture-plus-acceptance-rejection method for generating random variables. *ACM Trans. Math. Software* 10, 271–281.
- Kurosawa, R., Hillier, D.J., 2001. Tree-structured grid model of line and polarization variability from massive binaries. *Astron. Astrophys.* 379, 336–346.
- Laursen, P., Sommer-Larsen, J., Andersen, A.C., 2009. Ly α radiative transfer with dust: escape fractions from simulated high-redshift galaxies. *Astrophys. J.* 704, 1640–1656.
- Liaw, Y.-C., Leou, M.-L., Wu, C.-M., 2010. Fast exact k nearest neighbors search using an orthogonal search tree. *Pattern Recogn.* 43, 2351–2358.
- Lucy, L.B., 1977. A numerical approach to the testing of the fission hypothesis. *Astronom. J.* 82, 1013–1024.
- Marinacci, F., Pakmor, R., Springel, V., 2014. The formation of disc galaxies in high-resolution moving-mesh cosmological simulations. *Mon. Not. R. Astron. Soc.* 437, 1750–1775.
- Marsaglia, G., Bray, T.A., 1964. A convenient method for generating normal variables. *SIAM Rev.* 6, 260–264.
- Maselli, A., Ferrara, A., Ciardi, B., 2003. CRASH: a radiative transfer scheme. *Mon. Not. R. Astron. Soc.* 345, 379–394.
- Mathis, J.S., Whitney, B.A., Wood, K., 2002. Can reflection from grains diagnose the albedo? *Astrophys. J.* 574, 812–821.
- Mattila, K., 1970. Interpretation of the surface brightness of dark nebulae. *Astron. Astrophys.* 9, 53–63.
- Mavriplis, D.J., 1997. Unstructured grid techniques. *Annu. Rev. Fluid Mech.* 29, 473–514.
- Max, N., 2002. Consistent subdivision of convex polyhedra into tetrahedra. *J. Graph. Tools* 6, 29–36.
- Merritt, D., Fridman, T., 1996. Triaxial galaxies with cusps. *Astrophys. J.* 460, 136–161.
- Misiriotis, A., Kylafis, N.D., Papamastorakis, J., Xilouris, E.M., 2000. Is the exponential distribution a good approximation of dusty galactic disks? *Astron. Astrophys.* 353, 117–123.
- Möllenhoff, C., Popescu, C.C., Tuffs, R.J., 2006. Modelling the spectral energy distribution of galaxies. IV. Correcting apparent disk scalelengths and central surface brightnesses for the effect of dust at optical and near-infrared wavelengths. *Astron. Astrophys.* 456, 941–952.
- Monaghan, J.J., 1992. Smoothed particle hydrodynamics. *Ann. Rev. Astron. Astrophys.* 30, 543–574.
- Monaghan, J.J., Lattanzio, J.C., 1985. A refined particle method for astrophysical problems. *Astron. Astrophys.* 149, 135–143.
- Müller, M., Charypar, D., Gross, M., 2003. Particle-based fluid simulation for interactive applications. In: *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. pp. 154–159.
- Niccolini, G., Alcolea, J., 2006. 3D continuum radiative transfer. A new adaptive grid construction algorithm based on the Monte Carlo method. *Astron. Astrophys.* 456, 1–12.
- Pascucci, I., Wolf, S., Steinacker, J., Dullemond, C.P., Henning, T., Niccolini, G., Woitke, P., Lopez, B., 2004. The 2D continuum radiative transfer problem. Benchmark results for disk configurations. *Astron. Astrophys.* 417, 793–805.
- Pastrav, B.A., Popescu, C.C., Tuffs, R.J., Sansom, A.E., 2013a. The effects of dust on the derived photometric parameters of disks and bulges in spiral galaxies. *Astron. Astrophys.* 553, A80.
- Pastrav, B.A., Popescu, C.C., Tuffs, R.J., Sansom, A.E., 2013b. The effects of dust on the photometric parameters of decomposed disks and bulges. *Astron. Astrophys.* 557, A137.
- Pinte, C., Harries, T.J., Min, M., Watson, A.M., Dullemond, C.P., Woitke, P., Ménard, F., Durán-Rojas, M.C., 2009. Benchmark problems for continuum radiative transfer. High optical depths, anisotropic scattering, and polarisation. *Astron. Astrophys.* 498, 967–980.
- Plummer, H.C., 1911. On the problem of distribution in globular star clusters. *Mon. Not. R. Astron. Soc.* 71, 460–470.
- Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P., 2007. *Numerical Recipes*, third ed. In: *The Art of Scientific Computing*. Cambridge University Press.
- Retana-Montenegro, E., van Hese, E., Gentile, G., Baes, M., Frutos-Alfaro, F., 2012. Analytical properties of Einasto dark matter haloes. *Astron. Astrophys.* 540, A70.
- Roark, T., Roark, B., Collins II, G.W., 1974. Monte Carlo model of reflection nebulae: intensity gradients. *Astrophys. J.* 190, 67–72.
- Rocchini, C., Cignoni, P., 2000. Generating random points in a tetrahedron. *J. Graph. Tools* 5, 9–12.
- Romano, P.K., Forget, B., 2013. The openmc monte carlo particle transport code. *Ann. Nucl. Energy* 51, 274–281.
- Rycroft, C.H., 2009. Voro++: A three-dimensional voronoi cell library in c++. *Chaos* 19 (4).
- Saftly, W., Baes, M., Camps, P., 2014. Hierarchical octree and k-d tree grids for 3D radiative transfer simulations. *Astron. Astrophys.* 561, A77.
- Saftly, W., Camps, P., Baes, M., Gordon, K.D., Vandewoude, S., Rahimi, A., Stalevski, M., 2013. Using hierarchical octrees in Monte Carlo radiative transfer simulations. *Astron. Astrophys.* 554, A10.
- Schartmann, M., Wada, K., Prieto, M.A., Burkert, A., Tristram, K.R.W., 2014. Time-resolved infrared emission from radiation-driven central obscuring structures in active galactic nuclei. *Mon. Not. R. Astron. Soc.* 445, 3878–3891.
- Schaye, J., Crain, R.A., Bower, R.G., Furlong, M., Schaller, M., Theuns, T., Dalla Vecchia, C., Frenk, C.S., McCarthy, I.G., Helly, J.C., Jenkins, A., Rosas-Guevara, Y.M., White, S.D.M., Baes, M., Booth, C.M., Camps, P., Navarro, J.F., Qu, Y., Rahmati, A., Sawala, T., Thomas, P.A., Trayford, J., 2015. The EAGLE project: simulating the evolution and assembly of galaxies and their environments. *Mon. Not. R. Astron. Soc.* 446, 521–554.
- Schechtman-Rook, A., Bershady, M.A., Wood, K., 2012. The Three-dimensional distribution of dust in NGC 891. *Astrophys. J.* 746, 70.
- Scheuer, E.M., Stoller, D.S., 1962. On the generation of normal random vectors. *Technometrics* 4, 278–281.
- Semionov, D., Kodaira, K., Stonkute, R., Vasevičius, V., 2006. Radiative transfer problem in dusty galaxies: Photometric effects of dusty spiral arms. *Balt. Astron.* 15, 581–599.
- Sérsic, J.L., 1963. Influence of the atmospheric and instrumental dispersion on the brightness distribution in a galaxy. *Bol. Asoc. Argentina Astron. Plata Argentina* 6, 41–43.
- Springel, V., 2010a. E pur si muove: Galilean-invariant cosmological hydrodynamical simulations on a moving mesh. *Mon. Not. R. Astron. Soc.* 401, 791–851.
- Springel, V., 2010b. Smoothed particle hydrodynamics in astrophysics. *Ann. Rev. Astron. Astrophys.* 48, 391–430.
- Stalevski, M., Fritz, J., Baes, M., Nakos, T., Popović, L.Č., 2012. 3D radiative transfer modelling of the dusty tori around active galactic nuclei as a clumpy two-phase medium. *Mon. Not. R. Astron. Soc.* 420, 2756–2772.
- Stark, A.A., 1977. Triaxial models of the bulge of M31. *Astrophys. J.* 213, 368–373.
- Steinacker, J., Baes, M., Gordon, K.D., 2013. Three-dimensional dust radiative transfer. *Ann. Rev. Astron. Astrophys.* 51, 63–104.
- Stone, J.M., Norman, M.L., 1992. ZEUS-2D: A radiation magnetohydrodynamics code for astrophysical flows in two space dimensions. I—The hydrodynamic algorithms and tests. *Astrophys. J. Suppl.* 80, 753–790.
- Tremaine, S., Richstone, D.O., Byun, Y.-I., Dressler, A., Faber, S.M., Grillmair, C., Kormendy, J., Lauer, T.R., 1994. A family of models for spherical stellar systems. *Astronom. J.* 107, 634–644.
- Trujillo, I., Asensio Ramos, A., Rubiño-Martín, J.A., Graham, A.W., Aguerri, J.A.L., Cepa, J., Gutiérrez, C.M., 2002. Triaxial stellar systems following the $r^{-1/n}$ luminosity law: an analytical mass-density expression, gravitational torques and the bulge/disc interplay. *Mon. Not. R. Astron. Soc.* 333, 510–516.
- van den Bosch, R.C.E., van de Ven, G., Verolme, E.K., Cappellari, M., de Zeeuw, P.T., 2008. Triaxial orbit based galaxy models with an application to the (apparent) decoupled core galaxy NGC 4365. *Mon. Not. R. Astron. Soc.* 385, 647–666.
- Verhamme, A., Schaerer, D., Maselli, A., 2006. 3D Ly α radiation transfer. I. Understanding Ly α line profile morphologies. *Astron. Astrophys.* 460, 397–413.
- Vogelsberger, M., Genel, S., Springel, V., Torrey, P., Sijacki, D., Xu, D., Snyder, G., Nelson, D., Hernquist, L., 2014. Introducing the Illustris Project: simulating the coevolution of dark and visible matter in the Universe. *Mon. Not. R. Astron. Soc.* 444, 1518–1547.
- von Neumann, J., 1951. Various techniques used in connection with random digits. *J. Res. Nat. Bur. Stand.* 12, 36–38.
- Witt, A.N., 1977. Multiple scattering in reflection nebulae. I—A Monte Carlo approach. *Astrophys. J. Suppl.* 35, 1–6.
- Witt, A.N., Gordon, K.D., 1996. Multiple scattering in clumpy media. I. Escape of stellar radiation from a clumpy scattering environment. *Astrophys. J.* 463, 681–693.
- Wood, K., Mathis, J.S., Ercolano, B., 2004. A three-dimensional Monte Carlo photoionization code for modelling diffuse ionized gas. *Mon. Not. R. Astron. Soc.* 348, 1337–1347.
- Xilouris, E.M., Byun, Y.I., Kylafis, N.D., Paleologou, E.V., Papamastorakis, J., 1999. Are spiral galaxies optically thin or thick? *Astron. Astrophys.* 344, 868–878.
- Xilouris, E.M., Kylafis, N.D., Papamastorakis, J., Paleologou, E.V., Haerendel, G., 1997. The distribution of stars and dust in spiral galaxies: the edge-on spiral UGC 2048. *Astron. Astrophys.* 325, 135–143.