

“Groot Onderzoek” project: Looking at the details of SimpleX

Jakob van Bethlehem
Supervisor: Prof. Dr. R. van de Weygaert
Kapteyn Astronomical Institute Groningen

October 25, 2007

Abstract

SimpleX is a new numerical tool to study transport processes in general and radiation transport during the reionization epoch in particular. It is the first method to utilize an unstructured grid. The method hinges on the possibility to set up the grid in such a way that the lengths of the edges in the grid sample the mean free path of the medium. This direct relation between the physics and the grid is theoretically sound, but not yet verified in practice. In this report many details are studied that arise when using unstructured grids. We find some unexpected properties of the grid and confirm the correctness of other properties. The most unexpected result is that photons can travel on average per timestep over larger distances than the mean edge length. The speed even depends on the number of points N used in the grid. Such details need to be understood in order to perform detailed numerical calculations. Another requirement is the addition of a lot of physics that is not yet implemented. This is not a trivial task, because it has to be checked how the physics works on an unstructured grid. Some ideas about how to proceed to include a self consistent solution for the temperature are presented.

1 Introduction

Throughout history mankind has been trying to figure out the mechanisms that shaped the environment we live in. Many different civilizations have come up with even more explanations for the immense diversity of life on Earth and the splendours of the sky above. Some of these were based on scientific knowledge of the particular civilization, some were based on folklore and meant as entertainment or as behavioural and social lessons for youngsters. Some of these 'models' have intriguing similarities with modern day experimentally verified models, like for example the so-called *idols* (eidola) that Roman philosopher Lucretius writes about in his *de Rerum Nature* (About the nature of things)[16]. They were originally introduced by the Greek philosopher Epicurus. He described eidola as particles that stream from all objects and that have the same shape and colour as the object they come from. When eidola hit our eye, they show us the shape and colour of their originating object, very

similar to the concept of photons. Many of such historical insights can be found, but never have they been accepted as superior to other ideas because the available technology didn't admit experimental verification. The situation has changed dramatically during the twentieth century. Now we can actually see individual neutrons and protons and even quarks[2]. For astronomy in particular technological advances led to new telescopes and satellites going above Earth's atmosphere, giving humanity for the first time ever an impression of the enormous physical distances that separate us from the objects in the sky and discovering a plethora of new types of objects and unknown physics by looking at colours of light that are inaccessible to the human eye. This surge of new technology has also brought us the most recent explanation of the Universe: Concordance Cosmology.

The 'Concordance Cosmology' (CC) model combines five concepts into a single framework. The first is the Big Bang, a highly (infinitely) energetic singular event that gave birth to the Universe. The

second is inflation, a period right after the Big Bang during which the Universe expands by an enormous factor $\sim 10^{60}$, thereby flattening the local curvature in the Universe and blowing up Gaussian quantum fluctuations to astronomically scaled primordial density fluctuations. These would eventually collapse to form structures in the Universe. The third ingredient is the Cosmological Principle, which states that the Universe is globally homogeneous and isotropic. It is often overlooked that the WMAP data is *not* sufficient to prove the Cosmological Principle: WMAP only shows that space is *locally* isotropic and homogeneous. If the Universe has some non-trivial topology it is very well possible that the Universe looks locally isotropic while being globally anisotropic[14].

These three components set the global structure of the Universe which is actually not very well known. There are three interrelated questions that need to be answered to find out[9]: what is the spatial curvature, is space open or closed and what is the topology? The curvature can be negative, flat or positive. A positively curved space is automatically also closed, containing a finite amount of matter. Flat and negatively curved spaces can be open, containing an infinite amount of matter but can also have closed spatial sections if they have some nontrivial topology. This possibility is usually ignored, but there are some hints that it is wrong to do so. For example Roukema et al[29] found some clues in the WMAP first year data release that we live in a dodecahedral Universe. It is not clear whether this signal is still present in WMAP3. The simple reason for not understanding the global structure, is the lack of a theory that predicts it. We do have General Relativity, the fourth component of CC, but since that is a differential theory, it only describes the *geometry* which is a local thing. Since most geometries can be supported by many topologies[14] this is not enough. The purpose of General Relativity within the framework of CC is to predict the way that structure forms. And it does an extremely well job at this, together with the fifth component of CC: the long list of numbers that go into the models that tell for example how much matter there is and its nature (Λ CDM) and the power spectrum of the initial density perturbations.

Despite the lack of a precise understanding of the global structure of the Universe, we can make

some definite predictions about its behaviour given the fact that we exist and that we observe an expanding Universe of an enormous scale. The former means that gravity at some point has been able to locally overcome the global expansion caused by the Big Bang. The latter means that no physical mechanism has kicked in yet to overcome the explosive driving force of the Big Bang and, assuming no energy is leaking into our Universe somehow, that the temperature must have dropped. These simple facts are enough to do some astonishing calculations. The behaviour of the temperature for example allows a detailed calculation of nucleosynthesis which accurately predicts the observed values. Knowing that the temperature must have been dropping ever since the Big Bang, it is also inescapably that at some point matter becomes neutral, creating a surface of last scatter where photons have scattered of the last remaining electrons: the CMB. Remembering that gravity at some point must have overcome the global expansion we also know that first stars must have formed. The details of all this will depend on the precise properties of the Universe and in particular the timing of the occurrence of these processes will depend on the input parameters. However at some time in history the Universe must have become ionized again by the radiation coming from the first stars (well: unless the input parameters were such that gravity had overcome the global expansion before last scatter, which would have led to a very different Universe). This is the process of *reionization*.

The epoch of reionization saw two major changes in the Universe. The first is the formation of stars. These led to the formation of all kinds of new atoms (basically: all atoms heavier than Lithium) and eventually to life. The second effect is the slow, patchy ionization of the hydrogen in the Universe. Since the formation of the first stars is of such great importance for understanding the formation of the structure that is observed today, quite some attention has been paid to it. So far only theoretically though, because telescopes aren't able to see these stars. In the (near) future this hopefully will change dramatically, as three consortia are building radio telescopes that should be able to detect the disappearing 21 *cm* signal from neutral hydrogen as it is ionized:

LOFAR[15]:

built by ASTRON[4] and the 'Rekenentrum' of the University of Groningen[27], in the Netherlands

MWA[20]:

built by MIT[18] and the CSIRO Australia Telescope National Facility (ATNF)[5] in Australia

21CMA[1]:

built by the National Astronomical Observatories, Chinese Academy of Sciences (NAOC)[21] in Xinjiang, West China

The foresight that sometime soon observational data will be available for the reionization epoch has given rise to a large number of new numerical packages being developed to simulate this transition. Recently a comparison project was undertaken to compare 11 codes that calculate the radiative transfer of the radiation during the reionization epoch[13]. It showed that the general behaviour of the codes is quite comparable, but in detail they can give quite different results because they use different assumptions and simplifications for the general, seven dimensional radiative transfer problem. Despite the differences, all codes except one had in common that they use regular grids to solve their equations on. The exception is the topic of this research: SimpleX, developed by Ritzerveld[28].

Instead of fixing the grid that is used to solve the equations of radiative transfer (RT), SimpleX is the first code that uses an unstructured grid which automatically adapts to the density field without introducing difficult technical problems. In section 2 we'll introduce transport theory in general, of which the radiative transfer equation is a particular example. In the following section we'll explain how SimpleX deals with transport. In section 4 the input physics for SimpleX is considered. After that we take a look at some details of SimpleX by applying very simple test cases in order to get a solid understanding of the behaviour of SimpleX. In section 7 we proceed by applying SimpleX to simple reionization cases and we give a first go at implementing temperature dependences in SimpleX. We wrap up with discussion and conclusions.

2 Transport theory and radiative transfer

In this section we'll sketch the context for SimpleX. In the end of course it is supposed to solve the equations of radiative transfer, but these are a special case of the much more general class of transfer problems. We need to go back to this more general level to understand how SimpleX works.

But before doing this let's consider the equation we eventually wish to solve, the equation of radiative transfer:

$$\frac{dI_\nu(\vec{r}, \vec{\Omega}, t)}{d\vec{r}} = -\alpha_\nu(\vec{r}, t)I_\nu(\vec{r}, \vec{\Omega}, t) + j_\nu(\vec{r}, \vec{\Omega}, t) \quad (1)$$

It relates the change in the radiative intensity I_ν to the absorption coefficient α_ν and emission coefficient j_ν of the material that the radiation interacts with. Even though at first sight (1) looks like a rather simple first order differential equation, it has some properties that make it notoriously difficult to solve. The first difficulty is that it is a seven dimensional problem: the intensity can depend on place, direction and frequency and all these may depend on time. For particular emission processes the emission coefficient j_ν depends on the intensity I_ν , making (1) a highly non-linear integrodifferential equation. On top of that it is a highly non-local problem, because the photons that 'carry' the intensity can in principle travel to arbitrarily large distances before interacting.

In practice one therefore needs to make simplifying assumptions in order to be able to deal analytically with (1). Different assumptions are used for different specific problems. Some of the common assumptions like homogeneity and isotropy aim at reducing the dimensionality of the problem. Very often also the time dependency is neglected by considering only steady state solutions. But even then there are only very few cases that admit an analytical solution. In order to obtain solutions in more challenging situations numerical methods are needed.

Numerical methods aimed at solving the radiative transfer equation can be subdivided into two classes. Methods of the first class explicitly try to solve a set of differential equations that is derived from (1) after some simplifying assumptions. The differential equations and medium properties

are discretized and solved for using finite difference techniques. A second class of methods utilizes stochastic methods to solve (1). Methods in this class are called Monte Carlo methods and SimpleX falls in this category. In the following we'll take a closer look at the very general transport theory of which radiative transfer is a specific case to explain how Monte Carlo methods can be used to solve the problem of radiative transfer.

2.1 Transport theory[28]

Transport theory is the mathematical framework that describes the movement of particles through some medium. The basic description can be applied in many fields, since it doesn't depend on the type of particles or medium under consideration. That information has to be put in through a source function and collision term. Different forms of these lead to very different applications of transport theory. The general transport equation is readily derived from the even more general kinetic theory.

Kinetic theory aims at analyzing the collective behaviour of a large number of particles. At the basis are the Hamiltonian equations of motion for every particle i :

$$\begin{aligned}\frac{d\vec{v}_i}{dt} &= \frac{1}{m_i}\vec{F}(\vec{x}_i(t), t) \\ \frac{d\vec{x}_i}{dt} &= \vec{v}_i(t)\end{aligned}\quad (2)$$

Here \vec{x}_i and \vec{v}_i with $i = 1, \dots, N$ are the position and velocity of each particle and \vec{F} is the force exerted on the particle, all at time t . Note that in general the force can depend on all other particles as well. If the initial positions and momenta of all particles are known we can in principle solve (2) to find the position $\Gamma_N(t)$ of the system in the $6N$ dimensional phase space at all times t :

$$\Gamma_N(t) = (\vec{x}_1(t), \vec{v}_1(t), \dots, \vec{x}_N(t), \vec{v}_N(t)) \quad (3)$$

Considering that in general N is of the order of Avogadro's number ($N \sim 10^{23}$) it is immediately clear that in practice it is impossible to do this. Therefore the machinery of statistical mechanics is used to derive equations for macroscopic observables based on the microscopic details. This is a

process of 'coarse graining' by contracting and averaging parts of phase space. This is definitely not a trivial thing to do. For example the set (2) is symmetric in time t , leading to the conservation of energy through Noether's theorem. On the other hand we know that macroscopic systems evolve towards maximum entropy, breaking this symmetry. A lot of research is still going on regarding the best way to deal with this type of problems, which we'll not go into. Instead we proceed by introducing the concept of an ensemble, as developed by Gibbs. Instead of considering a single N -particle system and taking averages we can also consider an ensemble of systems. Each member of the ensemble has the same macroscopic properties, but an unspecified phase space distribution. Macroscopic variables in an ensemble are obtained by averaging over the ensemble distribution function $\rho(\Gamma_N, t)$. This ensemble space distribution function is used to obtain a single-particle phase space distribution function $f(\vec{x}, \vec{v}, t)$ by contraction:

$$f(\vec{x}, \vec{v}, t) \equiv \int d\Gamma_{N-1} \rho(\Gamma_N, t) \quad (4)$$

This distribution is used in the general transport equation by equating the full derivative of $f(\vec{x}, \vec{v}, t)$ to the change in density due to collisions and sources:

$$\frac{Df}{Dt} = \frac{\partial f}{\partial t} + \frac{\partial f}{\partial \vec{x}} \cdot \frac{\partial \vec{x}}{\partial t} + \frac{\partial f}{\partial \vec{v}} \cdot \frac{\partial \vec{v}}{\partial t} = \left(\frac{\partial f}{\partial t} \right)_{\text{coll}} + s(\vec{x}, \vec{v}, t) \quad (5)$$

Combining this equation with (2) gives the general transport equation:

$$\frac{\partial f}{\partial t} + \vec{v} \cdot \frac{\partial f}{\partial \vec{x}} + \frac{\vec{F}}{m} \cdot \frac{\partial f}{\partial \vec{v}} = \left(\frac{\partial f}{\partial t} \right)_{\text{coll}} + s(\vec{x}, \vec{v}, t) \quad (6)$$

To proceed from here, the collision and source term need to be specified. This is where all the physics of the particular application comes in. In the case of radiative transfer we deal with photons that are transported through some medium and $I_\nu = h\nu c f$. Substituting this expression for I_ν will result in (1)[8].

2.2 The mean free path

When interactions are localized the transport equation can be solved in terms of the *mean free path*

λ . For a homogeneous background it is given by:

$$\lambda^{-1}(\vec{x}, \vec{v}) = n(\vec{x}) \sigma(\vec{v}) \quad (7)$$

with $n(\vec{x})$ the number density of the background medium and $\sigma(\vec{v})$ the total microscopic cross section for all relevant physical processes. The mean free path is the average distance a particle travels between interactions. Using the concept of the mean free path the transport process can be viewed as a continuous loop over the two-step process as in figure 1. It is this realization that is at the basis of the stochastic approach that we'll discuss next.

repeat for each iteration i while experiment is running:

1. travel a distance s_i such that $\langle s_i \rangle = \lambda$
2. interact with the background

Figure 1: Using the concept of the free mean path, we can model the transport process as a continuous loop over a 'drift' step (step 1) and an interaction step (step 2). The average distance travelled between interactions in step 1 is the mean free path λ .

2.3 Monte Carlo methods

Instead of solving the general transport equation we can also try to explicitly model the transport process, thereby solving the transport problem without ever using the transport equation. Linear transport problems can be modelled analogous to a random walk, like depicted in figure 2. The extra complication is that the stepsize is not constant and interactions have to be put in. Let's see how to do that.

Consider a particle that is just sent in some direction from some position \vec{x} . We assume that the number density n of the background medium is locally constant and consider for simplicity only one type of interaction with cross section σ . Then the probability dp that the particle interacts after a distance ds is:

$$dp = -\frac{1}{\Lambda} p ds \quad (8)$$

where Λ is normalization constant. Solving this equation and normalizing such that $\int_0^\infty p(s) ds = 1$ gives:

$$p(s) = \frac{e^{-s/\Lambda}}{\Lambda} \quad (9)$$

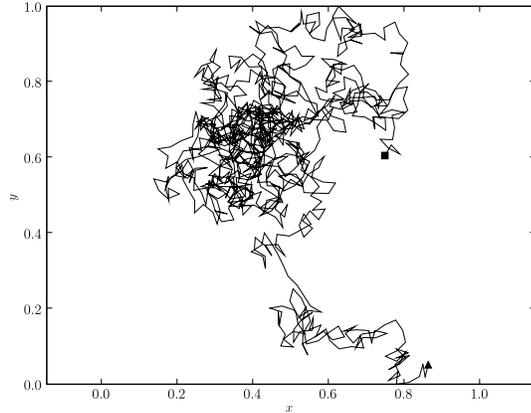


Figure 2: An example of a random walk in the plane for thousand steps. The walk started at the triangle and ended at the square. Every next point is determined by taking a step of fixed length in a randomly chosen direction.

This is the probability distribution for the path length that the particle traverses before it interacts. It's moments are not difficult to find:

$$\langle s^n \rangle = n! \Lambda^n \quad (10)$$

In particular, we see that the expectation value for the travelled distance $\langle s \rangle = \Lambda$. This means that Λ is just the mean free path as defined in (7):

$$\Lambda = \lambda = (n\sigma)^{-1} \quad (11)$$

So by sampling distribution (9) we find distances s_i that satisfy the condition in step 1 of figure 1. The sampling process for this particular distribution can be done in a straightforward way by invoking conservation of probability:

$$|\xi d\xi| = |p(s) ds| \Rightarrow \xi = P(s) \equiv \int_0^s p(t) dt \quad (12)$$

Here ξ is the uniform distribution over $[0, 1]$ and $P(s)$ is the cumulative distribution of (9). This method is also referred to as the direct inversion method[25] but it is really just a conservation law. Realizing that the distribution $(1 - \xi)$ is also a uniform distribution, the solution to (12) is:

$$s = -\lambda \log \xi \quad (13)$$

Given a uniformly distributed random number ξ_i which can be obtained from a random number generator, (13) gives the distance s_i the particle travels before it interacts. So we move the particle to $\vec{x} + \vec{s}_i$ where \vec{s} is a vector of length s_i in the direction the particle was sent to. We are now at step 2 in figure 1. At the new position we let the interaction terms act on the particle. For example consider the transport of photons through a homogeneous medium with an absorption cross section σ_a and a scattering cross section σ_s . The probability p_a that absorption takes places is $p_a = \sigma_a / (\sigma_a + \sigma_s)$ and similar for scattering $p_s = \sigma_s / (\sigma_a + \sigma_s)$. Since $p_a + p_s = 1$, we can get another uniformly distributed number ξ_j from another or the same random number generator and say that absorption has taken place if $0 \leq \xi_j \leq p_a$. If $p_a \leq \xi_j \leq 1$ we call it a scattering event. In the former case the photon is removed from the experiment and in the latter case the photon is re-emitted in some direction Ω . This would normally happen isotropically, but nothing permits us from doing this according to some distribution $g(\Omega)$. After performing the interactions we're back in step 1 and we get the next distance s_{i+1} to travel.

The simplicity of the concept in figure 1 makes Monte Carlo methods very powerful tools to study transport phenomena. With the ever increasing availability of computing power many runs of the experiment can be performed without any additional effort, making it very easy to run experiments with different initial conditions or interaction terms. There are however also some important caveats. The first possible caveat that cannot be stressed enough is the importance of a good random number generator. If subsequent random numbers are related somehow this will show up in the results. Also typically the experiment needs to be repeated many times in order to get some statistics on the possible outcomes. All random numbers that are generated in the process must be unique. So the randomness of separate random numbers must be extremely good, but also the cycle of the number generator must be large enough. Another problem occurs in the limit where the interaction cross section is small, ie $\sigma \rightarrow 0$. In this limit the mean free path $\lambda \rightarrow \infty$, and it becomes impossible to separate the interaction and drift steps. The last complication is the statistical nature of the method. This will introduce inherent statistical noise in the

calculated averages. As the number of runs N is increased, the noise goes as $N^{-1/2}$. With the fast increase in available computer power it shouldn't be problematic though to repeat the process often enough to get to the required noise levels.

Now that we have seen how Monte Carlo methods in general can be used to solve transport problems, let's see how these principles are applied in SimpleX.

3 Transport in the SimpleX way

SimpleX is a package that aims at numerically solving the equation of Radiative Transfer (1) during the recombination epoch using a Monte Carlo approach. The used method is more general however and can in principle be applied to any transport problem as explained in the previous section. All that needs to be changed is the physics that goes in. In this section we describe how SimpleX deals with the general transport problem. At this stage we could still replace any explicit reference to density distributions and photons with any other background medium or transported particle. In section 4 the specialization towards radiation transfer will be made.

3.1 Point Distributions

Any numerical method needs some grid to perform its calculations on. The one aspect that makes SimpleX stand out from the crowd is the use of an unstructured grid. This means that the cells of the grid have different shapes, unlike the standard Cartesian grids that consist only of squares (\mathcal{R}^2) or cubes (\mathcal{R}^3). Methods based on fixed grids have difficulties studying symmetries inherent in the problem because of the symmetry of the grid itself. On top of that they introduce spurious invariants (Ritzerveld[28] and references therein). Another problem is that fixed grids undersample regions with high variability and oversample regions of low variability. Undersampling means that details in the actual solution are missed and oversampling leads to unnecessary computational overhead. Adaptive refinement techniques have been adopted to deal with this problem, but these are

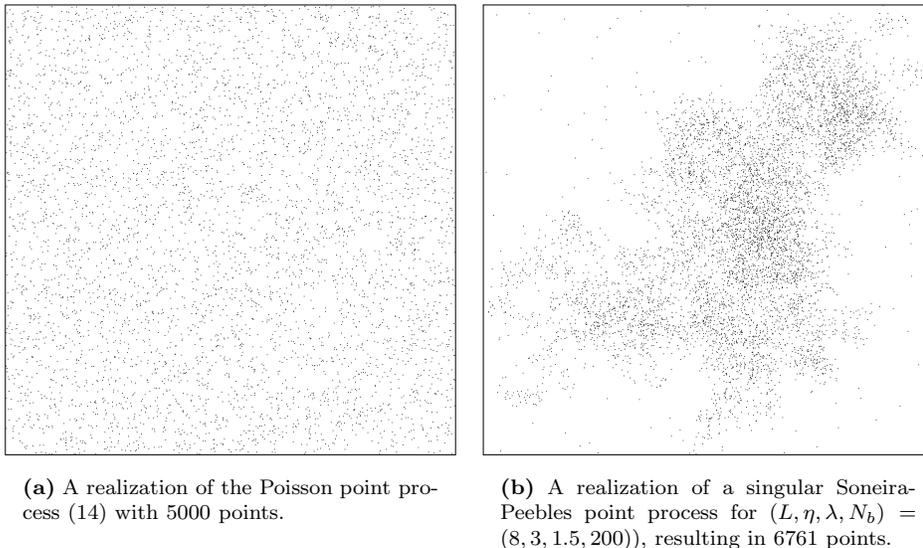


Figure 3: Examples of the two point processes that we'll use in our research.

difficult to implement and the problems with symmetries remain. The unstructured grids used in SimpleX fully adapt to the density field such that both overdense regions have higher resolution and underdense regions have lower resolution without any computational effort. Even more: calculations performed on the grid are independent on the resolution of the grid.

Unlike Cartesian grids, that must be *given* before performing a numerical experiment, unstructured grids are *constructed* from the properties of the medium under consideration using point processes to discretize the underlying medium. The resolution is adapted by using more/less points in dense/thin regions. In the current section we focus on the point processes. In section 3.2 the resulting grids are discussed.

In practice SimpleX will receive densityfields from hydrocodes, like GADGET[10] and solve the radiative transfer (RT) problem for these fields (although work is also going on to try to combine hydrodynamics and SimpleX into a single framework). The most important and critical aspect of SimpleX is the transformation of a given densityfield to the point distribution that is used as basis for the grid. If the density field is homogeneous and isotropic we can use a Poisson point process for this,

since a Poisson process has the same properties. For a given volume $S \subset \mathcal{R}^d$ and $N(A)$ the number of points in a non-empty subset A , the Poisson process defines the probability that the subset A has precisely x points as:

$$\Pr(N(A) = x) = \frac{n_p |A| e^{-n_p |A| x}}{x!} \quad N = 0, 1, 2, \dots \quad (14)$$

where n_p is the global, constant point intensity. In figure 3(a) a possible realization of this process is shown. Clearly this will not do for an inhomogeneous densityfield. In that case we would like to put more points in overdense areas and less points in underdense areas. This can be done by convolving a Poisson point distribution Φ with some well chosen function $f(x)$ of the density distribution $n(\vec{x})$ to get a new point distribution $n_p(\vec{x})$ [28]:

$$n_p(\vec{x}) = \Phi * f(n(\vec{x})) \quad (15)$$

The trick is to choose $f(x)$ such that the resulting point distribution is a good representation of the underlying density field. In section 3.2 we'll describe how a direct coupling between the resulting grid and the physical system can be made by choosing this function cleverly.

3.1.1 Soneira-Peebles point distributions

In order to simulate inhomogeneous density fields for simple test cases we'll use the Soneira-Peebles (SP) model[31]. This is a simple three parameter model that was used by Soneira and Peebles to reproduce the clustering statistics of the observed angular galaxy distribution. In figure 3(b) a possible realization is shown. Creating a SP point distribution is done by recursively adding circles inside other circles. The process starts with a single 'level zero' circle of given radius R . Then a number of η 'level-1' circles is created with the centers at randomly chosen positions within the level-0 circle. These circles have radius R/λ where λ is a constant $\lambda > 1$. Inside each of these circles η level-2 circles are created with a radius that has shrunk by another factor λ , ie $R_2 = R/\lambda^2$. In general we place η level- l circles inside each level- $(l-1)$ circle with radius $R_l = R/\lambda^l$. If L such levels are created, points are placed at the centers of all level- L circles and that is the distribution, which has η^L points. The start of this process is shown in figure 4. The process results in a family of models parametrized by three parameters: L , the number of levels, η , the number of circles to draw in each level and λ , the shrinkage factor. The three parameters give much freedom to vary. There are however also a number of degeneracies between them. Suppose we have a distribution with $N = \eta^L$ points. By simultaneously increasing η and decreasing L we can create distributions with the same N (although in practice it is impossible to get precisely the same N , since η and L must be integer numbers). For a fixed N changing η affects the dynamic range in spatial scales. For a low value of η more levels of circles are needed leading necessarily to smaller circles and therefore a higher dynamic range. A low value of η will also result in a low fillingfactor. For fixed η and L changing the λ parameter works precisely the other way around. By choosing a larger value of λ the circles at each level will be smaller, confining points into smaller regions, thereby boosting the dynamic range and reducing the fillingfactor. The effect of λ is stronger though and acts independent of the choice of N .

Some extensions can be made to this model. Firstly it is possible to create a series of SP point distributions and stack them on top of each other. Such a point distribution is called an extended SP

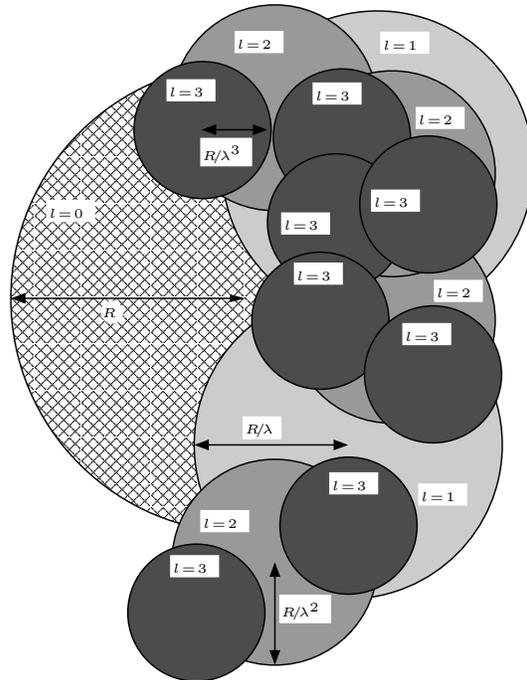
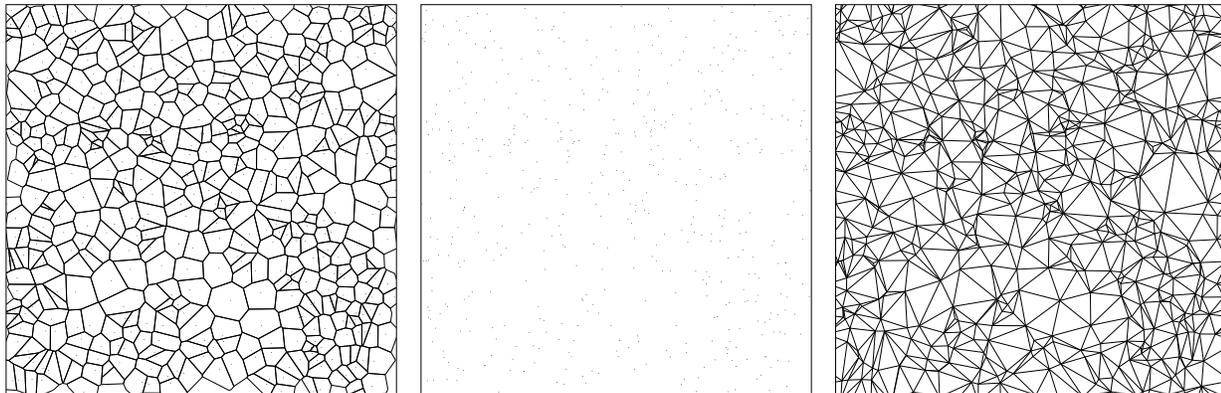


Figure 4: The SP recipe to create a point distribution. Start with a level-0 circle of radius R . Inside the circle choose centers for η level-1 circles with radius $R_1 = R/\lambda$. Repeat this process: create η level- l circles of radius R/λ^l inside each level- $(l-1)$ circle until the level $l = L$ is reached. That level contains η^L circles and the centers of these circles form the distribution. These become the SPS point distribution. The figure shows the possible result at $l = 3$ for a model with $\eta = 2$ and $\lambda \sim 1.5$.

point distribution (SPE). To distinguish this type of distribution from a single distribution, the latter is usually referred to as a singular SP point distribution (SPS). Another extension is the addition of a small Poissonian background. A very inhomogeneous SPS distribution will result in empty regions in the simulation box making it impossible to set up a grid there. Adding a small number N_b of Poissonian distributed points solves this problem. This is the type of distribution that will be used in the rest of this work. An example with $L = 8$, $\eta = 3$, $\lambda = 1.5$ and $N_b = 200$ is shown in figure 3(b).



(a) The Voronoi diagram corresponding to the Poisson point distribution in figure 5(b).

(b) A Poisson point distribution with 500 points. It is the basis for the Voronoi diagram in figure 5(a) and its dual, the Delaunay tessellation in figure 5(c).

(c) The Delaunay tessellation corresponding to the Poisson distribution in figure 5(b).

Figure 5: Comparison of the Delaunay tessellation and its dual, the Voronoi tessellation based on a Poisson point distribution.

3.2 The unstructured grid: the Delaunay tessellation

The process of building unstructured grids from a stochastic point process occurs in many very different places in physics and mathematics. In all these areas the same method has been reinvented over and over again: the Voronoi diagram and its dual, the Delaunay tessellation (because of the reinventions the method goes by many different names, see eg Van de Weygaert[32]). It is the most natural way of dividing up a spatial domain and it has the additional nice property that it conserves rotational and translational symmetries. Very detailed properties can be found in Okabe et al[22]. In astronomy the Voronoi diagram and Delaunay tessellation have been used in many applications: the modelling of galaxies by van de Weygaert[33], the reconstruction of continuous density fields from N-body simulations by Schaap (Delaunay Tessellation Field Estimator, DTFE[30]), the detection of voids in numerical simulations by Platen et al (the Watershed Void Finder, WVF[24]) and the detection of different morphological elements in N-body simulations by Aragon-Calvo et al (the Multiscale Morphology Filter[3]).

A tessellation in general is a collection of equally

shaped polytopes that fit together without overlaps or gaps to cover a given domain. So the Cartesian grid can be seen as a tessellation consisting of squares (\mathcal{R}^2) or cubes (\mathcal{R}^3). Suppose now that we have our distribution of N points $p_i, i \in [1, N]$ with coordinates \vec{x}_i in \mathcal{R}^d . The polytopes, or *cells* V_i , that make up the Voronoi diagram are those regions of space that are closer to p_i than to any other point p_j :

$$V(p_i) = \{\vec{x} \mid |\vec{x} - \vec{x}_i| \leq |\vec{x} - \vec{x}_j|, 1 \in [1, N], j \neq i\} \quad (16)$$

Here norms are just the Cartesian length of the vector. Note that the collection of all Voronoi cells is not strictly a tessellation because the cells have different shapes. Therefore usually the term *Voronoi diagram* is used. An example of a Voronoi diagram based on a Poisson point process in \mathcal{R}^2 is shown in figure 5(a). A Voronoi diagram is a planar graph, meaning that it can be drawn without intersecting edges. The consequence of this is that a *dual* graph must also exist. The dual graph is obtained by connecting all neighbouring cells of the Voronoi diagram with an edge. The result is the Delaunay tessellation, which is indeed a tessellation because the resulting polytopes are the d -dimensional analogue of triangles. The Delaunay tessellation is the un-

structured grid SimpleX is based on. The grid consists of triangles in \mathcal{R}^2 and tetrahedrons in \mathcal{R}^3 . The grid can be determined for any dimension d however and will then consist of the d -dimensional analogue of triangles. Such a generalized d -dimensional triangle is called a *simplex*, hence the name of the method.

3.2.1 Creating a periodic distribution

For the grid to be of any use it is very important to have a special layer of points at each boundary of the box that is not used for calculations, but is only there to assure that the grid behaves nicely near the edges. The problem is illustrated in figure 6(a). This figure shows the Delaunay tessellation for a Poisson point distribution with 200 points without boundary points. The results is that along the boundaries there will be very long edges, almost parallel to the edge of the box. If we were to transfer particles coming somewhere from the center of the box along these lines they would suddenly make almost a 90° turn and move further along the edges of the box. This is clearly not what we want. Another issue is that all analytical results about Delaunay tessellations are based on an 'infinite', or periodic distribution. So in order to compare results with theory, we need to make sure that the points that we use for the actual simulation look like a subset of such an 'infinite' set.

SimpleX gives two ways to do this. The first method doesn't require additional points and thus computer memory. The procedure is to flag any points inside a predefined buffersize along the edges of the box as a 'buffer point'. All particles that end up in one of these bufferpoints are assumed to have flown out of the simulation box and are removed from the experiment. The downside of this method is that the number of gridpoints used for the experiment is reduced. The idea is illustrated in figure 6(b). All points that are within a 10% size buffer of the simulation box are flagged as boundary point. They don't take part in the experiment and basically work as a kind of /dev/null. The effective simulation box is reduced to the smaller indicated box. A second method *adds* points in a given buffer size around the simulation box. These points are the periodic continuation of the points inside the simulation box. So if the buffer size is 10%, then the points that are placed 'right' of the simulation box,

are precisely the points that fall into the 'left' 10% of the simulation box. This process is illustrated in figure 6(c). The original points in the box in figure 6(a) are now inside the black box and the red points are periodically added. The advantage is that if N points are requested to be used, N points is what you get. The downside is that we need to store a lot more points in memory, limiting the number of points we can use.

3.3 Transport on the Delaunay grid

Delaunay tessellations based on Poisson point processes have been studied extensively and many properties are known[22], but only very few things allow analytical evaluation. One particular property that will prove to be extremely useful is the expectation value of the moments of the edglength L in the Delaunay tessellation[22]:

$$\langle L^k \rangle = \zeta(k, d) n_p^{-k/d} \quad (17)$$

Here $\zeta(k, d)$ is a geometrical constant that depends on the moment k and the dimension d . We can compare this relation with the expectation values $\langle s^k \rangle$ for the distance s a particle travels between interactions. In section 2.3 we found $\langle s^k \rangle \propto \lambda^k$. If we compare this with (17) it follows that if the point distribution would scale with n_p^d , the expectation value for the edglengths L scales in the same way as the expectation value for the physical distance s . In other words: if we choose the function $f(x)$ defined in (15) to behave as $f(x) \propto x^d$, such that:

$$n_p(\vec{x}) = \Phi * n^d(\vec{x}) \quad (18)$$

then all moments of the distribution of the lengths of the edges in the Delaunay tessellation are proportional with the moments of the distribution of the physical distance s :

$$\langle L^k \rangle(\vec{x}) = c(k) \lambda^k(\vec{x}) \quad (19)$$

Here $c(k)$ is a constant of proportionality that depends on k but besides that is a global constant. In particular we have that the average edglength of the Delaunay tessellation equals the mean free path multiplied by a global constant:

$$\langle L \rangle(\vec{x}) = c_1 \lambda(\vec{x}) \quad (20)$$

The consequence is that the edges of a Delaunay tessellation created from a Poisson point process

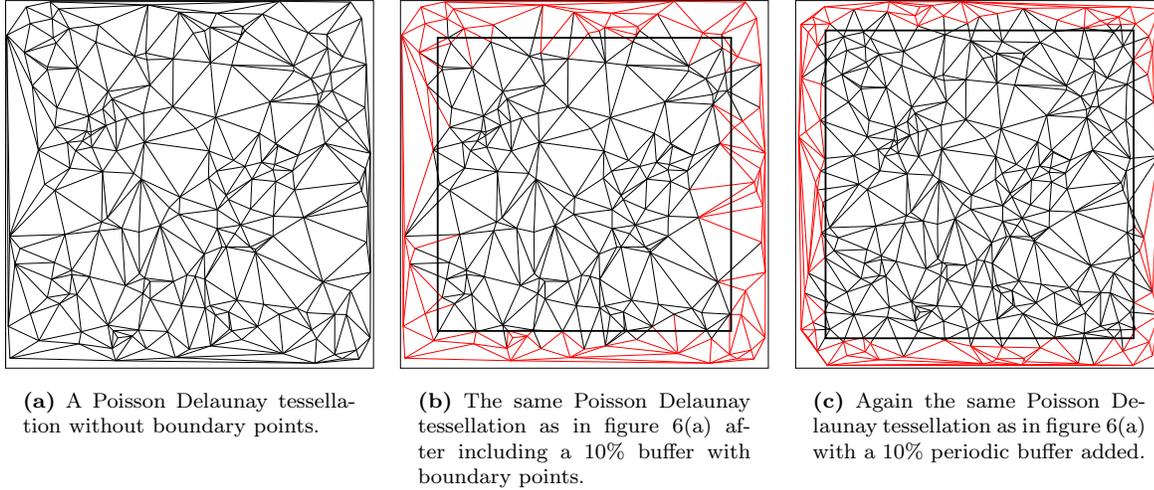


Figure 6: A Poisson Delaunay tessellation without boundary points gives rise to unwanted transport of particles parallel to the edges of the simulation box. This situation is illustrated in figure 6(a). In figure 6(b) this is resolved by flagging all points within a predefined buffer as boundary point. This buffer is plotted in red. Points in this boundary do not take part in the transport process, they remove any particle that enters from the simulation. The number of points used in the simulation is reduced to the points within the black box. Another method is to periodically add points at the boundary, in order to retain the number of simulation particles. This method is illustrated in figure 6(c). The red points are the periodically added points. Now the original simulation box is conserved inside the black box.

satisfying (18) will accurately sample the mean free path λ travelled by a particle (9).

With this in mind, let’s take a new look at 1. Performing a Monte Carlo simulation was said to be a two-step process. The first step was to get a sample of the distribution (9) for s and move the particle over that distance. The second was to apply the physics connected with the particular problem under consideration. Now that we have a grid with edges that are such that they sample the mean free path the first step becomes trivial. All we have to do for step 1 is to move particles from a given vertex in the grid to the neighbouring vertex in order to satisfy the condition $\langle s \rangle = \lambda$ and all that’s left to do is to include the physics. At this stage any physical process can still be put in. For SimpleX the physics consists of the transport of photons that interact with hydrogen. We will describe how this is included in the next section.

4 The physics

In this section we consider how the SimpleX way of transporting entities in a background medium is specialized to deal with photons traveling through a background density field. The current version of SimpleX deals with hydrogen only and is not able to solve for temperature. Instead it assumes a temperature of 200 K for neutral hydrogen and 10^4 K for ionized hydrogen. Further it deals with a single frequency, for which the ionization threshold of $13.6 \text{ eV}/h$ is used. We can take into account a general source spectrum S_ν by weighing physical quantities with the normalized source spectrum:

$$\tilde{S}_\nu \equiv \frac{S_\nu}{\int_{\mathcal{R}} S_\nu d\nu} \quad (21)$$

SimpleX allows the usage of a Blackbody source - $S_\nu = B_\nu(T_s)$ - for which the normalized Blackbody “distribution” looks like:

$$\tilde{B}_\nu(T_s) = \frac{15}{\pi^4} \left(\frac{h}{kT_s} \right)^4 \frac{\nu^3}{e^{h\nu/kT_s} - 1} \quad (22)$$

with T_s the (effective) temperature of the source in Kelvin.

4.1 Ionization cross section

The full blown expression for the ionization cross section of hydrogen is given by[11]:

$$\sigma_\nu(HI) = A_0 \left(\frac{\nu_0}{\nu}\right)^4 \frac{e^{4-4 \arctan \epsilon/\epsilon}}{1 - e^{-2\pi/\epsilon}} \text{ cm}^2 \quad \nu \geq \nu_0 \quad (23)$$

with:

$$\begin{aligned} A_0 &= 6.30 \cdot 10^{-18} \text{ cm}^2 \\ \epsilon &= \sqrt{\frac{\nu}{\nu_0} - 1} \quad \nu_0 = 3.29 \cdot 10^{15} \text{ Hz} \end{aligned}$$

and with ν_0 the ionization threshold $h\nu_0 = 13.6 \text{ eV}$. In practice we'll use the Kramers approximation[11]:

$$\sigma_\nu^{\text{Kramers}}(HI) = A_0 \left(\frac{\nu_0}{\nu}\right)^3 \text{ cm}^2 \quad \nu \geq \nu_0 \quad (24)$$

Both in (23) and (24) the cross section is zero for $\nu < \nu_0$. They are plotted in figure 7. The cross section peaks to A_0 at ν_0 and then falls off approximately like ν^{-3} . This approximation underestimates the cross section at the most 25% around 10^{16} Hz . It is not a very tight fit, but it is a reasonable tradeoff against the computational difficulties we would experience when using (23). We have to be careful however when we want to deal with much higher energies. Around 10^{17} Hz both relations are equal and for even higher frequencies the Kramers approximation overestimates the cross section. The ratio grows with a factor 10 every two decades in frequency (ie, at $\nu \approx 10^{19} \text{ Hz}$ the approximation overestimates the full expression by a factor ≈ 10).

We'll use the Kramers approximation to calculate the average ionization cross section:

$$\bar{\sigma}(T_s) = \int_{\nu_0}^{\infty} \tilde{B}_\nu(T_s) \sigma_\nu^{\text{Kramers}} d\nu \quad (25)$$

The resulting integral is evaluated in appendix B.2, equation (85):

$$\bar{\sigma}(T_s) = \frac{15A_0}{\pi^4} \left(\frac{h\nu_0}{kT_s}\right)^3 \ln\left(\frac{1}{1 - e^{-h\nu_0/kT_s}}\right) \text{ cm}^2 \quad (26)$$

The average ionization cross section is plotted in figure 8.

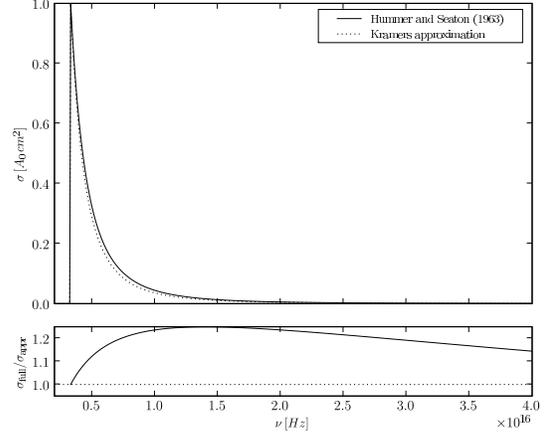


Figure 7: The hydrogen cross section. The full expression (23) is the solid line. The cross section peaks at the ionization threshold $\nu = 3.29 \cdot 10^{15} \text{ Hz}$ and then falls off approximately like ν^{-3} . The Kramers approximation (dashed line, eq (24)) follows this general behaviour. The ratio of the full expression and the approximation is also given. The approximation underestimates the cross section at the most 25% around 10^{16} Hz .

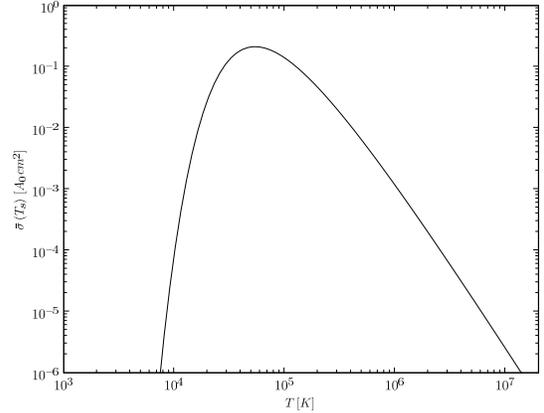


Figure 8: The hydrogen cross section as function of source temperature T_s , after averaging over frequency, weighed with a normalized Blackbody.

4.1.1 Ionization rate

Using the average hydrogen cross section, the optical depth encountered when moving from a point p_i in the tessellation to its neighbour p_j , is[28]:

$$\Delta\tau = n_{\text{HI}} \bar{\sigma}(T_s) \Delta L \quad (27)$$

with n_{HI} the number density of neutral hydrogen and ΔL the physical distance traveled. For a monochromatic flux of frequency $13.6 \text{ eV}/h$ the average cross section in this relation should be replaced by A_0 . The equation of radiative transfer (1) now tells us what happens with the photons that move along the edge between the neighbours. Rewriting (1) in terms of optical depth gives:

$$\frac{dI_\nu}{d\tau_\nu} = -I_\nu + S_\nu \quad (28)$$

where we left out all the explicit dependencies on \vec{r} , $\vec{\Omega}$ and t , $S_\nu = j_\nu/\alpha_\nu$ is the sourcefunction and $d\tau_\nu = \alpha |d\vec{r}|$. When considering radiation transport from p_i to p_j we are trying to solve (28) along a line for which $S_\nu = 0$, since radiation can only be created in gridpoints. Then the solution to (28) is:

$$I_\nu(\tau_\nu) = I_{0,\nu} e^{-\tau_\nu} \quad (29)$$

where $I_{0,\nu}$ is the radiation that is send by p_i in the direction of p_j . The radiation absorbed underway is then:

$$I_\nu^{\text{absorbed}}(\tau_\nu) = I_{0,\nu} (1 - e^{-\tau_\nu}) \quad (30)$$

This result is used to calculate the number of photons that is absorbed by neutral hydrogen. Given that N_{photons} is the number of photons that is send from p_i [28]:

$$N_{\text{photons}}^{\text{used}} = f_{\nu_0} N_{\text{photons}} (1 - e^{-\Delta\tau}) \quad (31)$$

where f_{ν_0} is the fraction of ionizing photons. In practice the multiplication with this factor is performed at the source such that only ionizing photons are distributed over the grid. For a monochromatic flux at threshold frequency we have $f_{\nu_0} = 1$. For a Blackbody source it becomes:

$$f_{\nu_0} = \frac{\int_{\nu_0}^{\infty} B_\nu(T_s) d\nu}{\int_0^{\infty} B_\nu(T_s) d\nu} = \int_{\nu_0}^{\infty} \tilde{B}_\nu(T_s) d\nu \quad (32)$$

We can evaluate this fraction numerically by using (92) from appendix B:

$$f_{\nu_0} = \frac{15}{\pi^4} I \left(3, \frac{h\nu_0}{kT} \right) \quad (33)$$

The actual number of newly ionized atoms is calculated from (31) as:

$$\Delta N_{\text{HII}} = \begin{cases} N_{\text{photons}}^{\text{used}} & N_{\text{photons}}^{\text{used}} \leq N_{\text{HI}} \\ N_{\text{HI}} & N_{\text{photons}}^{\text{used}} > N_{\text{HI}} \end{cases} \quad (34)$$

An important question is whether it is allowed to go directly from (29) to (30) in the described manner. Transporting a number of photons N_ν is not the same as transporting energy I_ν since:

$$N_\nu = \frac{I_\nu}{h\nu} \quad (35)$$

This relation suggests that we should use a cross section weighted with the normalized form of $B_\nu(T_s)/h\nu$ to calculate the correct optical depth. We will come back to this issue in section 7.1.

4.2 Recombination

A full blown expression for the case B recombination coefficient is given by[7]:

$$\alpha_B = 8.40 \cdot 10^{-11} T^{-1/2} T_3^{-0.2} (1 + T_6^{0.7})^{-1} \text{ cm}^3 \text{ s}^{-1} \quad (36)$$

where now T is the temperature of the *gas* and T_i is a shortcut for $10^{-i}T$. The coefficient is plotted in figure 9 as the solid black line. From the figure we see that the recombination rate has two limits. For temperatures $T \lesssim 10^5 \text{ K}$ it behaves as a powerlaw with a rather shallow exponent and for temperatures $T \gtrsim 10^5 \text{ K}$ it behaves as a powerlaw with a sharp exponent (for $T \rightarrow \infty$ we have $T_i \propto T$ and approximately $\alpha_{\text{HII}} \propto T^{(-0.5)+(-0.2)+(-0.7)} = T^{-1.4}$). Around 10^5 K we're in a transition region. Since one expects temperatures around 10^4 K for an ionized hydrogen gas, in SimpleX (36) is approximated by a single powerlaw:

$$\alpha_B = 3.22 \cdot 10^{-13} T_4^{-0.73} \text{ cm}^3 \text{ s}^{-1} \quad T \lesssim 10^5 \text{ K} \quad (37)$$

The constant was chosen such that (36) and (37) coincide at 10^4 K . This approximation is plotted as the dashed black line in figure 9. Note that the constants in (37) are slightly different than in the original version of SimpleX[28] (dotted black line in figure 9) to accomodate a better fit.

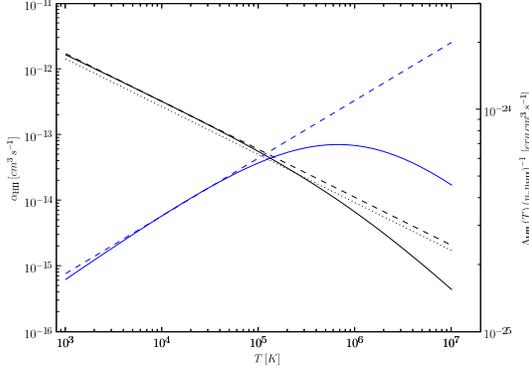


Figure 9: The recombination rate in black and the cooling in blue. The black solid line is expression (36) from [7]. Since temperatures around $10^4 K$ may be expected for an ionized hydrogen gas, in SimpleX the full expression is approximated by a single powerlaw (37), valid for $T \lesssim 10^5 K$ and chosen such that (36) and (37) coincide at $T = 10^4 K$. The dashed black line shows this approximation. The constants in this approximation are slightly different from the original version of SimpleX, which is plotted as the dotted black line. The blue solid line shows the recombination cooling rate (63). The region $T \lesssim 10^5 K$ can be approximated by a single powerlaw like the recombination rate, resulting in (64) which is plotted as the dashed blue line.

4.2.1 Recombination rate

Using the recombination coefficient the recombination rate is usually written as:

$$\frac{\partial n_{\text{HI}}}{\partial t} = \alpha_B n_{\text{HII}}^2 = \alpha_B x^2 n^2 \quad (38)$$

In SimpleX however we deal explicitly with the number of atoms, so we have to write for the number of recombined atoms N_{rec} :

$$N_{\text{rec}} = \alpha_B N_{\text{HII}} x n \quad (39)$$

The change in the number of neutral hydrogen atoms is now determined from:

$$\Delta N_{\text{HI}} = \begin{cases} N_{\text{rec}} & N_{\text{rec}} \leq N_{\text{HII}} \\ N_{\text{HII}} & N_{\text{rec}} > N_{\text{HII}} \end{cases} \quad (40)$$

5 The details of SimpleX

In the two previous sections 3 and 4 we described globally how SimpleX works. In this section we

take a look at the details and perform very simple test cases to see whether SimpleX behaves as expected.

5.1 QHull

The most striking difference between SimpleX and other radiative transfer codes is the underlying grid used to solve the radiative transfer problem. All traditional methods use Cartesian grids, whereas SimpleX utilizes an unstructured grid. This grid is the Delaunay tessellation corresponding to stochastically obtained points. These represent the density distribution in the simulation box in such a way that the lengths of the edges of the tessellation sample the mean free path λ a photon travels between interactions. But how to efficiently calculate the tessellation from the point distribution? Suppose we have N points $p_i, i \in [1, N]$ in d -dimensional space as input. In principle it is not very difficult to let the computer calculate the corresponding tessellation, but straightforward algorithms will typically have very bad time complexity. Consider for example the very simple algorithm in figure 10. This is a conceptually very simple approach that is easy to implement, but it has a time complexity of $\mathcal{O}(N^2)$ because for every point we need to check all other points. We can speed up the algorithm slightly by making sure that we don't look for an edge $p_j \rightarrow p_i$ if we have already found $p_i \rightarrow p_j$, but even then it is a $\mathcal{O}(N^2)$ algorithm, which is a huge problem for cosmological simulations where billions of particles are needed. Another important problem is that the algorithm has problems calculating distances between points that are very close together because of roundoff errors. This can potentially destroy the tessellation completely.

SimpleX uses the package QHull[26] to calculate the tessellation. This software is numerically relatively stable and runs in the theoretically best obtainable time complexity which is $\mathcal{O}(N \log N)$ [34] in \mathcal{R}^2 and \mathcal{R}^3 . It makes use of a remarkable relation between the convex hull of a set of points and the Delaunay tessellation discovered by Brown[6]. The convex hull of a set of points is the outer boundary such that all points are enclosed. An example in \mathcal{R}^2 is shown in figure 11. The QHull algorithm works as follows:

1. Start with points $\vec{x}_d^i = (x_d^1, x_d^2, \dots, x_d^d)$ in \mathcal{R}^d

For each point $i \in [1, N]$:

1. calculate the distances between i and all other points $j \in [1, N]/\{i\}$
2. select the d points $\{j_1, j_2, \dots, j_d\}$ with smallest distance to i

the set of points $\{i, j_1, \dots, j_d\}$ forms a Delaunay simplex; store this simplex

Figure 10: A straightforward algorithm to calculate the Delaunay tessellation of a point distribution consisting of N points in d -dimensional space. This algorithm is easy to implement, but has $\mathcal{O}(N^2)$ time complexity which becomes problematic even for not too high values of N . It will also suffer strongly from roundoff errors when calculating distances between points that are very close together.

2. Lift these points to a paraboloid in \mathcal{R}^{d+1} where $\tilde{x}_{d+1}^i = (x_d^1, x_d^2, \dots, x_d^d, \sum_{i=1}^d (x_d^i)^2)$. So we lift the points above the hyperplane \mathcal{R}^d in \mathcal{R}^{d+1} where the height equals the square of the distance each point is away from the origin
3. Compute the convex hull of the obtained paraboloid in \mathcal{R}^{d+1}
4. Project the downward facing faces of the convex hull. Brown[6] proved that this is the Delaunay tessellation we are looking for.

For \mathcal{R}^3 QHull determines both the Delaunay tessellation and the Voronoi diagram. For \mathcal{R}^2 QHull only calculates the Delaunay tessellation. We have to derive the Voronoi diagram from that ourselves. This problem is equivalent to finding the centers of the circumscribing circle for each Delaunay triangle (for details, see eg Okabe et al[22]). There are many ways to do this. SimpleX originally used a method based on the slope of the edge between points of the triangle. The problem with this is that the slope can evaluate to zero for some situations. Another method is given by Van de Weygaert[33]. We use a variation of that method that requires less algebraic calculations. The method is explained in appendix A.

In sections 5.1.1, 5.1.2 and 5.1.3 we perform a few simple tests on tessellations in a box $[0, 1]^d$ calculated by QHull in order to check its embedding in SimpleX and to find the limits of its applicability.

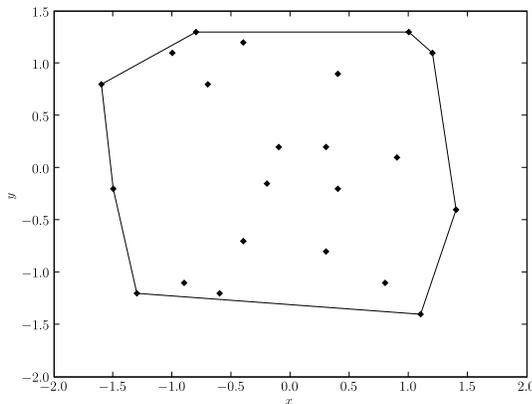


Figure 11: The convex hull of a set of points in \mathcal{R}^2 . The convex hull of a set of N points $p_i, i \in [1, N]$ in \mathcal{R}^d is the outer boundary of the set consisting of $(d-1)$ -dimensional hypersurfaces, such that all points fall inside the boundary.

5.1.1 Number of edges in vertex

There are not many properties of a general Delaunay tessellation that can be calculated analytically. But when we restrict ourselves to a Poisson Delaunay tessellation, ie a tessellation based on a Poisson point process, there are a few. One known property in \mathcal{R}^2 and \mathcal{R}^3 is the expectation value $E(E)$ for the number of edges E that intersect in a vertex (gridpoint). These edges correspond precisely to the number of $d-1$ hypersurfaces that make up the d dimensional Voronoi cell. They are [22]:

$$\begin{aligned} E(E)_{\mathcal{R}^2} &= 6 \\ E(E)_{\mathcal{R}^3} &= \frac{48\pi^2}{35} + 2 \approx 15.535 \end{aligned} \tag{41}$$

Despite the fact that the number of vertices E must be an integer number, the expectation value is only an integer for \mathcal{R}^2 . The way to test this prediction is to set up many tessellations, count the number of edges and create histograms of the distribution $p(E)$ of E . We create 100 tessellations from a Poisson point process with $N = 10^4$. For each tessellation we calculate a normalized histogram and then the average distribution. The resulting mean histograms are plotted in figures 12(a) and 12(b) for $d = 2$ and $d = 3$ respectively. The errorbars indicate the unbiased sample standard deviation.

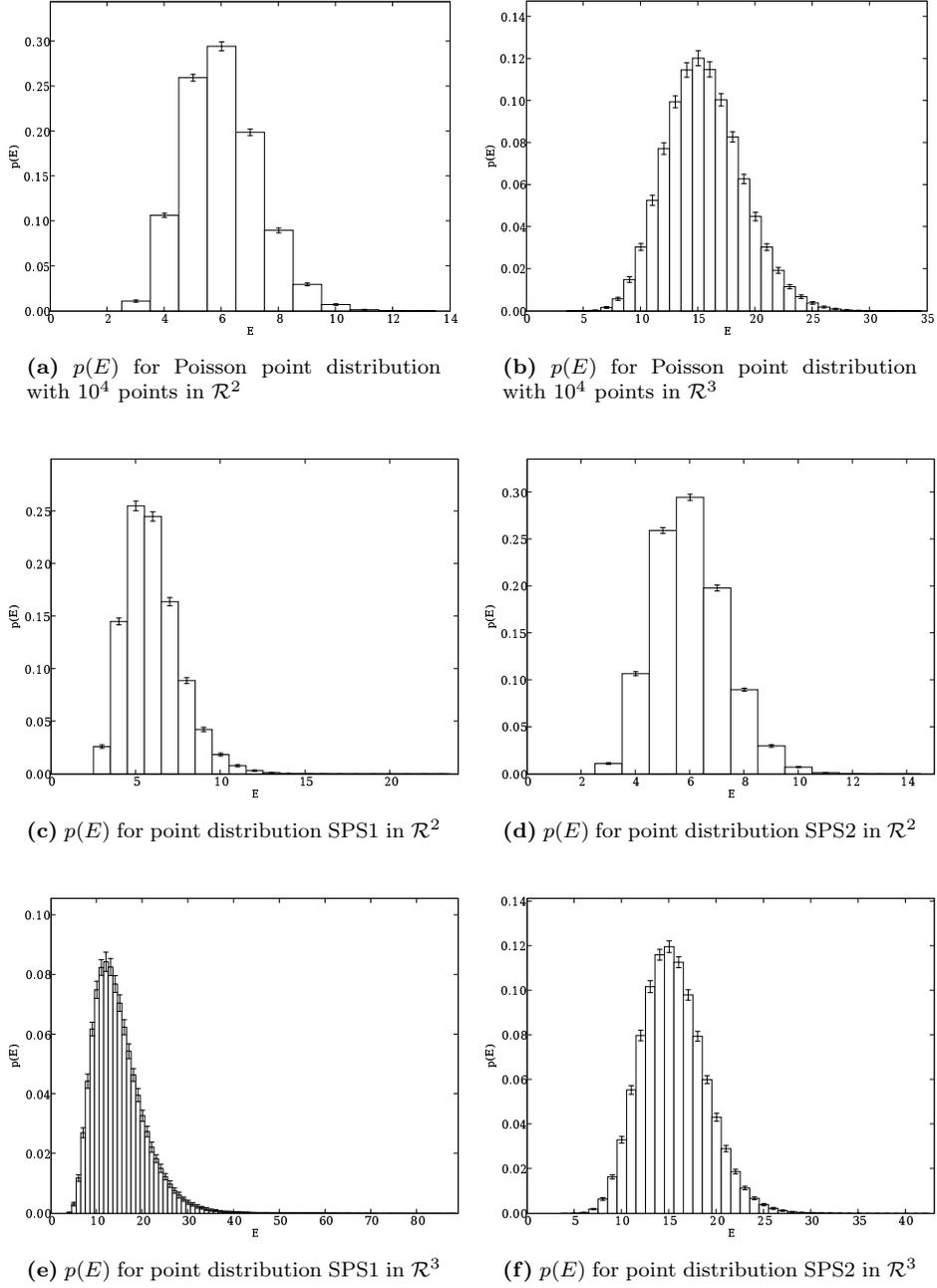


Figure 12: The distributions $p(E)$ of the number of edges E arriving at a vertex in a Poisson Delaunay tessellation with $N = 10^4$ and realizations of SPS1 and SPS2. For each case 100 realizations were averaged. The errorbars indicate the unbiased sample standard deviation. The expectation values are (42) for the Poisson cases and (43) and (44) for the SPS cases.

Given this distribution we can numerically calculate the expectation values for E :

$$\begin{aligned} E(E)_{2D} &= 6.000 \pm 0.052 \\ E(E)_{3D} &= 15.54 \pm 0.15 \end{aligned} \quad (42)$$

These results agree very well with the analytical results. We have also repeated the experiment for $N = 5 \cdot 10^3$. Since the statistical noise should go $\propto N^{-1/2}$ we expect the errorbars to increase by a factor $\sqrt{2}$ for this case and this is indeed observed.

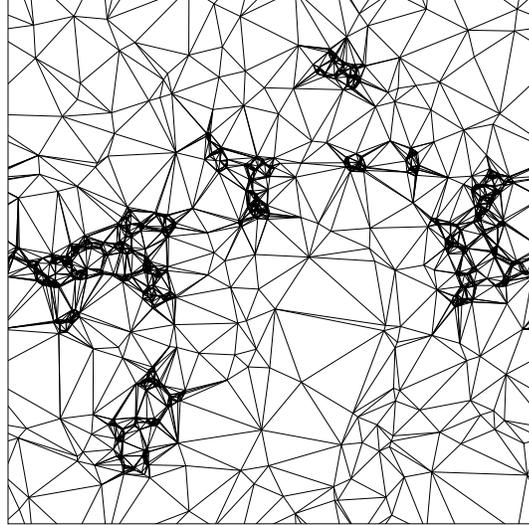
We can do the same test for SPS point distributions. In particular the case for $d = 2$ is very interesting because the expectation value for E should be independent on the point distribution. In other dimensions this is not generally true. We have tested this for two SPS distributions. The first - SPS1 - has $(L, \eta, \lambda, N_b) = (2, 13, 1.8, 200)$ giving $N = 8392$. This is an extremely inhomogeneous distribution with only a few very dense spots. The second - SPS2 - is more homogeneous with $(L, \eta, \lambda, N_b) = (5, 6, 1.2, 200)$ giving $N = 15825$. Examples of such distributions are shown in figure 13. In figures 12(c) and 12(d) we plot the resulting distributions in \mathcal{R}^2 . The expectation values are:

$$\begin{aligned} E(E)_{\text{SPS1}, \mathcal{R}^2} &= 6.000 \pm 0.058 \\ E(E)_{\text{SPS2}, \mathcal{R}^2} &= 6.000 \pm 0.039 \end{aligned} \quad (43)$$

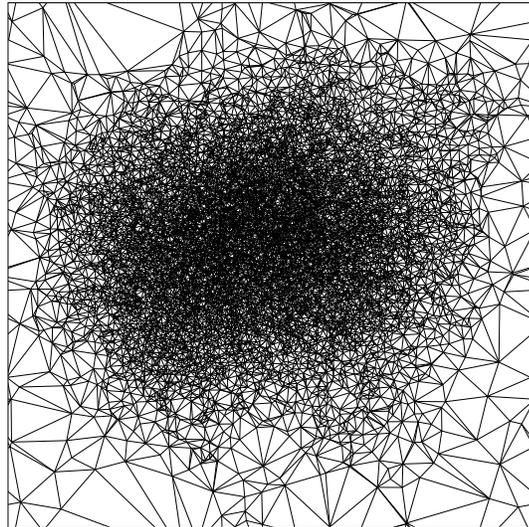
As expected the expectation value doesn't change, but if we look at figures 12(c) and 12(d) it is clear that the shape of the distribution *can* change. For the inhomogeneous SPS1 distribution the maximum of the distribution has shifted to $E = 5$ and the tail extends to much larger values of E . The SPS2 realization results in a distribution that is hardly distinguishable from the Poisson result in figure 12(a). Despite the deviating shape of $p(E)$ for SPS1, the uncertainties in (43) behave $\propto N^{-1/2}$ as they should.

To be complete we investigated the same SPS point distributions for $d = 3$ as well. The resulting distributions are shown in figure 12(e) and 12(f) and the expectation values are:

$$\begin{aligned} E(E)_{\text{SPS1}, \mathcal{R}^3} &= 14.87 \pm 0.17 \\ E(E)_{\text{SPS2}, \mathcal{R}^3} &= 15.46 \pm 0.12 \end{aligned} \quad (44)$$



(a) An example of a SPS1 distribution with $(L, \eta, \lambda, N_b) = (2, 13, 1.8, 200)$ giving $N = 8392$.



(b) An example of a SPS2 distribution with $(L, \eta, \lambda, N_b) = (5, 6, 1.2, 200)$ giving $N = 15825$.

Figure 13: Examples of the very inhomogeneous SPS1 distributions (figure 13(a)) and more Poisson-like SPS2 distributions (figure 13(b)).

Qualitatively the results in \mathcal{R}^3 are very comparable to \mathcal{R}^2 . For SPS1 the distribution shifts towards lower E and has a much longer tail. This time

however also the expectation value changes to lower E . The more homogeneous SPS2 distribution is virtually indistinguishable from the Poisson case, even though the expectation value seems to deviate more than expected when assuming it is constant. The uncertainties are again $\propto N^{-1/2}$.

5.1.2 Edge length distribution

One of the critical properties of the Delaunay tessellation is the distribution of the length L of the edges, because these lengths are supposed to sample the mean free path λ . In practice when considering the transport of photons along a given edge from point p_j to point p_i not the actual length of that edge is used, but the average length of all edges connected to p_i . Because the edges themselves should sample the mean free path, a good estimate of the mean free path itself ought to be obtained by explicitly taking the average. We can test this easily for a Poisson point distribution. Because of the homogeneity of the Poisson point distribution we can calculate the global average edgelenhth $\langle L \rangle$ and take this as the 'true' mean free path and then compare the results in all vertices with this global average.

We start by considering the edgelenhth distribution $p(L)$ for a Poisson point distribution with 10^4 points. We create 100 realizations and calculate the corresponding Delaunay tessellation. Then we calculate the lengths L of each edge and the averages \bar{L} of the lengths connected to each vertex p_i . For each realization the resulting distribution $p(L)$ is calculated by a binning process and finally the mean of these distributions is determined. Errors in the mean are estimated by the unbiased sample standard deviation. Also the cumulative distribution $P(L)$ is determined:

$$P(L) = \int_0^L p(L) dL \quad (45)$$

The results for \mathcal{R}^2 are presented in figures 14(a) and 14(b). In figures 14(c) and 14(d) are the results for \mathcal{R}^3 . The distribution of the edgelenhth L is plotted as a solid line with square datapoints and the distribution of the mean length \bar{L} is plotted as a dashed line with diamond datapoints in all figures. For the cumulative distributions the errors are about the size of the datapoints and the

errorbars are therefore left out. The lengths in all figures are normalized with respect to the global average edgelenhth $\langle L \rangle \approx \lambda$. We see that both in \mathcal{R}^2 and \mathcal{R}^3 the distribution of the averages \bar{L} has a much narrower peak around λ indicating that indeed the method of local averaging gives a better estimate for λ than all individual edgelenhths L . Another nice property is that the distribution of \bar{L} is more symmetric. If we compare between dimensions we see that the distributions for \mathcal{R}^3 are much narrower than for \mathcal{R}^2 . This is actually no surprise because this is known from the analytical distributions for L (which we didn't plot because it is very hard to do that, see eg Ritzerveld, equations (3.17) and (3.20)[28]). Next we calculate the expectation values of these distributions:

$$E(L)_{\mathcal{R}^2} = (0.9619 \pm 0.0064) \langle L \rangle$$

$$E(\bar{L})_{\mathcal{R}^2} = (0.9331 \pm 0.0091) \langle L \rangle \quad (46)$$

$$E(L)_{\mathcal{R}^3} = (0.9752 \pm 0.0041) \langle L \rangle$$

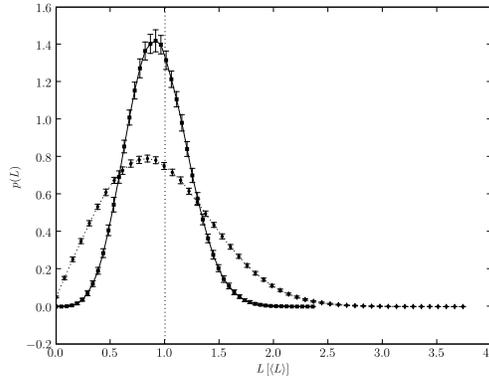
$$E(\bar{L})_{\mathcal{R}^3} = (0.9508 \pm 0.0094) \langle L \rangle \quad (47)$$

So even though from the figures the distributions of the mean seem to be closer to λ , the actual expectation value for the means are further away from λ . From the uncertainty it seems to be a very well determined offset though. SimpleX doesn't have a correction for this offset and we will add this soon. An important check before applying this correction is to look for possible dependencies of the expectation values on N . Therefore we repeated the full process for $N \in \{5e2, 1e3, 5e3\}$. The resulting expectation values are shown in figures 14(e) and 14(f). Even though we don't have many datapoints it seems fair to say that there is no dependency on N , both for the distributions of L and \bar{L} in \mathcal{R}^2 and \mathcal{R}^3 . Finally we can compare the results with analytical results. For \mathcal{R}^2 and \mathcal{R}^3 the expectation value of the moments k of the edgelenhth L is known[19]:

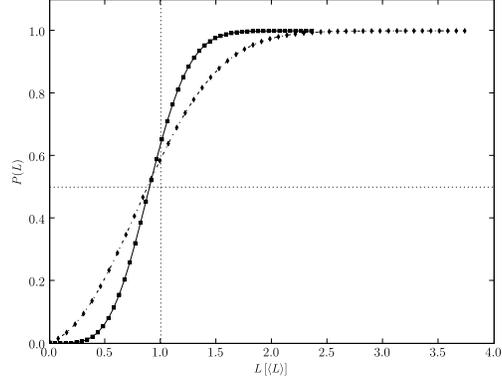
$$E(L^k)_{\mathcal{R}^2} = \frac{2^{k+1} (k+1) (k+3) \Gamma\left(\frac{k+1}{2}\right)}{3 (k+2) \pi^{(k+1)/2} N^{k/2}} \quad (48)$$

$$E(L^k)_{\mathcal{R}^3} = \frac{35}{32} \frac{(k+8) (k+6)}{(k+7) (k+5) (k+3)} \Gamma\left(3 + \frac{k}{3}\right) \left(\frac{6}{\pi N}\right)^{k/3} \quad (49)$$

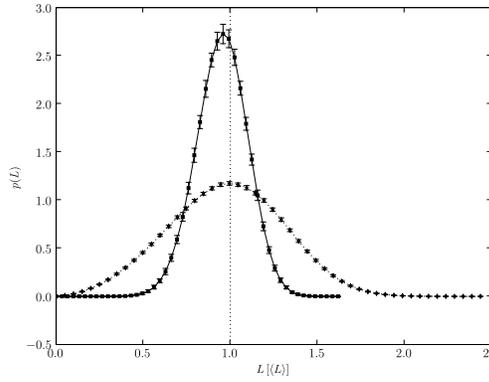
Note especially the brackets in the $\pi^{(k+1)/2}$ term for $E(L^k)$ in \mathcal{R}^2 . The brackets are missing both in



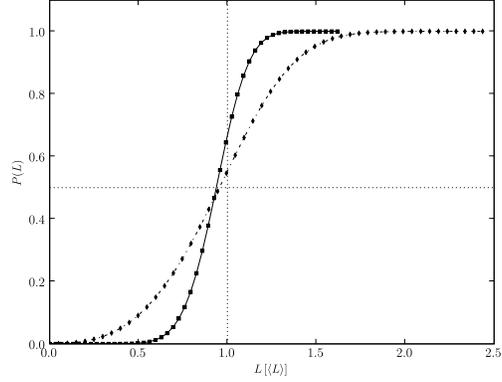
(a) Edgelenh distribution $p(L)$ for Poisson point distribution with 10^4 points in \mathcal{R}^2 .



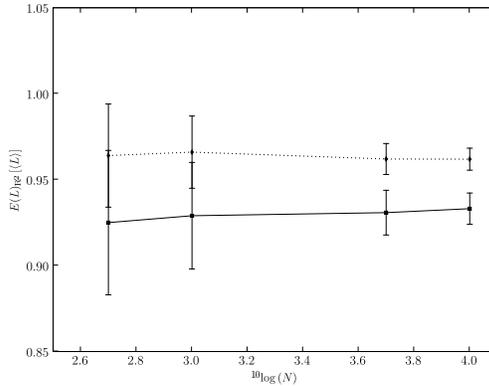
(b) Cumulative edgelenh distribution $P(L)$ for Poisson point distribution with 10^4 points in \mathcal{R}^2 .



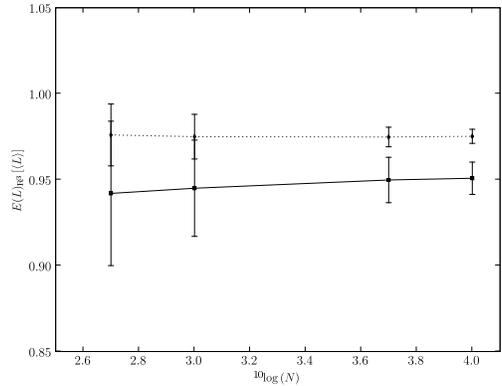
(c) Edgelenh distribution for Poisson point distribution with 10^4 points in \mathcal{R}^3 .



(d) Cumulative edgelenh distribution for Poisson point distribution with 10^4 points in \mathcal{R}^3 .



(e) Expectation value in \mathcal{R}^2



(f) Expectation value in \mathcal{R}^3

Figure 14: Figures 14(a), 14(b), 14(c) and 14(d) show the edgelenh distributions $p(L)$ in the Delaunay tessellation based on a Poisson point distribution. The datapoint in each bin is the average over 100 different realizations. The errorbars represent the unbiased sample standard deviation in each bin. For the cumulative distributions the errorbars are not plotted, because they are comparable in size to the datapoints. In all figures the solid line with square datapoints shows the distribution of the edgelenh L and the dotted line with the diamond datapoints shows the distribution of the local average edge length \bar{L} . In all figures the lengths are normalized with respect to the global average edgelenh $\langle L \rangle \approx \lambda$. In figures 14(e) and 14(f) the expectation values for L and \bar{L} are plotted for \mathcal{R}^2 and \mathcal{R}^3 respectively. Again the solid line with square points corresponds to edgelenhs and the dashed line with diamond points to the local means.

equation (3.18) of Ritzerveld[28] and in Ch 5.11 of Okabe et al[22]. Setting $k = 1$:

$$\begin{aligned} E(L)_{\mathcal{R}^2} &= \frac{32}{9\pi} n^{-1/2} \approx 1.132N^{-1/2} \\ E(L)_{\mathcal{R}^3} &= \frac{1715}{2304} \left(\frac{3}{4\pi}\right)^{1/3} \Gamma(1/3) N^{-1/3} \\ &\approx 1.237N^{-1/3} \end{aligned} \quad (50)$$

In figure 15 we plotted $\langle L \rangle$ for \mathcal{R}^2 in black with square datapoints and in red with diamond datapoints for \mathcal{R}^3 . The solid lines are the theoretical curves given by (50). The datapoints are the averages over the 100 realizations. The uncertainties in the points are about the same size as the points themselves. For \mathcal{R}^2 the datapoints fall right on top of the theoretical curve. For \mathcal{R}^3 there is a clear offset however. We also plotted the theoretical curve multiplied with the reciprocal of the offsets in (46) and (47) (dotted lines). After this correction the datapoints for \mathcal{R}^3 fall right onto the theoretical curve, but the datapoints in \mathcal{R}^2 are now all offset. It is not clear at this time why the situation is different in \mathcal{R}^3 and \mathcal{R}^2 and more attention is needed.

5.1.3 Angular distribution

Another property of Poisson Delaunay tessellation that is analytically known is the distribution $p(\theta)$ of the angle θ between adjacent edges in \mathcal{R}^2 . It is given by[12]:

$$p(\theta) = \frac{4 \sin \theta}{3\pi} (\sin \theta + (\pi - \theta) \cos \theta) \quad (51)$$

We calculated for 100 tessellations based on 10^4 Poissonian distributed points the angles between each set of adjacent edges. For each tessellation the distribution was calculated and then the average per bin was determined. The result is depicted in figure 16(a). The errorbars represent the unbiased sample standard deviation in each bin. The solid line is the analytical result (51). The general shape of the distribution is very well reproduced, but the peak is shifted slightly to lower values of θ . This is also very clearly visible in the cumulative distribution in figure 16(b) where the points are again the data and the solid line is the numerically obtained

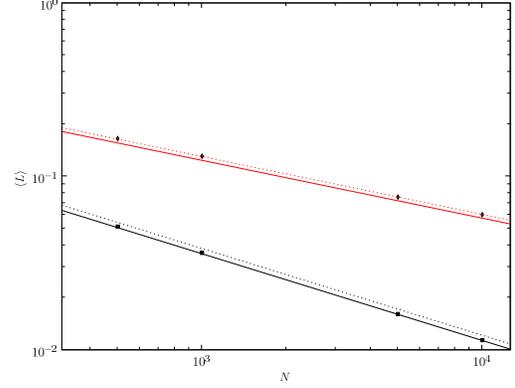
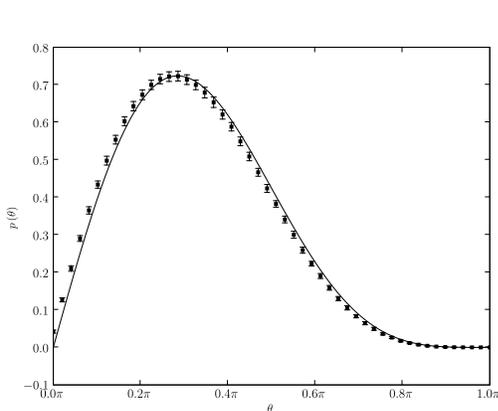


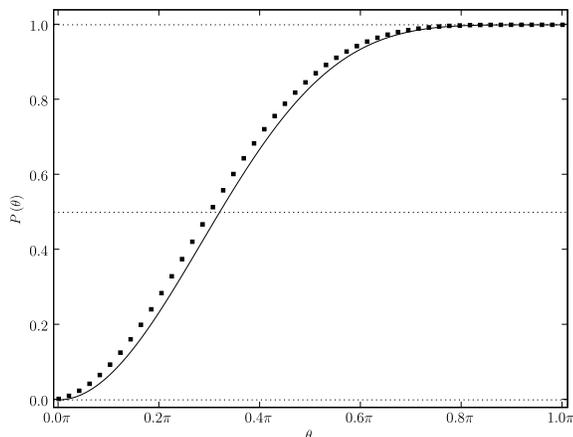
Figure 15: The mean of the global averages of the edgelenhth for 100 Poisson Delaunay tessellations for \mathcal{R}^2 (square black points) and \mathcal{R}^3 (diamond red points). The errors in the datapoints are about the size of the points. The overplotted solid lines are the theoretical expectation values for the edgelenhth given by (50). In \mathcal{R}^2 the data perfectly matches this expectation value, but for \mathcal{R}^3 there is an offset. If we apply the offsets found in (46) and (47) we get the dotted lines. Now the data in \mathcal{R}^3 matches perfectly, but in \mathcal{R}^2 there is an offset. It is unclear what causes this difference.

cumulative distribution for (51). The cause for the shift to smaller angles is not yet clear.

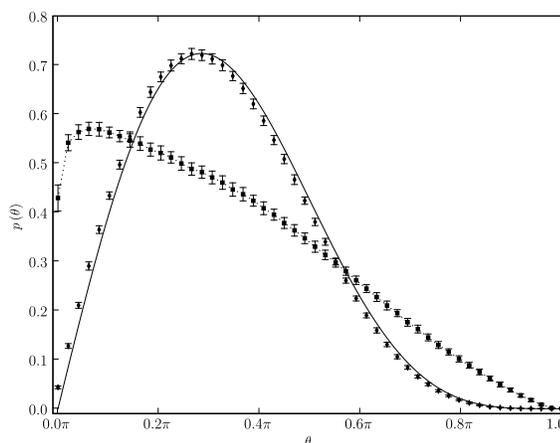
It is also interesting to see what we get for our SPS distributions SPS1 and SPS2 which we introduced in section 5.1.1. The results for SPS1 are plotted in figures 16(c) and 16(d) with square points with a dotted line. The results for SPS2 are plotted as diamond points and the solid lines are again the theoretical curve for a Poisson Delaunay tessellation, equation (51). As before the SPS2 results are very similar to the results for a Poisson point distribution, even the offset to smaller angles is also present for SPS2. The results for SPS1 however are completely different. The maximum probability for a SPS1 Delaunay tessellation occurs for a very small angle and then $p(\theta)$ drops off approximately linearly. So in comparison with a Poisson Delaunay tessellation we have much more very small angles, but also much more large angles in a SPS1 Delaunay tessellation. The latter must be true because in section 5.1.1 we saw that in \mathcal{R}^2 the number of edges emanating from a given vertex is independent of the point distribution. Since the sum of all angles surrounding a given vertex must



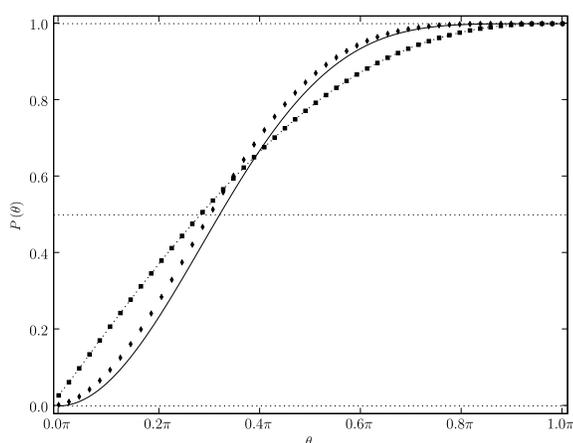
(a) The distribution $p(\theta)$ of the angle θ between adjacent edges. The errorbars indicate the unbiased sample standard deviation.



(b) The cumulative distribution $P(\theta)$ of the angle θ between adjacent edges. The errorbars are too small to picture.



(c) The distribution $p(\theta)$ for SPS1 (square points, dotted line) and SPS2 (diamond points) realizations. The theoretical curve (51) for Poisson Delaunay tessellation is plotted as a solid line. The errorbars are the unbiased sample standard deviation.



(d) The cumulative distribution $P(\theta)$ for SPS1 (square points, dotted line) and SPS2 (diamond points) realizations. The solid line is the numerically obtained cumulative distribution for the theoretical distribution (51). The errorbars are not plotted because they are about the size of the datapoints.

Figure 16: The distribution of the angle θ between adjacent edges in Poisson Delaunay tessellations (figures 16(a) and 16(b)) and SPS tessellations (figures 16(c) and 16(d)). The datapoints are the average in each bin over 100 realizations of each the tessellation. The solid line in all figures is the theoretical result (51). The Poisson data match the general shape of the theoretic distributions very well, but are slightly offset to smaller angles. It is not clear what causes this. Figures 16(c) and 16(d) show the results for SPS1 and SPS2 realizations, as defined in section 5.1.1. The SPS1 results are plotted as square points with a dotted line. The SPS2 results are the diamond points. The solid line in both figures is the theoretical results for a Poisson Delaunay tessellation. The SPS1 type of distribution is very inhomogeneous and results in many very small angles. In section 5.1.1 we showed that the number of edges emanating from a vertex is independent of the underlying point distribution and therefore there are also many large angles to make the sum 2π . The SPS2 distribution looks again very much like a Poisson distribution, including the offset to smaller angles.

always be 2π it follows automatically that small angles must be compensated for by large angles.

5.1.4 Limits of QHull

Throughout most of the last sections we have used around 10^4 points. For the Poisson distributions exactly this number and for the SPS1 and SPS2 realizations a number close to that (SPS1 has $N = 2^{13} + 200 = 8392$ and SPS2 has $N = 5^6 + 200 = 15825$ points). The reason is that for these type of numbers, in particular for \mathcal{R}^2 we can be quite safe that the resulting tessellation is correct. If we keep increasing N QHull will start to complain at some point that the points in the distribution are too close to each other. If we increase N even a bit further, than QHull fails and SimpleX crashes with a segmentation fault. The actual number N^{tot} that we can use as input depends on the settings of the buffers we discussed in section 3.2.1. What counts is of course the total number of points that goes into QHull. For an input of 10^4 Poisson points and a periodic buffer of 10% we have $N^{\text{tot}} \approx 1.44 \cdot 10^4$ and for a buffer of 50% even $N^{\text{tot}} \approx 4 \cdot 10^4$ points. The number of points actually used in the simulation is $N^{\text{sim}} \equiv N$ for this case. For a non periodic buffer we have $N^{\text{tot}} \equiv N$, but $N^{\text{sim}} \approx 0.64 \cdot 10^4$ for a 10% buffer and $N^{\text{tot}} = 0$ for a 50% buffer. We have tested QHull for Poisson distributions with different values of N^{tot} to see where numerical problems occur. In \mathcal{R}^2 the maximum allowed number of points is $N_{\mathcal{R}^2}^{\text{tot}} \approx 5 \cdot 10^5$ corresponding to a minimal edgelenhth of about $L_{\text{min}} \approx 1.4 \cdot 10^{-3}$. For \mathcal{R}^3 we went to $N^{\text{tot}} \approx 5 \cdot 10^6$ and at that point the computer ran out of memory. If we take the maximum possible value in \mathcal{R}^2 as an indication of the smallest allowed distance between two points, we can estimate that for \mathcal{R}^3 QHull should do well up to $N_{\mathcal{R}^3}^{\text{tot}} \approx (N_{\mathcal{R}^2}^{\text{tot}})^{3/2} \approx 3.5 \cdot 10^8$.

5.2 Impulse conservation

So far we have looked solely at properties of the tessellation. An important consequence of the application of the Delaunay tessellation as grid is that photons can take only discrete, precalculated paths through the medium. So let's consider a photon packet that is coming from points p_j and is moving towards point p_i , like in figure 18.

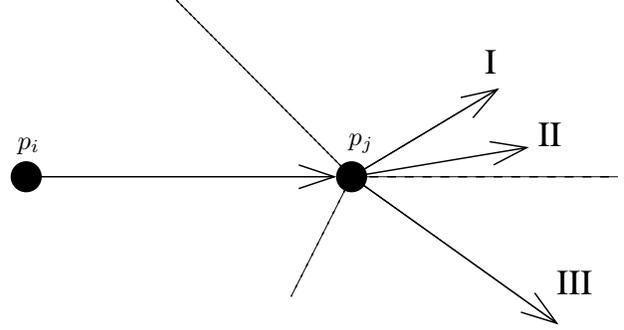


Figure 18: Photon packets that perform random walks over the Delaunay grid meet junctions all the time. Ideally photons that are not absorbed should travel further down the dashed line. This is not possible because of the discretization of the possible paths a photon packet can take. This is a side effect of using tessellations as grid. In SimpleX photon packets that arrive at a junction are split into d equally sized packets and are send along the 'straightest' edges, the edges that are closest to the dashed line. This C-method will send photon packets along edges I and II. Another method is the E-method (Enclosing). In this method the edges are chosen such that they enclose the dashed line, which is the original direction. For that scenario SimpleX would choose edges II and III. At this point it is not clear which method should be used.

Some photons of the packet will be absorbed (according to equation (31)). The remaining photons need be send further down the grid. Ideally we would like to send the remaining photons just straight on along the dashed line. But because the grid consists of a discrete number of edges such a path will generally not be there. So we have to choose an edge, let's say we pick the edge with the smallest deflection, edge II. Independent of the resulting deflection, the result is that we artificially introduce impulse in a direction perpendicular to the original direction. Given the fact that the angle between any two adjacent edges has an upper limit of 180° (in \mathcal{R}^2) this effect can be rather dramatic. For this reason SimpleX splits the photon package in two and sends the packets along the 'straightest' edges. In \mathcal{R}^d these are the d edges that deviate the least from the original direction. In figure 18 SimpleX would pick edges I and II. We will refer to this method as the C-method ('Closest' method). It is clear that this method doesn't reduce the impulse

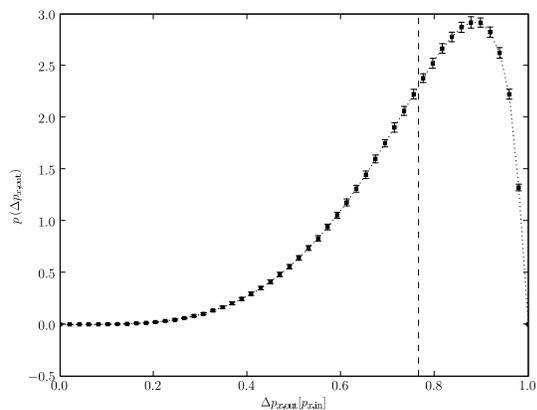
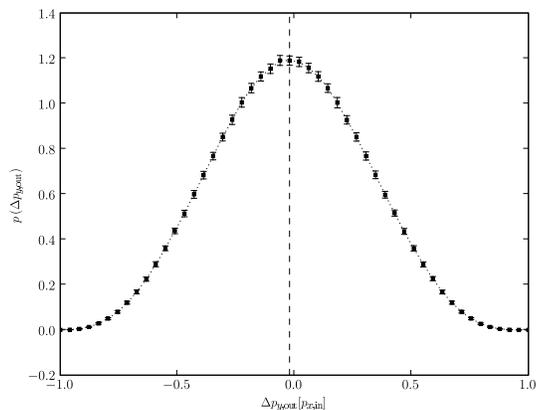
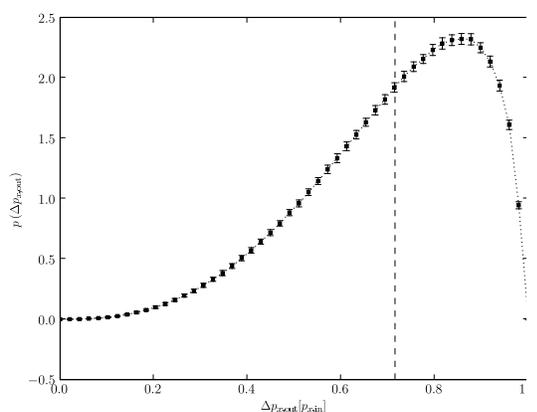
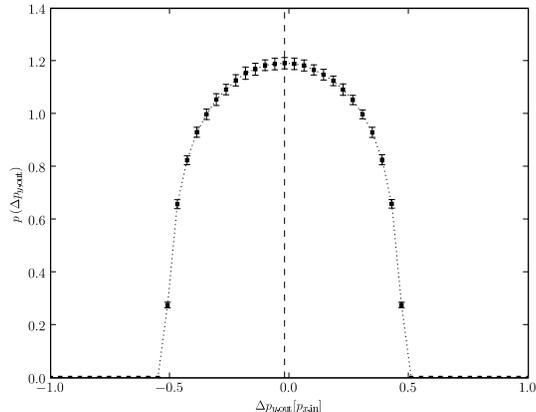
(a) The distribution of $\Delta p_{x,\text{out}}$ for the C-method.(b) The distribution of $\Delta p_{y,\text{out}}$ for the C-method.(c) The distribution of $\Delta p_{x,\text{out}}$ for the E-method.(d) The distribution of $\Delta p_{y,\text{out}}$ for the E-method.

Figure 17: The distribution of the total outgoing momentum in units of the incoming momentum. For each junction in a Poisson Delaunay tessellation the x -axis is placed along the incoming edge and the total outgoing momentum is determined along the x -axis and the y -axis. The standard method of choosing two edges in SimpleX is the C-method, where in \mathcal{R}^d the d edges are chosen that are closest to the original direction. The results for this method are shown in figure 17(a) and 17(b). Another possibility is to enforce that the original direction is 'enclosed' in the new directions, the E-method. The results for this case are shown in figure 17(c) and 17(d).

in the perpendicular direction(s) to zero locally, but the hope is that on average it does. We will investigate this by calculating the total change in momentum that would occur at each junction in a Poisson Delaunay tessellation with 10^4 points. For each junction we place the x -axis along the incoming edge and calculate the total momentum that goes out in units of the incoming impulse $p_{x,\text{in}}$ (so actually we are just calculating the sum of two cosinus and two sinus terms). Again we do this for 100 realizations and average the distributions obtained for each realization. The resulting distributions for the total outgoing impulse in the x -direction $p_{x,\text{out}}$ and y -direction $p_{y,\text{out}}$ are presented in figures 17(a) and 17(b). Ideally one would like to see spikes at $\Delta p_{x,\text{out}} = 1$ and $\Delta p_{y,\text{out}} = 0$. This is clearly not the case. The distribution of $\Delta p_{y,\text{out}}$ does look very symmetric though. Let's calculate the expectation values for these distributions:

$$\begin{aligned} E^{\text{C}}(\Delta p_{x,\text{out}}) &= 0.7655 \pm 0.0033 \\ E^{\text{C}}(\Delta p_{y,\text{out}}) &= -0.0205 \pm 0.0013 \end{aligned} \quad (52)$$

These values are plotted as the dashed lines in figures 17(a) and 17(b). The results show that the artificially introduced impulses perpendicular to the incoming direction do not average out. More important however is what happens if we also take into account the full orientation of the edges in the simulation box. That is what we'll do in section 5.3.

But before doing that it is interesting to see what happens if we make a small change in the method. The idea is inspired by the notion that the situation sketched in figure 18 where photons are sent in the same y -direction happens about once out of every 7 times. To be precise: the frequency of occurrence is $(13.92 \pm 0.12)\%$ for a Poisson Delaunay tessellation with 10^4 points (the value doesn't depend on N though). In those cases a massive amount of impuls can be introduced. We can enforce that the edges are chosen such that they 'enclose' the original direction. When this Enclosing, or E-method is applied, edges II and III would be selected. In figures 17(c) and 17(d) we plotted the resulting distributions for the impulses. The distribution for $p_{x,\text{out}}$ looks very similar, except that the expectation value has shifted to a slightly lower value. The distribution for $p_{y,\text{out}}$ on the other hand

has changed dramatically. The central value is still about the same, but the distribution has widened and now has a cut-off. So for the E-method it can be guaranteed that particles will never deflect more than 90° . More attention is needed to decide which of the methods should be used in SimpleX. We close this section with the expectation values for the E-method:

$$\begin{aligned} E^{\text{E}}(\Delta p_{x,\text{out}}) &= 0.7153 \pm 0.0031 \\ E^{\text{E}}(\Delta p_{y,\text{out}}) &= -0.0204 \pm 0.0010 \end{aligned} \quad (53)$$

5.3 Random walks

In the last section we saw that the distribution of $\Delta p_{y,\text{out}}$ is quite symmetric, when averaged over the full tessellation. However in practice photon packets execute a random walk along the grid and meet only a limited number of junctions. It remains to be seen whether the distribution stays as symmetric as before when following a photon packet along the grid. So let's test this. Choose the x -axis along the original direction of the photon packet. If the symmetry holds, the following following expression can be derived for the distance travelled after n steps of the random walk process[28]:

$$\Delta x = \sum_{i=1}^n \eta^i \rightarrow \frac{\eta}{1-\eta} \quad (54)$$

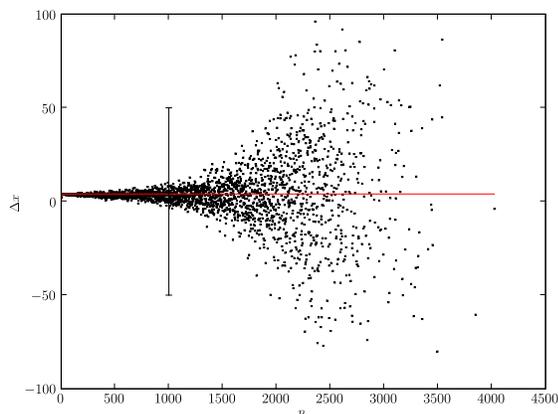
and

$$\Delta y = 0 \quad (55)$$

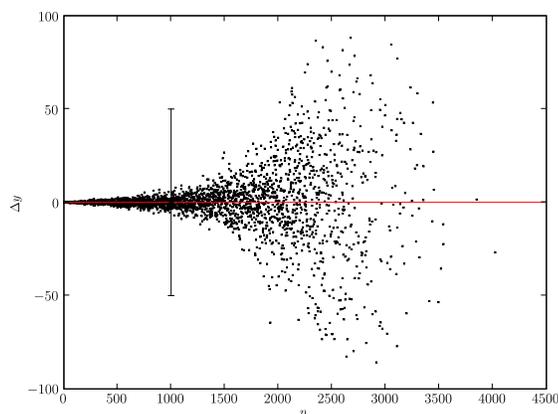
where:

$$\eta = \int_{-\pi}^{\pi} f(\theta) \cos \theta d\theta \quad (56)$$

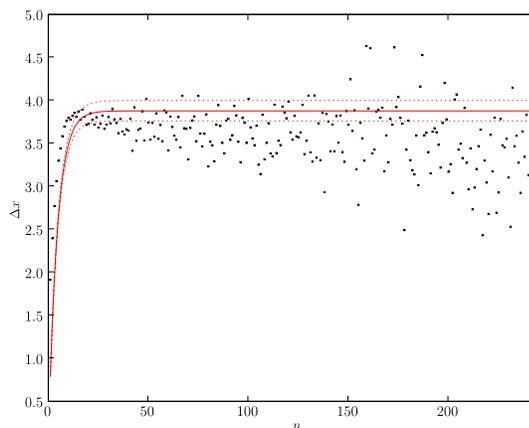
with $f(\theta)$ the distribution function of the deflection angle θ which has to be symmetric to get these results. In figures 19(a) and 19(b) we plotted the resulting average values of Δx and Δy after walks of different length n on a Poisson Delaunay tessellation with 10^4 points. The results are for the E-method. The results for the C-method are globally the same. We first consider Δy . The expected zero line is plotted in red. The averages are quite nicely centered around $\Delta y = 0$, but the scatter, even for $n \lesssim 1500$ is already $\sigma_{\Delta y} \approx 50$ as indicated by the single errorbar. The same problem



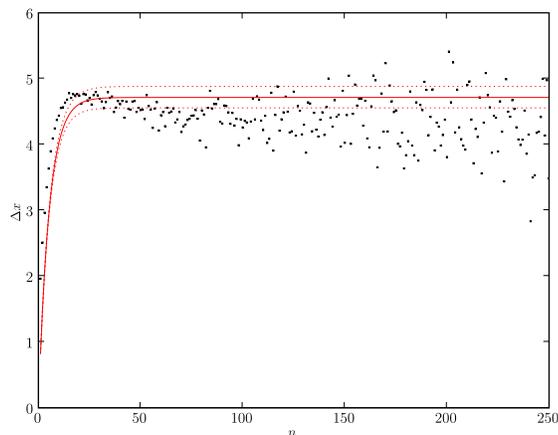
(a) Δx after random walks of n steps in a Poisson Delaunay tessellation using method E. The red line is the theoretically expected result (54) with $\eta = 0.795$. On average the data fits the theoretical curve quite well, but the scatter is very large.



(b) Δy after random walks of n steps in a Poisson Delaunay tessellation using method E. The red line is the theoretically expected zero line (55). On average the results seem to comply, but the scatter is very large.



(c) Zoom in for method E to the smallest values of n . The theoretical curve is overplotted in red for $\eta = 0.795$. The dotted lines are the theoretical curves for $\eta \pm 0.005$.



(d) Zoom in for method C to the smallest values of n . The theoretical curve is overplotted in red for $\eta = 0.825$. The dotted lines are the theoretical curves for $\eta \pm 0.005$.

Figure 19: The displacement with respect to an initial direction along the x -direction is an important factor for a transport routine. Figures 19(a) and 19(b) show the total displacement along the original direction Δx and the perpendicular direction Δy after random walks of different length n on a Poisson Delaunay tessellation with 10^4 points. The depicted results are for the E-method, but they are globally indistinguishable from the C-method. Only when we zoom in to very small n can we see differences. In figures 19(c) and 19(d) we show the results after zooming in for respectively the E-method and the C-method. The theoretical curve (54) is overplotted in red. The vertical position of the bend and the plateau in the theoretical curve depends very sensitively on the value of η . This allows a rather precise determination of that value. For these figure we have $\eta = 0.795$ for method E and $\eta = 0.825$ for method C with for both numbers $\sigma_\eta \approx 0.005$.

occurs for Δx . The red line depicts the theoretical curve for $\eta=0.795$ and the data follows that curve quite well. But also here the scatter is on the order of $\sigma_{\Delta x} \approx 50$. So in general equations (54) and (55) should not be used to give estimates on Δx and Δy . Their distributions are much too wide to get reliable estimates from theory and photons will scatter away much further than anticipated from the equations. Let's finally take a very close look at Δx . If we zoom in to the region $n \lesssim 250$ we can see the 'turn' in the distribution of Δx . This turn is quite sensitive to the value of η . In figures 19(c) and 19(d) the details in this region are shown for respectively the E-method and C-method. The red curves are again the theoretical curve (54). For the E-method we have $\eta^E = 0.795$ and for the C-method we have $\eta^C = 0.825$ to obtain these figures. An estimate of the uncertainty of this values is $\sigma_\eta \approx 0.005$. The curves for $\eta \pm \sigma_\eta$ are also included as dotted red lines. From the figures we can make quite a strong case for the statement $\eta^C > \eta^E$ and since Δx is a strictly increasing function of η also $(\Delta x)^C > (\Delta x)^E$. So the effect of trying to enforce impulse conservation, $\Delta p_y = 0$, seems to be that photons also travel less far in the original direction. Apparently photons are more often sent in extreme directions away from the original direction for case E. This is not a surprise, because the difference between case E and C occurs when two edges are close to the incoming direction, but both in the same y -direction. A third edge that is in the other y -direction must by definition have a larger deflection angle than the closest two edges, otherwise the third angle would have been selected as one of the edges with the smallest deflection. So choosing between method E and C is balancing between conserving impulse perpendicular to the original direction (case E) and conserving more impulse parallel to the original direction (case C). This is really just another way of saying that the expectation value for Δp_x is smaller in case E, which we already observed in section 5.2.

6 A very simple test case with a surprise: photon diffusion over the grid

In this section we consider an extremely simple test for SimpleX. We generate a Poisson Delaunay tessellation and place a source at the center. The photons from this source are sent along the grid without any obstruction. In each step all photons that arrive at a site are collected and isotropically sent away where isotropically means that an equal amount of photons is sent to each neighbour. It can be viewed as a diffusion process or a situation with 100% scattering. In figure 20 a few steps of this process in \mathcal{R}^2 are illustrated. A white color is high intensity and a dark, nearly black color is low intensity. Where the intensity is precisely zero the color is blue. This allows to see the difference between "no photons at all" and "just a few photons" much better. Qualitatively the results are precisely as expected. We see an expanding spherical wavefront that will in principle go on expanding forever, since there is no absorption. When we consider the speed at which the wavefront is moving there is a surprise however.

The wavefront is moving outwards because points that are precisely on the wavefront send photons in all directions and thus also to points just outside the wavefront. Since all points on the wavefront do this, the wavefront moves outwards. On average we expect the wavefront to move out a distance $\langle L \rangle$ during each iteration and likely even a bit slower because the edges in general won't be directed radially away from the central source. In figure 20 precisely the opposite happens. We measure the expansion speed by calculating for each iteration the location of the wavefront and calculating the slope of the resulting graph. The location of the wavefront is taken as the mean of the largest radius *with* radiation and the smallest radius *without* radiation. If the wavefront indeed moves away from the center with a speed of approximately a distance $\langle L \rangle$ per iteration, the slope of the graph should be ≈ 1 . In figure 21(a) the result is presented for a series of 50 realizations of the experiment. The solid line is a linear fit to the data of the form:

$$r(n) = a + bn \quad (57)$$

with n the number of iterations and $r(n)$ the radius

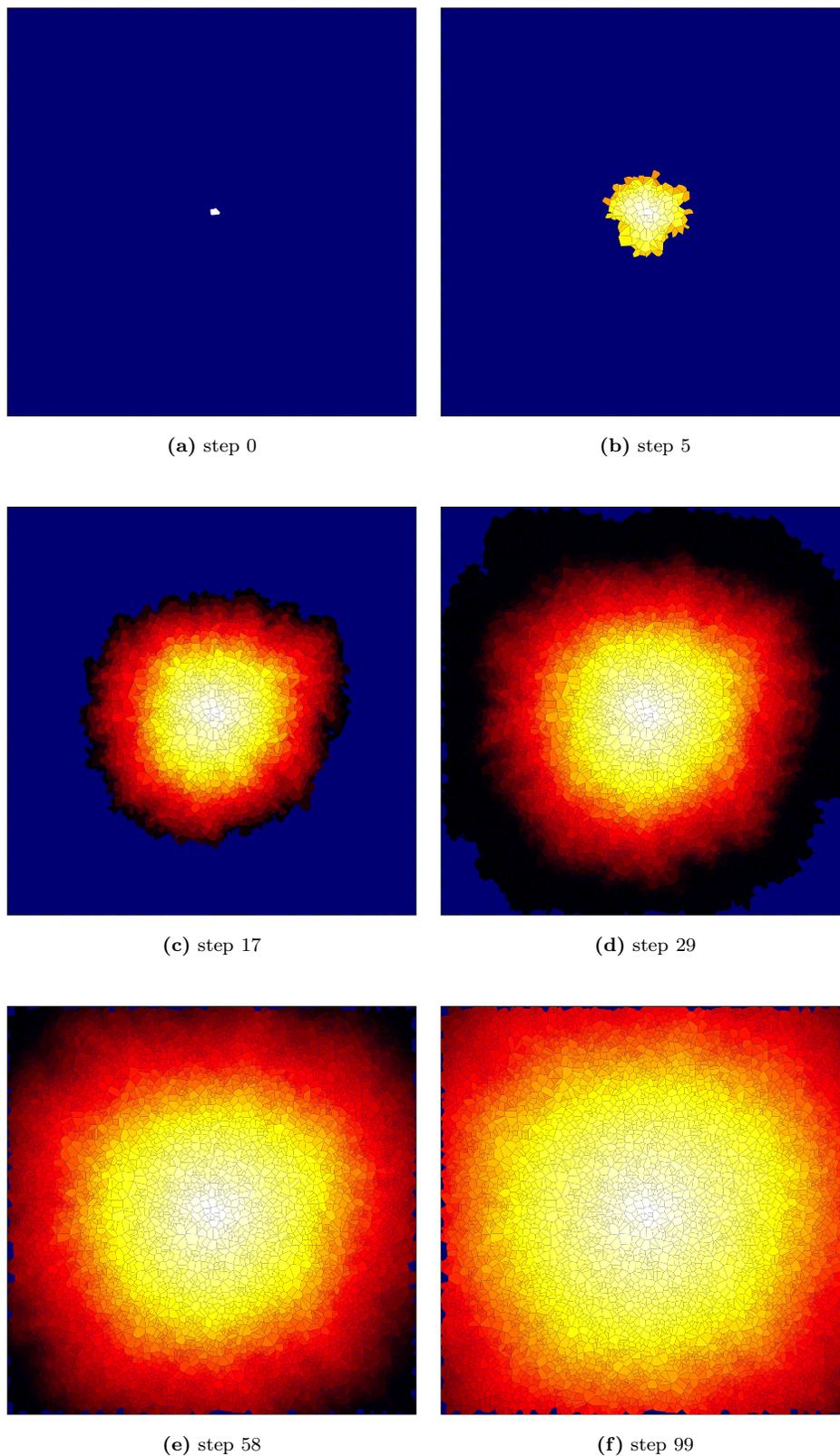


Figure 20: The process of simple photon diffusion over a Poisson Delaunay grid with 5000 points. In every step photons are collected locally and redistributed over all their neighbours. The photon source is a single Delaunay cell as close as possible to the center. The colorscheme is such that white has the highest values and black the smallest. Blue however is the zero-level. This level was introduced because the colorscheme in SimpleX is very insensitive to small changes in the low, black values, which made it impossible to see progress after a few steps into the process.

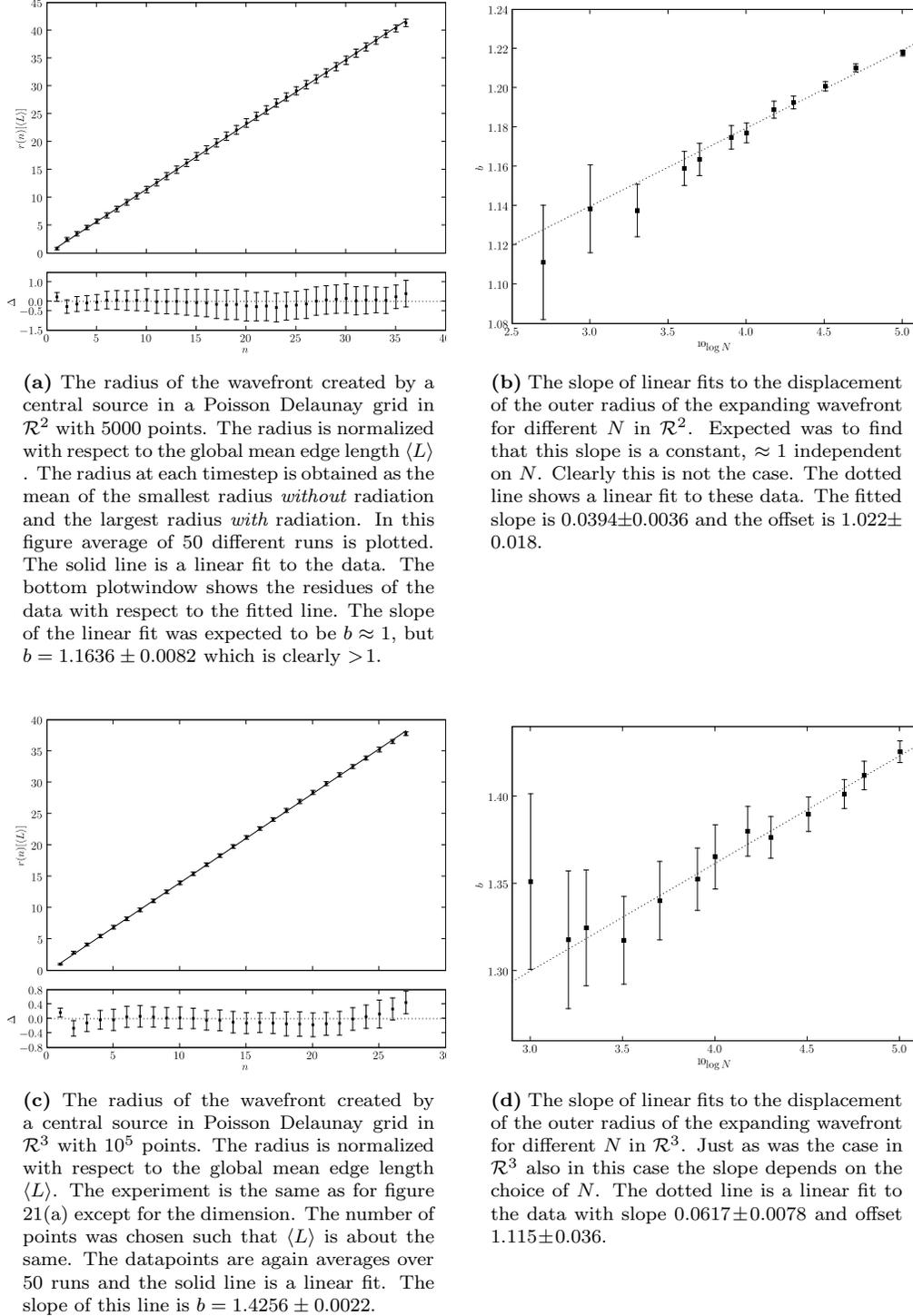


Figure 21: When photons are sent over the grid we expect the wavefront to move on average a distance $\langle L \rangle$ in each timestep. Surprisingly this is not what happens. Photons move away from the central source faster, both in \mathcal{R}^2 and \mathcal{R}^3 . Figures 21(a) and 21(b) show results for \mathcal{R}^2 . Figures 21(c) and 21(d) show results for \mathcal{R}^3 .

of the wavefront in units of the global average edge-length. The results for the data in figure 21(a) are $a = -0.13 \pm 0.14$ and $b = 1.1636 \pm 0.0082$. So the offset is more or less in agreement with $a \approx 0$, but the slope is very definitely not 1. In order to find out what causes this we repeated the experiment for other values of N . For each N 50 realizations of a Poisson Delaunay tessellation were calculated and photons were sent over the grid from a central source. The resulting slopes b are plotted in figure 21(b) versus $^{10}\log N$. The figure shows clearly that the resulting slopes are not only > 1 for all N considered, the slopes are also dependent on the choice of N . The latter is really problematic, because it means that the outcome of simulations depends on the choice of N . Another question we can ask is whether we have the same problem in \mathcal{R}^3 . So we repeated the simulation again for many values of N , but now in \mathcal{R}^3 . In figure 21(c) we plot the result for 50 runs with $N = 10^5$ points. This result is comparable to the result in \mathcal{R}^2 with $N = 5000$ in the sense that the average edge length is about equal. The resulting slope here is even larger than in \mathcal{R}^2 , namely $b = 1.4256 \pm 0.0022$. If we again extend the calculations to different N we get the results in figure 21(d). Also in \mathcal{R}^3 we see that the values found for b are dependent on N . The dependency is even stronger than in \mathcal{R}^2 . The cause of these effects is still a complete mystery.

7 The cosmological radiative transfer project

The promise of many observational data coming up for the epoch of reionization has ignited the development of a large number of numerical codes that aim at simulating this epoch. Some time ago many of these codes, including SimpleX, were compared in the *Cosmological Radiative Transfer Codes Comparison Project*[13]. The authors of the different codes were asked to solve a given set of cases and the results were compared. In this section we'll investigate the effect of the small change in the recombination coefficient (section 4.2). We'll also take a look at ways to incorporate temperature in SimpleX. We'll investigate these topics in terms of the case 0 of the comparison project that compared basic physics.

7.1 Case 0: basic physics

The solution of the radiative transfer equation depends strongly on the state of the gas, which is controlled by its temperature and degree of ionization. These in turn are controlled by the specific values for ionization rates, recombination rates and the corresponding cooling and heating rates that go into the code. This case is designed specifically to check this input and the robustness of the method.

A single optically thin zone of $1 pc^3$ with only hydrogen with a homogeneous density of $1 cm^{-3}$ is considered. The gas starts out completely neutral at a temperature $T_i = 100 K$ at time $t = 0$. Then a flux of $F = 10^{12} photons cm^{-2} s^{-1}$ is switched on that illuminates one face of the box homogeneously and heats and ionizes the gas. It has a $10^5 K$ blackbody spectrum and is on for $0.5 Myr$. After this the source is switched off and the gas is followed for another $5 Myr$ while it is cooling and recombining.

To work out this case for SimpleX we have to use a bit of trickery. In the fixed grid methods one can just set all opacities to zero in order obtain a fully optically thin zone. However, in SimpleX we need to sample the mean free path, which goes to infinity for $\tau \rightarrow 0$. Therefore we set up a simulation without using a grid. We define two variables that keep track of the total number of neutral and ionized hydrogen atoms and let our rate equations act on them. In this setup we also have to find another way to deal with ionization, because the procedure described in section (4.1.1) doesn't work here. We take the limit $\tau \rightarrow 0$ in (31):

$$N_{photons}^{used} \approx f_{\nu_0} N_{photons} \Delta\tau = f_{\nu_0} F A \bar{\sigma}(T_s) n_{HI} L$$

with $L = 1 pc$ the size of the box, $A = L^2$, the area of the face where the box is illuminated. Now we use that $ALn_{HI} = N_{HI}$ since AL is the volume of the box and find:

$$N_{photons}^{used} = \bar{\sigma}(T_s) f_{\nu_0} F N_{HI} \quad (58)$$

This is basically a first order approximation of (31) and should be equivalent with[7]:

$$\frac{\partial N_{HI}}{\partial t} = N_{HI} \int_{\nu_0}^{\infty} \pi \sigma_{\nu} \frac{B_{\nu}(T_s)}{h\nu} d\nu \quad (59)$$

We try to verify the equivalence of (58) and (59) by rewriting the integral in the latter:

$$\begin{aligned} & \int_{\nu_0}^{\infty} \pi \sigma_{\nu} \frac{B_{\nu}(T_s)}{h\nu} d\nu \\ &= \frac{\int_{\nu_0}^{\infty} \pi \sigma_{\nu} \frac{B_{\nu}(T_s)}{h\nu} d\nu}{\int_{\nu_0}^{\infty} \pi \frac{B_{\nu}(T_s)}{h\nu} d\nu} \frac{\int_{\nu_0}^{\infty} \pi \frac{B_{\nu}(T_s)}{h\nu} d\nu}{\int_0^{\infty} \pi \frac{B_{\nu}(T_s)}{h\nu} d\nu} \int_0^{\infty} \pi \frac{B_{\nu}(T_s)}{h\nu} d\nu \end{aligned}$$

The last term in this equation is precisely the total flux F in photons s^{-1} . The first and second term look a lot like $\bar{\sigma}(T_s)$ and f_{ν_0} respectively except for the fact that apparently we need to use $B_{\nu}/(h\nu)$ as weighing function instead of just B_{ν} . This is a result of the fact that we're using a photon flux as input, instead of an energy flux. We define the following new quantities:

$$\bar{\sigma}^p(T_s) \equiv \int_{\nu_0}^{\infty} \sigma_{\nu}^{\text{Kramers}} \left[\frac{B_{\nu}(T_s)}{h\nu} \right]_{\text{norm}} d\nu \quad (60)$$

$$f_{\nu_0}^p \equiv \int_{\nu_0}^{\infty} \left[\frac{B_{\nu}(T_s)}{h\nu} \right]_{\text{norm}} d\nu \quad (61)$$

where the superscript p indicates that these are quantities that are averaged with respect to the normalized Blackbody in photons s^{-1} , unlike the quantities defined in (26) and (32). With these definition we can rewrite (59) as:

$$\frac{\partial N_{\text{HII}}}{\partial t} = \bar{\sigma}^p f_{\nu_0}^p F N_{\text{HII}} \quad (62)$$

This is precisely the same expression as (58), except that the cross section and fraction f_{ν_0} are weighed in a different way. We feel that it is preferable to make the replacement $\bar{\sigma} \rightarrow \bar{\sigma}^p$ and $f_{\nu_0} \rightarrow f_{\nu_0}^p$ in all equations where these quantities have appeared so far, eg (27), (31) and (58). We're in the process of executing this replacement and investigating the magnitude of its effect. For the rest of the report we'll be using the original, energy weighted forms.

For the recombination rate we can use (40).

7.1.1 Cooling rate

Photons that are send out during recombination and that escape result in cooling. The resulting

cooling rate in $\text{erg cm}^{-3} \text{s}^{-1}$ is given by[7]:

$$\Lambda_{\text{HII}}(T) = 8.70 \cdot 10^{-27} T^{1/2} T_3^{-0.2} (1 + T_6^{0.7})^{-1} n_e n_{\text{HII}} \quad (63)$$

This relation is plotted as the solid blue line in figure 9. We see that also here the region $T \lesssim 10^5 \text{ K}$ can reasonably well be approximated by a single powerlaw. The powerlaw again is chosen to coincide with the full expression (63) at $T = 10^4 \text{ K}$:

$$\Lambda_{\text{HII}}(T) = 3.33 \cdot 10^{-27} T_4^{0.26} n_e n_{\text{HII}} \text{ erg cm}^{-3} \text{ s}^{-1} \quad (64)$$

This approximation is plotted as the dashed blue line in figure 9.

7.1.2 Heating rate

Electrons that are knocked out of an hydrogen atom by a photon with energy $\gamma_{\nu} h$ will carry along an energy $h\gamma_{\nu} - h\nu_0$ in the form of kinetic energy, thereby heating the gas. For a monochromatic flux of ionizing photons we have always $h\gamma_{\nu} = h\nu_0$ so such a flux can't heat the gas. Any non trivial source spectrum however will give rise to photons that have energies $> h\nu_0$ and therefore will result in heating.

Because we only know the total *number* of ionizing photons it is hard to write down an expression for the resulting heating. Another difficulty is that photons resulting from recombination have a different spectrum than photons resulting from the source. But for this experiment it is a reasonable approximation to say that all recombined photons go out. Then we can calculate the expectation energy for a single photon from the normalized Blackbody distribution (22):

$$\begin{aligned} \langle h\nu \rangle &= \frac{15}{\pi^4} \left(\frac{h}{kT_s} \right)^4 \int_{\nu_0}^{\infty} \frac{(h\nu) \nu^3 d\nu}{e^{h\nu/kT_s} - 1} \\ &= \frac{15}{\pi^4} I \left(4, \frac{h\nu_0}{kT_s} \right) kT_s \end{aligned}$$

Such that we can say to a first approximation:

$$\Gamma_{\text{HI}}(T_s) = N_{\text{photons}}^{\text{used}} \left[\frac{15}{\pi^4} I \left(4, \frac{h\nu_0}{kT_s} \right) kT_s - h\nu_0 \right] \quad (65)$$

where $N_{\text{photons}}^{\text{used}}$ comes from (58) and $I \left(4, \frac{h\nu_0}{kT_s} \right)$ is given by (93) in appendix B.

7.2 Case 0: results sofar

Sofar trials to use the cooling and heating relations (64) and (65) to solve for the temperature have been unsuccessful. The problem is that the expression for the heating gives a number that is much too large and completely destabilizes the procedure. This remains work for the near future.

The effect of changing the recombination coefficient can be clearly seen in figure 22. It shows the resulting neutral fraction for the case 0 test of the input physics. At the beginning of the experiment the source is switched on and the hydrogen is ionized until a balance is reached between ionization and recombination. The solid line is the result for the original value of the recombination coefficient, the dotted line is the result for the new value. Since the new recombination coefficient is slightly higher, the neutral fraction comes out also a bit higher. The difference can be understood precisely from analytical calculations. In steady state we have[28]:

$${}^{10}\log(1-x) = {}^{10}\log\left(\frac{\alpha_B(T_g)}{F\bar{\sigma}(T_s)}\right) \quad (66)$$

Using the old value of α_B with $T_g = 10^4 K$ the result becomes ${}^{10}\log(1-x) = -6.56$. With the new value it is ${}^{10}\log(1-x) = -6.45$. These values match precisely with figure 22. The issue during the Comparison Project was that all other methods had a value around ${}^{10}\log(1-x) = -7$. The expectation was that this must have something to do with the fixed temperature in SimpleX. Hopefully this claim can be tested soon. Another possibility is that the difference was not actually in the temperature, but in the definition of the frequency averaged ionization cross section. We already noted that it may be better to replace $\bar{\sigma}$ given by (26) with $\bar{\sigma}^P$ given by (60). This could just as well bridge the gap between SimpleX and other radiative transfer codes.

After $5 Myr$ the source is switched off and the gas can cool and recombine. For this epoch the results are practically indistinguishable.

8 Discussion

SimpleX is the first method that tries to use Delaunay tessellation as a grid for performing radiation

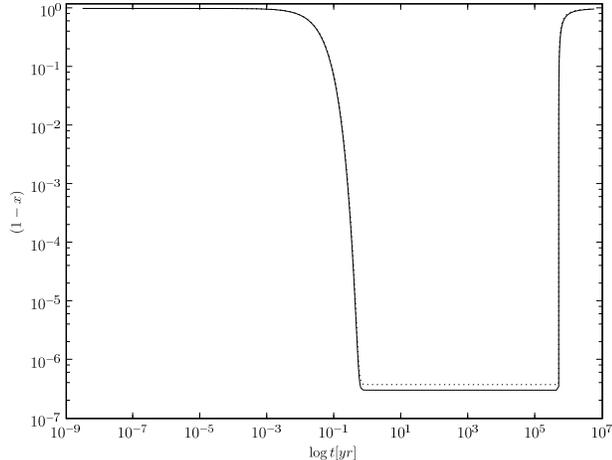


Figure 22: The resulting neutral fraction in time for the case 0 test of the Radiative Transfer Comparison Project. The solid line shows the results for the original version of SimpleX. The dotted line shows the results after the small change in the recombination coefficient (37).

transport. The logical consequence of this is that many properties and details of radiation transport have to be reinvented for applicability on such a grid. The general functioning of SimpleX is by now rather well understood and the results of the Comparison Project[13] have also shown that SimpleX does rather well between other codes. During the Comparison Project SimpleX never even got the chance to really show it's strength: the time complexity does *not* scale with the number of sources. This means that we could in principle put a source of photons in every site in the grid and all calculations would still be the same. This extreme property is very useful for studying the reionization epoch where many stars may have formed in small, dense regions of space. However in order to be able to do really precise numerical experiments there are still a lot of issues that need to be resolved, some of which we found in this work.

An important and critical element of SimpleX is it's availability to sample the mean free path of the transported particles. We have shown in section 5.1.2 that SimpleX does quite well for a homogeneous distribution but it remains interesting to see what happens for inhomogeneous distributions. Indications are that SimpleX works quite

well also then, but quantitative analysis has yet to be executed. The best test would be to let SimpleX sample a given density field, set up the Delaunay tessellation and make a comparison of the local mean free path in the original distribution and the edgelengths in the grid. This conceptually simple test has not been performed yet but has very high priority since it tests really the fundamental basis for the correctness of the method. If in practice it appears that it is not possible to make the sampling work such that (18) is fulfilled, the method loses its fundament. If the sampling criteria can be confirmed on the other hand, SimpleX potentially can become a very powerful tool for studying a range of transport processes. It does need a major rewrite before that though. Because SimpleX was developed pretty much on an ad-hoc basis in the sense that pieces were added when needed, the total package is not very well structured. This makes working with SimpleX rather tricky at times. If SimpleX is going to be an important player in numerical cosmology it will need a rather extensive rewrite to become a more mature package. The advantage of the pioneering work so far is that by now it is pretty clear how a package like SimpleX should be structured such that new parts can be added easily and such that parts that shouldn't need to communicate indeed don't communicate. Another interesting development is the work currently underway to embed a dynamical code for determining the tessellation into SimpleX. This allows SimpleX to work in a changing background. When using QHull or a similar code the tessellation would have to be fully recalculated when the background medium changes. The dynamical tessellation code is available to detect where points have shifted and updates the tessellation accordingly. This should make it possible to eventually couple SimpleX directly to for example hydrocodes, or even let SimpleX itself solve the hydrodynamical equations in parallel with radiative transfer.

But before getting carried away, we need to remind ourselves of the issues that still need resolving even in the current state of affairs. There are still many issues not yet very well understood. Consider for example the small offset in the angular distribution function (51) in section 5.1.3. This may be caused by QHull, it may be caused by a wrong use of QHull in SimpleX or yet another problem. Changing this distribution does change the trans-

port of photons though, because it will effect for example Δx and Δy as studied in section 5.3. These again will effect the way in which photons are distributed over the grid and will therefore also affect the reionization fronts we eventually want to study during the reionization epoch. But maybe even more important is to understand the effect noticed in section 6 that photons seem to be able to move on average over a longer distance than the average edgelength. And even more importantly: why is this effect dependent on N ? This is certainly very unexpected for a Poisson Delaunay tessellation because these are highly symmetric and scalefree. An original version of the experiment had multiple sites at the center as source. The suggestion at that point was that photons could travel faster away from the center for high values of N simply because on average there would be more sources and thus more edges leaving the center. In the current setup where there is precisely one vertex assigned as source this effect should not occur but the results haven't changed at all. This issue is part of the very basic understanding of the method and definitely needs to be resolved.

Besides details that are not yet fully understood SimpleX also lacks a lot of physics. It would be nice to be able to add other particles besides hydrogen to the simulation, like hydrogen and/or lithium. This is however not a trivial thing to do in SimpleX, since the mean free paths of such particles can be widely different from the mean free path for hydrogen. One could setup a tessellation for each particle that is added to the simulation, but that would result in an enormous increase of computer memory requirements. The same is true for the frequency. The reduction of the cross section to a single frequency cross section is rather coarse and should be improved upon. A first step towards this goal may be the utilization of a three photon model[23] where three frequency bins are created for frequencies $h\nu < h\nu_0$, $h\nu = h\nu_0$ and $h\nu > h\nu_0$. But again here is also the complication of computer memory, because at a different frequency the distribution of the mean free path changes for every particle. So we need yet another series of tessellations. Coupled with frequency is the temperature. The temperature in the simulation can be determined by calculating the cooling and heating. If there are only photons of frequency $h\nu_0$ no heating can occur, because they have no excess energy that can be used

to heat the gas. So we have to use some non-trivial source spectrum, but then is the question how to tell of a photon package what its spectrum is? If it's coming from the source the spectrum is known, but the photons could also come from recombinations in a neighbouring site. What is the spectrum of these photons? We could of course apply the on-the-spot-approximation but this would be a waste because SimpleX should be able to explicitly deal with recombination photons on the same level as source photons. (remember: all sites can be sources without adding additional overhead) If we could track the frequency of the photons it would also become possible to get a decent expression for the heating which would make it possible to also solve self consistently for the temperature within SimpleX.

Another possible addition is the physics involved in ionizations by very hard X-ray photons that may have been produced by the first stars that formed during the reionization epoch. The electron that is kicked out can give rise to secondary and even tertiary ionizations. It would be very interesting to be able to follow such processes. But also this will be a rather elaborate extension, because the space of mean free paths looks very different for electrons as for photons. So also here would we need a separate tessellation. Clearly there is more than enough room for improvements and additions.

9 Conclusions

SimpleX is a numerical method that aims to solve the equation of radiative transfer through a Monte Carlo approach. It is special in its application of an unstructured grid. The grid is generated from the background medium through which transport must take place instead of predetermined as for standard Cartesian grids. This in principle allows the grid to adapt to the density field in such a way that the edgelengths of the tessellation sample the mean free path of the particles. This principle can be used for any transport problem, but SimpleX is adapted for radiative transfer and study of the reionization epoch in particular.

It is known that SimpleX generally works very well[28] but there are still many details that are not very well understood or tested. One of the tests that has not yet been performed is the comparison of the local mean free path in a given den-

sity distribution with the edgelength distribution of the grid. By sampling the densityfield correctly the moments of both distributions should be proportional to each other. This test is fundamental for the method, since it the method depends completely on this proportionality and needs to be done soon.

There are still many pieces of physics that ought to be added to SimpleX. Some first thoughts about how to incorporate a selfconsistent solution for the temperature were presented, but no succesful solution was obtained sofar. Other features that are important to fully model the reionization epoch are the frequency dependence, other atoms besides hydrogen like helium and lithium and ultra high energetic photons which give rise to secondary and possibly tertiary ionizations.

SimpleX has the promise of becoming a very powerful method that is capable of dealing with all these issues in a very simple and straightforward fashion, but much work is still needed to get there.

Acknowledgements. I would like to thank R. van de Weygaert, G. Mellema and all the other people that are involved for giving me a fantastic opportunity to couple the masterthesis with the PhD project. I'm humbled.

A Determining the center of the circumscribing circle of a triangle

An important ingredient in determining Voronoi simplices is calculating the center of the circumscribing circle of a given triangle $\{\vec{x}_0, \vec{x}_1, \vec{x}_2\}$. The method in SimpleX uses the slope of the points \vec{x}_1 and \vec{x}_2 with respect to \vec{x}_0 to find the center. At some point a division by the slope is needed, which is problematic if a point happens to lay parallel to one of the Cartesian axes. The method by Van de Weygaert[33] (VW) does not suffer from this problem, but requires the evaluation of a series of outer products. Here we develop a variation of that method that requires much less algebraic calculations. Another benefit is that it works for any dimension d of space.

We begin the method by translating the triangle to the origin. Any of the \vec{x}_i , $i = 0, 1, 2$ can be used for this. We choose \vec{x}_0 , so $\vec{x} \mapsto \vec{x} - \vec{x}_0$. This trans-

lation is a general technique that could be used in any any method, because in all methods there will be many subtractions of whichever vector is used as basisvector in the calculations. All these subtractions during the calculation are now removed and replaced by precisely two subtractions and one addition (to inverse the translation for the calculated circumcenter). We now have the triangle $\{\vec{0}, \vec{r}_1, \vec{r}_2\}$ with $\vec{r}_i = \vec{x}_i - \vec{x}_0$, $i = 1, 2$ for which we calculate the circumcenter \vec{r}_c . In figure 23 the situation is illustrated. Note that in this figure we also rescaled the triangle but that is just to compress the size of the figure and is not part of the method. After calculating the circumcenter \vec{r}_c of this triangle the circumcenter of the original triangle can be found by inverting the translation: $\vec{x}_c = \vec{r}_c + \vec{x}_0$.

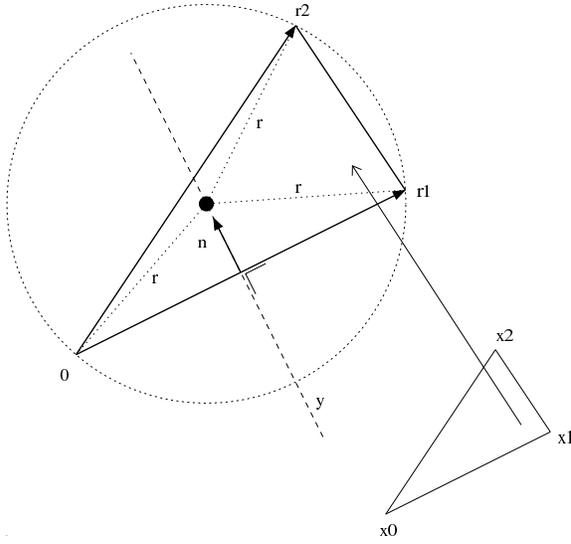


Figure 23: The layout of a triangle after translating to the origin.

The defining property of \vec{r}_c is that it has equal distances from all points of the triangle:

$$|\vec{r}_c| = |\vec{r}_c - \vec{r}_1| = |\vec{r}_c - \vec{r}_2| \equiv r \quad (67)$$

This means for example that \vec{r}_c must lay on the line that perpendicularly bisects \vec{r}_1 . We parametrize the points on this line as follows:

$$\vec{y} = \vec{n}t + \frac{1}{2}\vec{r}_1 \quad t \in \mathcal{R} \quad (68)$$

where \vec{n} is a vector that is perpendicular to \vec{r}_1 and t is a scaling factor. By choosing the correct value

of t we find \vec{r}_c :

$$\vec{r}_c = \vec{y}(t_*) = \vec{n}t_* + \frac{1}{2}\vec{r}_1 \quad (69)$$

So far we followed the method by VW. There the vector \vec{n} is calculated from a series of outerproducts and the value of t_* is calculated from the geometry. This is where our method will deviate.

Since we know that \vec{n} *must* lay in the plane that is spanned by \vec{r}_1 and \vec{r}_2 , we can write:

$$\vec{n} = a\vec{r}_1 + b\vec{r}_2 \quad (70)$$

This is just saying that \vec{n} must be a linear combination of \vec{r}_1 and \vec{r}_2 . The condition on \vec{n} is that it must be perpendicular to \vec{r}_1 :

$$\vec{r}_1 \cdot \vec{n} = 0 \quad (71)$$

$$\Rightarrow \frac{a}{b} = -\frac{\vec{r}_1 \cdot \vec{r}_2}{|\vec{r}_1|^2} \quad (72)$$

Additional constraints can be used to fix a and b . For example if we want \vec{n} to be normalized, we have the additional constraint:

$$\begin{aligned} |\vec{n}|^2 &= 1 = a^2 |\vec{r}_1|^2 + b^2 |\vec{r}_2|^2 + 2ab (\vec{r}_1 \cdot \vec{r}_2) \\ \Rightarrow a &= -\frac{\vec{r}_1 \cdot \vec{r}_2}{|\vec{r}_1| \sqrt{|\vec{r}_1|^2 |\vec{r}_2|^2 - (\vec{r}_1 \cdot \vec{r}_2)^2}} \\ b &= \frac{|\vec{r}_1|}{\sqrt{|\vec{r}_1|^2 |\vec{r}_2|^2 - (\vec{r}_1 \cdot \vec{r}_2)^2}} \end{aligned} \quad (73)$$

For our method the length of \vec{n} is not important. Changing the length of \vec{n} will just shift around the value of t_* in (69). So we choose the coefficients such that we minimize the amount of required calculations:

$$\begin{aligned} a &= -(\vec{r}_1 \cdot \vec{r}_2) \\ b &= |\vec{r}_1|^2 \end{aligned} \quad (74)$$

Note that by definition of \vec{n} and \vec{y} we have:

$$|\vec{y} - \vec{0}|^2 = |y|^2 = |\vec{y} - \vec{r}_1|^2 \quad (75)$$

To find t_* we have to solve (67):

$$|\vec{r}_c - \vec{r}_2|^2 = |\vec{y}(t_*) - \vec{r}_2|^2 = |\vec{y}(t_*)|^2 \quad (76)$$

The (unique) solution to this equation is:

$$t_* = \frac{1}{2} \frac{|\vec{r}_2|^2 - \vec{r}_1 \cdot \vec{r}_2}{\vec{n} \cdot \vec{r}_2} \quad (77)$$

If we substitute (74) this becomes:

$$t_* = \frac{1}{2} \frac{|\vec{r}_2|^2 - \vec{r}_1 \cdot \vec{r}_2}{|\vec{r}_1|^2 |\vec{r}_2|^2 - (\vec{r}_1 \cdot \vec{r}_2)^2} \quad (78)$$

The denominator of this expression looks familiar from (73) and in fact, if we make the combination $t_* \vec{n}$ we need in (69) the resulting expression is precisely the same for either choice of a and b . So normalizing \vec{n} is not only numerically expensive because of the squareroot, it is also unnecessary, since we're ultimately calculating (78) anyhow.

We still have to check one thing: what happens if the denominator in (78) is zero? It is easily verified that this happens precisely when \vec{x}_1 and \vec{x}_2 are aligned, ie when $\vec{r}_2 = \alpha \vec{r}_1$ for some $\alpha \in \mathcal{R}$. In this case we don't have a triangle to begin with and no Voronoi vertex exists.

Let's talk practice now. In order to find the circumcenter \vec{x}_c of a given triangle $\{\vec{x}_0, \vec{x}_1, \vec{x}_2\}$, we take the following steps:

1. translate to origin: $\vec{r}_i = \vec{x}_i - \vec{x}_0$, $i = 1, 2$
2. calculate $|\vec{r}_1|^2$, $|\vec{r}_2|^2$ and $\vec{r}_1 \cdot \vec{r}_2$
3. check: $|\vec{r}_1|^2 |\vec{r}_2|^2 - \vec{r}_1 \cdot \vec{r}_2 \neq 0$
4. calculate \vec{r}_c :

$$\vec{r}_c = \left(\frac{1}{2} - t_* (\vec{r}_1 \cdot \vec{r}_2) \right) \vec{r}_1 + |\vec{r}_1|^2 t_* \vec{r}_2 \quad (79)$$

with t_* given by (78).

5. set $\vec{x}_c = \vec{r}_c + \vec{x}_0$

The resulting method works for any dimension and requires only very few algebraic calculations, since we only need to find $|\vec{r}_1|^2$, $|\vec{r}_2|^2$ and $\vec{r}_1 \cdot \vec{r}_2$ once, which can be done very easily.

B 'Blackbody-type' integrals

When dealing with integrals over the Planck curve the following type integral appears often for integer n :

$$I_n(x_0) \equiv \int_{x_0}^{\infty} \frac{x^n dx}{e^x - 1} \quad (80)$$

B.1 $x_0 = 0$

The situation $x_0 = 0$ is needed when we need the normalized version of (80). We'll focus on this type of integral first. Rewrite (80) as:

$$I_n(0) = \int_{\mathcal{R}} \frac{x^n e^{-x} dx}{1 - e^{-x}} \quad (81)$$

Note that $0 < e^{-x} < 1 \forall x > 0$, such that $(1 - e^{-x})^{-1}$ can be written as a convergent powerseries:

$$I_n(0) = \int_{\mathcal{R}} x^n \sum_{k=0}^{\infty} e^{-kx} dx \quad (82)$$

Since each separate term in the integral converges absolutely, the summation and integral sign can be switched. After substituting $u = kx$ we get:

$$I_n(0) = \sum_{k=1}^{\infty} \frac{1}{k^{n+1}} \int_{\mathcal{R}} u^n e^{-u} du \quad (83)$$

This is the product of the Riemann zeta function $\zeta(n+1)$ and the Gamma function $\Gamma(n+1)$, so finally:

$$\begin{aligned} I_n(0) &= \int_{\mathcal{R}} \frac{x^n dx}{e^x - 1} = \zeta(n+1) \Gamma(n+1) \\ &= n! \zeta(n+1) \end{aligned} \quad (84)$$

where in the last step we used that for integer n the equality $\Gamma(n+1) = n!$ holds. Note that this formula is only valid for integers $n > 0$, since the zeta function $\zeta(s)$ only converges for $s > 1$. In table 1 we list the results for $n \in [1, 5]$.

n	$\zeta(n+1)$	$\Gamma(n+1)$	$I_n(0)$
1	$\frac{\pi^2}{3}$	1	$\frac{\pi^2}{3} \approx 3.29$
2	≈ 1.20	2	2.40
3	$\frac{\pi^4}{90}$	6	$\frac{\pi^4}{15} \approx 6.49$
4	≈ 1.04	24	24.9
5	$\frac{\pi^6}{945}$	120	$\frac{120\pi^6}{945} \approx 122$

Table 1: The value of integral $I_n(0)$ as given in (80) for $n = 1, \dots, 5$.

B.2 $x_0 \neq 0$

In the previous section we found that we can't calculate $I_0(0)$ because the zeta function diverges for

$n=0$. For the general case $x_0 \neq 0$ however we can write down a primitive explicitly:

$$\begin{aligned} I_0(x_0) &= \int_{x_0}^{\infty} \frac{dx}{e^x - 1} \\ &= \ln(1 - e^{-x}) \Big|_{x_0}^{\infty} \Rightarrow \\ I_0(x_0) &= -\ln(1 - e^{-x_0}) \end{aligned} \quad (85)$$

In order to evaluate (80) when $x_0 \neq 0$ for $n > 0$ we have to revert to numerical techniques. A method that is very easy to implement can be developed by going back to (82) but now with $x_0 \neq 0$. Then again the separate terms in the sum are absolutely convergent, so we may write:

$$I_n(x_0) = \sum_{k=1}^{\infty} \int_{x_0}^{\infty} x^n e^{-kx} dx \quad n > 0 \quad (86)$$

Let's focus on the integral at the right hand side. We perform integration by parts:

$$\int_{x_0}^{\infty} x^n e^{-kx} dx = \frac{x_0 e^{-kx_0}}{k} + \frac{n}{k} \int_{x_0}^{\infty} x^{n-1} e^{-kx} dx$$

Continuing we can reduce the power of x by one in every partial integration, obtaining:

$$\int_{x_0}^{\infty} x^n e^{-kx} dx = e^{-kx_0} \sum_{j=0}^n \frac{n!}{j!} \frac{x_0^j}{k^{n+1-j}} \quad n > 0 \quad (87)$$

Now that we have this relation we can also verify it by induction on n . First test $n = 1$:

$$\begin{aligned} \int_{x_0}^{\infty} x^n e^{-kx} dx &= e^{-kx_0} \left(\frac{x_0}{k} + \frac{1}{k^2} \right) \\ e^{-kx_0} \sum_{j=0}^1 \frac{1!}{j!} \frac{x_0^j}{k^{2-j}} &= e^{-kx_0} \left(\frac{1}{k^2} + \frac{x_0}{k} \right) \end{aligned}$$

Now assume that (87) is correct for some n and see what happens for $n + 1$:

$$\begin{aligned} \int_{x_0}^{\infty} x^{n+1} e^{-kx} dx &= \frac{x_0^{n+1} e^{-kx_0}}{k} + \frac{n+1}{k} \int_{x_0}^{\infty} x^n e^{-kx} dx \\ &= \frac{x_0^{n+1} e^{-kx_0}}{k} + \frac{n+1}{k} e^{-kx_0} \sum_{j=0}^n \frac{n!}{j!} \frac{x_0^j}{k^{n+1-j}} \end{aligned}$$

$$= \frac{x_0^{n+1} e^{-kx_0}}{k} + e^{-kx_0} \sum_{j=0}^n \frac{(n+1)!}{j!} \frac{x_0^j}{k^{n+2-j}}$$

The first term in this last expression is obtained precisely when substituting $j = n + 1$ in the sum, which finishes the proof of (87) by induction on n .

Substituting (87) back into (86) then gives:

$$I(n, x_0) = \sum_{k=1}^{\infty} \sum_{j=0}^n \frac{e^{-kx_0}}{k^{n+1}} \frac{n!}{j!} (x_0 k)^j \quad (88)$$

We separate out the polynomial of degree n in this sum:

$$P_n(x) \equiv \sum_{j=0}^n \frac{n!}{j!} x^j \quad (89)$$

For values $x_0 \geq 1$ the sum with respect to k converges quit rapidly. In general k_{\max} , the last term needed to get to a given accuracy ϵ will depend on n and ϵ . So finally we can write for the numerical approximation of (80):

$$I_n(x_0) \approx \sum_{k=1}^{k_{\max}(n, \epsilon)} \frac{e^{-kx_0}}{k^{n+1}} P_n(kx_0) \quad (90)$$

For values $x_0 \gtrsim 1$ this series converges very rapidly. For small values of x_0 the number of required terms grows quickly. More about this in section B.2.1.

$$P_2(x) = 2 + 2x + x^2 \quad (91)$$

$$P_3(x) = 6 + 6x + 3x^2 + x^3 \quad (92)$$

$$P_4(x) = 24 + 24x + 12x^2 + 4x^3 + x^4 \quad (93)$$

B.2.1 Obtaining $k_{\max}(n, \epsilon)$ [17]

In order to use (90) we still need to know where to cut off the summation, given a required accuracy ϵ . The error term $E(n, \epsilon)$ consists of the sum of the remaining terms:

$$E(n, \epsilon) = \sum_{k=k_{\max}+1}^{\infty} \frac{e^{-kx_0}}{k^{n+1}} P_n(kx_0) \quad (94)$$

Define $p = n + 1 - j$ and substitute (89):

$$E(n, \epsilon) = \sum_{j=0}^n \frac{n!}{j!} x_0^j \sum_{k=k_{\max}+1}^{\infty} \frac{e^{-kx_0}}{k^p} \quad (95)$$

There are two ways to proceed from here. The first method uses a recurrence relation for the term

in the second sum, the second method uses the Schwarz inequality for this sum.

Using recurrence:

Define $A(k) \equiv e^{-kx_0}/k^p$. Then for $k \geq k_{\max} > 0, p > 0$:

$$\frac{A(k+1)}{A(k)} = e^{-x_0} \left(\frac{1}{1+k^{-1}} \right)^p < e^{-x_0}$$

Repeating this process for $A(k+l)$ for general integer $l > 1$ results in:

$$A(k+l) < e^{-lx_0} A(k)$$

Putting this into (95) gives:

$$E(n, \epsilon) < \sum_{j=0}^n \frac{n!}{j!} x_0^j \frac{e^{-(k_{\max}+1)x_0}}{(k_{\max}+1)^p} \sum_{l=0}^{\infty} e^{-lx_0} \quad (96)$$

Since $e^{-x_0} < 1$ for $x_0 > 0$ the sum over l converges to $(1 - e^{-x_0})^{-1}$:

$$E(n, \epsilon) < \frac{e^{-(k_{\max}+1)x_0}}{(k_{\max}+1)^{n+1}} \frac{P_n[(k_{\max}+1)x_0]}{1 - e^{-x_0}} \quad (97)$$

So the error made by cutting off the summation in (90) at $k_{\max}(n, \epsilon)$ is bounded by the next term in the summation, multiplied with a factor $(1 - e^{-x_0})^{-1}$.

Using Schwarz inequality:

The Schwarz inequality states:

$$\left(\sum_i a_i b_i \right)^2 \leq \sum_i a_i^2 \sum_i b_i^2 \quad (98)$$

Consider again the sum over k in (95). Using Schwarzs inequality:

$$\left(\sum_{k_{\max}+1}^{\infty} \frac{e^{-kx_0}}{k^p} \right)^2 \leq \sum_{k_{\max}+1}^{\infty} e^{-2kx_0} \sum_{k_{\max}+1}^{\infty} \frac{1}{k^{2p}} \quad (99)$$

The first term reduces to a powerseries with argument $e^{-2x_0} < 1$:

$$\sum_{k_{\max}+1}^{\infty} e^{-2kx_0} = \frac{e^{-2(k_{\max}+1)x_0}}{1 - e^{-2x_0}}$$

For the second term write $g(x) = x^{-2p}$, then:

$$g(k) \geq g(x) \geq g(k+1) \Leftrightarrow k \leq x \leq k+1$$

Since $g(x)$ is a strictly decreasing function we can also integrate from $x=k \rightarrow x=k+1$:

$$g(k) \geq \int_k^{k+1} g(x) dx \geq g(k+1)$$

Summing from $k = k_{\max} + 1$ then gives:

$$\sum_{k=k_{\max}+1}^{\infty} g(k) \geq \int_{k_{\max}+1}^{\infty} g(x) dx \geq \sum_{k=k_{\max}+1}^{\infty} g(k+1)$$

The last term can be rewritten as:

$$\sum_{k=k_{\max}+1}^{\infty} g(k+1) = \left[\sum_{k=k_{\max}+1}^{\infty} g(k) \right] - g(k_{\max}+1)$$

Performing the integral in the middle:

$$\int_{k_{\max}+1}^{\infty} g(x) dx = \frac{(k_{\max}+1)^{1-2p}}{2p-1}$$

Combining these results gives the upperlimit we want:

$$\sum_{k_{\max}+1}^{\infty} \frac{1}{k^{2p}} \leq (k_{\max}+1)^{-2p} \left[\frac{k_{\max}+1}{2p-1} + 1 \right] \quad (100)$$

Put all results into (99) and take the squareroot:

$$\sum_{k_{\max}+1}^{\infty} \frac{e^{-kx_0}}{k^p} \leq \frac{e^{-(k_{\max}+1)x_0}}{\sqrt{1 - e^{-2x_0}} (k_{\max}+1)^p} \left[\frac{k_{\max}+1}{2p-1} + 1 \right]^{1/2} \quad (101)$$

Substituting this into (95) then gives a second error estimate for the approximation (90):

$$E(n, \epsilon) \leq \frac{e^{-(k_{\max}+1)x_0}}{(k_{\max}+1)^{n+1}} \frac{P_n^{(k_{\max}+1)}(x_0)}{\sqrt{1 - e^{-2x_0}}} \quad (102)$$

The general form of this result is very similar to (97), except that now the overall prefactor is $(1 - e^{-2x_0})^{-1/2}$ and the resulting polynomial has other coefficients than before:

$$P_n^k(x_0) \equiv \sum_{j=0}^n \frac{n!}{j!} (kx)^j \left[\frac{k}{2(n + \frac{1}{2} - j)} + 1 \right]^{1/2} \quad (103)$$

So when to use (97) and when to use (102)? Generally (102) will result in smaller values for k_{\max} for small values of x_0 and (97) for large values of x_0 . Another consideration is the computational work of evaluating (102). The particular choice should be adapted to the application.

References

- [1] <http://21cma.bao.ac.cn/>.
- [2] F. et al. (397 coauthors) Abe. Evidence for top quark production in $p\bar{p}$ collisions at $\sqrt{s}=1.8$ TeV. *Phys. Rev. D*, 50:2966–3026, September 1994.
- [3] M. A. Aragon-Calvo, B. J. T. Jones, R. van de Weygaert, J. M., and v. der Hulst. The Multi-scale Morphology Filter: Identifying and Extracting Spatial Patterns in the Galaxy Distribution. *ArXiv e-prints*, 705, May 2007.
- [4] <http://www.astron.nl/>.
- [5] <http://www.atnf.csiro.au/>.
- [6] D. Brown. Voronoi diagrams from convex hulls. 9:223–228, 1979.
- [7] R. Cen. A hydrodynamic approach to cosmology - Methodology. *ApJS*, 78:341–364, February 1992.
- [8] S. Chandrasekhar. *Radiative transfer*. New York: Dover, 1960, 1960.
- [9] G. F. R. Ellis. Cosmology: The shape of the Universe. *Nature*, 425:566–567, October 2003.
- [10] <http://www.mpa-garching.mpg.de/gadget/>.
- [11] D. G. Hummer and M. J. Seaton. The ionization structure of planetary nebulae, I. Pure hydrogen nebulae. *MNRAS*, 125:437–+, 1963.
- [12] V. Icke and R. van de Weygaert. Fragmenting the universe. *A&A*, 184:16–32, October 1987.
- [13] I. T. Iliev, B. Ciardi, M. A. Alvarez, A. Maselli, A. Ferrara, N. Y. Gnedin, G. Mellema, T. Nakamoto, M. L. Norman, A. O. Razoumov, E.-J. Rijkhorst, J. Ritzerveld, P. R. Shapiro, H. Susa, M. Umemura, and D. J. Whalen. Cosmological radiative transfer codes comparison project - I. The static density field tests. *MNRAS*, 371:1057–1086, September 2006.
- [14] M. Lachieze-Rey and J. Luminet. Cosmic topology. *arXiv:gr-qc/9605010 v2*, January 2003.
- [15] <http://www.lofar.org/>.
- [16] Lucretius. http://classics.mit.edu/Carus/nature_things.html.
- [17] T. E. Michels. Planck functions and integrals; methods of computation. *NASA technical note TN D-4446*, 1968.
- [18] <http://www.haystack.mit.edu/>.
- [19] L. Muche. Distributional properties of the three-dimensional poisson Delaunay cell. *J. Stat. Physics*, 84:147–167, 1996.
- [20] <http://www.haystack.mit.edu/ast/arrays/mwa/>.
- [21] <http://www.bao.ac.cn/english/>.
- [22] A. Okabe, editor. *Spatial tessellations : concepts and applications of voronoi diagrams*, 2000.
- [23] V. Petrosian, J. Silk, and G. B. Field. A Simple Analytic Approximation for Dusty STRÖMGREN Spheres. *ApJ*, 177:L69+, October 1972.
- [24] E. Platen, R. van de Weygaert, and B. J. T. Jones. A cosmic watershed: the WVF void detection technique. *MNRAS*, 380:551–570, September 2007.
- [25] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical recipes in FORTRAN. The art of scientific computing*. Cambridge: University Press, —c1992, 2nd ed., 1992.
- [26] <http://www.qhull.org/>.
- [27] <http://www.rug.nl/rc>.
- [28] J. Ritzerveld. *the simplicity of transport*. PhD thesis, Sterrewacht Leiden of the University of Leiden, the Netherlands, 2007.
- [29] B. F. Roukema, B. Lew, M. Cechowska, A. Marecki, and S. Bajtlik. A hint of Poincaré do-decahedral topology in the WMAP first year sky map. *A&A*, 423:821–831, September 2004.
- [30] W. Schaap. *DTFE: The Delaunay Tessellation Field Estimator*. PhD thesis, Kapteyn Astronomical Institute of the University of Groningen, the Netherlands, 2007.
- [31] R. M. Soneira and P. J. E. Peebles. A computer model universe - Simulation of the nature of the galaxy distribution in the Lick catalog. *AJ*, 83:845–849, July 1978.
- [32] R. van de Weygaert. *Voids and the geometry of Large Scale Structure*. PhD thesis, Sterrewacht Leiden of the University of Leiden, the Netherlands, 1991.
- [33] R. van de Weygaert. Fragmenting the Universe. 3: The constructions and statistics of 3-D Voronoi tessellations. *A&A*, 283:361–406, March 1994.
- [34] A. C.-C. Yao. A lower bound to finding convex hulls. *J. ACM*, 28:780–787, 1981.