MNRAS **000**, 1–7 (2018) Preprint 18 April 2018 Compiled using MNRAS LATEX style file v3.0

TREVR: A general $O\left(N \log^2 N\right)$ radiative transfer algorithm

J. J. Grond, R. M. Woods, J. Wadsley ⋆ and H. M. P. Couchman
Department of Physics and Astronomy, McMaster University, Hamilton, Ontario L8S 4M1, Canada

Accepted XXX. Received YYY; in original form ZZZ

ABSTRACT

We present TREVR (Tree-based Reverse Ray Tracing), a general algorithm for computing the radiation field in astrophysical simulations. TREVR prioritizes the ability to handle large numbers of sources and computational speed whilst maintaining a specified level of accuracy via adaptive refinement. TREVR is based on a *tree* data structure similar to many gravity and hydrodynamics solvers. This allows for computation of the radiation field in $O(N \log N_{\rm source})$ time without absorption and $O(N \log N_{\rm source} \log N)$ time with absorption. This impressive scaling is made possible by merging sources of radiation according to an opening angle criteria and walking the tree structure to trace a ray. The computational complexity we quote accounts for the use of adaptive refinement, a main feature that is unique to TREVR among other radiative transfer methods. We provide a suite of tests demonstrating the algorithm's ability to accurately compute fluxes, ionization fronts and shadows. We also analyze the algorithm's computational complexity, in how it scales with the number of sources and sinks. Further examinations of how the aforementioned refinement criterion's value affects computational cost and accuracy are presented. Finally, we discuss strengths and shortcomings of this algorithm, how they constrain the types of problems TREVR can handle and how these shortcomings can be remedied if possible.

Key words: radiative transfer – methods: numerical

1 INTRODUCTION

Radiation is arguably the most important physical phenomena to the field of astrophysics. Almost all of the information we receive from outer space comes in the form of photons we detect on or around earth. Understanding the process of radiative transfer (RT) is key in interpreting this information, as the photons are affected by the medium they travel through on their way to our telescopes and detectors. Interactions between photons and the medium do not only affect the photons themselves but the matter as well. Photons and baryons exchange energy and momentum, affect the excitation and ionization states of said baryons and thus determine the chemical and thermodynamic properties of the bulk medium. This in turn makes radiation a driving factor in many of the physical processes we study.

On galaxy scales, the question of how feedback mechanisms affect star and galaxy formation is one of these physical processes we can study. Stellar feedback comes in the form of photoionization by ultraviolet (UV) radiation, stellar winds and supernovae (cite here), the latter of which has been a main focus in simulations in previous years (cite a bunch of SN feedback papers?). It is important to note that even though supernovae might be spectacularly powerful events, ionizing radiative output from stellar populations in the UV regime contribute two orders of magnitude more power at early times and about 50 times more energy over the course of a stellar population's lifetime. This is made evident in Fig. 1, a plot of luminosity output per solar mass as a function of time from stellar populations created via output from the stellar evolution code Starburst99 (Leitherer et al. 1999). However, the way in which this massive output of UV radiation is deposited and consequently affects the interstellar medium (ISM) is still unclear (cite... from fervent paper - conflicting results from Murray 2011, Dale et al. 2012, Faucher-Gigu'ere et al. 2013, Hopkins et al. 2014).

With such a large amount of energy input from stellar sources and questions still left open about how it affects the ISM, it may then come as a surprise that RT has historically been treated rather poorly in most large scale astrophysical simulations, usually as some imposed uniform background. This is not because of carelessness or lack of effort, but because RT is intrinsically a complicated problem. The complexity of this problem is evident in the classical RT equation (e.g. Mihalas & Mihalas 1984),

$$\left[\frac{1}{c}\frac{\partial}{\partial t} + \hat{\mathbf{n}} \cdot \nabla\right] I\left(\mathbf{x}, \hat{\mathbf{n}}, t, \nu\right) = \epsilon\left(\mathbf{x}, \hat{\mathbf{n}}, t, \nu\right) - \alpha\left(\mathbf{x}, \hat{\mathbf{n}}, t, \nu\right) I\left(\mathbf{x}, \hat{\mathbf{n}}, t, \nu\right), \quad (1)$$

⋆ E-mail: wadsley@mcmaster.ca

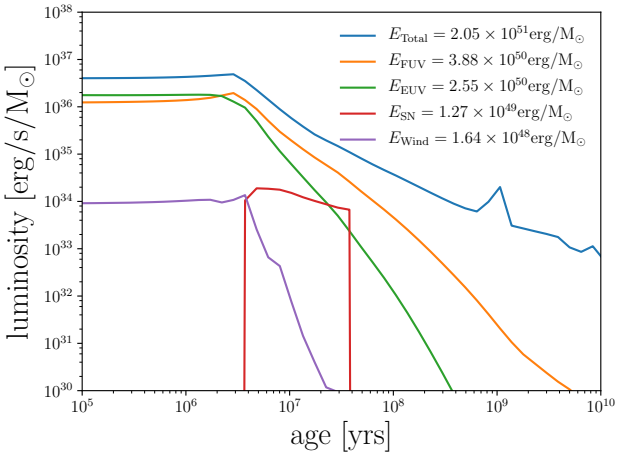

Figure 1. Luminosity per solar mass as a function of time for a stellar population having a Chabrier initial mass function (Chabrier 2003).

where I, ϵ and α are the intensity, emissivity and extinction coefficients respectively and all depend on position $\mathbf{x}$, unit direction of light propagation $\hat{\mathbf{n}}$, time t and frequency ν. Apart from being a seven dimensional problem, RT has a high characteristic speed of c, the speed of light. Also, unlike a force at a distance problem such as gravity, RT depends on the properties of the intervening material which is handled by the extinction coefficient α.

Because of this complexity, a naive numerical solution to the RT problem scales with the number of resolution elements like $O\left(N^{7/3}\right)$. This costly scaling is simply a result of the three major parts that go into computing the radiation field in a simulation. Firstly, a radiation field is represented by a simulation's N resolution elements, so the field intensity needs to be computed at N points in the simulation volume. Secondly, each resolution element's intensity value is made up of contributions from $N_{\rm source}$ sources of radiation. This can also be thought of as $N_{\rm source}$ rays of light being computed per resolution element. This leads to an already nasty scaling for the total number of rays computed in a simulation going as $N_{\rm ray} = N \times N_{\rm source}$, which is like $O\left(N^2\right)$ assuming the $N_{\rm source} = N$. This fact alone limits rudimentary RT methods to only small scale problems (such as...?), where the number of sources is only a handful, to avoid $O\left(N^2\right)$ scaling. Finally, each ray of light interacts with the medium along its path as mentioned earlier. Since the medium is represented by the simulation's N resolution elements and N is proportional to the simulation volume, a one dimensional ray intersects with $O\left(N^{1/3}\right)$ resolution elements. Using the total number of resolution elements interacted with as a measure of computational cost we get that the computational cost is proportional to $N \times N_{\rm source} \times N^{1/3}$, which is like $O\left(N^{7/3}\right)$. This poor scaling with resolution elements makes it unfeasible to simulate RT along with gravity and hydrodynamics methods that scale like $O\left(N \log N\right)$. After breaking down the source of computational complexity in the RT problem it is evident that something needs to be done about the strong dependence on $N_{\rm source}$ and the $N^{1/3}$ cost of computing each ray to attain a feasible RT method.

To avoid the intrinsic complexity of the RT problem, a feasible method would have to solve a simplified RT problem. The first one of these simplifications divides RT methods into two different categories based on how they treat c in Eq. 1. For methods that use a finite c, which is often a reduced speed of light, the partial time derivative remains in Eq. 1 and the radiation field is advected or "evolved". Methods that solve the RT equation in this way, which we will call evolutionary methods, include moment methods like OTVET (Gnedin & Abel 2001) and RAMESE-RT (Rosdahl & Teyssier 2015) as well as photon packet propagation methods like TRAPHIC (Pawlik & Schaye 2008), SPHRAY (Altay et al. 2008) and SimpleX2 (Paardekooper et al. 2010). On the other hand, in limit where c is taken to be infinite, the partial time derivative in Eq. 1 goes to zero and the radiation field can be computed instantaneously. Methods that solve the RT equation in this way, which we will refer to as instantaneous methods, include forward ray tracers such as C^{2}Ray (Mellema et al. 2006), Moray (Wise & Abel 2011) and Fervent (Baczynski et al. 2015) as well as reverse ray tracers such as TreeCol (Clark et al. 2012) and URCHIN (Altay & Theuns 2013).

Instantaneous methods come in the form of ray tracers. Ray tracers are the most simple, natural way to go about solving the RT problem. Forward ray tracers trace many rays outward from sources of radiation, similarly to the actual phenomena, in hope that they will intersect resolution elements for which the radiation field will be computed. Each source needs to compute a number of rays comparable to the number of resolution elements to ensure accuracy, meaning forward ray tracers scale with number of resolution elements like $O\left(N_{\rm source} N N^{1/3}\right)$. This scaling limits forward ray tracers to problems with few sources to avoid $O(N^2)$ like scaling. This also rules out the inclusion of scattering in the method as scatterings are treated as re-emission events and thus all sinks would have to be treated as sources as well.

Recently there has been some focus on reverse ray tracing methods (Clark et al. 2012; Altay & Theuns 2013). Reverse ray tracers trace rays from the sink particle directly to the sources. This way of tracing the rays has a couple of benefits over forward ray tracing. Firstly, tracing from the sinks guarantees the density distribution is well sampled near the resolution element as apposed to forward ray tracing where one would have to increase the number of rays per sink to guarantee accuracy. Put simply, radiation is computed exactly where it is needed. This is especially advantageous in smoothed particle hydrodynamics (SPH) simulations, as low density regions are represented by few SPH particles, and thus extra work is not done to resolve said regions. Another benefit is that sub time steps can be used. However, just performing a naive reverse ray trace does not negate $O\left(N_{\rm source} N N^{1/3}\right)$ scaling with resolution elements, and so the inability to model many sources remains the most significant barrier current instantaneous methods face when trying to solve the general RT problem.

Evolutionary methods do not suffer from the linear dependence on number of sources. The main benefit of evolutionary methods is that they have no dependence on the number of sources, and just scale like $O(N)$ with the number of resolution elements, allowing them to handle large numbers of sources and scattering. Moment methods are limited

to the optically thick diffusive limit. They also lack sharp directionality, resulting in poor shadows behind optically thick objects. This also makes moment methods reliant on the partitioning of space into uniform grids. If implemented in a smooth particle hydrodynamics (SPH) like scheme, the lack of resolution elements in less dense regions would only exacerbate the directionality problem.

Photon packet propagation methods, specifically TRAPHIC (Pawlik & Schaye 2008), perform better in the optically thin regime. TRAPHIC introduces virtual particles (ViPs) to propagate their photon packets in less dense, optically thin regions lacking in SPH particles. They also preserve directionality quite well, however Monte Carlo aspects of how they propagate their photon packets introduce significant noise into their computed radiation field. Monte Carlo resampling is shown to reduce this noise but is quite expensive and deteriorates the initially sharp shadows. Both of these methods scale linearly with resolution elements as mentioned before, but are also forced to operate on every resolution element. In moment methods the radiation field for every grid cell needs to be computed, and in photon packet propagation methods the photon packets hop particle to particle. In the case of TRAPHIC, their N is even greater than the number of SPH particles including the addition of ViPs. Regardless, TRAPHIC is arguably the best general RT method due to its ability to handle both the optically thick and thin regimes with feasible scaling.

We hope from this introduction to the state of the art in RT methods it is apparent that there is room for improvement. Although promising work has been done with reverse ray tracers like TreeCol, a general implementation of one has yet to be published. There is also the problem of scaling with sources in instantaneous methods. If this could be solved, instantaneous ray tracers could compete with the feasibility and improve upon the accuracy of evolutionary codes like TRAPHIC.

2 METHOD

2.1 Simplifications to the full RT problem

Before describing TREVR, let's first define the simplified version of the classical RT equation the method solves. Since TREVR is an instantaneous method, c is set to infinity eliminating the partial time derivative in 1 leaving us with the instantaneous RT equation,

$$\hat{\mathbf{n}} \cdot \nabla I\left(\mathbf{x}, \hat{\mathbf{n}}, t, \nu\right) = \epsilon\left(\mathbf{x}, \hat{\mathbf{n}}, t, \nu\right) - \alpha\left(\mathbf{x}, \hat{\mathbf{n}}, t, \nu\right) I\left(\mathbf{x}, \hat{\mathbf{n}}, t, \nu\right). \quad (2)$$

The emissivity term in the above equation describes a continuous emitting medium. TREVR is a numerical method that assumes sources of radiation are discrete point sources. In this case the emissivity term can be dropped and the solution to the RT equation becomes a linear combination of contributions from all sources. Also, since we are considering sources one by one we can start using the path length s between a source and resolution element as our integration element

$$\frac{dI}{ds} = -\alpha I. \quad (3)$$

In this paper we only consider the extinction contributions from absorption. We can then combine the path length and absorption coefficient to solve for intensity by integrating

$$d\tau = \kappa \rho ds, \quad (4)$$

the optical depth due to absorption, where κ is the opacity due to absorption and ρ is density. This leaves us with

$$\frac{dI}{d\tau} = -I, \quad (5)$$

the final version of the RT problem this method solves. The solution to this equation is

$$I(s) = I(s_0) e^{-\tau(s)}, \quad (6)$$

where $I(s_0)$ is the intensity of the source and $\tau(s)$ is the only quantity to be integrated in our method

$$\tau(s) = \int_{s_0}^{s} \kappa(s) \rho(s) ds. \quad (7)$$

We can use the radiation intensity directly in cooling functions and then take moments of the intensity to obtain useful quantities such as radiation flux

$$\mathbf{F} = \int I cos\theta d\Omega \hat{\mathbf{n}}, \quad (8)$$

and radiation pressure

$$\mathbf{p} = \frac{1}{c} \int I cos^2\theta d\Omega \hat{\mathbf{n}}. \quad (9)$$

Because we assume our sources of radiation to be true point sources, $cos\theta = 1$. If the angular size of a resolution element relative to the source of radiation is small enough such that the small angle approximation holds true, it should also hold that there is a one to one correspondence between intensity and flux. Thus, flux can be written simply as

$$\mathbf{F} = I(s_0) e^{-\tau(s)} \hat{\mathbf{n}} = \frac{L}{4\pi s^2} e^{-\tau(s)} \hat{\mathbf{n}}, \quad (10)$$

where L is the luminosity of the source of radiation. For the remainder of paper we represent the radiation field as flux values computed via the above expression.

2.2 Algorithm

Please note that although TREVR has been initially implemented in the Smoothed Particle Hydrodynamics (SPH) code Gasoline (Wadsley et al. 2004), TREVR is not specific to Gasoline or SPH. The method only requires that the simulation volume is hierarchically partitioned in space.

The TREVR algorithm is based around a tree data structure which partitions the simulation volume hierarchically in space. The smallest resolution elements, SPH gas particles in our case, live in the leaf nodes of the tree data structure. The maximum number of resolution elements per leaf node or "bucket" is a fixed parameter that is set to 10 for all runs in this paper. N resolution elements hold radiation intensity values and represent the radiation field TREVR computes.

2.2.1 Source Merging

As mentioned in the introduction, a naive algorithm would compute interactions between a resolution element and all

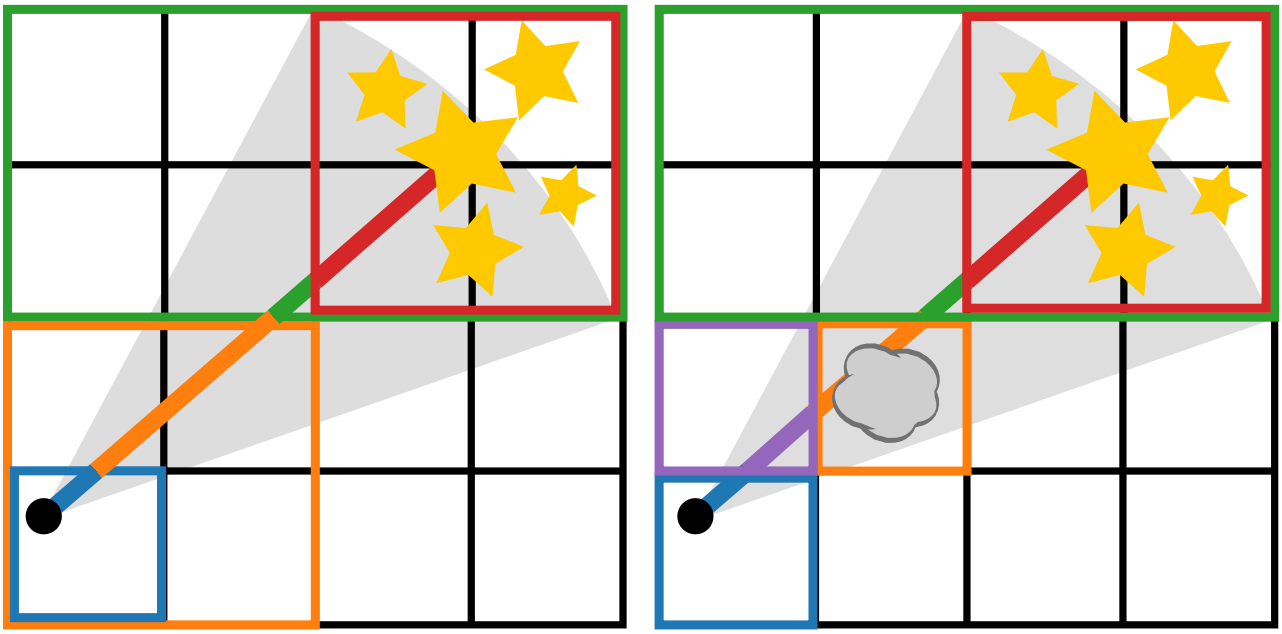

Figure 2.

sources of radiation (SPH star particles in our case). If we assume the number of resolution elements is equal to the number of sources, an infeasible number of interactions would need to be computed, scaling like $O\left(N^2\right)$. To mitigate this N^2 scaling TREVR employs source merging similar to particle merging in the Barnes & Hut (1986) tree-based gravity solver which has remained common place in astrophysical simulations (Benz 1988; Vine & Sigurdsson 1998; Springel et al. 2001; Wadsley et al. 2004; Hubber et al. 2011). Sources of radiation are merged together at their centre of luminosity if they meet an "opening angle" criteria. This criteria is defined as

$$\theta_{\text{open}} > l/s, \tag{11}$$

where l is the side length of a tree cell, r is the distance between the centre of luminosity of a source and centre of mass of a resolution element and θ_{open} is the opening angle, a fixed accuracy parameter. If a group of sources occupy the biggest tree cell that meets this criteria, all sources in that cell are merged into one, considerably reducing the number of interactions TREVR computes. This is illustrated in the left panel of Fig. 2, where the grey angle represents a cell whose angular size meets the opening angle criterion.

The cost savings of source merging can be quantified by integrating to compute the number of tree cells that are "opened" according to the opening angle criteria. These opened cells will have their sources merged making them a count of N_{source}. We can do this by integrating spherical shells of thickness dr along the distance from a resolution element r, and then dividing the sphere volume by the volume of an opened cell, $V_{\text{cell}}(r)$.

$$N_{\text{source}} = \int_{R_0}^{R} \frac{4\pi r^2}{V_{\text{cell}}(r)} dr \tag{12}$$

The bounds of the above integral go from R_0, the length of a bucket cell, to R, the length of the simulation volume. Because the number of particles in a simulation is proportional to the simulation volume, the lower integration limit can be expressed using particle numbers via

$$\frac{R_0}{R} = \sqrt[3]{\frac{N_B}{N}}, \tag{13}$$

the cubed root of the ratio of the average number of particles per bucket, N_B, to the total number of simulation particles. The opened cell volume can also be rewritten by cubing the opening angle criteria

$$V_{\text{cell}}(r) = l^3 = \theta_{\text{open}}^3 r^3. \tag{14}$$

Substituting gives us the following integral and its solution,

$$N_{\text{source}} = \int_{\left(\frac{N}{N_B}\right)^{-\frac{1}{3}}}^{R} \frac{4\pi r^2}{\theta_{\text{open}}^3} dr \propto \log N/N_B. \tag{15}$$

This results means that the number of interactions scales like $O\left(N \log N\right)$. This is also the total cost scaling in the optically thin regime, which is unsurprising given the RT problem is almost identical to the gravity problem in the absence of intervening material. That brings up the next part of the algorithm, what to do about tracing a ray in the optically thick regime.

2.2.2 *Tracing Rays*

In the presence of absorbing material along a ray, the optical depth needs to be computed along said ray by computing the optical depth integral introduced in Eq. 7. To solve this integral numerically, we traverse the tree between the interacting source and resolution element to build up the optical

depth. This is possible because the tree is partitioned in and fills space, thus all the intervening material should be contained in the sub-tree we traverse. Making use of properties computed during the tree build, we can compute the optical depth of a piece of the ray using the geometry of the cell and ray as well as the average density and opacity in the cell

$$\tau_i = \bar{\rho}_i \bar{\kappa}_i s_i. \tag{16}$$

The total optical depth is then summed up during the tree walk,

$$\tau = \sum_i \tau_i, \tag{17}$$

giving us everything needed to evaluate Eq. 10. This process is also illustrated in the left panel of Fig. 2, where there are two important things to note. Firstly, since we are performing a reverse ray trace similar to that of URCHIN (Altay & Theuns 2013), the resolution element denoted by the black circle is intrinsically well resolved at the bucket cell (denoted by the blue cell) level. However, the second point is that as the tree is walked upwards space becomes less resolved. In Fig. 2 the colour of highlighted cells corresponds to which pieces of the ray their properties are used to compute. It should be apparent that the central parts of the ray are less resolved (the green cell) and as you move towards the source or resolution element the ray becomes more resolved (the red and orange cells). This can be looked at in two ways. If the medium is uniform, the algorithm can be extremely efficient while still being able to resolve a sharp feature in the radiation field such as an ionization front. However, if the medium is highly irregular along the ray the algorithm will not be to resolve sharp density distributions which could significantly alter the optical depth. Adaptive refinement is needed during the tree walk to accurately resolve the medium along the ray.

2.2.3 *Adaptive Refinement*

Consider the right panel in Fig. 2. A dense blob of gas to be resolved resides in the orange highlighted cell. At the point in the tree walk where we reach the orange highlighted cell in the left panel, a decision needs to be made on whether the current cell sufficiently represents the media. This decision is made by a refinement criteria. If the cell passes the criteria to refine, rather than using its average properties we recursively check the cell's children until the criteria fails. Thus building a better resolved section of the ray.

Difficulty comes in choosing a refinement criteria that is both accurate and efficient. Ideally, the criteria should be true when an average optical depth in a region may not be accurate to the true distribution, such as a clumpy medium where the average density and opacity is much higher than the "effective" density and opacity (Városi & Dwek 1999; Hegmann & Kegel 2003). For this reason we have chosen optical depth to be the basis of our refinement criteria.

Our optical depth based refinement criteria is unique to TREVR. Consider two rays through through a large cell as in Fig. 3 (note that his description is simplified to 2D). These rays represent what the case would be if properties of the children were used instead of the parent cell. We can compute the minimum and maximum absorption coefficients $\alpha_{\rm min}$ and $\alpha_{\rm max}$, via their average density and opacity values

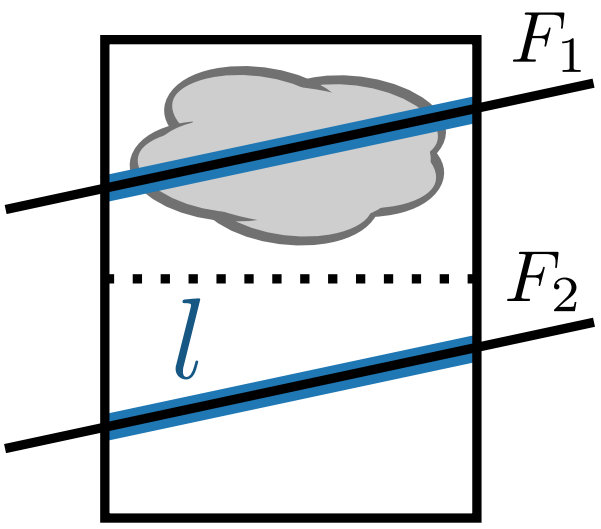

Figure 3.

computed during the tree build. This multiplied by the intersection length l gives us the minimum and maximum optical depths, $\tau_{\rm min}$ and $\tau_{\rm max}$. We can then test the following refinement criteria

$$\tau_{\rm refine} < \tau_{\rm max} - \tau_{\rm min}, \tag{18}$$

and refine if it is true. The fractional error in flux for a chosen value of $\tau_{\rm refine}$ is

$$\frac{F_1 - F_2}{F_1} \leq 1 - e^{-\tau_{\rm max} - \tau_{\rm max}} < \tau_{\rm refine}, \tag{19}$$

for small *tau*, making the refinement criteria a convenient choice of parameter for guaranteeing accuracy.

If the refinement criteria passes at the bucket level, individual particles within a bucket are considered. An straight forward N^2 ray tracing scheme similar to SPHRay (Altay et al. 2008) can be performed. This should contribute negligible cost to the algorithm as the number of particles in a bucket is less than 10.

Fully characterizing the computational cost of the algorithm including the addition of adaptive refinement follows the same method as used earlier. However, now instead of integrating the number of sources we integrate the total number of ray segments computed. We will look at two cases, not refining at all and fully refining down to the bucket level. This will give us upper and lower bound for the algorithms scaling as characterizing the refinement between these extremes depends on the specific density and opacity distributions being operated on.

First let's consider the case where the refinement criteria always passes and all rays are resolved down to the bucket level. The number of segments per ray is then just the length of a ray divided by the size of a bucket. We can express this as

$$N_{\rm seg} = \frac{r}{R_0} = \frac{r}{R}\left(\frac{N}{N_B}\right)^{\frac{1}{3}} \tag{20}$$

after substituting Eq. 13 in for R_0. Since $N_{\rm source}$ is also the

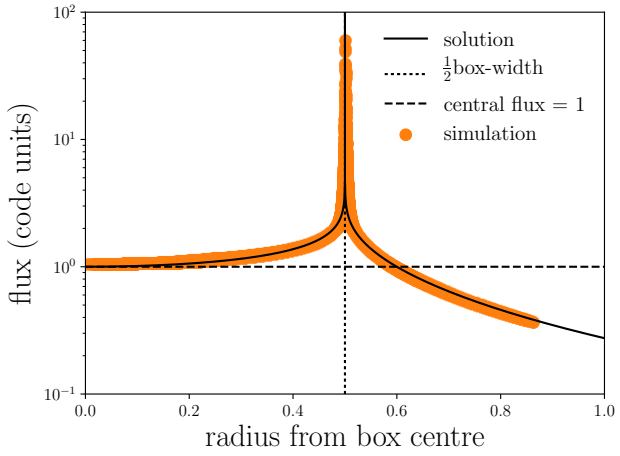

Figure 4.

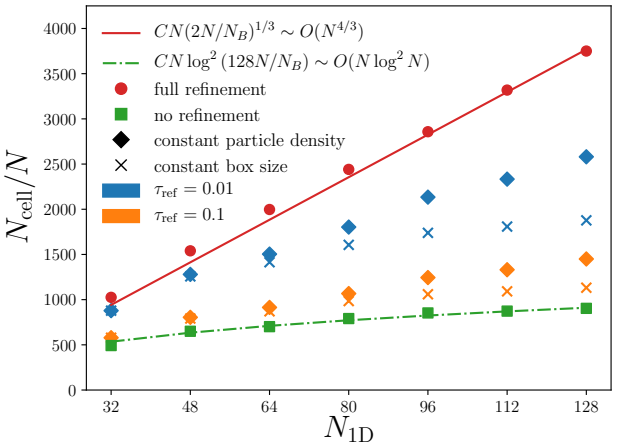

Figure 5.

number of rays computed, the total number of ray segments computed is just Eq. 12 multiplied by the number of ray segments

$$N_{\rm seg} = \int_{\left(\frac{N}{N_B}\right)^{-\frac{1}{3}}}^{R} \frac{4\pi r^2}{\theta^3_{\rm open}} \frac{r}{R} \left(\frac{N}{N_B}\right)^{\frac{1}{3}} dr \propto N(2N/N_B)^{\frac{1}{3}}. \quad (21)$$

This results means that the total cost of the algorithm scales like $O\left(N^{4/3}\right)$ in the worst case.

In the case were the refinement criteria never passes, the ray is split into segments made up of the cells traversed in the tree walk of the sub-tree going from source to resolution element. The number of cells traversed in a tree walk is equal to the logarithm of the number of leaf nodes contained within the sub-tree. The number of leaf nodes in the sub-tree is also given by Eq. 20, so by taking the logarithm of Eq.20 and adding two for the two buckets on eiter side of the sub-tree we come to

$$N_{\rm seg} = \log_2 \left[\frac{r}{R} \left(\frac{N}{N_B}\right)^{\frac{1}{3}} \right], \quad (22)$$

where the logarithm is base two as TREVR is implemented with a binary tree. Like before we multiply Eq. 12 by the number of ray segments and integrate the following

$$N_{\rm seg} = \int_{\left(\frac{N}{N_B}\right)^{-\frac{1}{3}}}^{R} \frac{4\pi r^2}{\theta^3_{\rm open}} \log_2 \left[\frac{r}{R} \left(\frac{N}{N_B}\right)^{\frac{1}{3}} \right] dr \propto N\log^2(128N/N_B). \quad (23)$$

This results means that the total cost of the algorithm scales like $O\left(N\log^2 N\right)$ in the best case.

2.3 Cosmological Background Radiation

In order to treat cosmological simulations properly we must account for the radiation coming from the rest of the universe outside of the simulation volume. Most current codes apply a constant UV field to the entire box, essentially the lowest order approximation possible. Some specialized codes like URCHIN altayTheuns13 do a reverse ray trace to the edge of the box, where the background flux is assumed to be coming from. Others, such as TRAPHIC pawlikSchaye08 allow their ray trace to be periodic. We believe that this periodic treatment is problematic for reasons we will explain at the end of this subsection (well... we haven't figured it out yet).

Instead, we have implemented a method involving "background sources". "Background" particles are distributed in a spiral pattern on the surface of a sphere at the very edge of the simulation volume (or at a large distance if required) and the number of sources can be varied to match the required angular resolution of the background. Finding the flux at the centre of a sphere of sources is a problem akin to Newton's Shell Theorem. However, because the intensity does not cancel like force, the solution differs and is as follows:

$$F(r) = \frac{L}{8\pi R} \ln\left(\frac{R+r}{R-r}\right), \quad (24)$$

where L is the total luminosity of the emitting shell, R is the radius of the sphere and r is the radius the flux is being computed at. The shape of the function can be seen in Figure 4 where we have plotted the flux as a function of radius for a homogeneous, optically thin test volume.

Note that due to the logarithm in equation 24, the flux is nearly constant at small radii. Since most cosmological zoom in simulations only consider gas at a fairly small radius, this setup of background sources is an acceptable method to provide a cosmological background flux. A benefit of this method is that we can use all of the existing machinery described in the methods section, and only have to add temporary background star particles as the source of the background radiation. This way, there is no need to create periodic copies of the simulation volume (talk to James, explain <-this here).

3 CODE TESTS

3.1 Scaling with number of resolution elements

To test the scaling claims made in the previous section, a particle distribution that both our source merging and adaptive refinement procedures can latch onto is necessary. We

have opted to use a sinusoidally perturbed glass as the initial condition (IC) for tests benchmarking TREVR's scaling, opening angle and refinement criteria. This IC began life as a unit cube glass of particles with an average density and opacity of 1. The glass was then perturbed by applying 24 sinusoidal modes resulting in a smoothly varying density distribution where 49% of particles have a density between 0 & 1, 43% between 1 & 2, 5% between 2 & 3, 1% between 3 & 4 and the remaining 2% of particles are between densities of 4 & 24. Exact details on this IC can be found in Appendix A.

Fig. 5 is a plot of the number of ray segments computed per resolution element as a function of the number of resolution elements in the previously described test. The number of resolution elements making up the IC is increased along the x-axis from 32^3 to 128^3 in steps of 16^3 in two different ways. Firstly, we increase the particle number while keeping the box size constant. This represents a case similar to increasing the resolution of an isolated galaxy or the like and is denoted by the x marker. Secondly, we increase the box size while keeping the *particle* density constant. This represents a case similar to increasing the simulation volume of a cosmological simulation and is denoted by the diamond marker.

Four different levels of refinement are also shown in this plot. Green plot markers represent not refining, red markers

3.2 Opening Angle Testing

3.3 Refinement Criteria Testing

3.4 Isothermal Spheres

3.5 Strömgren Sphere Test

4 DISCUSSION AND CONCLUSION

ACKNOWLEDGEMENTS

REFERENCES

Altay G., Theuns T., 2013, MNRAS, 434, 748
Altay G., Croft R. A. C., Pelupessy I., 2008, MNRAS, 386, 1931
Baczynski C., Glover S. C. O., Klessen R. S., 2015, MNRAS, 454, 380
Barnes J., Hut P., 1986, Nature, 324, 446
Benz W., 1988, Computer Physics Communications, 48, 97
Chabrier G., 2003, PASP, 115, 763
Clark P. C., Glover S. C. O., Klessen R. S., 2012, MNRAS, 420, 745
Gnedin N. Y., Abel T., 2001, New Astron., 6, 437
Hegmann M., Kegel W. H., 2003, MNRAS, 342, 453
Hubber D. A., Batty C. P., McLeod A., Whitworth A. P., 2011, A&A, 529, A27
Leitherer C., et al., 1999, ApJS, 123, 3
Mellema G., Iliev I. T., Alvarez M. A., Shapiro P. R., 2006, New Astron., 11, 374
Mihalas D., Mihalas B. W., 1984, Foundations of radiation hydrodynamics. Courier Corporation
Paardekooper J.-P., Kruip C. J. H., Icke V., 2010, A&A, 515, A79
Pawlik A. H., Schaye J., 2008, MNRAS, 389, 651
Rosdahl J., Teyssier R., 2015, MNRAS, 449, 4380
Springel V., Yoshida N., White S. D. M., 2001, New Astron., 6, 79
Városi F., Dwek E., 1999, ApJ, 523, 265
Vine S., Sigurdsson S., 1998, MNRAS, 295, 475
Wadsley J. W., Stadel J., Quinn T., 2004, New Astron., 9, 137

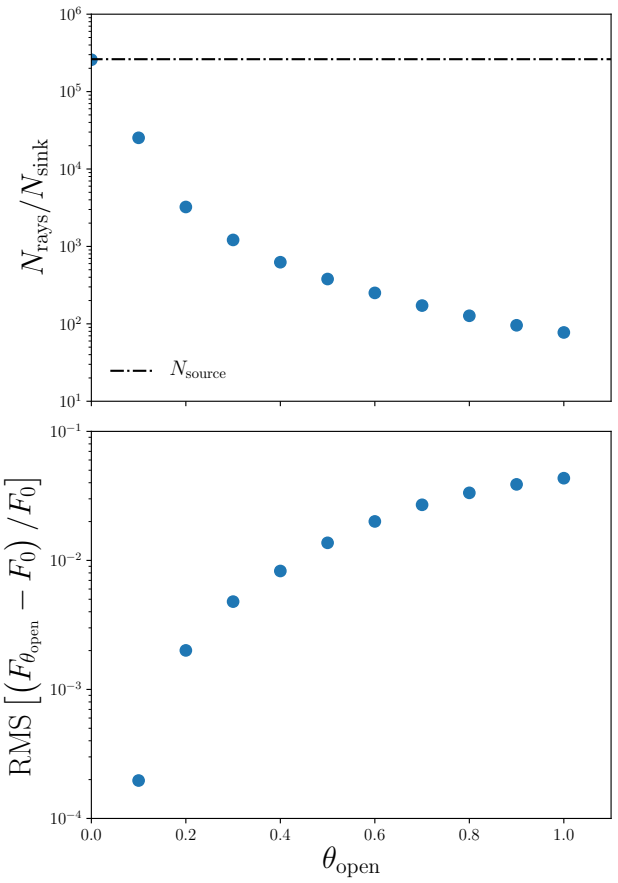

Figure 6.

Wise J. H., Abel T., 2011, MNRAS, 414, 3458

APPENDIX A: TEST INITIAL CONDITIONS

This paper has been typeset from a TEX/LATEX file prepared by the author.

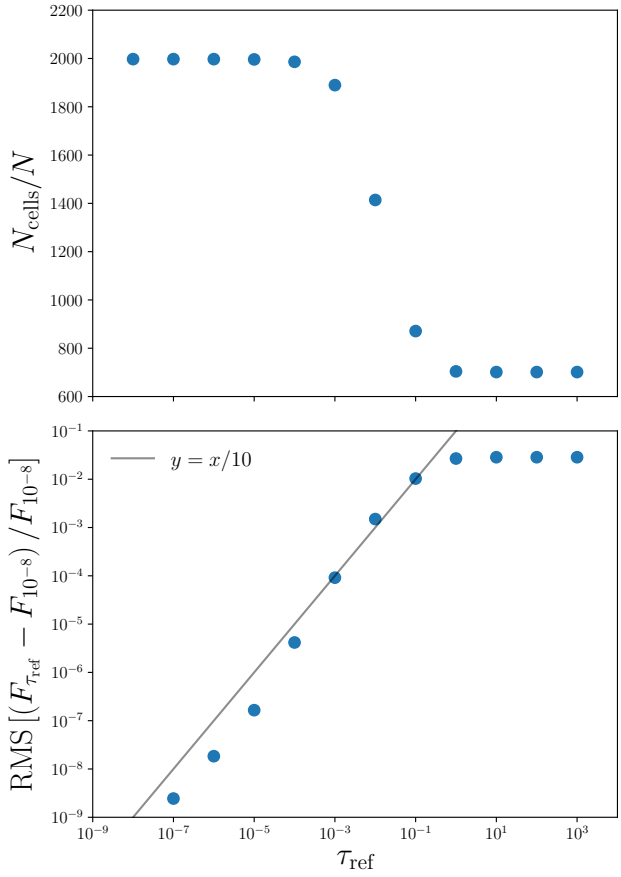

Figure 7.

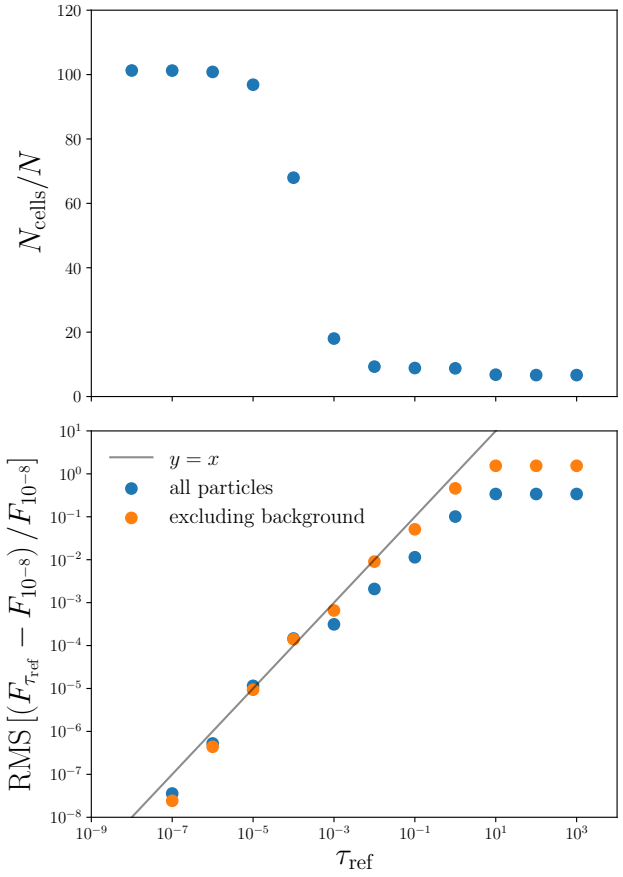

Figure 8.

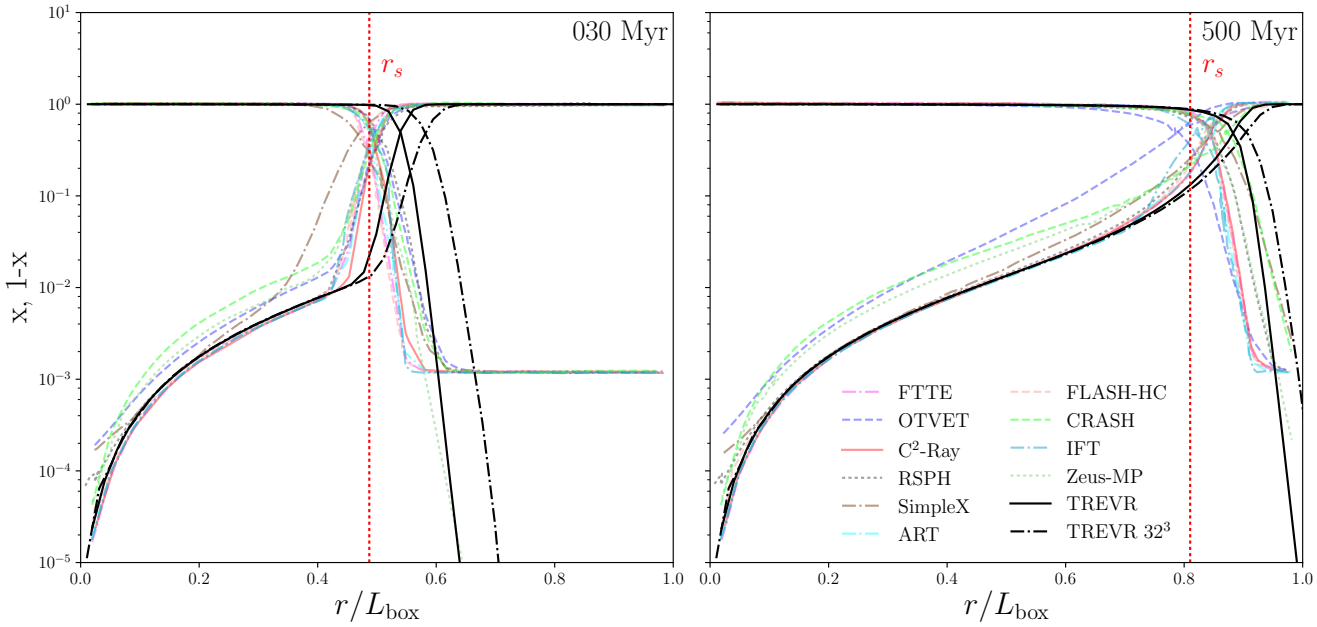

Figure 9.

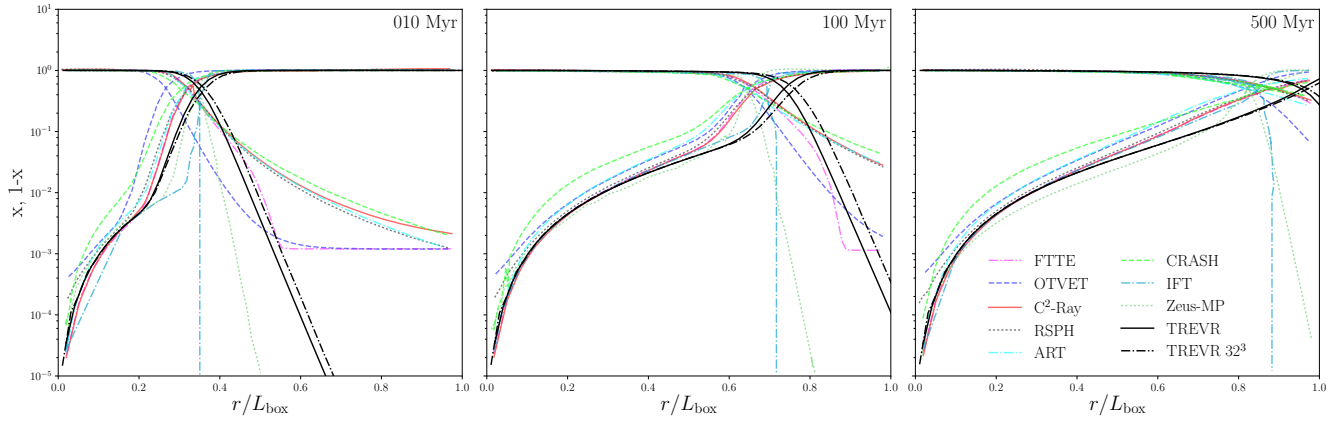

Figure 10.

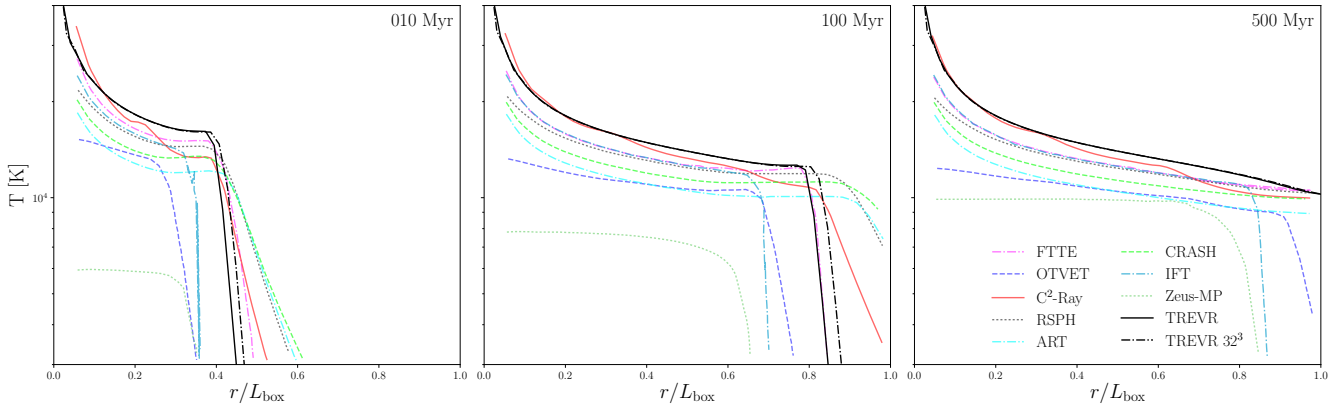
010 Myr
100 Myr
500 Myr
T [K]
10^4
$r/L_{\rm box}$
0.0
0.2
0.4
0.6
0.8
1.0
FTTE
OTVET
C^2-Ray
RSPH
ART
CRASH
IFT
Zeus-MP
TREVR
TREVR 32^3

Figure 11.