

# Real Time Visualization of Debris Disks in Scattered Light and Thermal Radiation

Robin Leman<sup>a,b,1</sup>, Eve J. Lee<sup>a,c,d,2</sup>, and Clark Verbrugge<sup>b,2</sup>

<sup>a</sup>Department of Physics, McGill University, Montréal, Québec, Canada; <sup>b</sup>School of Computer Science, McGill University, Montréal, Québec, Canada; <sup>c</sup>McGill Space Institute, McGill University, Montréal, Québec, Canada; <sup>d</sup>Institute for Research on Exoplanets, Montréal, Québec, Canada

April 2022

This report builds on the work of Lee & Chiang (2016) (1) to visualize and characterize the morphologies of debris disks in solar systems. By using 3-dimensional computer simulation, debris disks were accurately rendered in scattered light and thermal radiation. The minimal model simulated is a "narrow ring of parent bodies, secularly forced by a single planet on a possibly eccentric orbit, colliding to produce dust grains that are perturbed by stellar radiation pressure" (Lee & Chiang (2016) (1)). In scattered light, the rendering accuracy was improved by using a linear combination of three Henyey-Greenstein phase functions. By assuming the disk to be optically thin, thermal radiation was found successful at probing larger dust masses and removing scattering artifacts. Overall, the model was found useful at accurately characterizing the disk morphologies in scattered light and thermal radiation. By leveraging real time computer graphics techniques, we were able to render up to  $1e7$  particles at  $61 \pm 4$  FPS (Frame Per Second), with an initial loading time of  $1.56 \pm 0.05$  s.

## 1. INTRODUCTION

A debris disk is a collection of dust particles, remnants of planetesimal collisions, orbiting a solar system older than 8 Myrs (Michel et al (2021) (2)). These dust particles are gravitationally perturbed by the planets of the solar system, directly affecting their orbits (Cataldi (2016) (3)). This results in various debris disk shapes, characterized by the planets' orbits, the viewing angle of the disk, and the observed wavelength. Analyzing the morphology of a debris disk can thus give critical information on the presence of exoplanets in the system. In previous papers, different morphologies have been discovered: "moths", "needles", "ship-and-wakes", "bars" and "eccentric rings" (Lee & Chiang (2016) (1)).

In this report, 3-dimensional rendering techniques are used to visualize the morphologies of a debris disk orbiting a solar system composed of a single planet, with a possibly eccentric orbit. Building up on the work of Lee & Chiang (2016) (1), a more performant and practical tool is presented to understand the various shapes of a debris disk at different viewing angles, in both scattered light and thermal emission.

Background theory is detailed in Section 2. In Section 3, the model used for the simulation is described, with a key emphasis on the improvements from the work of Lee & Chiang (2016) (1). This model has two main features: scattered light, using a linear combination of phase functions, and thermal emission, to visualize debris disks at sub-millimeter wavelengths. The rendering methods are presented in Section 4, demonstrating the use of real time computing techniques to make the simulation more practical and performant. Finally, the results and performances of the tool are analysed and discussed in Sections 5 and 6.

## 2. THEORY

### A. Celestial Dynamics

To visualize the morphology of a debris disk, the orbits of the dust particles composing the disk have to be computed. To represent an orbit, Keplerian elements are used, as described by Murray & Dermott (1999) (4).

These Keplerian elements arise when solving the two body

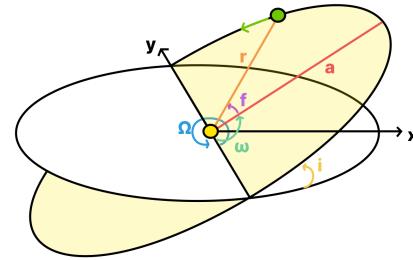


Fig. 1. Schematic of the Keplerian elements used to represent an orbit.

problem is celestial dynamics (4). Considering two bodies with positions  $\vec{r}_1, \vec{r}_2$ , and masses  $m_1, m_2$ , the gravitational forces  $F_1, F_2$ , acting respectively on  $m_1, m_2$  due to  $m_2, m_1$  are

$$\vec{F}_1 = G \frac{m_1 m_2}{r^3} \vec{r} = m_1 \ddot{\vec{r}}_1 \quad [1]$$

$$\vec{F}_2 = -G \frac{m_1 m_2}{r^3} \vec{r} = m_2 \ddot{\vec{r}}_2 \quad [2]$$

with  $G$  the universal gravitational constant, introducing the differential equation

$$m_1 \ddot{\vec{r}}_1 + m_2 \ddot{\vec{r}}_2 = 0. \quad [3]$$

This is reduced to the differential equation

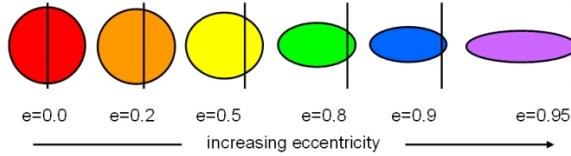
$$\ddot{\vec{r}} + \alpha \frac{\vec{r}}{r^3} = 0 \quad [4]$$

by letting  $\vec{r} = \vec{r}_2 - \vec{r}_1$  and  $\alpha = G(m_1 + m_2)$ .

This differential equation is solved by

<sup>1</sup>Main Author

<sup>2</sup>Project Supervisor



**Fig. 2.** Schematic showing the various shapes of an orbit depending on the eccentricity  $e$ . The vertical line represents an axis of reference with the focus at its center. From COSMOS - The SAO Encyclopedia of Astronomy

$$r = \frac{a(1 - e^2)}{1 + e \cdot \cos f} \quad [5]$$

where  $a$  is defined as the semi major axis,  $e$  the eccentricity, and  $f$  the true anomaly. These elements describe the orbit of the celestial body, and are visually represented in Figure 1. An interactive simulation of orbits can be accessed [here](#).

The semi-major axis is the sum of the distance from the perihelion to the aphelion divided by two. The perihelion is the closest point on the orbit from the focus, and the aphelion is the furthest point. The eccentricity  $e$  describes the shape of the orbit, as seen in Figure 2. The debris disks of interest are produced with particles on elliptical orbits, so  $0 \leq e < 1$ . The true anomaly  $f$  defines the position of the orbiting body at a time  $t$ : it is the angle between the radius  $r$  and the direction of perihelion. The true anomaly is defined in reference to the argument of perihelion  $\omega$ , the angle between the direction of perihelion and some reference plane, in the plane of the orbit.

We need two other parameters to represent the orbit relative to the reference plane. The inclination  $i$  represents the vertical tilt of the orbit, and the longitude of the ascending node  $\Omega$  represents where the orbital plane crosses the reference plane.

To relate the true anomaly  $f$  to a time  $t$ , the mean anomaly  $M$  and eccentric anomaly  $E$  are introduced. The mean anomaly is simply

$$M = \frac{2\pi}{T}(t - \tau) \quad [6]$$

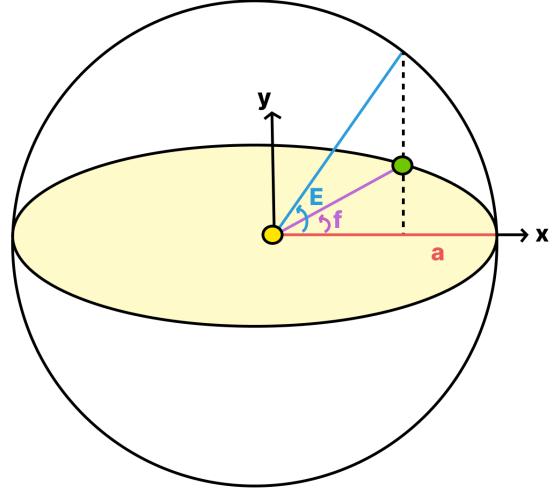
with  $T$  the orbit period and  $\tau$  the time of passage at the perihelion. The eccentric anomaly is defined as the angle between the semi-major axis and the distance from the center to the intersection point on the circumscribed circle, as shown in Figure 3.

The true, mean and eccentric anomalies are related by

$$M = E - e \sin E, \quad [7]$$

$$\tan \frac{f}{2} = \sqrt{\frac{(1+e)}{(1-e)}} \tan \frac{E}{2}. \quad [8]$$

This will be useful to draw the dust particles on an orbit by uniformly distributing bodies on the true or mean anomaly. Distributing an orbit uniformly on the true anomaly divides the orbit into equal amount of distances, and distributing an orbit uniformly on the mean anomaly divides the orbit into equal amount of times.



**Fig. 3.** Schematic of the geometrical meanings of the true and eccentric anomalies.

## B. Formation of Debris Disks

The formation of planets, according to the core accretion model (Maoz (2016) (5)), is the result of planetesimal collisions in the protoplanetary disk of a solar system. A rocky core forms through the coagulation of small planetesimals colliding, until it is massive enough to accrete a gaseous envelope. These collisions are not perfect mergers, which means that two bodies colliding will result in some degree of shattering. These ejected mass fragments will then collide with other fragments, starting a collisional cascade grinding the material to sub-micron sized dust particles.

Depending on radiation pressure and the systems' gravitational potential, these dust particles can be ejected from the system if their size  $s$  is smaller than a threshold blowout size  $s_{\min}$  (Hughes et al. (6)). On the contrary, particles with sizes larger than the blowout size are bound and periodically orbit the system. Since radiation forces and stellar pressure continuously eject particles of sizes  $s < s_{\min}$ , the detection of a debris disk implies that the system was capable of producing bodies hundreds of kilometers in size (Hughes et al. (6)).

When the gas of the protoplanetary disk has been dispersed, at around 8 Myrs old (Michel et al (2021) (2)), the orbiting dust particles form a debris disk surrounding the solar system.

## C. Distribution of Radiation

Radiation has a major role in both the orbit and the light intensity of a dust particle composing the debris disk. In this model, the effect of radiation is computed similarly to Lee & Chiang (2016) (1). The ratio of the force of stellar radiation  $F_{\text{rad}}$  and stellar wind  $F_{\text{wind}}$  to that of stellar gravity  $F_{\text{grav}}$  is defined as:

$$\beta = \frac{F_{\text{rad}} + F_{\text{wind}}}{F_{\text{grav}}}. \quad [9]$$

This relates  $\beta$  to the size  $s$  of a dust particle as described by Silsbee & Draine (2016) (7). Gravity scales as the mass of the dust particle, which itself scales as its volume:  $F_{\text{grav}} \propto s^3$ .  $F_{\text{rad}}$  and  $F_{\text{wind}}$  are both proportional to the cross sectional

area of the dust particle, which scales as  $F_{\text{wind}} + F_{\text{wind}} \propto s^2$ . This results in

$$\beta \propto s^{-1}. \quad [10]$$

Assuming the grain size to follow a power law distribution  $dN/ds \propto s^{-7/2}$  (Mathis et al (1977) (8)), we obtain a  $\beta$ -distribution following a power law  $dN/d\beta \propto \beta^{3/2}$ . However, this distribution underestimates grains on high eccentricities and high semi-major axis that avoided inter-particle collisions. To correct this, the distribution is scaled by the orbital period  $T_d$  of the dust particle, as described by Strubbe & Chiang (2006) (9). This gives the distribution

$$\frac{dN}{d\beta} \propto \beta^{3/2} T_d. \quad [11]$$

### 3. MODEL

#### A. Resolving dust orbits in Keplerian coordinates

The orbits of the dust particles forming the debris disk are computed using the same methods as Lee & Chiang (2016) (1). The model starts from parent bodies, defined as the bodies at the top of the collision cascade, grinding the material down to dust particles.

We instantiate  $N_p = 1000$  parent body orbits in a uniformly distributed ring located just outside the planet's elliptical orbit:

$$a_{\text{inner}} = a_{\text{planet}} \left[ 1 + 2(M_{\text{planet}}/M_{\star})^{2/7} \right], \quad [12]$$

corresponding to the planet's chaotic zone (Lee & Chiang (2016) (1)), with  $a_{\text{inner}}$  the inner semi-major axis of the ring. The disk has a width  $0.1a_{\text{inner}}$  and is apsidally aligned with the planet's orbit. The eccentricity of a parent body  $e_p$  is computed as the sum of its forced and free eccentricities, assuming the parent bodies to be secularly forced by the planet (Murray & Dermott (1999) (4)). From the parent body orbits,  $N_{\text{launch}} = 100$  true anomalies are uniformly distributed along the orbit. A dust particle orbit is created at each true anomaly, with Keplerian parameters

$$a_d = \frac{a_p(1 - e_p^2)(1 - \beta)}{1 - e^2 - 2\beta(1 + e_p \cos f_p)} \quad [13]$$

$$e_d = \frac{\sqrt{e_p^2 + 2\beta e_p \cos f_p + \beta^2}}{1 - \beta} \quad [14]$$

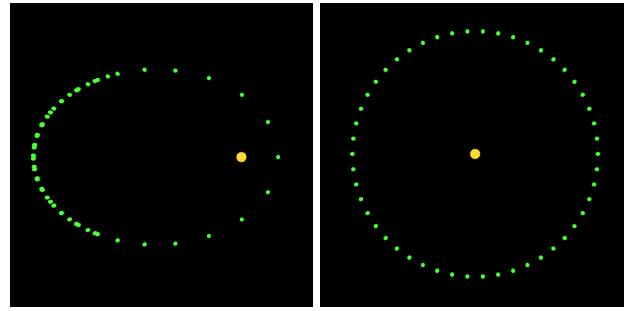
$$\omega_d = \omega_p + \arctan \left( \frac{\beta \sin f_p}{e_p + \beta \cos f_p} \right) \quad [15]$$

$$i_d = i_p \quad [16]$$

$$\Omega_d = \Omega_p \quad [17]$$

where the  $p$  and  $d$  subscripts respectively represent the parent and dust orbital parameters, with  $a$  the semi-major axis,  $e$  the eccentricity,  $\omega$  the argument of perapse,  $\Omega$  the longitude of the ascending node,  $f_p$  the launch true anomaly, and  $\beta$  drawn randomly from the  $\beta$ -distribution described in Section 2.

For a more performant simulation, the dust particle orbits are pre-computed on a CPU and their Keplerian parameters stored in a text file. At the start of the tool, the orbits are loaded from the text file and converted to particles in Cartesian coordinates.



**Fig. 4.** Effect of uniformly distributing 50 bodies along the mean anomaly of an orbit. On the left, an orbit with eccentricity 0.7, on the right, eccentricity of 0.

#### B. Conversion from Keplerian orbits to Cartesian particles

Since the simulation is static, the full orbit of each dust particle has to be drawn to correctly visualize the morphology of the disk. In order to draw the full orbit,  $N_d = 100$  dust particles are drawn along each dust particle orbit.

These dust particles are distributed uniformly on the mean anomaly  $M_d$  of the particle orbit. Distributing the mean anomaly is equivalent to dividing the orbit into equal interval of times. This will concentrate the dust particles far from the central star, around the apoapse, as this is where they spend most of their time on the orbit. This is visually represented in Figure 4.

Once the mean anomaly  $M_d$  is distributed, Equation 7 is solved for the eccentric anomaly  $E_d$  using Newton's method. The iteration is stopped either when the new iteration diverges from the previous by less than a threshold  $\delta = 0.0001$  or when the number of iterations overpasses 100. Finding  $E_d$  allows to compute the true anomaly  $f_d$  with Equation 8. The orbit radius  $r_d$  is then computed from Equation 5.

Finally, the particles' positions are converted to Cartesian coordinates:

$$\begin{aligned} x_d &= r_d \cdot (\cos \Omega_d \cos(\omega_d + f_d) - \sin \Omega_d \sin(\omega_d + f_d) \cos i_d) \\ y_d &= r_d \cdot (\sin \Omega_d \cos(\omega_d + f_d) - \cos \Omega_d \sin(\omega_d + f_d) \cos i_d) \\ z_d &= r_d \cdot \sin(\omega_d + f_d) \cos i_d \end{aligned}$$

from Schwartz (2017) (10) and Murray & Dermott (1999) (4). Since the dust particles are static, we compute their positions on the CPU at the launch of the software, before the first rendered frame. This function has a linear complexity  $O(n)$ , with  $n$  the number of particles. Since it has to be repeated for 100,000 dust orbits, and since each conversion is independent, multi-threading is used to split the workload equally along the available machine's threads. A detailed description of the hardware used for the simulation can be found in Appendix B.

### 4. RENDERING

The intensity  $I_d$  of every dust particle is computed according to the chosen rendering mode, scattered light or thermal radiation, as described in the following sections.

This intensity is then mapped to a 1-dimensional texture, or colormap, outputting a corresponding colour  $C$ . The colormaps for the different rendering modes are presented in Appendix A.

The dust particles are then projected onto a square pixel grid of the desired resolution, from the point of view of a camera. The camera is constrained to a spherical plane of radius  $d$ , azimuth angle  $\varphi$  and polar angle  $\theta$ , with the origin defined at the star of the solar system. Those parameters allow us to simulate the position of the debris disk from the point of view of an observer looking from a distance  $d$  at the star of the solar system.

The dust particles in a same pixel contribute to the same amount  $\alpha$  of alpha blending (Eck (2021) (11)). An additive blending method is used where, if the current colour of a pixel is  $C_{\text{current}}$ , and a dust particle with colour  $C_d$  is added onto the pixel, the resulting pixel colour will be

$$C_{\text{result}} = C_d \cdot \alpha + C_{\text{current}}. \quad [18]$$

We render the dust particles in a real-time environment, taking user input for the position of the camera. We compute the dust particle intensity and draw the particles at every frame, with a simulation running at an adaptive frame rate: a new frame starts as soon as the previous one has been rendered.

This process can be summarized by the Engine Loop Algorithm 1.

---

**Algorithm 1** Engine Loop

---

```

createWindow()
instantiateDustParticles()
while isRunning do
    processInput()
    render()
shutdownWindow()
```

---

The rendering is performed with the OpenGL (12) graphics API for a multi-platform software. The window is managed by the GLFW library. GLM is used as the math library for its compatibility with OpenGL.

The software contains two types of user input. The first one, through peripherals, allows for direct control of the camera, locked in a spherical plane centered at the star of the system. The second one is through an editor, managed through the ImGUI library.

The rendering pipeline is as follows. Firstly, the vertex stage iterates over every particles and projects them onto a pixel grid. Then, it computes the intensity  $I_d$  of the particle. Secondly, the fragment stage maps the intensity  $I_d$  to the colormap texture, and performs alpha blending to add the contribution of every particle to the pixel. The rendering pipeline is performed on a GPU.

### A. Scattered Light

In scattered light, the intensity  $I_d$  of a dust particle is computed by simulating a ray of light leaving the host star, colliding with the dust particle, and scattering to the observer.

For this, the probability that a ray scatters at an angle  $\theta_d$  needs to be computed. Here,  $\theta_d$  is defined as the angle between the star, the dust particle, and the observer. This probability follows an angular distribution called a phase function. In 1941, Henyey-Greenstein (13) empirically derived a phase function

$$\tilde{\beta}_{HG}(g, \theta_d) = \frac{1}{4\pi} \frac{1-g^2}{(1+g^2 - 2g \cos \theta)^{3/2}} \quad [19]$$

that mimics light scattering of small interstellar dust. A positive  $g$  mimics forward scattering, and a negative  $g$  mimics backward scattering.

By directly measuring the scattering phase function of the Saturn G and D rings, Hedman & Stark (2015) (14) found that the best fit for nearby debris disk in scattered light was a linear combination of three Henyey-Greenstein functions:

$$\begin{aligned} \tilde{\beta}(\theta_d) = & 0.754 \tilde{\beta}_{HG}(0.995, \theta_d) \\ & + 0.151 \tilde{\beta}_{HG}(0.585, \theta_d) + 0.095 \tilde{\beta}_{HG}(0.005, \theta_d). \end{aligned}$$

The dust particle intensity is then computed as

$$I_d \propto \tilde{\beta}(\theta_d) \frac{1}{\beta_d^2 r_d^2} \quad [20]$$

where  $1/\beta_d^2$  accounts for the scattering cross section of each grain, and  $1/r_d^2$  for the inverse square law.

The alpha blending introduced in Equation 18 sums the intensity of all dust particles in a same pixel.

### B. Thermal Radiation

At sub-millimeter wavelengths, radio images probe the thermal emission of the debris disk (Draine (2006) (15)). By considering the disk to be optically thin, meaning that light is not easily absorbed by the disk and passes through easily, the intensity of dust thermal emission is proportional to the column density:

$$I_d = \kappa_\nu^{abs}(s) \cdot \Sigma_d \cdot B_\lambda(T_d) \quad [21]$$

where  $\kappa_\nu^{abs}(s)$  is the dust absorption opacity,  $s$  the dust grain size,  $\Sigma_d$  the dust surface density, and  $B_\lambda(T_d)$  Planck's Function with  $T_d$  the effective temperature of the dust grain and  $\lambda$  the observed wavelength.

To simplify this, an average grain size  $s$  is used for all particles, assuming that the disk follows a power law distribution

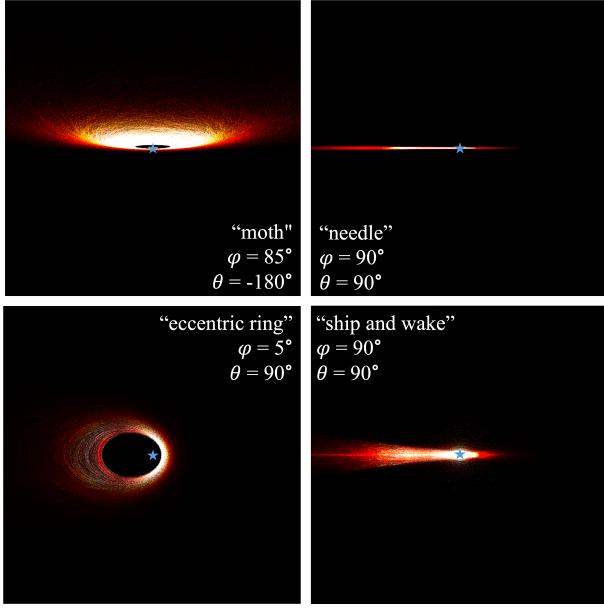
$$\frac{dN}{ds} \propto \begin{cases} s^{-7/2}, & \text{for } s_{\min} \leq s \leq s_{\max}. \\ 0, & \text{otherwise.} \end{cases} \quad [22]$$

As described by Tilman Birnstiel et al. (2018) (16), a constant dust absorption opacity  $\kappa_\nu^{abs}$  is used for  $s_{\min} = 10^{-5}$  cm and  $s_{\max} = 1$  mm.

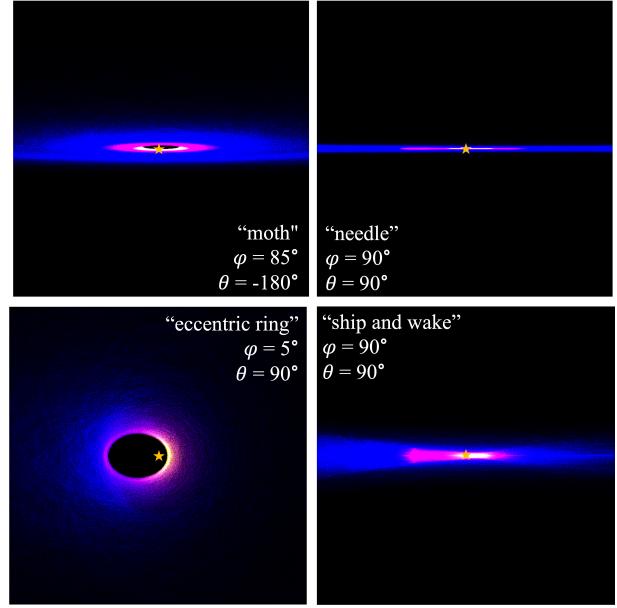
In the simplified treatment of thermal emission, all dust particles have effectively the same size, so the surface density  $\Sigma_d$  of a single dust particle is constant. The total surface density of a pixel is accounted for by the alpha blending in Equation 18, with  $\alpha$  equal to the area of a pixel. Here, the role of alpha blending is to scale the pixel intensity proportionally to the number of particles per pixel. The intensity of a dust particle is thus reduced to:

$$I_d \propto B_\lambda(T_d). \quad [23]$$

The effective temperature of a dust particle  $T_d$  is computed assuming that each grain is a black-body and radiated by the central star. For a star of radius  $R_S$  and effective temperature  $T_S$ , the Stefan-Boltzmann law gives the total radiation power emitted at the surface of the star:



**Fig. 5.** Various debris disk morphologies rendered in scattered light. The top images feature a system with low planet eccentricity and low inclination variation. The lower images feature a system with high planet eccentricity and high inclination variation.



**Fig. 6.** Same morphologies and systems as Figure 5 but rendered in thermal radiation. The morphologies are easily recognizable, at the exception of the "ship and wake" and "moth" morphologies.

$$P_S = \sigma T_S^4 4\pi R_S^2 \quad [24]$$

with  $\sigma$  the Stefan-Boltzmann constant. At a distance  $r_d$  from the star, the energy flux received by the dust particle is

$$S_d = \frac{P_S}{4\pi r_d^2} \quad [25]$$

by the inverse square law. So,

$$S_d = \sigma T_S^4 \frac{R_S^2}{r_d^2}. \quad [26]$$

By using the Stefan-Boltzmann law,

$$S_d = \sigma T_d^4. \quad [27]$$

By equating the last two equations, the effective temperature  $T_d$  of the dust particle is

$$T_d = T_S \sqrt{\frac{R_S}{r_d}}. \quad [28]$$

## 5. RESULTS

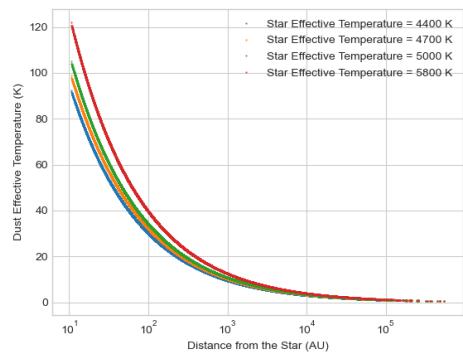
A total of  $N_p \cdot N_{\text{launch}} \cdot N_d = 10^7$  dust particles are rendered on a 1080x1080 pixel grid, in order to conserve a square window of standard screen resolution. For the presented results, the dust particles' intensities are uniformly scaled up or down. This allows for a better image contrast and better classification of the morphologies.

For thermal radiation, a sub-millimeter wavelength of  $\lambda = 0.087$  cm is chosen as described by Tilman Birnstiel et al (2018) (16), for a corresponding average absorption opacity  $\kappa_\nu^{\text{abs}} = 3.558 \text{ cm}^2 \text{ g}^{-1}$ . The star is assumed to have a radius  $R_S = 1R_\odot$  and effective temperature  $T_S = 4700$  K. The distribution of dust temperatures for different stellar effective temperatures is shown in Figure 7.

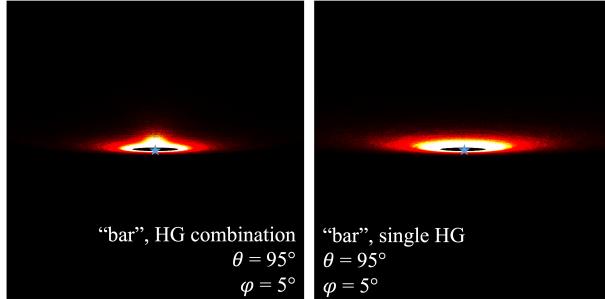
Different disk morphologies, depending on the viewing angle and the orbital properties of the system, are presented in scattered light Figure 5 and in thermal emission Figure 6.

These results show the dependence of debris disk morphology on both the viewing angle of the disk and the orbital properties of the system. When viewed face on, a disk with high planet eccentricity will appear as an "eccentric ring" (Figure 5, bottom left), but a disk with low planet eccentricity will appear as a "ring" (Figure 9, left). When seen edge on, a system with low inclination variation will appear as a "needle" (Figure 5, top right), but a system with high inclination variation will appear as a "ship and wake" (Figure 5, bottom right).

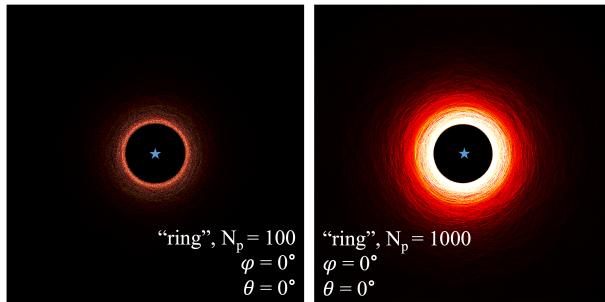
The morphologies are recognizable in both scattered light and thermal radiation, at the exception of the "ship and wake" and "moth". Indeed, the scattering effect seems to have a big impact on the appearance of morphologies when seen almost edge on.



**Fig. 7.** Dust effective temperature when irradiated by a central star. The temperature follows an inverse square law according to the grain position.



**Fig. 8.** A "bar" morphology, coming from a disk with a "ring" morphology when seen face on. Here, two rendering techniques are compared: on the left, a linear combination of three Henyey-Greenstein phase functions, on the right, a single Henyey-Greenstein phase function with parameter  $g = 0.5$ .



**Fig. 9.** A debris disk view face on with a "ring" morphology. On the left, rendered with  $10^7$  particles. On the right, rendered with  $10^8$  particles (more specifically,  $N_p = 1000$ ). A high particle count allows for a smoother render and a larger rendering intensity. However, the disk morphology is still recognizable on a lower particle count.

As opposed to the work of Lee & Chiang (2016) (1), this render uses a combination of three Henyey-Greenstein phase functions. A side by side comparison of the improved Henyey-Greenstein phase function is presented in Figure 8. The linear combination of three Henyey-Greenstein functions appears significantly different than the single Henyey-Greenstein function used in Lee & Chiang (2016) (1) when seen almost edge on. The linear combination creates a scattering effect, making the original "bar" morphology difficult to recognize.

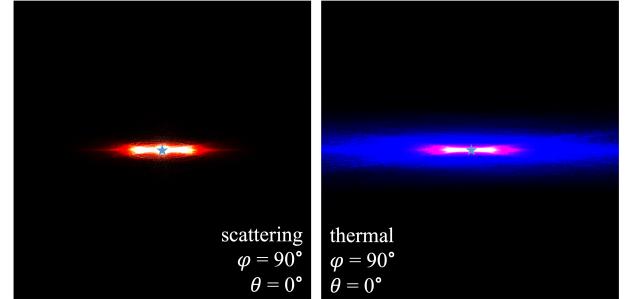
The effect of an increase in the number of drawn particles on the render is presented in Figure 9 by comparing a render with  $10^7$  and  $10^8$  particles. Even though a high particle count allows for a better contrast and a smoother render, the disk morphology is still recognizable with  $10^7$  particles.

Finally, a comparison of a scattered light render and a thermal radiation render is shown in Figure 10. It shows that thermal radiation allows for a larger probing of the dust mass, better revealing the extent of the debris disk.

## 6. PERFORMANCES

The performance statistics for  $10^7$  drawn particles are averaged out over multiple runs, and compiled in Table 1. Performance analysis is performed with the Tracy profiler.

The total initial loading time is defined as the time between the launch of the program and the first rendered frame. During that time are performed various operations, particularly the loading of pre-computed dust orbits from a file,



**Fig. 10.** A debris disk seen in thermal radiation (right) and scattered light (left). Thermal radiation allows for a larger probing of the dust mass.

FPS mean per run	$61 \pm 4$ FPS
FPS median per run	$73.2 \pm 0.5$ FPS
Loading Orbits From File	$0.06 \pm 0.01$ s
Converting Orbits To Particles	$0.54 \pm 0.05$ ms
Total Initial Loading Time	$1.56 \pm 0.05$ s

**Table 1. Performance statistics for  $10^7$  particles drawn on 8 threads.**

and the conversion from dust orbits to particles in Cartesian coordinates.

A plot of the mean FPS (Frame Per Second) per run versus particle count is shown in Figure 11. For renders containing more than  $2 \cdot 10^7$  particles, the mean FPS per run is less than the 30 FPS minimum for real time software. The performances increase exponentially as the number of particles decreases.

The bottleneck of the initial loading time is the conversion from dust orbits to particles in Cartesian coordinates, function that is repeated for 100,000 orbits. Even though it is not possible to improve its complexity, the use of multi-threading greatly accelerates the process, as shown in Figure 12. A horizontal asymptote is observed at  $1.56 \pm 0.05$  s beyond 8 threads, suggesting the inefficiency of using more than 8 threads.

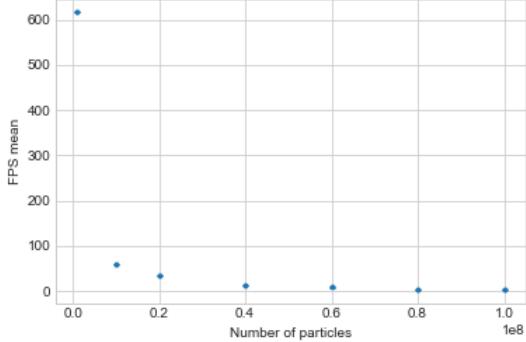
## 7. DISCUSSION

The model is successful at producing similar morphologies to the ones of Lee & Chiang (2016) (1) for a same disk in scattered light.

Using a linear combination of three Henyey-Greenstein phase functions produces a more accurate render in scattered light. When a disk is viewed face on, using a linear combination is similar to a single Henyey-Greenstein function. When viewed almost edge on, the combination creates a protrusion, making harder the classification of morphologies.

Thermal radiation is successful at probing a larger dust population, by directly probing the dust mass distribution. The scattering effect is a major actor in creating disk morphologies and thermal radiation alone might fail at identifying morphologies like the "ship and wake" or the "moth". This is expected, as it has been observed that scattering features disappear in thermal emission (Meredith A. MacGregor et al (2018) (17)).

The render might be improved by distributing the grain size over the debris disk instead of taking an average grain size. This might help visualizing debris disks containing multiple arms, like the disk around the M1V star GSC 07396-00759, as demonstrated by Cronin-Coltsmann & Kennedy (2022) (18).



**Fig. 11.** FPS mean of a run, depending on the number of particles in the scene. Error bars are too small to be seen.

The  $61 \pm 4$  mean FPS per run is well above the 30 FPS threshold of real-time software for  $10^7$  particles. However, the mean has a large standard deviation, hinting at bottlenecks in the simulation. This large deviation could be due to user input, causing fluctuations in the simulation. User input leads to an increase in the number of operations performed per frame, lowering the mean FPS. To balance the effect of user input, the FPS median can be analysed. The FPS median is  $73.2 \pm 0.5$  FPS per run. Its standard deviation is much lower than the mean FPS, and the FPS median is much higher. This means that the user can expect the simulation to run at  $73.2 \pm 0.5$  FPS when not moving the camera.

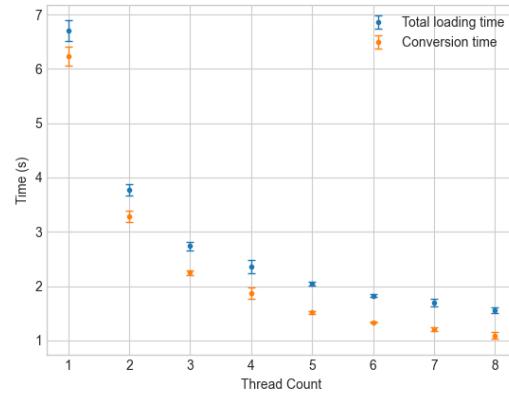
Multi-threading is successful at accelerating the initial loading time of the software from  $6.7 \pm 0.2$  s to  $1.56 \pm 0.05$  s. However, the multi-threading optimization hits an asymptotic behaviour with more than 8 threads, limiting the optimization.

Drawing more than  $2 \cdot 10^7$  particles lowers the mean FPS to less than 30 FPS. Even though the simulation can be operable up to  $10^8$  particles with 3 FPS in average, it is impractical and inefficient. To increase the FPS mean per run, approximations of the computations would have to be made. For example, the Schlick approximation (19) can be used for the Henyey-Greenstein phase function. However, it has been shown that  $10^7$  drawn particles were enough to accurately classify debris disk morphologies.

## 8. CONCLUSION

Overall, the tool is useful at categorizing debris disk morphologies depending on the viewing angle and the orbital properties of the system, with results agreeing with Lee & Chiang (2016) (1) in scattered light. It has been shown that a simple solar system containing a single planet on a possibly eccentric orbit could exhibit multiple morphologies. For a debris disk viewed face on, a "ring" or "eccentric ring" could be observed for respectively low and high planet eccentricity. When viewed edge on, a "needle" will be observed for a disk with low inclination variation, and a "ship and wake" for high inclination variation. A "bar" or a "moth" can be observed for disks viewed almost edge on, depending on the system's orbital parameters.

In scattered light, the rendering was made more accurate by using a linear combination of three Henyey-Greenstein phase functions. The simulation was improved by adding a thermal radiation rendering mode, allowing for a larger probing of the



**Fig. 12.** Speed up curve of the conversion time from a dust orbital to dust particles in Cartesian coordinates, depending on the number of threads used.

dust mass and the analysis of the disk morphology without scattering artifacts.

By leveraging modern rendering techniques, the model was optimized to  $61 \pm 4$  FPS in average, allowing for a practical simulation in a real-time environment, with an initial loading time of  $1.56 \pm 0.05$  s.

## REFERENCES

1. EJ Lee, E Chiang, A primer on unifying debris disk morphologies. *The Astrophys. J.* **827**, 11 pp (August 2016).
2. A Michel, N van der Marel, BC Matthews, Bridging the gap between protoplanetary and debris disks: Separate evolution of millimeter and micrometer-sized dust. *The Astrophys. J.* **921** (2021).
3. G Cataldi, *Debris disks and the search for life in the universe*. (Department of Astronomy, Stockholm University, ISBN 978-91-7649-366-3), 1 edition, (2016).
4. CD Murray, SF Dermott, *Solar System Dynamics*. (Cambridge University Press, ISBN 0-521-57597-4), 1 edition, (1999).
5. D Maoz, *Astrophysics in a Nutshell*. (Princeton University Press, ISBN 978-0-691-16479-3), 2 edition, (2016).
6. AM Hughes, G Duchêne, BC Matthews, Debris disks: Structure, composition, and variability. *Annu. Rev. Astron. Astrophys.* **56**, 541–591 (2018).
7. K Silsbee, BT Draine, Radiation pressure on fluffy submicron-sized grains. *The Astrophys. J.* **818** (2016).
8. JS Mathis, W Rumpl, KH Nordström. (year?).
9. LE Strubbe, EI Chiang, Dust dynamics, surface brightness profiles, and thermal spectra of debris disks: The case of au mic. *The Astrophys. J.* **648** (2006).
10. MR Schwarz, Keplerian orbit elements → cartesian state vectors (2017).
11. DJ Eck, Introduction to computer graphics (2021).
12. OpenGLWiki, Rendering pipeline overview (2021).
13. LG Henyey, JL Greenstein, Diffuse radiation in the galaxy. *The Astrophys. J.* **93**, 70–83 (1941).
14. MM Hedman, CC Stark, Saturn's g and d rings provide nearly complete measured scattering phase functions of nearby debris disks. *The Astrophys. J.* **811** (September 2015).
15. BT Draine, On the submillimeter opacity of protoplanetary disks. *The Astrophys. J.* **636** (2006).
16. T Birnstiel, et al, The disk substructures at high angular resolution project (dsharp). v. interpreting alma maps of protoplanetary disks in terms of a dust model. *The Astrophys. J.* **869** (December 2018).
17. MA MacGregor, et al, Alma detection of extended millimeter halos in the hd 32297 and hd 61005 debris disks. *The Astrophys. J.* **869** (2018).
18. PF Cronin-Coltsmann, GM Kennedy, Alma's view of the m-dwarf gsc 07396-00759's edge-on debris disc: Au mic's coeval twin. *Royal Astron. Soc.* **536** (February 2022).
19. C Schlick, An inexpensive brdf model for physically-based rendering. *Comput. Graph. Forum* **3** (1994).

## A. COLORMAP

A 1-dimensional texture is used to map the intensity of a dust particle to a color. For scattered light, the colormap is taken from Matplotlib's hot texture, Figure 13. For thermal radiation, an ironbow colormap is chosen, Figure 14.



**Fig. 13.** Matlab "hot" colormap as 1-dimensional texture for scattered light. From [kbinani](#).



**Fig. 14.** "ironbow" colormap as 1-dimensional texture for thermal radiation. From [kbinani](#).

## B. HARDWARE SPECIFICATIONS

All measurements were made on the same hardware, with specifications detailed in Table 2.

Processor	Intel(R) Xeon(R) CPU E5-2667 v3 @ 3.20GHz
RAM	12.0GB
System Type	64-bit Operating System
GPU	Nvidia Quadro P5000
CPU Topology	8 Cores (4 Cores with 2 hyperthreads)

**Table 2. Hardware specifications.**

## C. SOURCE CODE AND RAW IMAGES

The code source of the simulation and all images in high resolution can be found at <https://github.com/RobinLmn/DebrisDiskSimulation>.

## D. ORBIT VISUALIZER

To better understand the Keplerian elements used to represent an orbit, an interactive simulation was created. It can be accessed through a browser at <https://robinlmn.itch.io/orbit-visualizer>.