Method Article

Methods for ray-tracing thermal modelling of Saturn’s main rings✩,✩✩

Ángel M. García-Reyesa, Alberto Flandeses

aPosgrado en Ciencias de la Tierra, Universidad Nacional Autónoma de México, Coyoacán 04510, Mexico City, Mexico
bCiencias Espaciales, Instituto de Geofísica, Universidad Nacional Autónoma de México, Coyoacán 04510, Mexico City, Mexico

A B S T R A C T

The thermal emission and temperatures of the main rings of Saturn depend on the energy the ring particles absorb, reflect and scatter and/or on their Bond albedo, emissivity, thermal inertia, rotation rate and porosity. However, the energy that each particle absorbs also depends on the amount of energy (e.g., solar energy) that reaches its surface and this latter on the local optical depth, that controls the mutual eclipsing between neighbouring particles and, in general, all shadowing effects on the rings. On the other hand, thermal models of the rings of Saturn based on the energy balance equation strongly depend on a function that described how the non-shadowed area of ring particles changes with solar elevation. Experimental and analytical shadowing functions have been proposed by [6] and [1], respectively. In this work, we propose shadowing functions based on the creation of 3D arrays of spherical particles that simulate specific regions of the main rings of Saturn. The methods implemented to obtain these shadowing functions follow the next general steps:

• Arrays are created as a collection of spherical particles with a size distribution that follows a power law constrained to the optical depth of the region of study based on the UVIS instrument data.
• The particles of the arrays are then reordered to add some relevant dynamical features observed in actual rings (e.g., wake structures in the case of optically-thick rings).
• Under different illumination geometries, images of these arrays are rendered using ray tracing. From these images, an analysis of their pixel brightness values allows us to determine the non-shadowed fractional area of the particles in order to compose the corresponding shadowing functions.

© 2021 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)

✩ Direct Submission or Co-Submission: Co-submissions are papers that have been submitted alongside an original research paper accepted for publication by another Elsevier journal

Co-Submission: Flandes, Alberto; García-Reyes, Ángel M.; Spilker, Linda. Ray-tracing thermal modelling of Saturn’s main rings, Icarus, Volume 354, article id. 114063. DOI: 10.1016/j.icarus.2020.114063, Bibcode:2021Icar..35414063F 2021,

DOI of original article: 10.1016/j.icarus.2020.114063

✩✩ Corresponding author.
E-mail address: alberto_flandes@igofisica.unam.mx (A. Flandes).

https://doi.org/10.1016/j.mex.2021.101457
2215-0161/© 2021 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)
1 Introduction

From the thermal behaviour of the main rings of Saturn one can derive properties related to their structure, dynamics and composition. The current work focuses on the creation of arrays that represent synthetic rings with which we analyse the thermal behaviour with solar elevation of selected regions of the main rings. Our analysis is based on data from the Cassini mission instruments Composite Infrared Spectrometer (CIRS) and Ultraviolet Imaging Spectrograph (UVIS), which we use as reference to select specific ring regions of study that we reproduce with arrays of lambertian spherical particles. In the end, the analysis is done through an energy balance model, which depends on the so-called shadowing function obtained from the created synthetic rings using ray tracing. The methods of the current work use Interactive Data Language (IDL 8.8) and Autodesk Maya software (2018). The relevant parameters involved in the construction of the arrays are the optical depth, \( \tau \), and the self-gravity wake ratios for the optically-thick ring regions wake vertical thickness to wake width, \( H/W \), and wake separation to wake width, \( S/W \) (from estimations by [7]). The arrays we create do not derive from any N-body dynamical code (such as [8]), but we create them based on the general structure features derived from observations and must be taken as snapshots of the average structure of selected regions of the main rings, though they implicitly posses dynamical features. In the end, these arrays are useful in the description and analysis of some actual thermal processes in the rings as one can see in [4] and [5].

The current methods follow four stages that are indicated and summarised in the graphical abstract (flow diagram) shown in Fig. 1. Stage 1 comprises the selection of the ring regions that will be replicated based on their optical depth (\( \tau \)) profiles. In this stage, we define the number

![Fig. 1. Comparison between: (a) a preliminary array (B1 region) from Output 2 and (b) the corresponding final array from Output 4. (a) shows a somewhat flat distribution of particles across the layer, while (b) shows a distribution where most of the mass concentrates closer to the central plane of the array and larger particles concentrate closer to the axis of the wake as well, as can be seen on the bottom edge-on view. Note also that the interwakes' distribution is quite similar in both cases and both are similar to the C1 and CD1 regions as well.](image-url)
Table 1
Variables and parameters used in this work. See the text for explanations and details.

| Variable | Description |
|----------|-------------|
| $f$      | Particles rotation parameter. |
| $\tau$  | Optical depth. |
| $\tau_{\text{dyn}}$ | Dynamical optical depth. |
| $\tau_{w}$ | Wake optical depth. |
| $\tau_{s}$ | Interwake optical depth. |
| $B'$ | Solar elevation with respect to the ring plane. |
| $B$ | Spacecraft elevation. |
| $\alpha$ | Phase angle. |
| $R$ | Radial distance (spacecraft-FOV) |
| $C(B', \tau)$ | Shadowing function. |
| $\psi$ | Local hour angle. |
| $N$ | Number of particles of arrays. |
| $A$ | Cross sectional area of box where arrays are created. |
| $A_e$ | Effective area occupied by the particles of the arrays. |
| $L_T$ | Depth of arrays. |
| $W$ | Width of wake/bars. |
| $S$ | Wake separation. |
| $H_0$ | Preliminary vertical height of arrays. |
| $H$ | Final vertical height of arrays. |
| $\phi_w$ | Wake inclination angle. |
| $a$ | Particle radius. |
| $a_i$ | Radius of the $i$th particle of the array. |
| $a_{\text{max}}$ | Maximum radius of particle of the array. |
| $a_{\text{min}}$ | Minimum radius of particle of the array. |
| $\sigma$ | Standard deviation. |
| $T_p$ | Temperature of a particle |

and size distribution or particles according to power laws with specific indices, $q$. We also define the preliminary and general geometrical dimensions of the arrays of particles, width ($W$), depth and height ($H$). These dimensions are particularly relevant for optically thick rings or where wakes are present.

In Stage 2, the positions of the particles of the preliminary arrays generated in Stage 1 are re-ordered to match the structure of the different ring regions and account for their general dynamical features like self gravity effects in optically thick rings.

In Stage 3, the output of Stage 2 is used to create 3D arrays and images through ray tracing. Finally, in Stage 4, the set of images generated in Stage 3 are analysed to derive the shadowing function, $C(B', \tau)$, of each array/region. Note that the temperature ($T_p$) of a given particle in the rings, is calculated through the energy balance equation:

$$T_p = \sqrt{[4]^{1/4}(c_1 + c_2 A_V) C(B', \tau) + c_3 (1 - A_V) + c_4}, \quad (1)$$

where $c_1$, $c_2$, $c_3$ and $c_4$ are constants (see [4] for details). The temperature of a whole region (e.g., the field of view of CIRS) is assumed to be equal to the temperature of a particle of the same region, granted that all the particles of the region have the same thermal behavior. The rotation rate $f$ and the bond albedo, $A_V$, of the particles are used as fitting parameters of the model and their values help to evaluate the validity of this approach. We highlight that details on the temperature derivation method (i.e., Stage 4), are not given in this work given that this part is explained in detail in our main work [4].

In Table 1 we present all the parameters that we use in this works and that are explained along the text.

The arrays of particles that we deal with in this work have few tens of thousands of particles that we managed with a computer with a 3.2 GHz Intel Core i7 (8 cores, 16 threads) processor, 64 GB RAM memory at 2400 MHz and a GeForce GTX 740 graphic card (2Gb).
The operative system used was OS. In general, the computation time is directly proportional to the optical depth and the number of particles (N) of the arrays or regions of study. With the computer we used, the creation of arrays shown by Flandes et al., 2021 required from few minutes (e.g., C1) to nearly one hour for the most demanding arrays (τ > 0.6). The rendering process (generation of the images of the arrays, lightning included) required from less than a minute (for the less demanding) to around two hours per image (for the most demanding arrays).

Along the paper and as examples, we include some codes in IDL and Maya Embedded Language (MEL) to help understand and reproduce our procedures. The codes have to be compiled in the order they are presented after the parameters of each specific region had been declared in advance. In any case, we surmise that the user has some programming abilities and a basic knowledge of the software Maya.

2 Stage 1: Selection of ring regions of study

The optical depth, τ, defines each of the main rings, thus one can draw the general structure of the ring regions we deal with from the optical depth measurements retrieved from the UVIS instrument data [2]. We choose ten regions (see Fig. 1 of [4]) that will be reproduced and analysed and whose optical depth is assumed constant. For this latter reason, we take the average value of the whole region in each case. The regions are as follows: One region of the C ring outside of the plateaus that we call C1 and one region of the Cassini Division ring that we designate CD1. From the B ring, we only consider the lowest optical depth or less dense regions that we designate B1, B2 and B5. The A ring is separated into five regions, that is, A1 to A5. The regions’ boundaries and τ average values are our Input 0 and can be seen in Table 2.

3 Stage 1: Creation of synthetic ring regions

We create preliminary arrays in order to determine the number, N and the size distribution, a_i, of the particles. For this purpose, we require four input parameters. First, the average optical depth, τ, from the regions of study we chose. Second and third, a minimum (a_{min}) and maximum (a_{max}) particle size. Fourth, we assume that the size distributions follow a power-law \( dn(a) = n(a)da \sim a^{-q}da \) [3], thus we have to define an index (q) for the power law for each array as well. In general and to a first approximation, the input parameters \( a_{min} \) and \( a_{max} \) lie in the interval [0.1, 10] m, while the index \( q \) has values in the interval [2.75, 3.11].

In order to implement the power law, it is convenient to use the function that comes as part of the different software, like IDL, but as reference, we include an example function that can also be adapted:

```
;Code 1: RANDOM POWER LAW FUNCTION.
Function power_law, N, q, a_max, a_min
  return, (randomu(s,N)*(a_max^q(1-q)-a_min^q(1-q))+a_min^q(1-q))^q(1/(1-q))
END
```

Table 2
Radial boundaries of the ring regions studied in this work.

| Ring regions | \( R_{\text{min}} \times 10^3 \text{km} \) | \( R_{\text{max}} \times 10^3 \text{km} \) | \( \tau \) (average) |
|--------------|---------------------------------|---------------------------------|---------------------|
| C1           | 79.0                            | 84.0                            | 0.09                |
| B1           | 93.0                            | 96.0                            | 1.15                |
| B2           | 96.5                            | 100.0                           | 1.94                |
| B5           | 110.5                           | 116.5                           | 3.42                |
| CD1          | 118.0                           | 119.0                           | 0.13                |
| A1           | 122.5                           | 125.0                           | 1.09                |
| A2           | 125.5                           | 127.5                           | 0.69                |
| A3           | 128.0                           | 130.5                           | 0.60                |
| A4           | 131.0                           | 133.0                           | 0.60                |
| A5           | 134.0                           | 136.0                           | 0.63                |
One of our assumptions is that the optical depth from the measurements corresponds to the dynamical optical depth, granted that, in any case, the particle centers are randomly distributed. Each preliminary array can be seen as a box with cross sectional area, \( A = W L_y \) (with \( L_y \) the depth of the box), filled with spherical particles of different sizes at random positions regardless of their sizes (i.e., a uniform random distribution) and whose theoretical optical depth is defined as:

\[
\tau_{\text{dyn}} = \sum_i \pi a_i^2/A = A_e/A
\]

where \( A_e \) is the total effective area occupied by the cross sections of the particles. In the construction process, there is a difference between optically-thin and -thick arrays. For example, the creation of optically-thick arrays require wake parameters that describe their bar-like structure. The different treatment of both is as follows:

### 3.1 Optically thin rings (C1 and CD1) and Output 1

For the optically-thin arrays that represent either the C1 and CD1 regions, we create a single array, given that their structures are somewhat uniform. The output of this latter process, that we designate Output 1, returns the number of particles (N) in the array and the size distribution, where the size of each particle is specified as \( a_i \). In Table 3, we show the values of the input parameters used for these two regions (This table, as well as the following Tables 4 and 5, were directly copied from [4], since their parameters are necessary for the understanding of the methods we explain in this work).

### 3.2 Optically-thick rings (B and A ring regions) and Output 2

For every optically-thick array that represents either A or B ring regions, we need to create two boxes. Each pairs represents the a wake (or bar-like structure) and a region between two consecutive
Table 5
A ring regions simulation parameters.

| Parameter/Ring | A1   | A2   | A3   | A4   | A5   |
|----------------|------|------|------|------|------|
| \( \tau_w \)   | 1.09 | 0.69 | 0.60 | 0.60 | 0.63 |
| \( \tau_s \)   | 0.14 | 0.12 | 0.12 | 0.15 | 0.21 |
| \( q' \)       | -2.75| -2.75| -2.75| -2.75| -2.75|
| \( a_{w_{\text{max}}} \) [m] | 4.94 | 4.87 | 4.87 | 4.71 | 8.93 |
| \( a_{s_{\text{max}}} \) [m] | 0.59 | 0.62 | 0.62 | 0.60 | 1.13 |
| \( a_{\text{min}} \) [m]  | 0.13 | 0.13 | 0.13 | 0.13 | 0.25 |
| \( N \)         | 24010| 9593 | 6090 | 6220 | 2740 |
| \( N_i \)       | 21200| 19300| 18100| 29000| 20600|
| \( H \) [m]     | 4.77 | 5.08 | 4.29 | 4.33 | 7.04 |
| \( (H/W)^i \)   | 0.11 | 0.19 | 0.22 | 0.23 | 0.26 |
| \( (S/W)^i \)   | 0.76 | 1.39 | 1.65 | 2.05 | 2.41 |
| \( \phi_{\text{w}} \) | 66.90| 68.70| 60.50| 65.70| 62.70 |
| \( W \) [m]     | 40.14| 25.06| 21.03| 19.44| 28.10 |
| \( S \) [m]     | 30.62| 36.00| 34.69| 39.92| 67.85 |

wakes or bars, that we call *interwake*. Each with a different optical depth. Ahead, the optical depths of wakes and interwakes are identified as \( \tau_w \) and \( \tau_s \) respectively, where \( \tau_w > \tau_s \). As reference, in this work, \( \tau_w \geq 0.60 \) and \( \tau_s \geq 0.35 \).

The relative sizes of wakes and interwakes are based on the estimated \( H/W \) and \( S/W \) ratios from which the specific values of \( H \) and \( S \) are determined given that the area, \( A \), has been defined. See Tables 4 & 5 for the values of the input parameters we use (note that these tables also show output parameters that are explained throughout the text). In this case, the output, that we designate Output 2, returns the number of particles, \( N \), of each array, the size of each particle, \( a_i \), the width of the wakes/bars, \( W \), and the separation between them, \( S \) (for each array). It also returns the vertical thickness of the arrays, but consider that this value is preliminary and designated \( H_0 \). Since the particles of the arrays are re-arranged (as it is explained in the next section), this value changes to some extent.

The following is an example of the routine we used for the creation of preliminary arrays. Note that this code uses Code1.

```plaintext
; Code 2: Estimation of N and H_0.
A_e = W*y * (1/Tau_dyn)
N = long(100.0)
Radial = Power_law(N, -q , a_max, a_min)
repeat begin
N=N+10.0 & Radial = Power_law(N, -q , a_max, a_min)
A_sim = total(Radial^2)*|p|)
endrep until A_sim gt A_e
Radial = Radial[reverse(sort(Radial))]
H_0 = total([2*(Radial]^3)/A_e)
```

Fig. 1a shows a visualization in Autodesk Maya of an example Output 2 that corresponds to the preliminary array for the B1 region. Note that the wake shows a somewhat flat distribution and the side interwakes are similar to optically-thin arrays.

4 Stage 2: Re-arrangement of particles

Preliminary arrays follow no weight or mass rule for the positioning of particles, which is unrealistic. In actual Main rings, larger particles tend to concentrate at or closer to the ring’s mid plane and, in particular, in the case of optically-thick rings, closer to or at the axis of the wake according to the self-gravity behaviour. For this latter reason, we re-arrange the position of the particles of the arrays according to the following:
4.1 Optically-thin rings (C1 and CD1) and Output 3

First, Using Output 1, the particles are set again in the box or layer of particles, now according to their radii size in descending order, however, –second– the z-coordinates of the particles are constrained to follow a normal distribution (in this case we use the randomN IDL routine) centered at the xy-plane or midplane, whilst the x and y positions simply follow a uniform random distribution (where we use the randomU IDL routine).

The result of this procedure is Output 3 and contains the final positions of all particles as \((a_i, x_i, y_i, z_i)\).

4.2 Optically-thick rings (B and A ring regions) and Output 4

For these arrays we use Output 2 as input, which contains the relative dimensions of wakes and interwakes. Interwake arrays are simply generated with the same rules applied to the C1 and CD1. Wake arrays follow the former two rules as well, however the x-coordinates of their particles follow a normal distribution centered at the xz-plane, assuming that the xz plane divides each box vertically (we call this third rule). The second and third rules guarantee that most particles, and particularly the largest, which are set first, concentrate along and about the axis of the wake. With this procedure, we obtain Output 4, which also returns the individual positions of particles of each array as the vector \((a_i, x_i, y_i, z_i)\), that implicitly contains the geometrical properties, W, S and H and their ratios S/W and H/W. We designate these final arrays. Fig. 1b shows the final array for the B1 region.

An example of the general routine we use for the re-arrangement of particles see Code 3 (Note that this code uses Code 2).

```
;--------------------------------------------------
;Code 3: FINAL ARRAYS: RE-ARRANGEMENT OF PARTICLES.

for i = 0, 0, N-1 do begin
increment=0.0
REPEAT BEGIN
if wake_conlord 0 then begin
x[i]=RandomU(s,1)*((W_x-2*a[i])+a[i]) ;x position with uniform random distribution.
y[i]=RandomU(s,1)*((L_y)-(2.0)) ;y position with uniform random distribution.
z[i]=RandomN(s,1)*((H_z)/(8.0)) ;z position with Gaussian random distribution.
endif
if wake_conlord 1 then begin
   weight = (a[i]-min(a))/(max(a)-min(a)) ;Mass weight condition variable.
y[i]=RandomU(s,1)*((L_y)-(2.0)) ;y position with uniform random distribution.
repeat x[i] = (RandomN(s,1))((W_x-2*a[i]))/(2.0.)+W_x/2. $ ;X position with Gaussian random distribution.
until (x[i] le W_y-x[a[i]] and x[i] ge a[i])
z[i]=RandomN(s,1)*((H_z + (increment/10e3))/(8.0)) ;z position with Gaussian random distribution.
endif
if glt 0 then begin
for j = 0, i-1. do begin
   d[j] = SQRT((x[[j]]-x[i])^2+(y[j]-y[i])^2 + (z[i]-z[j])^2 ) ;Distance-Particle Matrix.
endfor
   dist_part = where d[j] ge Radii[i] \(5^*H\) then $ ;Mass-weight H increment until \(5^*H\) limit.
   increment=increment + 2.0
endif
ENDREP UNTIL n_elements(dist_part) ge i
ENDFOR
```

One detail that can be seen in Fig. 1, is that we consider different maximum radii for the particles of interwakes and wakes. The maximum radius of the interwakes are calculated as \(a_{max} = \bar{a}_{w} + \sigma (a_{w})\).
where $\bar{a}_w$ is the average and $\sigma(\bar{a}_w)$ is the standard deviation of the radii values of the wake. Also note that, whilst the dimensions $W$ and $S$ are unchanged, the vertical height of the final arrays change from $H_0$ to $H$. $H$ is defined as the distance between the centres of the two farthest particles below and above the midplane ($H = z_{\text{max}} - z_{\text{min}}$). For the A ring arrays, $|H - H_0| < 0.1$, while for the B ring regions, the difference is between 10% (for B1. See Fig. 1) and near 30% (for B2 and B5). This guarantees that the $S/W$ and $H/W$ ratios taken are roughly the same.

5 Stage 3: MEL scripts and MAYA scenes/Output 5

Outputs 3 and 4 are used to write a MEL (Maya Embedded Language) script. This script is called Output 5 and it is used to create the scenes in the MAYA software environment from which images of the arrays are rendered in order to derive the shadowing functions necessary to estimate the temperature of the rings with the analytical model implemented.

Besides the commands to create the arrays of each region, the scripts contain additional lines to create other elements of the scene. In general, Output 5 includes subroutines to:
- Create the arrays based on Output 3 and 4.
- Create a point-light source that simulates the Sun and its motion in the range of the elevation angles we consider.
- Create a virtual observer (camera) that simulates the spacecraft.

For simplicity, Output 5 is generated in IDL, but has got to be directly compiled in the script editor of the MAYA interface. In the following subsections we provide three codes that may help to obtain Output 5. Nevertheless, we also provide instructions for users who prefer working directly with the MAYA interface.

5.1 Arrays based on Output 3 & 4

We highlight that Outputs 3 and 4 contain the basic elements of the arrays, but in order to render images of these arrays, we need to create a view of the arrays in Maya. This can be done with the following code.

```plaintext
;---------------------------------------
;Code 4: CREATION OF ARRAYS IN THE MAYA SCENE.
For i = 0.0, N-1 do begin
  printf, 1, "polySphere -r", Radii[i], "; rename pSphereShape1 e", i, "; move",x[i], y[i], z[i], ";"," endfor
;---------------------------------------
```

Code 4 creates N spherical objects (made out of polygons) with radius $a_i$. It also assigns a name and a number to each sphere and sets it at a position given by $(x_i, y_i, z_i)$. By default, Maya creates spheres with lambertian surfaces, which in our case is convenient, since these surfaces reflect light in a diffuse fashion like the surfaces of actual ring particles due to their roughness. Actually, the images shown in Fig. 2 correspond to arrays visualized in Maya using Code 4.

5.2 Virtual Sun

It is necessary to create a point-light source to simulate the Sun. The light source must also be able to move from $22^\circ$ to $0^\circ$ with respect to the ring plane. If this is done with the interface, one may follow the next simple steps:

(a) From the Create dropdown list in the Maya interface, select Nurbs primitive->circle. Set the circle plane perpendicular to the central plane of the array and make its radius equal to the distance at which the point-light source should move. This distance/radius should be many times the size of the array.
(b) Also from the Create dropdown menu select Point-light source and set it at elevation $22^\circ$ (with respect to the array/ring plane) on the circular trajectory already created.
(c) Select the light-source and then the trajectory and click the P key to parent the source to the trajectory.

(d) Select the light source and the trajectory simultaneously and then select Frame 0 (zero) in the timeline below, then press the S key. This defines the initial keyframe of the motion.

(e) Select the trajectory and rotate it 22° degrees (about its own central axis) from the channels menu. Since the light source is coupled to the trajectory, it will be dragged and moved to elevation 22° as well.

(f) Select Frame 22 in the timeline and press the S key again to set the end of the motion of the light source. Now, when one scrolls through the timeline, the light source moves along the circular trajectory in the required solar elevation interval.

The same can be done through the script editor with the help of Code 5. As Code 4, Code 5 must be compiled first in the IDL console or terminal to produce a MEL script that executes the same steps described above.

;-------------------------------------------
;Code 5: CREATION OF THE VIRTUAL SUN.
center = [c_x, 0, c_z]
R = 3000.0

printf, 1, 'pointLight -pos '+string(center[0])+'+0 '+string(R+center[2])+';'
printf, 1, 'circle -c '+string(center[0])+'+0 '+string(center[2])+'+nrx 0 -nry 1 -r '+string(R)+'+n circle2;'
printf, 1, 'select pointLight1 circle2; parent;'
printf, 1, 'rotate -y '+string(-Phi_w)+'+ -cp;'

for w=-22 , 22 do begin
  printf, 1, 'select circle2;'
  printf, 1, 'currentTime '+string(w+23)+';'
  printf, 1, 'rotate -x '+string(w)+'+ -cp;'
  printf, 1, 'setKeyframe -breakdown 0 -hierarchy none -controlPoints 0 -shape 0 "{circle2}";
endfor

;-------------------------------------------
5.3 Virtual observer/camera

A camera is created to simulate the spacecraft or observer. The camera is positioned at an elevation angle (i.e., spacecraft elevation, B) of 22° with respect to the ring/array plane. Note that the CIRS data we use correspond to observations of the south side of the rings (B and B’ 0°), but in our arrays, positive or negative elevations make no difference.

In the Maya interface, the camera can be easily created through Create->cameras->camera. We can also use the Channels menu (on the right side of the interface) to locate, aim and adjust its field of view (e.g., focal length). We can also reproduce the latter steps with Code 6.

;-------------------------------------------------------------------------
; Code 6: CREATION OF THE CAMERA/OBSERVER IN THE SCENE.
center = [c_x, 0, c_z] ;Centre of FOV
R = 3000.0 ;Suggested camera/observer radial distance.
prinf, 1, ’circle -c '+ string(center[0]) + ' 0 '+ string(center[2]) + ' -nrz 0 -nrz 1 -r '+ string(R/20.) + ' -n circle1;
prinf, 1, ’camera -p '+ string(center[0]) + ' 0 '+ string(R/20.+center[2]) + ' -n camera1 -fl 500;
prinf, 1, ’select camera1 circle1; parent;
prinf, 1, ’rotate -y '+ string(-Phi_w)+' -cp;
prinf, 1, ’rotate -x -22 -cp;
prinf, 1, ’viewLookAt -position '+ string(center)+' 0 '+string(center[2])+' camera1; '

;-------------------------------------------------------------------------

It is worth noting that it is necessary to consider the orientation of the actual wakes in the rings of Saturn or wake angle, $\phi_w$ (see Tables 4 & 5). In order to include this angle, in Code 6, we simply rotate the camera (indicated as Phi_w) with respect to the plane of the point-light source trajectory.

Code 6 also has to be compiled inside IDL and its output can be directly run inside Maya.

6 Stage 3: Image rendering/Output 6

To prepare the render process (or image creation) in Maya, first, one has to adjust the render settings on the window with the same name, which can be displayed by clicking the blue dot clapperboard icon on the upper menu. In the Render Settings window one can define the size (e.g., a minimum of 640 × 480 pixels) and resolution (we chose a minimum of 72) of the images among other properties. In this case, we choose Raytracing as the type of rendering as follows: Maya Software tab > Raytracing Quality section> enable Raytracing. Also on the light source attributes enable RayTrace Shadows Attributes > Use Ray Trace Shadows. Note that we chose Raytracing, since with this routine the software calculates the path of individual light rays between the camera and the light source and allows to accurately reproduce the shadow profiles. On this window menu, it is also important to select the camera we created to be sure that the rendered images show its field of view.

According to Code 6, every frame in the timeline corresponds to a particular solar elevation angle (of course, one may decide to create many more images as per degree), thus we should set frame 0 as the starting render frame and frame 22 as the end render frame. One important detail though, is that the render process may depend on the Maya version used. If Maya 2016 or earlier, one has to use Mental Ray. If Maya 2017 or later Maya comes with the Arnold built-in render engine, which has got its own RenderView.

Once the settings have been defined, the image rendering can be started by pressing the keyboard space bar to display the extended Maya menu and choosing Render->Batch Render.

Fig. 2 shows an example of the result of the ray tracing rendering process, where the lightning and shadowing can be seen, for the same B1 region shown in Fig. 1. The render has been done with the light source at an elevation $B' = 12^\circ$ to highlight shadows.

7 Stage 3: Image pixel analysis: shadowing function/Output 7

We highlight that the pixel analysis of the images is aimed to determine the non-shadowed surface area or shadowing function, $C(B', \tau)$) of the particles of each array. The procedure simply consists in
calculating the fraction of white pixels of each image. For simplicity, shadowed or black pixels are defined by pixels with brightness values equals zero or close to zero and above this threshold value (defined as CT) pixels are taken as white or non-shadowed pixels. In the end, the fraction of white pixels corresponds to a point value of the C(\(B', \tau\)) function of a given region. The image analysis is done in IDL based on the Image statistics routine with the help of Code 7.

```
;Code 7: IMAGE PIXEL ANALYSIS:
;fname = 'array.jpg'
imageSize = [width, height]   ;Image dimensions, e.g., 640X480.
read_jpeg, file, array, /GRAYSCALE ;File selection.
WINDOW, uu, XSIZE = imageSize[0], YSIZE = imageSize[1], $
TITLE = 'array' ;Window to display image.
TV, maya, TRUE =1 ;Image display (turn-off to reduce compilation time).
IMAGE STATISTICS, array, COUNT = pixelNumber, $
DATA SUM = pixelTotal, MAXIMUM = pixelMax, $
MEAN = pixelMean, MINIMUM = pixelMin, $
STDDEV = pixelDeviation, $
SUM OF SQUARES = pixelSquareSum, $
VARIANCE = pixelVariance
CT = 20 ;Color threshold separates black pixels from white pixels.
Black = where(array le CT) ;Any pixel with value below CT is considered a black pixel.
White = where(array gt CT) ;Any pixel with value above CT is considered a white pixel.
Nwhite = n_elements(white) ;Calculation of white pixels.
Nblack = n_elements(black) ;Calculation of black pixels.
TOTALP = Nblack + Nwhite ;Total number of pixels.
NONSHADOWED=Nwhite/ TOTALP ;Non-shadowed fraction.
```

7 Stage 4: Temperature derivation

Finally, the theoretical temperature of each region is directly obtained entering the corresponding C(\(B', \tau\)) function values into Eq. (1) and fitting the resulting temperature curves to the CIRS temperature data of the region in the same range of solar elevations with the rotation factor, f, and Bond albedo, \(A_b\), of particles as only fitting parameters as it is explained in Section 5.2 and shown in Fig. 6 both in Flandes et al., 2021. For the curves shown in Fig. 6 of that same work, the fit was done with the IDL program mpfit.pro (https://idlastro.gsfc.nasa.gov/ftp/pro/markwardt/mpfit.pro). We refer the reader to our main work [4] for further details on the temperature derivation stage.

8 Final remarks

This work shows an alternate and simple method to create synthetic rings for the study of some thermal properties of the main rings of Saturn. The effectiveness of this approach can be evaluated from Figs. 6 & 7 shown in Flandes et al., 2021, where the model is compared to observed temperatures and albedos.

It should be noted that no examples of optically-thin regions were shown, because their construction is similar to the interwakes of optically-thick regions (see Fig. 1 and Fig. 2) and thus, the codes we presented can be used for their creation as well.

The software versions used in this work are IDL 8.1 and 8.8 and Maya 2016 and 2020, though earlier versions of both may work as well. We find that IDL and Maya are efficient software for the purpose of this work, but other equivalent software may be used and our model codes may simply be translated.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.
Acknowledgements

This work was supported by DGAPA/PAPIIT IN104721 and IN105818.

References

[1] N. Altobelli, L. Spilker, C. Leyrat, S. Pilorz, S. Eddgington, B. Wallis, A. Flandes, Thermal phase curves observed in Saturn’s main rings by Cassini- Cir: detection of an opposition effect? Geophys. Res. Lett. 36 (10) (2009) L10105 (CiteID L10105), doi:10.1029/2009GL038163.

[2] J.E. Colwell, P.D. Nicholson, M.S. Tiscareno, C.D. Murray, R.G. French, E.A. Marouf, in: The Structure of Saturn’s Rings, Springer, 2009, p. 375. Ch. 13.

[3] Cuzzi, J., Clark, R., Filacchione, G., French, R., Johnson, R., Marouf, E., Spilker, L., 2009. Ring Particle Composition and Size Distribution, Saturn from Cassini-Huygens, Ed. Dougherty), Michele K. and {Esposito}, Larry W. and {Krimigis}, Stamatios M., ISBN 978-1-4020-9216-9. Springer Science+Business Media B.V., 2009, p. 459, DOI: 10.1007/978-1-4020-9217-6_15.

[4] A. Flandes, A.M. García-Reyes, L.J. Spilker, Ray-tracing thermal modelling of Saturn’s main rings, Icarus 354 (2021) 14063, doi:10.1016/j.icarus.2020.114063.

[5] A. Flandes, L. Spilker, R. Morishima, S. Pilorz, C. Leyrat, N. Altobelli, S. Brooks, S.G. Edgington, Brightness of Saturn’s rings with decreasing solar elevation, Planet. Space Sci. 58 (2010) 1758–1765.

[6] L. Froidevaux, Saturn’s rings: infrared brightness variation with solar elevation, Icarus 46 (1981) 4–17.

[7] R.G. Jerousek, J.E. Colwell, L.W. Esposito, P.D. Nicholson, M.M. Hedman, Small particles and self-gravity wakes in Saturn’s rings from UVIS and VIMS stellar occultations, Icarus 279 (2016) 36–50.

[8] H. Salo, J. Schmidt, N-body simulations of viscous instability of planetary rings, Icarus 206 (2) (2010) 390–409.