

# <sup>1</sup> growpacity: A computationally efficient dust opacity model suitable for coagulation models

<sup>3</sup> **Alexandros Ziampras**  <sup>1</sup> and **Tilman Birnstiel** <sup>1</sup>

<sup>4</sup> 1 Ludwig-Maximilians-Universität München, Universitäts-Sternwarte, Scheinerstr. 1, 81679 München,  
<sup>5</sup> Germany 2 Max Planck Institute for Astronomy, Königstuhl 17, 69117 Heidelberg, Germany 3  
<sup>6</sup> Exzellenzcluster ORIGINS, Boltzmannstr. 2, 85748 Garching, Germany

DOI: [10.xxxxxx/draft](https://doi.org/10.xxxxxx/draft)

## Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Editor: [Open Journals](#) 

Reviewers:

- [@openjournals](#)

Submitted: 01 January 1970

Published: unpublished

## License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#))

## <sup>7</sup> Summary

<sup>8</sup> growpacity is a Python and C toolkit for calculating dust opacities in astrophysical environments. It provides a class (`OpacityCalculator`) for generating, storing, and interpolating <sup>9</sup> mean opacities (Rosseland and Planck) over customizable grids of grain size distributions, <sup>10</sup> temperature, and power-law exponents. This package is designed as a wrapper around [OpTool](#), <sup>11</sup> allowing users to specify grain composition and obtain temperature-, maximum grain size-, <sup>12</sup> and dust size distribution-dependent mean opacities. The resulting tables are lightweight and <sup>13</sup> optimized for use in radiation hydrodynamics and dust coagulation models, where grain size <sup>14</sup> distributions evolve dynamically.

## <sup>16</sup> Statement of need

<sup>17</sup> Radiation hydrodynamics simulations often require a dust opacity model under the assumption of a particular grain size or size distribution. It is however not straightforward to implement a <sup>18</sup> dust opacity model that accounts for the dynamical evolution of this grain size distribution. <sup>19</sup> With growpacity, we outline a method to construct such a model and provide a python <sup>20</sup> package that implements it. The model is a wrapper around the OpTool package, with the user <sup>21</sup> providing a grain composition and receiving temperature-, maximum grain size-, and dust size <sup>22</sup> distribution-dependent mean opacities, tabulated in a lightweight format and complemented <sup>23</sup> with efficient interpolation methods for usage in radiation hydrodynamics simulations.

## <sup>25</sup> Method

<sup>26</sup> For a given grain composition and assuming a grain size distribution with number density  $n(a) \propto$  <sup>27</sup>  $a^q$  for grains with size  $a$  between minimum and maximum grain sizes  $a_{\min}$  and  $a_{\max}$ , the [OpTool](#) <sup>28</sup> package [[@DominikEtal:2021](#)] can compute the absorption and scattering opacities  $\kappa_{\text{abs}}(\nu)$  <sup>29</sup> and  $\kappa_{\text{sca}}(\nu)$  (in  $\text{cm}^2/\text{g}$ ) as well as the asymmetry factor  $g(\nu)$  [[@HenyeyGreenstein:1941](#)] over <sup>30</sup> a frequency grid  $\nu$ . This calculation is done using the Distribution of Hollow Spheres method <sup>31</sup> (DHS) [[@MinEtal:2005](#)]. The Rosseland and Planck mean opacities  $\kappa_R$  and  $\kappa_P$  can then be <sup>32</sup> computed as

$$\kappa_P(T) = \frac{\int_0^\infty \kappa_{\text{abs}} B_\nu(T) d\nu}{\int_0^\infty B_\nu(T) d\nu}, \quad \kappa_R(T) = \frac{\int_0^\infty u_\nu(T) d\nu}{\int_0^\infty [\kappa_{\text{abs}} + (1-g) \kappa_{\text{sca}}]^{-1} u_\nu(T) d\nu}, \quad u_\nu(T) = \left. \frac{dB_\nu}{dT} \right|_T,$$

<sup>33</sup> where  $B_\nu(T)$  is the Planck function at temperature  $T$ . By fixing the grain composition and <sup>34</sup>  $a_{\min}$ , OpTool can be used to calculate the absorption and scattering opacities over a grid

35 of  $a_{\max}$  and  $q$ . The above equation can then be used to compute and tabulate  $\kappa_R$  and  $\kappa_P$   
 36 over  $a_{\max}$ ,  $q$ , and  $T$ . The resulting 3D tables can be used in any context where  $a_{\max}$  and  
 37  $q$  are dynamically evolved according to a dust coagulation model [[@BirnstielEtal:2017](#);  
 38 [@StammllerBirnstiel:2022](#); [@PfeilEtal:2024](#); [@RobinsonEtal:2024](#)].

### 39 Interpolation algorithm

40 Once the 3D tables of  $\kappa_R(q, a_{\max}, T)$  and  $\kappa_P(q, a_{\max}, T)$  have been computed, they can be  
 41 interpolated within the range of tabulated values. We interpolate for  $\log \kappa$  as a function of  $q$ ,  
 42  $\log a_{\max}$ , and  $\log T$ , with regular sampling in this space (i.e., logarithmic spacing for  $a_{\max}$  and  
 43  $T$ ). This works best when the mean opacities follow a powerlaw with respect to temperature  
 44  $\kappa \propto T^b \Rightarrow \log \kappa \propto b \log T$ , which is a reasonable approximation and especially holds for small  
 45 grains [[@BellLin:1994](#); [@SemenovEtal:2003](#)]. It also ensures that the interpolated opacities  
 46 are always positive in case extrapolation is needed.

47 We use a trilinear interpolation scheme, which is fast and simple to implement, and takes  
 48 advantage of the fact that the arrays  $q$ ,  $\log a_{\max}$ , and  $\log T$  are sorted and regularly spaced  
 49 to efficiently locate the indices of the grid points that surround the point of interest. For  
 50 each array  $x \in \{q, \log a_{\max}, \log T\}$  and for a target value  $x_t$ , we first find the index  $i$  such  
 51 that  $x_i \leq x_t < x_{i+1}$  as  $i = \lfloor (x_t - x_0) \Delta x^{-1} \rfloor$ , where  $x_0$  is the first (smallest) value in the  
 52 sampling space and  $\Delta x = x_{i+1} - x_i$  is the (constant) sampling spacing. As this information  
 53 is known *a priori*, this reduces the complexity of finding the required indices from  $\mathcal{O}(\log N)$  to  
 54  $\mathcal{O}(1)$ , where  $N$  is the number of grid points in  $x$ , and is especially efficient for larger grids. Of  
 55 course, care must be taken to ensure that the indices are within the bounds of the grid.

### 56 Implementation

57 The above method is implemented in the `growpacity` package. The package provides a simple  
 58 python interface to `OpTool` to compute accurate dust opacities for a given grain composition  
 59 (provided in `OpTool` format) and for a fixed  $a_{\min}$ , over a grid of  $a_{\max}$ ,  $q$ , and  $T$ . The resulting  
 60 3D tables of  $\kappa_R$  and  $\kappa_P$  are saved in binary files that can be easily loaded with python. We  
 61 also provide an implementation of the interpolating function in C, that can be readily used in  
 62 radiation hydrodynamics codes. These files are especially lightweight: a typical calculation  
 63 involving  $q \in [-4.5, -2.5]$  with  $\Delta q = 0.25$ ,  $a_{\max} \in [0.1 \mu\text{m}, 1 \text{ m}]$  sampled twice per decade,  
 64 and  $T \in [1, 2000] \text{ K}$  with 100 points results in two arrays of  $9 \times 15 \times 100$  elements, or  
 65 about 220 kB of memory. This choice of spacing means that opacities are exactly evaluated  
 66 for  $q \in [-3.75, -3.5, -3]$ , corresponding to dust size distributions in equilibrium due to  
 67 small/large-scale turbulence, radial drift, or in a non-equilibrium growth regime. For more  
 68 information, see [[@PfeilEtal:2024](#)].

69 The code and documentation are available at  
 70 <https://github.com/alexziab/growpacity>.

### 71 Limitations and extensions

72 We underscore that our intent is not to provide a new or more realistic dust opacity model,  
 73 but rather one suitable for use in coagulation models, where dust densities and distributions  
 74 can vary as a function of position and time—the applicability of the model depends on  
 75 entirely user-defined choices. The method can be easily extended to include gas opacities  
 76 [[@SemenovEtal:2003](#); [@MalyginEtal:2014](#)] and prescriptions for the sublimation of dust  
 77 species [[@IsellaNatta:2005](#)], for a more complete opacity model in regimes where gas  
 78 opacities are significant.

## <sup>79</sup> Acknowledgements

- <sup>80</sup> AZ and TB acknowledge funding from the European Union under the Horizon Europe Research  
<sup>81</sup> and Innovation Programme 101124282 (EARLYBIRD).  
<sup>82</sup> Views and opinions expressed are those of the authors only.

DRAFT