

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

<sup>3</sup> **Alexandros Ziampras**  <sup>1,2</sup> and **Tilman Birnstiel**  <sup>1,3</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Published: unpublished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](#)). 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 dust opacity model that accounts for the dynamical evolution of this grain size distribution. This shortcoming is especially relevant in environments where dust coagulation is important, such as protoplanetary disks.

## <sup>7</sup> Summary

<sup>8</sup> growopacity is a Python and C toolkit for calculating dust opacities in astrophysical environments. It provides a framework for generating, storing, and interpolating Rosseland and Planck mean opacities over customizable grids of grain size distributions, temperatures, and powerlaw exponents. This package is designed as a wrapper around [OpTool](#), allowing users to specify a grain composition of their choice and obtain tabulated mean opacities. The resulting tables are lightweight and optimized for use in radiation hydrodynamics and dust coagulation models, where grain size distributions evolve dynamically.

## <sup>15</sup> Statement of need

<sup>17</sup> Dust coagulation models evolve the dust grain size distribution as a function of position and time, with grain sizes ranging from  $\sim 0.1 \mu\text{m}$  to  $\sim \text{cm}$ —12 orders of magnitude in mass. The complexity of modeling dust dynamically in this manner is exacerbated by the vastly different opacities between small and large grains, with small grains dominating the opacity budget but also being subject to rapid depletion due to coagulation. A dust opacity model that can account for these effects is therefore essential in accurately capturing the thermal structure of dusty astrophysical environments such as protoplanetary disks, where the temperature profile directly influences processes such as planet formation, disk chemistry, and observational signatures of substructure. With recent strides in dust coagulation modeling (e.g., [Pfeil et al., 2024](#); [Robinson et al., 2024](#); [Stammler & Birnstiel, 2022](#)), there is a growing need for an accurate, computationally efficient, flexible, and easy-to-implement dust opacity model that can be integrated into such simulations.

<sup>33</sup> With growopacity, we outline a method to construct such a model and provide a python package that interfaces to [OpTool](#) to compute temperature-, maximum grain size-, and dust size distribution-dependent mean opacities, tabulated in a lightweight format and complemented with efficient interpolation methods for usage in radiation hydrodynamics simulations.

## 37 Method

38 For a given grain composition and assuming a grain size distribution with number density  
 39  $n(a) \propto a^q$  for grains with size  $a$  between minimum and maximum grain sizes  $a_{\min}$  and  $a_{\max}$ ,  
 40 the `OpTool` package (Dominik et al., 2021) can compute the absorption and scattering opacities  
 41  $\kappa_{\text{abs}}(\nu)$  and  $\kappa_{\text{sca}}(\nu)$  (in  $\text{cm}^2/\text{g}$ ) as well as the asymmetry factor  $g(\nu)$  (Henyey & Greenstein,  
 42 1941) over a frequency grid  $\nu$ . This calculation is done using the Distribution of Hollow Spheres  
 43 method (DHS, Min et al., 2005). The Rosseland and Planck mean opacities  $\kappa_R$  and  $\kappa_P$  can  
 44 then be computed as

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

45 where  $B_\nu(T)$  is the Planck function at temperature  $T$ . By fixing the grain composition and  
 46  $a_{\min}$ , `OpTool` can be used to calculate the absorption and scattering opacities over a grid of  $q$   
 47 and  $a_{\max}$ . The above equation can then be used to compute and tabulate  $\kappa_R$  and  $\kappa_P$  over  
 48  $q$ ,  $a_{\max}$ , and  $T$ . The resulting 3D tables can be used in any context where  $q$  and  $a_{\max}$  are  
 49 dynamically evolved according to a dust coagulation model (Birnstiel et al., 2017; Pfeil et al.,  
 50 2024; Robinson et al., 2024; Stammler & Birnstiel, 2022).

## 51 Interpolation algorithm

52 Once the 3D tables of  $\kappa_R(q, a_{\max}, T)$  and  $\kappa_P(q, a_{\max}, T)$  have been computed, they can be  
 53 interpolated within the range of tabulated values. We interpolate for  $\log \kappa$  as a function of  $q$ ,  
 54  $\log a_{\max}$ , and  $\log T$ , with regular sampling in this space (i.e., logarithmic spacing for  $a_{\max}$  and  
 55  $T$ ). This works best when the mean opacities follow a powerlaw with respect to temperature  
 56  $\kappa \propto T^b \Rightarrow \log \kappa \propto b \log T$ , which is a reasonable approximation and especially holds for small  
 57 grains (Bell & Lin, 1994; Semenov et al., 2003). It also ensures that the interpolated opacities  
 58 are always positive in case extrapolation is needed.

59 We use a trilinear interpolation scheme, which is fast and simple to implement, and takes  
 60 advantage of the fact that the arrays  $q$ ,  $\log a_{\max}$ , and  $\log T$  are sorted and regularly spaced  
 61 to efficiently locate the indices of the grid points that surround the point of interest. For  
 62 each array  $x \in \{q, \log a_{\max}, \log T\}$  and for a target value  $x_t$ , we first find the index  $i$  such  
 63 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  
 64 sampling space and  $\Delta x = x_{i+1} - x_i$  is the (constant) sampling spacing. As this information  
 65 is known *a priori*, this reduces the complexity of finding the required indices from  $\mathcal{O}(\log N)$  to  
 66  $\mathcal{O}(1)$ , where  $N$  is the number of grid points in  $x$ , and is especially efficient for larger grids. Of  
 67 course, care must be taken to ensure that the indices are within the bounds of the grid.

## 68 Implementation

69 The above method is implemented in the `growpacity` package. The package provides a simple  
 70 python interface to `OpTool` to compute accurate dust opacities for a given grain composition  
 71 (provided in `OpTool` format) and for a fixed  $a_{\min}$ , over a grid of  $a_{\max}$ ,  $q$ , and  $T$ . The resulting  
 72 3D tables of  $\kappa_R$  and  $\kappa_P$  are saved in binary files that can be easily loaded with python. We  
 73 also provide an implementation of the interpolating function in C, that can be readily used in  
 74 radiation hydrodynamics codes. These files are especially lightweight: a typical calculation  
 75 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,  
 76 and  $T \in [1, 2000] \text{ K}$  with 100 points results in two arrays of  $9 \times 15 \times 100$  elements, or  
 77 about 220 kB of memory. This choice of spacing means that opacities are exactly evaluated  
 78 for  $q \in [-3.75, -3.5, -3]$ , corresponding to dust size distributions in equilibrium due to

<sup>79</sup> small/large-scale turbulence, radial drift, or in a non-equilibrium growth regime (for more  
<sup>80</sup> information, see [Pfeil et al., 2024](#)).

<sup>81</sup> The code and documentation are available at  
<sup>82</sup> <https://github.com/alexziab/growpacity>.

## <sup>83</sup> Limitations and extensions

<sup>84</sup> We underscore that our intent is not to provide a new or more realistic dust opacity model,  
<sup>85</sup> but rather one suitable for use in coagulation models, where dust densities and distributions  
<sup>86</sup> can vary as a function of position and time—the applicability of the model depends on entirely  
<sup>87</sup> user-defined choices. The method can be easily extended to include gas opacities ([Marigo et](#)  
<sup>88</sup> [al., 2024](#); e.g., [Malygin et al., 2014](#); [Semenov et al., 2003](#)) and prescriptions for the sublimation  
<sup>89</sup> of dust species (e.g., [Isella & Natta, 2005](#)), for a more complete opacity model in regimes  
<sup>90</sup> where gas opacities are significant.

## <sup>91</sup> Acknowledgements

<sup>92</sup> AZ and TB acknowledge funding from the European Union under the Horizon Europe Research  
<sup>93</sup> and Innovation Programme 101124282 (EARLYBIRD). TB acknowledges funding by the  
<sup>94</sup> Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's  
<sup>95</sup> Excellence Strategy - EXC-2094 - 390783311. Views and opinions expressed are those of the  
<sup>96</sup> authors only.

## <sup>97</sup> References

- <sup>98</sup> Bell, K. R., & Lin, D. N. C. (1994). Using FU Orionis outbursts to constrain self-regulated  
<sup>99</sup> protostellar disk models. *ApJ*, 427, 987–1004. <https://doi.org/10.1086/174206>
- <sup>100</sup> Birnstiel, T., Klahr, H., & Ercolano, B. (2017). *TWO-POP-PY: Two-population dust evolution*  
<sup>101</sup> *model*. Astrophysics Source Code Library, record ascl:1708.015.
- <sup>102</sup> Dominik, C., Min, M., & Tazaki, R. (2021). *OpTool: Command-line driven tool for creating*  
<sup>103</sup> *complex dust opacities*. Astrophysics Source Code Library, record ascl:2104.010.
- <sup>104</sup> Henyey, L. G., & Greenstein, J. L. (1941). Diffuse radiation in the Galaxy. *ApJ*, 93, 70–83.  
<sup>105</sup> <https://doi.org/10.1086/144246>
- <sup>106</sup> Isella, A., & Natta, A. (2005). The shape of the inner rim in proto-planetary disks. *A&A*,  
<sup>107</sup> 438(3), 899–907. <https://doi.org/10.1051/0004-6361:20052773>
- <sup>108</sup> Malygin, M. G., Kuiper, R., Klahr, H., Dullemond, C. P., & Henning, Th. (2014). Mean gas  
<sup>109</sup> opacity for circumstellar environments and equilibrium temperature degeneracy. *A&A*, 568,  
<sup>110</sup> A91. <https://doi.org/10.1051/0004-6361/201423768>
- <sup>111</sup> Marigo, P., Addari, F., Bossini, D., Bressan, A., Costa, G., Girardi, L., Pastorelli, G., Trabucchi,  
<sup>112</sup> M., & Volpato, G. (2024). *ÆSOPUS 2.1: Low-temperature Opacities Extended to High*  
<sup>113</sup> *Pressure*. 976(1), 39. <https://doi.org/10.3847/1538-4357/ad7b27>
- <sup>114</sup> Min, M., Hovenier, J. W., & de Koter, A. (2005). Modeling optical properties of cosmic dust  
<sup>115</sup> grains using a distribution of hollow spheres. *A&A*, 432(3), 909–920. <https://doi.org/10.1051/0004-6361:20041920>
- <sup>117</sup> Pfeil, T., Birnstiel, T., & Klahr, H. (2024). TriPoD: Tri-Population size distributions for Dust  
<sup>118</sup> evolution: Coagulation in vertically integrated hydrodynamic simulations of protoplanetary  
<sup>119</sup> disks. *A&A*, 691, A45. <https://doi.org/10.1051/0004-6361/202449337>

- <sup>120</sup> Robinson, A., Booth, R. A., & Owen, J. E. (2024). Introducing CUDISC: a 2D code for  
<sup>121</sup> protoplanetary disc structure and evolution calculations. *MNRAS*, 529(2), 1524–1541.  
<sup>122</sup> <https://doi.org/10.1093/mnras/stae624>
- <sup>123</sup> Semenov, D., Henning, Th., Helling, Ch., Ilgner, M., & Sedlmayr, E. (2003). Rosseland and  
<sup>124</sup> Planck mean opacities for protoplanetary discs. *A&A*, 410, 611–621. <https://doi.org/10.1051/0004-6361:20031279>
- <sup>126</sup> Stammmer, S. M., & Birnstiel, T. (2022). DustPy: A Python Package for Dust Evolution in  
<sup>127</sup> Protoplanetary Disks. *ApJ*, 935(1), 35. <https://doi.org/10.3847/1538-4357/ac7d58>

DRAFT