Introducing cuDisc: a 2D code for protoplanetary disc structure and evolution calculations

Alfie Robinson,[1]★ Richard A. Booth[2] and James E. Owen[1]
[1]Astrophysics Group, Imperial College London, Prince Consort Road, London SW7 2AZ, UK
[2]School of Physics and Astronomy, University of Leeds, Leeds, LS2 9JT, UK

Accepted XXX. Received YYY; in original form ZZZ

ABSTRACT

We present a new 2D axisymmetric code, cuDisc, for studying protoplanetary discs, focusing on the self-consistent calculation of dust dynamics, grain size distribution and disc temperature. Self-consistently studying these physical processes is essential for many disc problems, such as structure formation and dust removal, given that the processes heavily depend on one another. To follow the evolution over substantial fractions of the disc lifetime, cuDisc uses the CUDA language and libraries to speed up the code through GPU acceleration. cuDisc employs a second-order finite-volume Godonuv solver for dust dynamics, solves the Smoluchowski equation for dust growth and calculates radiative transfer using a multi-frequency hybrid ray-tracing/flux-limited-diffusion method. We benchmark our code against current state-of-the-art codes. Through studying steady-state problems, we find that including 2D structure reveals that when collisions are important, the dust vertical structure appears to reach a diffusion-settling-coagulation equilibrium that can differ substantially from standard models that ignore coagulation. For low fragmentation velocities, we find an enhancement of intermediate-sized dust grains at heights of ∼ 1 gas scale height due to the variation in collision rates with height, and for large fragmentation velocities, we find an enhancement of small grains around the disc mid-plane due to collisional "sweeping" of small grains by large grains. These results could be important for the analysis of disc SEDs or scattered light images, given these observables are sensitive to the vertical grain distribution.

Key words: protoplanetary discs – methods: numerical – stars: pre-main sequence

1 INTRODUCTION

The past few decades have seen the study of young planetary systems and their formation environments become a rapidly evolving field with major interest within the astrophysics community. This is largely due to an unprecedented wealth of observations made possible by recent observatories such as ALMA (Wootten & Thompson 2009) and the continual discovery of diverse exoplanetary systems (e.g Mayor et al. 2011; Batalha et al. 2013; Winn & Fabrycky 2015; Madhusudhan 2019; Zhu & Dong 2021). Protoplanetary discs, the discs of gas and dust that form around young stars, are the birthplace of such planetary systems and have, therefore, been subject to extensive theoretical interest. Various questions relating to their nature have arisen in recent years that remain at least partly unanswered by current theoretical models. Examples of such problems include: the nature of the mechanisms behind "sub-structure" formation in protoplanetary discs, as observations have shown these objects to exhibit diverse features from axisymmetric rings and gaps to non-axisymmetric arcs (Andrews 2020; Bae et al. 2022); the connection between the spatial distribution and evolution of chemical species in discs to the eventual compositions of planetary cores and atmospheres (Öberg et al. 2011; Booth et al. 2017; Madhusudhan 2019; Eistrup 2023); and mechanisms for the dispersal of protoplanetary discs after their observed lifetimes of ∼ a few Myr (Ercolano & Pascucci 2017; Owen & Kollmeier 2019).

Exploring these problems requires sophisticated numerical modelling, given the plethora of physical processes that govern the structure and evolution of protoplanetary discs. Our understanding of discs has primarily been advanced through the use of state-of-the-art codes for studying the dynamics and thermodynamics of the gas and dust that comprise the disc material. 2D and 3D simulations have typically been used to study discs on short, dynamical time-scales, given their computational cost, whilst 1D models have often been used to study discs on longer, secular time-scales. Work done using these models has hugely advanced our understanding of protoplanetary discs; however, it has become evident that for certain problems, the interplay of each of the facets of disc physics - dynamics, thermodynamics and the dust size distribution - must be studied self-consistently over secular time-scales. The 1D code DustPy (Stammler & Birnstiel 2022) is the current state-of-the-art for studying problems of this nature; however, it cannot be used if the problem depends on the intricacies of the disc vertical structure. Examples of such problems include temperature instabilities (Watanabe & Lin 2008; Wu & Lithwick 2021; Fuksman & Klahr 2022) where 2D temperature solvers have been used but dust dynamics and growth neglected, snow line instabilities (Owen 2020) where 1D temperature and dynamics solvers have been used, and problems relating to disc dispersal such as the removal of dust from discs via radiation pressure (Owen &

★ E-mail: a.robinson21@imperial.ac.uk

Kollmeier 2019; Krumholz et al. 2020), which has been studied with both 1D and 2D solvers but without dust growth/fragmentation, or entrainment in photoevaporative or magnetic winds (Franz et al. 2020; Booth & Clarke 2021; Hutchison & Clarke 2021; Rodenkirch & Dullemond 2022), again with 1D and 2D approaches but with dust grain growth neglected.

This paper presents a new code, `cuDisc`, that includes 2D multi-fluid dust dynamics coupled to both 2D temperature evolution and 1D secular gas evolution. Dust and gas can be evolved, whilst simultaneously evolving the dust grain size distribution and the resulting temperature structure. This self-consistent calculation means fewer assumptions about the system's state for a particular scenario must be made. The code uses a 2D grid in the poloidal plane, meaning that assumptions about vertical structure in the dust do not have to be made as the structure can be calculated self-consistently. In order to allow evolution calculations to be run for significant fractions of the typical disc lifetime (~ Myr) in computationally feasible time-scales, `cuDisc` utilises GPU acceleration via the employment of the CUDA language and libraries. In the rest of this paper, Sections 2, 3, 4 and 5 outline the numerical methods used by `cuDisc` whilst Section 6 outlines the code structure and Section 7 shows some example science calculations made using `cuDisc` where we study grain growth in a steady-state transition disc.

2 GRID STRUCTURE

`cuDisc` is a 2D code in the poloidal plane, with axisymmetry adopted about the disc's rotation axis. The grid is structured in a fashion that makes certain features of disc physics easier to calculate; cell interfaces are defined along lines of constant polar angle above the mid-plane, θ, and constant cylindrical radius, R, as shown in Fig. 1. This means that the basis vectors of the grid coordinate system are spherical radius, $\hat{\boldsymbol{r}}$, and cylindrical height, $\hat{\boldsymbol{Z}}$, although vector quantities (such as velocity) are dealt with in the standard cylindrical components. This grid structure allows for ray-tracing from the central star for calculating quantities such as optical depth whilst also making calculations that require vertical integration easy, such as computing hydrostatic equilibrium. Cell spacing can be arbitrary in principle, but the standard implementation is logarithmic in R with a power law in θ. The individual cell structure is shown in Fig. 2. Physical quantities are stored at cell centres, and edge coordinates are required for the reconstruction of the cell-centred quantities to the cell interfaces for the advection routine (see later sections). Cell volumes, $V_{i,j}$, and interface areas $A^R_{i,j}$ & $A^Z_{i,j}$ are given by

$$V_{i,j} = \frac{1}{3} d(R^3)_i d(\tan\theta)_j, \tag{1}$$

$$A^R_{i,j} = (R^e_i)^2 d(\tan\theta)_j, \tag{2}$$

$$A^Z_{i,j} = \frac{1}{2}\frac{d(R^2)_i}{\cos\theta^e_j}. \tag{3}$$

These geometric factors are written in such a way as to avoid divergence as $R \rightarrow 0$. Explicitly, $d(R^n)_i = (R^e_{i+1})^n - (R^e_i)^n$ and $d(\tan\theta)_j = \tan\theta^e_{j+1} - \tan\theta^e_j$. The Z coordinates are then given by $Z^e_{i,j} = R^c_i \tan\theta^e_{i,j}$ and $Z^c_{i,j} = R^c_i \tan\theta^c_{i,j}$. Due to the non-orthogonality of the grid coordinate bases, when calculating fluxes through interfaces, the dot-product of velocity with the interface normals must be calculated to only account for the velocity component orthogonal to the interface.

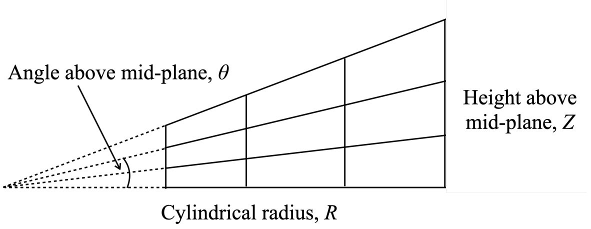

Figure 1. The computational grid used in `cuDisc`. Cell boundaries exist along lines of constant radius and constant angle above the mid-plane, θ.

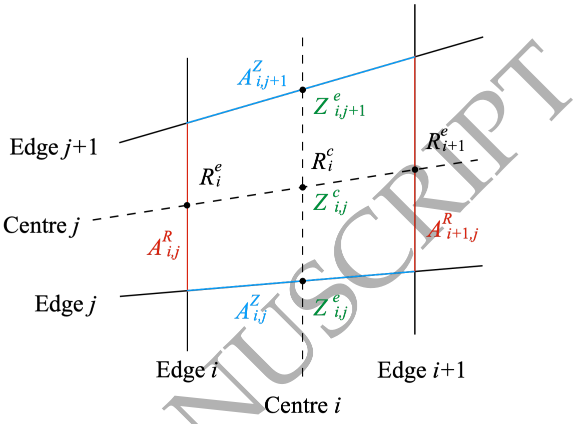

Figure 2. The cell structure and indexing implemented in `cuDisc`. The indices in the R and Z directions are i and j, respectively.

At each edge of the active cell domain, ghost cells are employed in order to set the boundary conditions. The conditions along each of the four boundaries can be set independently as either: outflow, where ghost cell quantities are set by the values in the first adjacent active cell, zero, where ghost cell quantities are set to floor values, or closed, where ghost cell quantities are set by the values in the adjacent active cell but with zero flux over the boundary. The default setup assumes vertical symmetry in the disc about the mid-plane and, therefore, uses a minimum θ of 0 with a closed boundary condition and outflow boundary conditions for the three other boundaries.

3 DYNAMICS SOLVER

In `cuDisc`, the dust species are evolved by treating each grain size as a pressureless fluid and solving their associated advection-diffusion equations. We employ a second-order finite-volume Godonuv scheme for solving the set of equations (Stone & Gardiner 2009). We write our equations in terms of vector fields of the conserved quantities, $\boldsymbol{Q}$, their associated fluxes, $\boldsymbol{F}(\boldsymbol{Q})$, and any source terms, $\boldsymbol{S}$. For our system, these fields are given by

$$\boldsymbol{Q}_i = \begin{pmatrix} \rho_i \\ \rho_i v_{R,i} \\ \rho_i v_{\phi,i} R \\ \rho_i v_{Z,i} \end{pmatrix}, \tag{4}$$

where ρ_i and $\boldsymbol{v}_i = (v_{R,i}, v_{\phi,i}, v_{Z,i})$ are the volume density and velocity of dust species i respectively,

$$\boldsymbol{F}_i = \begin{pmatrix} \rho_i \boldsymbol{v}_i + \boldsymbol{F}_{\mathrm{diff},i} \\ v_{R,i}(\rho_i \boldsymbol{v}_i + \boldsymbol{F}_{\mathrm{diff},i}) \\ v_{\phi,i} R(\rho_i \boldsymbol{v}_i + \boldsymbol{F}_{\mathrm{diff},i}) \\ v_{Z,i}(\rho_i \boldsymbol{v}_i + \boldsymbol{F}_{\mathrm{diff},i}) \end{pmatrix}, \tag{5}$$

where the two terms for each conserved quantity are the fluxes generated by advection, $\rho_i \boldsymbol{v}_i$, and diffusion, $\boldsymbol{F}_{\mathrm{diff},i}$,

$$\boldsymbol{S}_i = \begin{pmatrix} 0 \\ \dfrac{\rho_i v_{\phi,i}^2}{R} - \rho_i \Omega^2 R + f_{\mathrm{drag},i,R} + f_{\mathrm{ext},i,R} \\ f_{\mathrm{drag},i,\phi} \\ -\rho_i \Omega^2 Z + f_{\mathrm{drag},i,Z} + f_{\mathrm{ext},i,Z} \end{pmatrix}, \tag{6}$$

where Ω is the Keplerian angular velocity, $\sqrt{GM_*/(R^2+Z^2)^{3/2}}$, and $\boldsymbol{f}_{\mathrm{drag}}$ and $\boldsymbol{f}_{\mathrm{ext}}$ are source terms that arise due to dust-gas drag and any external forces (e.g. radiation pressure). We formulate diffusion in the momentum equations as the diffusive flux acting to diffuse the advective quantities. We do not currently consider the other terms discussed in recent works, such as Huang & Bai (2022); these being the advection of the diffusive flux and the time-dependent diffusive flux. Diffusion related terms arise out of modelling turbulence by writing the density and velocity components as a short-term average plus a short-term fluctuation, then averaging the resulting equations over the short-term, keeping any terms that include correlations between fluctuations (Reynolds averaging, see e.g. Cuzzi et al. 1993). Note that we solve an azimuthal equation even though all ϕ derivatives are zero by axisymmetry, as the advection of angular momentum throughout the (R, Z) plane could be important in some problems; this is sometimes referred to as 2.5D. Also, note that we write this equation in angular momentum conserving form as this removes the Coriolis force term, which has been shown to lead to loss of angular momentum conservation (see e.g. Kley 1998).

The quantities, $\boldsymbol{Q}$, are updated by solving the advection-diffusion equations given by

$$\frac{\partial \boldsymbol{Q}}{\partial t} + \nabla \cdot \boldsymbol{F} = \boldsymbol{S}. \tag{7}$$

We solve this set of equations in two stages via operator splitting: a transport step where the advection-diffusion equations are solved as homogeneous hyperbolic equations and a source step where the quantities are updated through the source terms. For the transport step, the equations are integrated over volume to find the flux-conservative form. Discretising this gives the equation used to evolve a quantity Q in cell i, j to time $n+1$,

$$Q_{i,j}^{n+1} = Q_{i,j}^{n} - \frac{\Delta t}{V_{i,j}} (FA)_{i,j}^{n+\frac{1}{2}}, \tag{8}$$

where $(FA)_{i,j}$ is the total area-weighted-flux exiting through the interfaces between cell i, j and adjacent cells, given by

$$(FA)_{i,j} = F_{i+1,j}^{R} A_{i+1,j}^{R} - F_{i,j}^{R} A_{i,j}^{R} + F_{i,j+1}^{Z} A_{i,j+1}^{Z} - F_{i,j}^{Z} A_{i,j}^{Z}. \tag{9}$$

The time index of $n+\frac{1}{2}$ indicates that the fluxes should be calculated at the half time-step to represent the average flux over the full time-step Δt. Following Stone & Gardiner (2009), we calculate these half-time fluxes in the following manner:

(i) The primitive quantities, $(\rho_i, \boldsymbol{v}_i)$, are reconstructed at both sides of the cell interfaces from the conserved quantities at cell centres using a first-order donor cell method and a Riemann solver is used to calculate first-order advective fluxes through each interface at time n.

(ii) The diffusive fluxes at each interface at time n are calculated from the cell-centred conserved quantities and added to the advective fluxes to get the total area-weighted flux exiting each cell at time n, $(FA)_{i,j}^{n}$.

(iii) The conserved quantities are advanced by a half time-step using these fluxes:

$$Q_{i,j}^{n+\frac{1}{2}} = Q_{i,j}^{n} - \frac{1}{2}\frac{\Delta t}{V_{i,j}} (FA)_{i,j}^{n}, \tag{10}$$

(iv) These half-updated quantities are given a half time-step source update (described in detail in subsection 3.1).

(v) Primitive quantities at time $n+\frac{1}{2}$ are now reconstructed from these half-updated quantities using a second-order piecewise linear method with a modified van Leer limiter detailed by Mignone (2014). These half-time primitive quantities are used to calculate second-order advective fluxes through each interface at time $n+\frac{1}{2}$.

(vi) The diffusive fluxes at each interface at time $n+\frac{1}{2}$ are calculated from the half-updated cell-centred quantities and added to the second-order advective fluxes to get the total area-weighted flux exiting each cell at time $n+\frac{1}{2}$, $(FA)_{i,j}^{n+\frac{1}{2}}$.

In steps (i) and (v), where the advective fluxes are calculated, the Riemann problem has to be solved at the cell interfaces. For pressureless fluids, collisions between fluids with different velocities lead to "delta shocks", which travel at the Roe-averaged velocity (Roe 1981),

$$\hat{v} = \frac{\sqrt{\rho_l}\boldsymbol{v}_l \cdot \hat{\boldsymbol{n}} + \sqrt{\rho_r}\boldsymbol{v}_r \cdot \hat{\boldsymbol{n}}}{\sqrt{\rho_l} + \sqrt{\rho_r}}, \tag{11}$$

where r and l denote the values of the reconstructed quantities to the left and right of the interface in question. The reconstructed velocities are dotted with the interface normal, $\hat{\boldsymbol{n}}$, to find the velocity orthogonal to each interface; this is important given that we use a non-orthogonal grid. Numerically, we capture this process according to Leveque (2004) by taking the upwind flux depending on the sign of the Roe-averaged velocity:

$$F_{\mathrm{adv}} = \begin{cases} Q_l \boldsymbol{v}_l \cdot \hat{\boldsymbol{n}} & \text{if } \hat{v} > 0 \\ Q_r \boldsymbol{v}_r \cdot \hat{\boldsymbol{n}} & \text{if } \hat{v} < 0 \\ \frac{1}{2}(Q_l \boldsymbol{v}_l \cdot \hat{\boldsymbol{n}} + Q_r \boldsymbol{v}_r \cdot \hat{\boldsymbol{n}}) & \text{if } \hat{v} = 0 \\ 0 & \text{if } \boldsymbol{v}_l \cdot \hat{\boldsymbol{n}} < 0 < \boldsymbol{v}_r \cdot \hat{\boldsymbol{n}}, \end{cases} \tag{12}$$

The diffusive fluxes are then added to these advective fluxes to find the total flux.

3.1 Dust source terms and diffusion

The source terms given by Eqn. 6 include curvature terms, gravitational terms, drag terms and any other external forces. Drag is caused by dust-gas interactions and is given by

$$\boldsymbol{f}_{\mathrm{drag}} = -\frac{\rho}{t_s}(\boldsymbol{v} - \boldsymbol{v}_g), \tag{13}$$

where $\boldsymbol{v}_g$ is the gas velocity and t_s is the stopping time that conveys how well-coupled dust grains are to the gas flow; short stopping times imply well-coupled dust grains that are entrained in the gas flow. For the problems that we intend to study using cuDisc, the largest dust grains have radii ∼ cm size, meaning that we are in the regime of Epstein drag. This gives the stopping time as

$$t_s = \frac{\rho_m s}{\rho_g v_{\rm th}}, \tag{14}$$

where ρ_m is the dust grain internal density, s is the dust grain radius, ρ_g is the gas mass density and $v_{\rm th}$ is the thermal velocity of the gas, usually given by $(8/\pi)^{1/2} c_s$, where c_s is the isothermal gas sound speed.

Including source term updates, the full time-step is given by:

(i) $\boldsymbol{Q}^*_{i,j} = \boldsymbol{Q}^n_{i,j} - \frac{1}{2}\frac{\Delta t}{V_{i,j}}(\boldsymbol{FA})^n_{i,j}$

(ii) $\boldsymbol{Q}^{**}_{i,j} = \boldsymbol{Q}^*_{i,j} + \frac{1}{2}\Delta t \boldsymbol{S}_{\rm exp}(\boldsymbol{P}^n_{i,j})$

(iii) $\boldsymbol{Q}^{**}_{i,j} \rightarrow \boldsymbol{P}^{**}_{i,j}$

(iv) $\boldsymbol{P}^{n+\frac{1}{2}}_{i,j} = \left(\rho^{**}_{i,j},\ \boldsymbol{v}^{**}_{i,j} - \frac{\Delta t/2}{\Delta t/2 + t_s}(\boldsymbol{v}^{**}_{i,j} - \boldsymbol{v}_{g,i,j})\right)$

(v) $\boldsymbol{Q}^{\dagger}_{i,j} = \boldsymbol{Q}^n_{i,j} - \frac{\Delta t}{V_{i,j}}(\boldsymbol{FA})^{n+\frac{1}{2}}_{i,j}$

(vi) $\boldsymbol{Q}^{\dagger\dagger}_{i,j} = \boldsymbol{Q}^{\dagger}_{i,j} + \Delta t \boldsymbol{S}_{\rm exp}(\boldsymbol{P}^{n+\frac{1}{2}}_{i,j})$

(vii) $\boldsymbol{Q}^{\dagger\dagger}_{i,j} \rightarrow \boldsymbol{P}^{\dagger\dagger}_{i,j}$

(viii) $\boldsymbol{P}^{n+1}_{i,j} = \left(\rho^{\dagger\dagger}_{i,j},\ \boldsymbol{v}^{\dagger\dagger}_{i,j} - \frac{\Delta t}{\Delta t + t_s}(\boldsymbol{v}^{\dagger\dagger}_{i,j} - \boldsymbol{v}_{g,i,j})\right)$

where $\boldsymbol{Q}$ & $\boldsymbol{P}$ represent conserved quantities and primitive quantities respectively, $\boldsymbol{S}_{\rm exp}$ is the vector of source terms that are solved explicitly (curvature, gravity and external forces) and quantities with an asterisk ($*$) or a dagger ($\dagger$) signify intermediate states. Steps (iii) and (vii) signify conversions of conserved quantities to primitive quantities. Steps (iv) and (viii) are the drag updates to the primitive velocities; these are solved implicitly, as if they were solved explicitly, then short stopping times would prohibitively limit the length of an explicit time-step. As the implicit update only includes the drag terms, the Jacobian for the implicit system of equations is linear and diagonal, meaning the solution has a simple, exact analytic form. Boundary conditions are set before step (i) and again after step (iv).

The diffusive mass flux discussed in the previous section arises due to the diffusion of dust grains via turbulent gas motions. This diffusive velocity is calculated in a way analogous to molecular diffusion (Clarke & Pringle 1988; Takeuchi & Lin 2002), where the diffusive flux is given by

$$\boldsymbol{F}_{{\rm diff},i} = -\frac{\rho_g \nu}{{\rm Sc}_i}\nabla\left(\frac{\rho_i}{\rho_g}\right), \tag{15}$$

where ν is the kinematic viscosity of the disc, and ${\rm Sc}_i$ is the Schmidt number, a dimensionless number that represents the strength of the dust-gas coupling for a given grain. In the calculations presented in this paper, ${\rm Sc}_i$ is set to unity for all dust species, however the code allows any choice for the form of ${\rm Sc}_i$, which may vary arbitrarily with both position and dust properties. Given that we formulate the diffusion in the momentum equations as the diffusive flux acting to diffuse the advective quantities, the diffusive momentum flux at an interface is given by $\boldsymbol{F}_{{\rm diff},i}$ multiplied by the left or right advective velocity depending on the sign of the diffusive flux itself,

$$F_{\rm diff,mom} = \begin{cases} F_{\rm diff}\boldsymbol{v}_l \cdot \hat{\boldsymbol{n}} & \text{if } F_{\rm diff} > 0 \\ F_{\rm diff}\boldsymbol{v}_r \cdot \hat{\boldsymbol{n}} & \text{otherwise.} \end{cases} \tag{16}$$

These diffusive fluxes are then added to the advective fluxes defined earlier to give the total flux.

The time interval Δt for each time-step is determined from the Courant-Friedrichs-Lewy (CFL; Courant et al. 1928) condition:

$$\Delta t_{\rm CFL} = \min\left(\Delta t^{\rm adv}_{\rm CFL}, \Delta t^{\rm diff}_{\rm CFL}\right), \tag{17}$$

where $\Delta t^{\rm adv}_{\rm CFL}$ & $\Delta t^{\rm diff}_{\rm CFL}$ are the CFL time-steps for advection and diffusion respectively, given by

$$\Delta t^{\rm adv}_{\rm CFL} = C^{\rm adv}\min\left(|dR_i/v^s_{R,i,j}|, |dZ_{i,j}/v^s_{Z,i,j}|\right), \tag{18}$$

where the minimum is over all cells i, j and species s, and

$$\Delta t^{\rm diff}_{\rm CFL} = C^{\rm diff}\min\left(|(dR_i)^2/(\nu_{i,j}/{\rm Sc}_s)|, |(dZ_{i,j})^2/(\nu_{i,j}/{\rm Sc}_s)|\right). \tag{19}$$

Both time-steps have associated Courant numbers, $C^{\rm adv}$ & $C^{\rm diff}$, which are safety factors set by default to 0.4 and 0.2, respectively.

3.2 Diffusion

For both the dust dynamics and the radiative diffusion problem presented in Section 4, we need to calculate diffusive fluxes. In general, a diffusive flux can be written as

$$\boldsymbol{F} = -D\nabla u, \tag{20}$$

where D is the diffusion constant and u is the quantity being diffused.

Calculating the diffusive flux requires care in `cuDisc` given the non-orthogonal coordinate system in use. This is because using the standard difference formulae would give gradients of u that are not perpendicular to the interface in question and therefore using them directly would provide incorrect fluxes across the interfaces. To deal with this we need a method that calculates the 2D gradient of u that we can dot-product with the interface normal to find the flux across the interface. To this end, we use the second-order accurate, conservative method detailed by Wu et al. (2012) for diffusion problems on arbitrary mesh structures. Here, we present this method specifically for our mesh.

To compute the gradients of u within a cell, Wu et al. introduce interpolation points on the cell interfaces. The location of these interpolation points and accompanying values of u are determined through the conditions (i) u must be continuous everywhere and (ii) the flux across the interface must be continuous, i.e. $D\hat{\boldsymbol{n}} \cdot \nabla u$ must be continuous at the interface, $\hat{\boldsymbol{n}}$ being the interface normal vector. In this paper, we describe how these interpolation points are found; for readers interested in why these criteria lead to the interpolation points, we refer to the appendix of Wu et al. (2012). The flux at an interface is built up from one-sided fluxes on each side of an interface. Take a radial interface σ between cells k and l (see Fig. 3). The net flux across the face from the perspective of cell k is given by

$$\tilde{F}_\sigma = w_{k\sigma}F_{k\sigma} - w_{l\sigma}F_{l\sigma}, \tag{21}$$

where $w_{k\sigma}$ and $w_{l\sigma}$ are weights assigned to the one-sided flux contributions from the cells on each side of the interface, $F_{k\sigma}$ and $F_{l\sigma}$, where these fluxes are directed out of cells k/l across the interface σ. The weights are defined as

$$w_{k\sigma} = \frac{\mu_{k\sigma}}{\mu_{k\sigma} + \mu_{l\sigma}}, \quad w_{k\sigma} + w_{l\sigma} = 1, \tag{22}$$

where $\mu_{k\sigma} = \frac{d_{k\sigma}}{D_k}$, $d_{k\sigma}$ being the shortest (perpendicular) distance

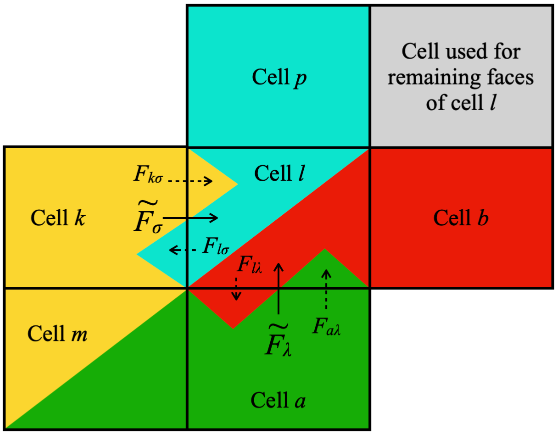

Figure 3. The cell stencil used for calculating diffusive fluxes over interfaces. The cells are shown as orthogonal for illustrative ease. The total flux $\tilde{F}_\sigma$ through the interface σ that connects cells k and l is calculated using the one-sided fluxes, $F_{k\sigma}$ and $F_{l\sigma}$. The same can be said for the total flux $\tilde{F}_\lambda$ for interface λ but with the one-sided fluxes $F_{a\lambda}$ and $F_{l\lambda}$. The cell colours indicate which cells are used to construct the one-sided fluxes: cells with yellow for $F_{k\sigma}$, cyan for $F_{l\sigma}$, green for $F_{k\lambda}$, and red for $F_{l\lambda}$.

between the cell centre and the plane of the interface σ and D_k the diffusion constant associated with cell k.

To include all flux orthogonal to the interface σ, the one-sided flux $F_{k\sigma}$ is calculated by decomposing the interface normal vector $\boldsymbol{n}_\sigma$ into two vectors, $\boldsymbol{v}_{kl}$ and $\boldsymbol{v}_{km}$, that point from cell centre k to the interpolation points on the cell interface in question and the clockwise-adjacent interface that joins cell k to cell m, as shown in Fig. 4. To find the interpolation point $\boldsymbol{v}_{kl}$ we first find two vectors that connect cell centre k to the points on the interface that are closest to cell centres k and l, shown as $\boldsymbol{dv}_{kl,1}$ and $\boldsymbol{dv}_{kl,2}$ in Fig. 4. $\boldsymbol{v}_{kl}$ is then constructed using a weighted sum of $\boldsymbol{dv}_{kl,1}$ and $\boldsymbol{dv}_{kl,2}$,

$$\boldsymbol{v}_{kl} = w_{k\sigma}\boldsymbol{dv}_{kl,1} + w_{l\sigma}\boldsymbol{dv}_{kl,2}. \tag{23}$$

The vector $\boldsymbol{v}_{km}$ is constructed similarly but with vectors and weights associated with interface τ that connects cells k and m. The interface normal is then decomposed into these two vectors by solving:

$$c_{kl}\boldsymbol{v}_{kl} + c_{km}\boldsymbol{v}_{km} = \boldsymbol{n}_\sigma \tag{24}$$

for the coefficients c_{kl} and c_{km}. These coefficients allow us to calculate the gradients across the interface without explicitly calculating the values of u at the interpolation points. The one-sided flux F_k is then calculated via the weighted sum of these gradients:

$$F_{k\sigma} = -D_k\left[w_{l\sigma}c_{kl}(u_l - u_k) + w_{m\tau}c_{km}(u_m - u_k)\right]. \tag{25}$$

To calculate the net flux $\tilde{F}_\sigma$ from Eqn. 21, $F_{l\sigma}$ is constructed similarly to $F_{k\sigma}$ but from the perspective of cell l. For polar/height interfaces such as interface λ in Fig. 3, the process is the same except that the stencil for constructing one-sided fluxes uses the interface in question and the anti-clockwise adjacent interface as opposed to clockwise.

3.3 Gas

In `cuDisc`, the 1D viscous evolution equation for gas surface density is solved, and the 2D gas distribution is calculated through hydrostatic equilibrium. Currently, there is no feedback from the dust to the gas; however, it may be added in the future. The viscous evolution equation is given by (Pringle 1981)

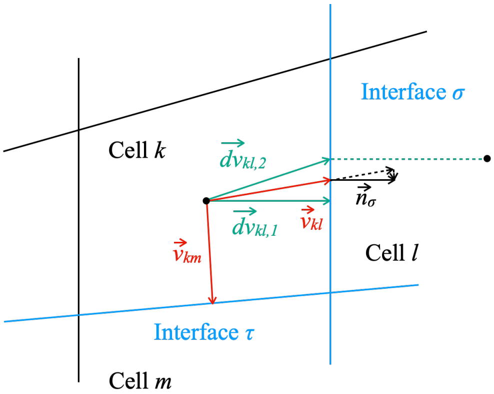

Figure 4. The vector decomposition used for the cell interface normals when calculating diffusive fluxes. Green vectors show those that point from cell centre k to the points on the interface σ that are closest to cell centres k and l. The red vector $\boldsymbol{v}_{kl}$ is then constructed via a weighted sum of the two green vectors. $\boldsymbol{v}_{km}$ is constructed from vectors for the interface between cells k and m but are not shown for simplicity.

$$\frac{\partial \Sigma_g}{\partial t} = \frac{3}{R}\frac{\partial}{\partial R}\left(R^{1/2}\frac{\partial}{\partial R}(R^{1/2}\nu\Sigma_g)\right) + S_g, \tag{26}$$

where S_g represents any source terms (e.g. mass loss via photoevaporative winds). The equation is solved in two stages via operator splitting; in the first stage, the first term on the right-hand side is solved as a diffusion problem using an explicit forward-time-centered-spatial scheme, and in the second stage, the source terms are solved using explicit Eulerian updates.

The equations we use to calculate the gas velocities are found from viscous theory (see e.g Urpin 1984; Balbus & Papaloizou 1999), written in the coordinate grid basis of `cuDisc`, $(\hat{\boldsymbol{r}}, \hat{\boldsymbol{Z}})$. These velocities are needed explicitly for their influence on dust dynamics - they are not used for evolving the gas. Under the assumption that $v_Z \ll v_R, v_\phi$ (justified for thin discs as $v_Z \sim (H/R)v_R$ where H is the gas scale height, see Urpin 1984), we find

$$v_\phi = \left[\frac{GM_* \cos^2\theta}{r} + \frac{r}{\rho_g}\left(\frac{\partial P}{\partial r} - \sin\theta\frac{\partial P}{\partial Z}\right)\right]^{1/2}, \tag{27}$$

$$v_R = r\cos\theta\left[\frac{\partial(rv_\phi)}{\partial r} - r\sin\theta\frac{\partial v_\phi}{\partial Z}\right]^{-1} \times$$
$$\left\{\frac{1}{\rho_g}[\nabla\cdot\boldsymbol{T}]_\phi - v_{Z,\mathrm{cyl}}\frac{\partial v_\phi}{\partial Z}\right\}, \tag{28}$$

where the derivatives are performed along the grid directions of spherical r and cylindrical Z. $v_{Z,\mathrm{cyl}}$ is the physical velocity in the Z-direction. We do not include a form for the vertical velocity due to viscosity here as it depends on assumptions made about the turbulence model (see e.g. Philippov & Rafikov 2017); it can also be governed by other processes such as disc winds. For these reasons, by default (and for the tests in this paper) the vertical gas velocity is assumed to be 0. Different models of vertical gas motions can be included depending on the problem in question. θ is the angle above the disc mid-plane, P is the gas pressure, given by the ideal gas law, and $\boldsymbol{T}$ is the viscous stress tensor. To calculate $[\nabla\cdot\boldsymbol{T}]_\phi$, the $r\phi$ and $Z\phi$ components of $\boldsymbol{T}$ are required. Assuming a Navier-Stokes-like viscosity, these have the forms:

$$T^{r\phi} = \frac{\rho_g \nu}{\cos^2\theta}\left[r\frac{\partial}{\partial r}\left(\frac{v_\phi}{r}\right) - \sin\theta\frac{\partial v_\phi}{\partial Z}\right], \tag{29}$$

$$T^{Z\phi} = \frac{\rho_g \nu}{\cos^2\theta}\left[\frac{\partial v_\phi}{\partial Z} - r\sin\theta\frac{\partial}{\partial r}\left(\frac{v_\phi}{r}\right)\right], \tag{30}$$

and the tensor divergence in the ϕ direction is explicitly written as

$$[\nabla\cdot\boldsymbol{T}]_\phi = \frac{1}{r^3}\frac{\partial(r^3T^{r\phi})}{\partial r} + \frac{\partial T^{Z\phi}}{\partial Z}. \tag{31}$$

These derivatives are calculated using finite differencing on the grid variables to calculate cell-centred values for the gas velocities.

3.4 Kinematic viscosity

The kinematic viscosity ν is important for governing the gas' evolution and dust diffusion. It is often parameterised following Shakura & Sunyaev (1973) as $\alpha c_s H$ where H is the gas scale height, and α is a constant that is used to control the strength of the effective viscosity generated by turbulent processes. With a constant α, this assumes that the viscosity varies with temperature changes in the disc. However, α need not be assumed constant; it can depend on other quantities such as ionisation fraction or magnetic field strength (see e.g Jankovic et al. 2021). A different way to parameterise the viscosity that removes this variation with the temperature profile calculated by `cuDisc` is $\nu = \nu_0(R/R_0)$, where the linear dependence on radius comes from assuming a mid-plane temperature profile proportional to $R^{-1/2}$. Both methods of setting the viscosity are available in `cuDisc`. By default, ν_0 is calculated assuming a mid-plane temperature of 100 K at 1 AU, but the user can edit this.

3.5 Hydrostatic equilibrium

The 2D gas density structure is calculated by solving the hydrostatic equilibrium equation,

$$\frac{dP}{dZ} = -\frac{GM_*}{(R^2+Z^2)^{\frac{3}{2}}}\rho_g Z. \tag{32}$$

First, we rewrite the equation using the ideal gas law to obtain

$$d\log P = \frac{GM_*}{c_s^2}d\left(\frac{1}{r}\right), \tag{33}$$

where we have utilised $ZdZ/r^3 = dr/r^2 = -d(1/r)$, r being the spherical radius, and c_s is the isothermal sound speed. We then take the exponential of Eqn. 33 and in each cell i, j calculate

$$\frac{P_{i,j}}{P_{i,j-1}} = \exp\left[GM_*\left\langle\frac{1}{c_s^2}\right\rangle_{i,j}\left(\frac{1}{r_{i,j}} - \frac{1}{r_{i,j-1}}\right)\right], \tag{34}$$

where $\left\langle\frac{1}{c_s^2}\right\rangle_{i,j}$ is the average reciprocal of the sound speed squared,

$$\left\langle\frac{1}{c_s^2}\right\rangle_{i,j} = \frac{1}{2}\left[\frac{1}{c^2_{s,i,j}} + \frac{1}{c^2_{s,i,j-1}}\right]. \tag{35}$$

The pressure in each cell relative to the cell at $j = 0$ is then found by multiplicatively scanning over Z,

$$\frac{P_{i,j}}{P_{i,j=0}} = \prod_{k=1}^{j}\frac{P_{i,k}}{P_{i,k-1}}. \tag{36}$$

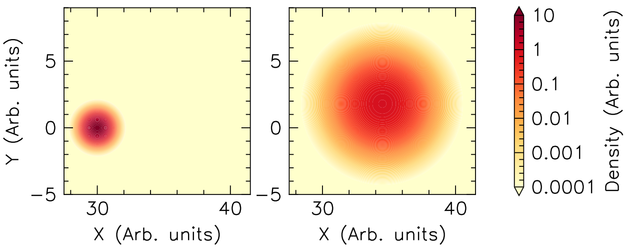

Figure 5. Advection-diffusion of a 2D gaussian pulse using the dynamics routine in `cuDisc`. The left and right panels show the initial and final states, respectively. This simulation had a resolution of 512 × 512 cells.

This method enforces positivity of the calculated pressure. The pressure is then converted to density and normalised via the gas surface density,

$$\rho_{g,i,j} = \frac{\Sigma_{g,i}(RdR)_i}{\sum_k \omega_{i,k}V_{i,k}}\omega_{i,j}, \tag{37}$$

where $\omega_{i,j}$ is the unnormalised density and $(RdR)_i$ is calculated as $(RdR)_i = \frac{1}{2}d(R^2)_i = \frac{1}{2}[(R^e_{i+1})^2 - (R^e_i)^2]$.

3.6 Dynamics Tests

To demonstrate that the code is second-order accurate, we now show test problems run using 2D Gaussian pulses with the analytic form

$$\rho(x,y,t) = \frac{A}{t}\exp\left(-\frac{(x-x_p(t))^2 + (y-y_p(t))^2}{4Dt}\right), \tag{38}$$

where $x_p(t) = x_0 + v_x(t - t_0)$ with an equivalent expression for y_p. For these tests, we used the following parameter choices: $A = D = 1$, $x_0 = 30$, $y_0 = 0$, $t_0 = 0.1$, $v_x = 5$ and $v_y = 2$. These pulses were advected in both R and Z directions whilst undergoing diffusion from $t = t_0$ to $t = 1$. Cells were logarithmically-spaced in R between 10 and 50 and linearly-spaced in θ between $-\pi/6$ and $\pi/6$. Cartesian areas and volumes were used, and the boundaries were set to outflow conditions. An example run can be seen in Fig. 5. The L2 error norm, $\sqrt{\sum_{ij}(\rho_{ij} - \rho_{\mathrm{an},ij})^2/(N\times N)}$, was calculated using the analytic solution, ρ_{an} compared to the computed solution, ρ. Fig. 6 shows how this error decreases with number of cells N on a 2D $N\times N$ grid. As shown, the L2 error is proportional to $1/N^2$, demonstrating that our method is second-order accurate.

To test our code on disc-related problems, we compared simulated steady-state results to analytic results given by Takeuchi & Lin (2002). The gas was initialised using their analytic density and velocity profiles. Three dust species (10 μm, 100 μm and 1 mm) were initialised as being well-mixed with the gas (i.e. the same density and velocity profiles) with a standard dust-to-gas ratio of 0.01 applied to the density. In order to study vertical settling, radial fluxes were set to 0 for comparison with Takeuchi & Lin's analytic profiles, and the code was run for 100,000 years to ensure a steady-state in the dust, as the settling timescale for the smallest grains in this set-up was 90,000 years. 300 cells were used in the vertical dimension. Fig. 7 shows the computed equilibrium dust-to-gas profiles of the dust species compared to the analytic solutions, showing strong agreement. A small discrepancy is visible for the smallest grains at large Z; this is due to our inclusion of the advection of momentum and because we do not make a thin disc approximation in the gravitational potential

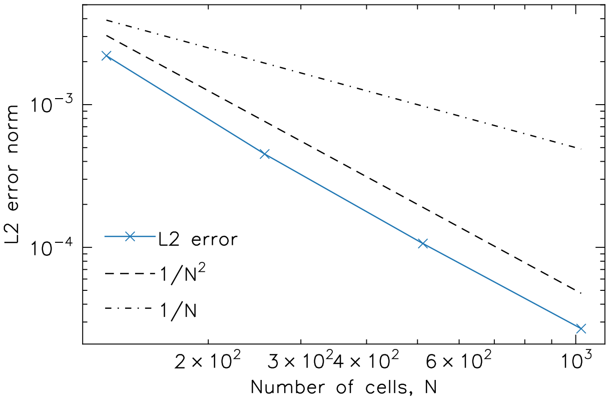

Figure 6. The L2 error norm for advection-diffusion of a 2D gaussian pulse using the dynamics routine in `cuDisc`. Slopes of $1/N$ and $1/N^2$, where the 2D grid has size $N \times N$, are shown for reference.

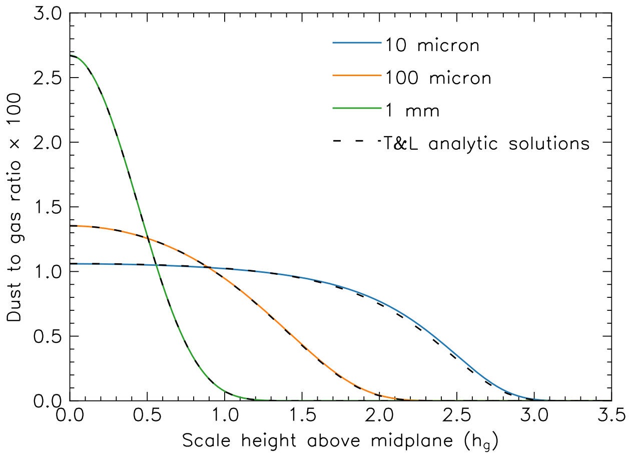

Figure 7. Steady-state dust-to-gas ratios as a function of height above the disc mid-plane (in units of the gas scale height, h_g) at a radius of 10 AU for three dust species. The analytic solutions given by Takeuchi & Lin (2002) are overplotted for comparison.

(Takeuchi & Lin invoke $Z << R$ to simplify the problem). With radial velocities turned on, we also match the analytic profiles for radial velocity as a function of height, as shown in Fig. 8.

4 TEMPERATURE SOLVER

Radiative transfer is a complex problem, and various numerical methods can be employed whose results differ depending on the weights attributed to accuracy versus computation time. For certain problems, calculated temperatures may need to be highly accurate and need not evolve (e.g. comparing line fluxes to observations), allowing long computation times to be a viable option. Other problems may not require such high accuracy in the calculation of temperature but instead may require fast calculation to assess the thermal evolution of a system.

Since the aim of `cuDisc` is to study the evolution of protoplanetary discs over long timescales ($\sim 10^5$ yr or longer), we require a fast temperature solver. For this reason, our method uses a hybrid of ray-tracing and multi-band flux-limited diffusion as opposed to exact techniques such as Monte Carlo. We benchmark our solver against `RADMC-3D` (Dullemond et al. 2012), a Monte Carlo code that requires longer computation times for more accurate results.

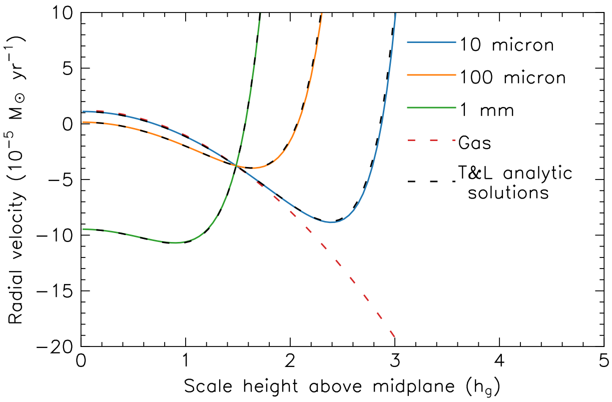

Figure 8. Steady-state radial velocities as a function of height above the disc mid-plane (in units of the gas scale height, h_g) at a radius of 10 AU for three dust species. The analytic solutions given by Takeuchi & Lin (2002) are overplotted for comparison.

4.1 Hybrid ray-tracing + multi-band flux-limited diffusion

In `cuDisc`, we use a hybrid radiative transfer method that employs ray-tracing from the star for heating due to stellar radiation and multi-band flux-limited diffusion (FLD) for treating re-emitted/scattered radiation. FLD is a model developed by Levermore & Pomraning (1981) that aims to solve the radiative transfer problem by treating the transport of radiative energy as a diffusion problem. The diffusive flux is limited to ensure that radiation does not propagate faster than the speed of light. The limiter is chosen to tend towards the correct physical forms in limits of high and low optical depth. The following scheme used in `cuDisc` is based on similar schemes detailed by Kuiper et al. (2010); Commerçon et al. (2011); Bitsch et al. (2013). In conservative form, the equations governing the internal energy density, ϵ, and radiative energy density in a given wavelength band b, E_R^b, are given by

$$\frac{\partial \epsilon}{\partial t} = -\sum_b \left(\sum_s \rho_s \kappa_{P,s}^b\right)\left(f^b(T)B(T) - cE_R^b\right) + S_{\rm heat}, \qquad (39)$$

$$\frac{\partial E_R^b}{\partial t} + \nabla \cdot \boldsymbol{F}^b = \left(\sum_s \rho_s \kappa_{P,s}^b\right)\left(f^b(T)B(T) - cE_R^b\right) + S_{\rm sca}^b, \qquad (40)$$

where ρ_s is the mass density of species s, $\kappa_{P,s}^b$ is the Planck mean opacity in wavelength band b for species s, $B(T) = 4\sigma T^4$ accounts for radiative cooling, $S_{\rm heat}$ accounts for heating via stellar irradiation and viscous processes and $S_{\rm sca}^b$ is the scattered stellar radiation. As Eqn. 40 is for a particular wavelength band, a Planck factor f^b is included that accounts for the fraction of total radiated thermal energy that is emitted in the particular band in question. We do not include advective terms in our formulation. $\boldsymbol{F}^b$ is the flux of radiative energy in a given band, and this term is calculated in flux-limited diffusion as

$$\boldsymbol{F}^b = -\frac{c\lambda^b}{\rho\kappa_R^b}\nabla E_R^b, \qquad (41)$$

where κ_R^b is the Rosseland mean opacity in the given band and λ^b the flux-limiter. In our multi-band scheme, the Planck opacities in Eqns. 39 & 40 are calculated using absorption opacities, whilst the Rosseland opacity is calculated using the total opacity, absorption plus scattering. The choice of flux-limiter used for `cuDisc` is that given by Kley (1989),

$$\lambda^b = \begin{cases} \dfrac{2}{3 + \sqrt{9 + 10R_b^2}}, & \text{for } R_b < 2 \\[2ex] \dfrac{10}{9 + 10R_b + \sqrt{81 + 180R_b}}, & \text{for } R_b > 2 \end{cases} \tag{42}$$

where

$$R_b = \frac{1}{\rho\kappa_R^b} \frac{|\nabla E_R^b|}{E_R^b}. \tag{43}$$

Taking the limits of high and low optical depth ($\rho\kappa_R \gg 1$ & $\rho\kappa_R \ll 1$) the flux tends to the diffusion limit, $\boldsymbol{F} = -(c/3\rho\kappa_R)\nabla E_R$, and the free-streaming limit, $\boldsymbol{F} = -cE_R(\nabla E_R/|\nabla E_R|)$, respectively.

The heating term $S_{\rm heat}$ is split into stellar heating and viscous heating. The viscous heating is calculated explicitly as

$$S_{\rm visc} = \frac{9}{4}\alpha\rho_g c_s^2 \Omega, \tag{44}$$

where α parameterises the effective turbulent viscosity of the disc (Shakura & Sunyaev 1973), c_s is the isothermal sound speed, and Ω is the Keplerian angular velocity. Stellar heating in each cell is calculated via ray tracing from the star using

$$S_* = \sum_B \frac{L_B}{4\pi r^2 \Delta r} e^{-\tau_B}(1 - e^{-\rho\kappa_B\Delta r})(1 - a_B), \tag{45}$$

where the sum is over the wavelength bands used for stellar radiation, L_B is the stellar luminosity in each band, and Δr is the (spherical) radial length of the cell. By default, the number of bands used for stellar heating (∼ 100) is larger than the number used for the FLD routine detailed above (∼ 20) as the heating calculation is computationally cheap. For each wavelength band, the $e^{-\tau_B}$ term accounts for the flux that has been attenuated (removed through extinction; i.e. absorption plus scattering opacity) by the material along the radial path between the star and the cell of interest, whilst $(1 - e^{-\rho\kappa_B\Delta r})$ accounts for the amount of remaining flux that is attenuated in the cell. κ_B is the extinction of stellar radiation at the centre of each wavelength band. The term $(1 - a_B)$ includes the albedo, a_B, to take into account the effect of scattering. The albedo is calculated as the ratio of scattering opacity to total extinction at each wavelength. The diffusive flux generated by scattering at each wavelength is therefore given by

$$S_{{\rm sca},B} = \frac{L_B}{4\pi r^2 \Delta r} e^{-\tau_B}(1 - e^{-\rho\kappa_B\Delta r}) a_B. \tag{46}$$

This is then binned into the wavelength bins used for the FLD routine and included in Eqn. 40.

The internal energy is given by $\epsilon = \rho c_V T$, where ρ is the total density of all species and c_V is the specific heat at constant volume, meaning that Eqn. 39 can be written in terms of B,

$$\frac{\rho c_V}{16\sigma T^3}\frac{\partial B(T)}{\partial t} = -\rho\kappa_P(B(T) - cE_R) + S. \tag{47}$$

This allows the system of equations to be set up and solved implicitly for the quantities B and E_R. This method ensures the correct equilibrium is reached for large Δt - this being the standard case for our simulations. We also note that the Commerçon et al. (2011) linearization writes $B(T)$ in terms of T, instead of T in terms of $B(T)$ as we do. The discretised forms of Eqns. 47 & 40 for cell i, j at time-step n are

$$\begin{aligned} &\frac{\rho_{i,j}^n c_V}{16\sigma(T_{i,j}^n)^3\Delta t} B_{i,j}^{n+1} V_{i,j} \\ &\quad + \sum_b \left(\sum_s \rho_{s,i,j}^n \kappa_{P,s,i,j}^b\right)\left(f_{i,j}^b B_{i,j}^{n+1} - cE_{R,i,j}^{b,n+1}\right) V_{i,j} \\ &\qquad = \frac{\rho_{i,j}^n c_V}{16\sigma(T_{i,j}^n)^3\Delta t} B_{i,j}^n V_{i,j} + S_{{\rm heat},i,j}^n V_{i,j}, \end{aligned} \tag{48}$$

and

$$\begin{aligned} &\frac{E_{R,i,j}^{b,n+1}}{\Delta t} V_{i,j} - \left(\sum_s \rho_{s,i,j}^n \kappa_{P,s,i,j}\right)\left(f_{i,j}^b B_{i,j}^{n+1} - cE_{R,i,j}^{b,n+1}\right) V_{i,j} + \\ &\quad \sum_{k=i-1}^{i+1} \sum_{l=j-1}^{j+1} \tilde{D}_{kl}^b E_{R,k,l}^{b,n+1} = \frac{E_{R,i,j}^{b,n}}{\Delta t} V_{i,j} + S_{{\rm sca},i,j}^{b,n} V_{i,j}. \end{aligned} \tag{49}$$

$\tilde{D}_{kl}^b$ is the diffusion matrix, constructed by summing all terms that apply to cell k, l when finding $\tilde{F}_\sigma A_\sigma$ (Eqn. 21) for each of the four interfaces around cell i, j. The construction of the elements of $\tilde{D}_{kl}$ can be found in more detail in Appendix A. We time-lag the diffusion constant in Eqn. 41 and the Planck factor, f^b (i.e. they are evaluated at t^n instead of t^{n+1}). The system of equations defined by Eqns. 48 & 49 are solved using incomplete-LU preconditioning and the Bi-Conjugate Gradient Stabilized (BiCGStab) solver described in Naumov (2011). The convergence criterion is set by comparing the maximum fractional residual between the left and right-hand sides of each equation at each iteration to a user-controlled relative tolerance, set by default to 10^{-4}.

4.2 Opacities

Dust absorption and scattering opacities can be set in various ways in `cuDisc`. Simple, functional forms that approximate the opacity dependence on grain size and wavelength may be used, or opacity tables generated using packages such as the DSHARP opacity library (Birnstiel et al. 2018) can be read in. If tables are used, `cuDisc` can interpolate the opacity data to the user-defined grain sizes and wavelength bins using piecewise-cubic Hermite interpolation (Fritsch & Carlson 1980) in both grain size and wavelength space. In general, more wavelength bins are used for the cheaper stellar heating calculation in Eqn. 45 than are used in the FLD scheme. For conversion between these two wavelength grids, the opacities are binned by calculating the Planck mean opacity over the range of wavelengths corresponding to each coarse bin - these partial means result in a better estimate of the temperature in optically thin regions when a small number of bins are used.

4.3 Radiative transfer tests

To test our temperature solver, we ran modified versions of standard benchmark tests used in the literature for comparing temperature solvers: the Pascucci et al. (2004) and Pinte et al. (2009) tests. Fig. 9 shows the comparison of mid-plane and surface temperatures calculated using `cuDisc` and the Monte Carlo radiative transfer code

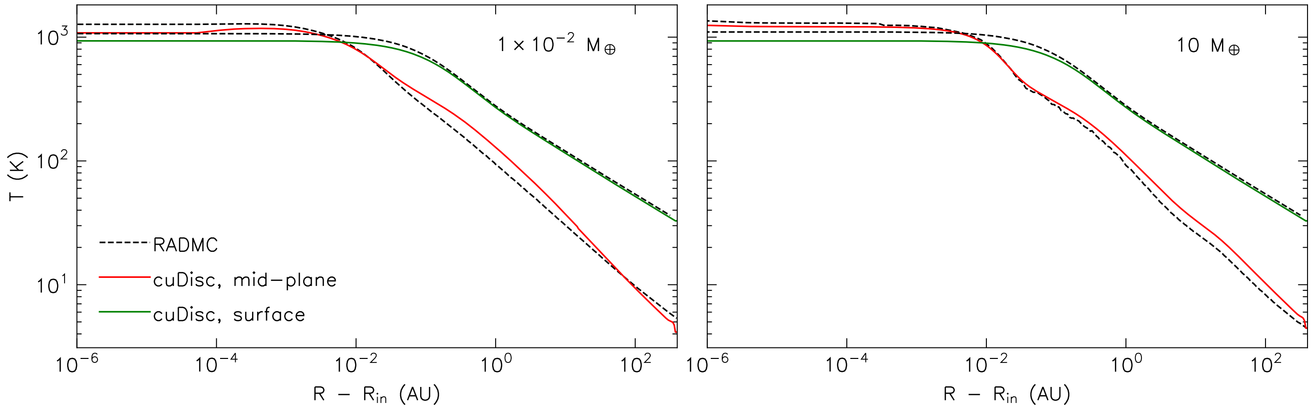

Figure 9. Comparison of mid-plane and surface temperatures calculated by `cuDisc` and `RADMC-3D` for the Pinte et al. (2009) benchmark adapted to have a grain distribution. The x-axis is plotted with units of cylindrical radius minus the inner disc radius (0.1 AU). The left and right plots show models with total dust masses of 1×10^{-2} $M_\oplus$ and 10 $M_\oplus$, respectively.

`RADMC-3D` for the Pinte test, modified to include a distribution of grain sizes. The density of each grain species i was set by

$$\rho_i(R,Z) = F_i \frac{\Sigma_{\rm tot}(R)}{\sqrt{2\pi}h_i(R)} \exp\left(-\frac{Z^2}{2h_i^2(R)}\right), \qquad (50)$$

where F_i is the fraction of the total surface density $\Sigma_{\rm tot}$ distributed to each grain size and h_i the scale height for each grain. The grain size distribution was set according to the MRN (Mathis et al. 1977) distribution with a maximum grain size of 0.5 cm, and the total surface density was set according to Pinte et al. (2009) as

$$\Sigma_{\rm tot} = \Sigma_0 (R/{\rm AU})^{-1.625}, \qquad (51)$$

where Σ_0 was set by the total desired disc mass. The scale height for each grain size was set as the approximate height reached once grains are settled (Dubrulle et al. 1995),

$$h_i = h_g \sqrt{\frac{1}{1+{\rm St}_i/\alpha}}, \qquad (52)$$

where h_g is the gas scale height, ${\rm St}_i$ is the grain Stokes number, set simply in this problem as ${\rm St}_i = 0.05 a_i (R/{\rm AU})$ where a_i is the grain radius, and α is the gas turbulence parameter, set to 1×10^{-3}. The gas scale height was set according to Pinte et al. (2009) as $h_g = 10(R/100~{\rm AU})^{1.125}$ AU. The opacities of the dust grains were set using the Birnstiel et al. (2018) tool for calculating opacities, together with the dielectric constants given by Draine (2003). Scattering was treated as being isotropic as `cuDisc` is currently unable to treat anisotropic scattering. Fig. 9 shows the results for two different total disc masses (1×10^{-2} & 10 $M_\oplus$). These masses corresponded to mid-plane optical depths from the star to the observer at 0.81 μm of 87 and 8.7×10^4, respectively. These values are just over an order of magnitude smaller than the optical depths for the Pinte test at the same disc masses, as our use of a grain distribution lowers the overall opacity of the disc due to the presence of large grains. The inner and outer bounds of the grids were set to (0.1, 400) AU in R and (0, $\pi/4$) rad in θ. 150 equally-spaced cells were used in θ, whilst the number of cells in R was adapted depending on the disc mass. For both disc masses, 200 logarithmically-spaced cells were used between 0.15 & 400 AU, whilst to minimise the optical depth in cells very close to the inner edge, we follow the method detailed by Ramsey & Dullemond (2015) and use 40 & 96 logarithmically-spaced cells between 0.1 & 0.15 AU for the lower & higher disc mass respectively. 100 logarithmically-spaced wavelengths between 0.1 & 3000 μm were used for stellar heating and binned down to 20 bands for the FLD calculations. Apart from the binned wavelength bands (which are not required), these same grid structures were used in `RADMC-3D` for consistency. Resolution tests were performed, which confirmed that convergence was reached for the grid resolutions quoted here.

We find that our results in the optically thin surface layers are accurate to within a few percent for the entirety of the disc apart from the very inner region between 0.1 and 0.2 AU. At the mid-plane, the accuracy varies but remains within ~20 % for most of the disc - a level of accuracy high enough for our problems.

5 DUST GROWTH AND FRAGMENTATION

Dust coagulation is performed using the method outlined in Brauer et al. (2008) but without the vertical integration that they use to convert the problem to 1D along the disc mid-plane. In this method, the Smoluchowski coagulation equation (Smoluchowski 1916) becomes

$$\dot{\rho}_k = \sum_{i=0}^{N-1}\sum_{j=0}^{i} \dot{\rho}_{{\rm gain},ijk} - \sum_{i=0}^{N-1} \dot{\rho}_{{\rm loss},ik} \qquad (53)$$

where dots represent time derivatives, N is the total number of dust species, ρ_k is the density of the k-th dust species, and $\dot{\rho}_{{\rm gain},ijk}$ and $\dot{\rho}_{{\rm loss},ik}$ are the gains in density due to collision products of other grains (through coagulation and fragmentation) and losses in density due to collisions between the k-th dust species and all others respectively. The loss is given by

$$\dot{\rho}_{{\rm loss},ik} = m_k n_i n_k K_{ik} (p_{{\rm coag},ij} + p_{{\rm frag},ij}) \qquad (54)$$

where m_k is the mass of the k-th grain, K_{ij} is referred to as the kernel, and $p_{{\rm coag},ij}$ and $p_{{\rm frag},ij}$ are the probabilities of coagulation and fragmentation, which follow

$$p_{{\rm coag},ij} + p_{{\rm frag},ij} = 1 \qquad (55)$$

in the absence of bouncing (as assumed here). The form of this kernel is taken from Birnstiel et al. (2010) and is given by

$$K_{ij} = \sigma_{ij} \Delta v_{ij}, \qquad (56)$$

where σ_{ij} is the collision cross-section of the grains and Δv_{ij} is the relative velocity of the grains, given by

$$\Delta v_{ij}^2 = v_{\mathrm{BM},ij}^2 + v_{\mathrm{turb},ij}^2 + (\Delta v_{\mathrm{lam},ij})^2, \tag{57}$$

where $v_{\mathrm{BM},ij}$, $v_{\mathrm{turb},ij}$, and $\Delta v_{\mathrm{lam},ij}$ are the relative velocities due to Brownian motion, gas turbulence, and laminar dust motion respectively. Here $\Delta v_{\mathrm{lam},ij}$ is the root-mean-square relative velocity between the dust particles, calculated using the velocities determined by the dynamical solver. Brownian motion most strongly affects small grains and is given by

$$v_{\mathrm{BM},ij} = \sqrt{\frac{8k_B T(m_i + m_j)}{\pi m_i m_j}}, \tag{58}$$

where k_B is the Boltzmann constant and T the disc temperature. The form of $v_{\mathrm{turb},ij}$ is taken from Ormel & Cuzzi (2007) and is given by

$$v_{\mathrm{turb},ij} = \sqrt{\alpha} c_s V_{\mathrm{rel},ij}, \tag{59}$$

where $V_{\mathrm{rel},ij}$ is the relative velocity between grains scaled to the eddy velocity $\sqrt{\alpha}c_s$,

$$V_{\mathrm{rel},ij} = \sqrt{\Delta V_{\mathrm{I}}^2 + \Delta V_{\mathrm{II}}^2}, \tag{60}$$

where ΔV_{I}^2 & ΔV_{II}^2 are the relative velocities induced by “slow” class I and “fast” class II eddies, respectively given by Eqns. 17 & 18 in Ormel & Cuzzi (2007). For computational speed, we use a quadratic fit to find the Stokes number of the boundary between class I & II eddies for a given pair of grains, St* (Eqn. 21d in Ormel & Cuzzi (2007)). This fit is accurate to within 2%.

In a collision, grains can coagulate to form more massive grains or fragment to produce a remnant grain and a set of fragments that are distributed over lower-mass grains. Combining all of these collision products gives us the density gain for a dust species as

$$\dot{\rho}_{\mathrm{gain},ijk} = n_i n_j K_{ij}\left[p_{\mathrm{coag},ij}(m_i+m_j)C_{ijk} \right. \\ \left. + p_{\mathrm{frag},ij}(m_{\mathrm{rem},ij}R_{ijk} + m_{\mathrm{frag},ij}F_{ijk})\right], \tag{61}$$

where C_{ijk}, R_{ijk} and F_{ijk} are the fractions of the mass associated with coagulation products, remnant products and fragment products that are distributed into bin k, and $m_{\mathrm{rem},ij}$ and $m_{\mathrm{frag},ij}$ are the masses of remnants and fragments produced by the collision. Note to avoid double counting we replace $K_{ij} \to \frac{1}{2}K_{ij}$ when $i = j$.

The particle velocities are assumed to follow a Maxwell-Boltzmann distribution around Δv_{ij} to calculate the fragmentation probability. This is a simplified version of the calculations done by Garaud et al. (2013), who separated the stochastic (turbulent and Brownian) motions from the laminar motions; we are therefore assuming that stochastic motions are larger than any laminar relative motions. This is also the approach taken by Stammler & Birnstiel (2022). In this way, the fragmentation probability is given by

$$p_{\mathrm{frag},ij} = \left[\frac{3}{2}\left(\frac{v_{\mathrm{frag}}}{\Delta v_{ij}}\right)^2 + 1\right] \exp\left[-\frac{3}{2}\left(\frac{v_{\mathrm{frag}}}{\Delta v_{ij}}\right)^2\right]. \tag{62}$$

The coefficients C_{ijk}, R_{ijk} and F_{ijk} add the collision products into the appropriate mass bins. Since our standard choice of the mass grid is logarithmic, these products do not map directly to other mass bins and must be split over the two bins on either side of the product mass. Defining the integers $\ell(m)$ and $u(m)$ as: $m_{\ell(m)}$ being the largest mass such that $m_{\ell(m)} < m$ and $m_{u(m)}$ being the smallest mass such that $m_{u(m)} > m$, we may write

$$C_{ijk} = \begin{cases} \varepsilon(m_i+m_j) & k = \ell(m_i+m_j) \\ 1-\varepsilon(m_i+m_j) & k = u(m_i+m_j) \\ 0 & \text{otherwise,} \end{cases} \tag{63}$$

and

$$R_{ijk} = \begin{cases} \varepsilon(m_{\mathrm{rem},ij}) & k = \ell(m_{\mathrm{rem},ij}) \\ 1-\varepsilon(m_{\mathrm{rem},ij}) & k = u(m_{\mathrm{rem},ij}) \\ 0 & \text{otherwise,} \end{cases} \tag{64}$$

where

$$\varepsilon(m) = \frac{m_{u(m)} - m}{m_{u(m)} - m_{\ell(m)}}. \tag{65}$$

The distribution of fragments, F_{ijk}, is calculated following Rafikov et al. (2020). First, we assume that the number density distribution of fragments is given by a power law,

$$n(m)dm \propto m^{-\eta}dm \tag{66}$$

where $n(m)dm$ is the number of particles per unit volume in the mass range $[m, m+dm]$ and η is the fragmentation parameter that controls how fragments are distributed amongst the smaller mass grains; by default, we take the standard value of 11/6 found in the steady-state solutions of Dohnanyi (1969); Tanaka et al. (1996). Converting to mass density and integrating over the bins to find the fraction of mass distributed into each bin gives

$$N_k(m_{\max}) = \frac{\left(m_{k+1}^e\right)^{2-\eta} - \left(m_k^e\right)^{2-\eta}}{m_{\max}^{2-\eta} - \left(m_0^e\right)^{2-\eta}}, \tag{67}$$

where $m_{\max}$ is the mass of the largest fragment and where m_{k+1}^e and m_k^e are the upper and lower edges of the k^{th} mass bin.

We then place each $m_{\mathrm{frag},ij}$ into the bin l by finding the smallest l such that $m_{l+1}^e > m_{\mathrm{frag},ij}$ and setting $m_{\max,ij} = m_{l+1}^e$. From this we arrive at $F_{ijk} = N_k(m_{\max,ij})$. However, F_{ijk} is not used directly. Instead, we first determine the total amount of fragments falling into each l-bin before distributing them according to $N_k(m_{l+1}^e)$. This results in a much more efficient algorithm, as outlined in Rafikov et al. (2020).

The proportion of mass ending up as fragments is defined as a multiple of the mass of the smaller of the two colliding grains. Denoting the more massive particle as the target with mass, m_{tar} and the smaller as the impactor with mass, m_{im}, we define the χ_{im} such that $m_{\mathrm{frag},ij} = \min(\chi_{\mathrm{im}}m_{\mathrm{im}}, m_{\mathrm{tar}})$. The remnant mass is then given by

$$m_{\mathrm{rem}} = \begin{cases} m_{\mathrm{tar}} - \chi_{\mathrm{im}}m_{\mathrm{im}} & \text{for } \chi_{\mathrm{im}}m_{\mathrm{im}} < m_{\mathrm{tar}} \\ 0 & \text{otherwise,} \end{cases} \tag{68}$$

where χ_{im} is a factor controlling the amount of material the impactor can remove relative to its mass, usually taken to be unity. The fragment mass is then given by the total mass of the colliding grains minus the remnant mass.

To integrate Eqn. 53, `cuDisc` employs the Bogacki-Shampine 3(2) embedded Runge-Kutta method with error estimation (Bogacki & Shampine 1989) that adaptively calculates the time-step required to meet user-specified relative and absolute tolerances. As default, these are a 1% relative tolerance and an absolute tolerance of 1×10^{-10} multiplied by the sum over all grain densities in a cell.

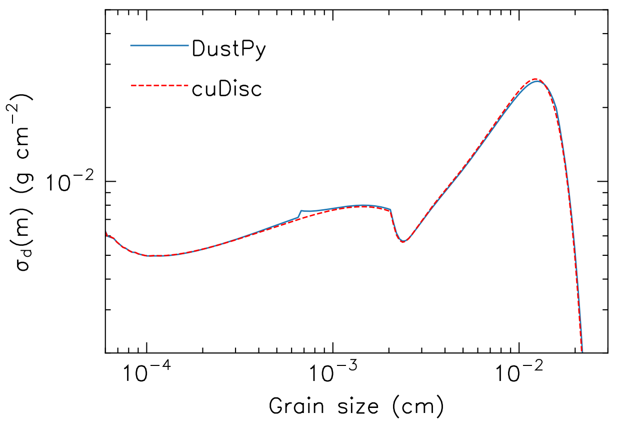

Figure 10. Comparison of steady-state dust grain size distributions calculated by `cuDisc` and `DustPy`. The simulation parameters used were R = 20 AU, T = 36 K, μ = 2.4, $\alpha = 10^{-3}$ and $v_{\rm frag}$ = 100 cm s^{-1}.

5.1 Coagulation tests

To test our coagulation routine, we produced direct comparisons to the 1D disc evolution code `DustPy` using a vertically integrated kernel (Birnstiel et al. 2010), which can be seen in Fig. 10. The vertically integrated kernel is used to study the growth and fragmentation of dust surface densities instead of volume densities and is constructed by dividing the coagulation kernel (Eqn. 56) by $\sqrt{2\pi(h_i^2 + h_j^2)}$, where h_i & h_j are the scale heights of the dust grains, assuming a Gaussian vertical density profile. The relative vertical velocity is set according to the difference in terminal settling velocities of the dust grains in question, taking the dust grain scale height as its representative height Z above the mid-plane. Each simulation was run with dynamics turned off, at a single radius in the disc, with the same initial conditions. The simulations were run until a steady-state was reached in the dust grain size distribution. To compare the two codes, we plot the mass-grid independent densities for each grain, given by

$$\sigma_d(m) = m\frac{\partial \Sigma_d}{\partial m}. \qquad (69)$$

200 grain species were used for both simulations, as this is comparable to the typical amount used for science calculations. The two codes show strong agreement. The only slight difference at $\sim 7 \times 10^{-4}$ cm is due to different methods used by the two codes to fit the relative turbulent velocities described in Ormel & Cuzzi (2007). The difference occurs because the approximation used in `DustPy` is not continuous in this region.

We also ran tests using a constant kernel, the results of which were consistent with the analytic solutions.

6 CODE STRUCTURE

The overall time evolution is controlled by the CFL condition on the dust dynamics. Gas, temperature and coagulation updates are then performed when required depending on the evolution time-scales associated with each physical process. A schematic of the code structure can be seen in Fig. 11. By default, gas surface density updates and subsequent hydrostatic equilibrium calculations are also performed at each CFL time-step due to their low computational cost. For temperature calculations, an update is performed on the first time-step of the simulation, and the percentage change with respect to the initial state is calculated. How often the temperature solver should be employed is then controlled by choosing a percentage tolerance for temperature updates that allows the code to calculate the approximate time period that can be allowed to elapse before the next update is required; by default, this is set to 0.1%. However, this tolerance should be chosen depending on the expected time-scales of temperature evolution for each specific problem. Coagulation updates are managed differently, given that the coagulation routine is explicit and, as such, performs its own sub-steps within each global time-step (i.e. sub-cycling). To control how often these updates are performed, the final sub-time-step within the coagulation routine is extracted and used to compare to global time-steps. By default, the coagulation routine is triggered when one sub-time-step taken from the last coagulation step has elapsed in global simulation time; however, this can be changed by the user.

7 GRAIN GROWTH IN A STEADY-STATE TRANSITION DISC

To show a simple science test case and compare it to other works, we ran simulations of a disc with an inner hole evolving towards a steady-state, representative of a "transition" disc structure (e.g. Owen 2016). To compare to `DustPy`, one simulation was run with a vertically isothermal temperature profile set to the mid-plane temperature adopted by `DustPy`, whilst one had the 2D temperature solver switched on. We will refer to these simulations as isothermal and non-isothermal from here on. The gas surface density was set to have a mass within 100 AU of 0.017 $M_\odot$ with a power law radial dependence and a sharp exponential cut off at 5 AU; i.e. $\Sigma_g = \Sigma_0 R_{\rm AU}^{-1} \exp\left[-(5/R_{\rm AU})^{10}\right]$. For these tests, the gas surface density was not evolved; however, the vertical gas density profile was updated for the non-isothermal test to maintain hydrostatic equilibrium. Radial gas velocities were set to 0, whilst azimuthal velocities were updated with the temperature to maintain force balance. 150 logarithmically-spaced cells were used in R between 3 & 20 AU, and a total of 150 cells were used in θ between 0 & $\pi/4$, with 92 linearly-spaced cells between 0 & $\pi/9$ with double the resolution of the remaining linearly-spaced 58 cells. This sub-division was used to enhance the resolution at the mid-plane. The viscosity, ν, was set according to Shakura & Sunyaev (1973) via $\alpha c_s H$. The parameters chosen for each of the simulations are given in Table 1. The stellar radiation was assumed to be a blackbody at the effective temperature given in Table 1. Note that simulations were run for fragmentation velocities of 100 and 1000 cm s^{-1}. For the 100 cm s^{-1} runs, 100 dust species with grain sizes logarithmically spaced between 0.1 μm and 0.5 cm were initialised with a size distribution according to the MRN distribution (Mathis et al. 1977) and spatially distributed as being well-mixed with the gas at a local dust-to-gas ratio of 0.01. For the 1000 cm s^{-1} runs, the mass grid was adjusted to 135 logarithmically spaced grain sizes between 0.1 μm and 20 cm to account for the larger grains present. 100 & 135 mass bins were used respectively to give m_{n+1}/m_n = 1.38 for both sets of simulations. This choice maintains > 7 bins per mass decade, a requirement for accuracy in the coagulation routine (Ohtsuki et al. 1990). For these calculations, changes in dust composition throughout the disc due to sublimation were neglected. In this section we discuss the 100 cm s^{-1} runs, whilst the 1000 cm s^{-1} runs are discussed in Section 7.2.

Fig. 12 shows the 2D dust density profiles of three grain sizes

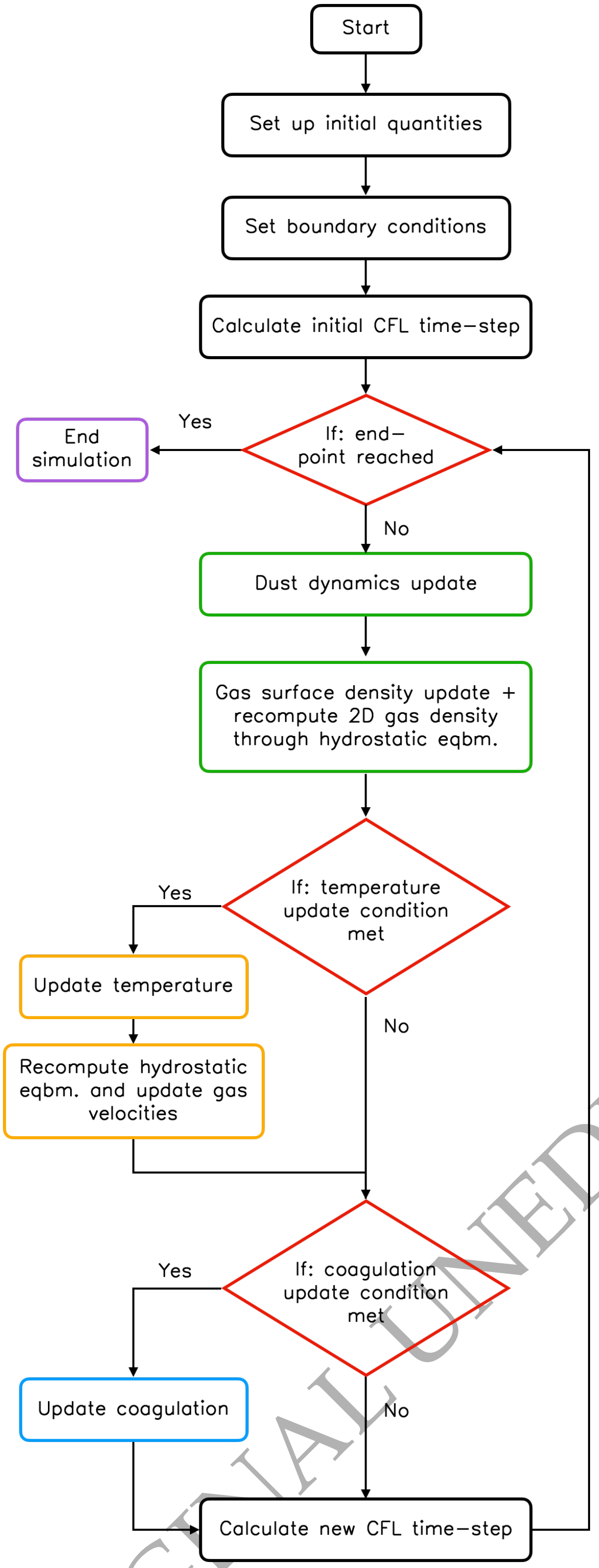
Start
Set up initial quantities
Set boundary conditions
Calculate initial CFL time−step
If: end−point reached
Yes
End simulation
No
Dust dynamics update
Gas surface density update + recompute 2D gas density through hydrostatic eqbm.
If: temperature update condition met
Yes
Update temperature
Recompute hydrostatic eqbm. and update gas velocities
No
If: coagulation update condition met
Yes
Update coagulation
No
Calculate new CFL time−step

Figure 11. The schematic for `cuDisc`. Dust & gas dynamics are updated every time-step whilst temperature and coagulation updates are performed when certain conditions are met. These conditions can be controlled by the user in order to align with the timescales relevant to the physical problem at hand.

Parameter	Value
α	10^{-3}
μ	2.4
$v_{\rm frag}$	100, 1000 cm s^{-1}
M_*	1 $M_\odot$
R_*	1.7 $R_\odot$
$T_{*,\rm eff}$	4500 K
$M_{\rm disc}$	0.017 $M_\odot$
Σ_0	250 g cm^{-2}
ρ_m	1.6 g cm^{-3}
Dust opacities	DSHARP mix (Birnstiel et al. 2018)

Table 1. Parameters used for the transition disc simulations detailed in Section 7.

after 1 Myr of evolution, at which time the simulations had reached a steady state, for both the isothermal disc and non-isothermal disc, whilst Fig. 13 shows the temperature profile of the non-isothermal disc after 1 Myr. Dust settling is apparent from the stratification of the different grains. The non-isothermal disc also exhibits an increase in the proportion of large grains due to the cool interior that allows for the coagulation of grains up to larger sizes. This is because the strength of the gas turbulent velocity is proportional to the sound speed when assuming a Shakura & Sunyaev (1973) viscosity, and reduced turbulent velocities allow particles to grow larger before reaching the fragmentation limit. The temperature structure of the non-isothermal disc also exhibits a super-heated surface layer with temperatures greater than the blackbody equilibrium temperature; this is due to the small grains that occupy the upper regions of the disc whose opacities mean they are good absorbers of stellar optical photons, but bad emitters of their own infra-red photons (see e.g. Chiang & Goldreich 1997). Above the super-heated layer, the dust densities drop to the floor value and the temperature is set by the opacities given to the gas. By default, these opacities are set as the dust-to-gas ratio floor value multiplied by the opacities of the smallest dust grains, as this leads to a smooth temperature transition in the upper regions of the disc. In reality, the temperature in the gaseous atmosphere above the dust disc is controlled by other processes (see e.g. Woitke et al. 2009). In principle, these processes can be included in the framework if necessary.

Figs. 14 & 15 compare the vertically integrated dust grain size distributions for the different simulations at a radius of 6 AU and for the entire disc, respectively. Vertically-integrated mass-grid independent densities were calculated from the volume densities computed by `cuDisc` through

$$\sigma_d(m) = \int_{-\infty}^{\infty} m \frac{\partial \rho_d}{\partial m} dz. \tag{70}$$

Fig. 15 shows that the spatial distribution of dust is very similar for all runs; the maximum grain size is set by fragmentation throughout the entire disc due to the fairly low choice of 100 cm s^{-1} as the fragmentation velocity. The non-isothermal run is slightly more peaked at 6 AU. This is because the equilibrium spatial distribution is set by the balance between diffusion and radial drift, and, as previously mentioned, the non-isothermal disc has lower turbulence and, therefore, less diffusion of the dust out of the dust trap. Fig. 14 shows that the simulations follow very similar distributions up to $\sim 30\mu$m, with the non-isothermal run having an overall higher density across all grain sizes due to the increased surface density at the dust trap. The `DustPy` simulation exhibits a much more prominent trough at

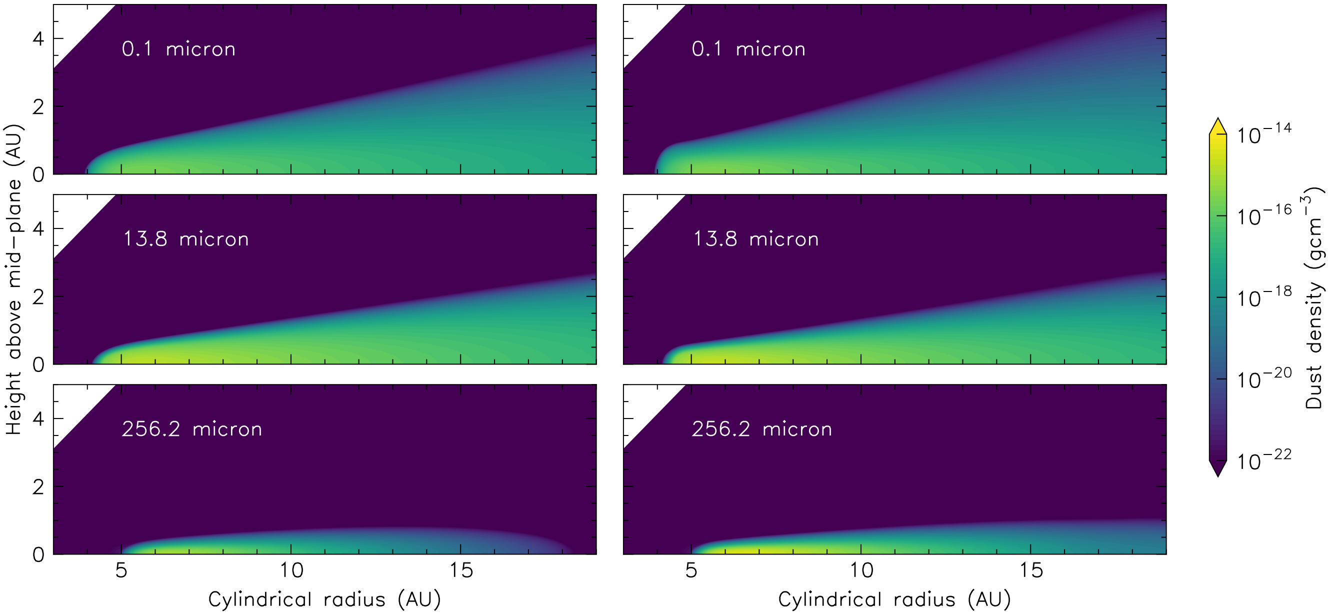

Figure 12. Dust profiles for three grain sizes after 1 Myr of evolution for a vertically isothermal disc (left) and a disc with the 2D temperature solver switched on (right). These runs were set with $v_{\rm frag}$ = 100 cm s^{-1}.

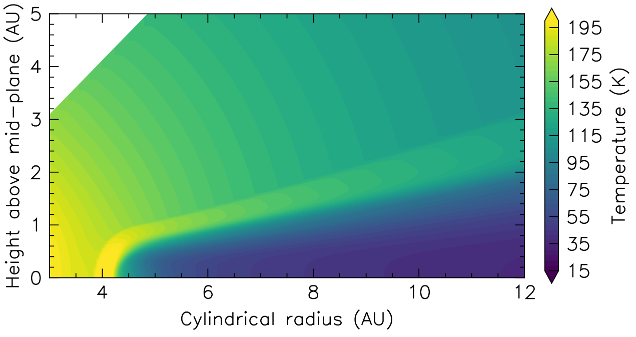

Figure 13. Temperature profile after 1 Myr of evolution for a disc with the 2D temperature solver switched on and $v_{\rm frag}$ = 100 cm s^{-1}.

30 - 40 μm than the `cuDisc` simulations, and the upper cut-off in the size distribution differ across the simulations; the `DustPy` and non-isothermal runs show similar cut-offs whilst the isothermal run is lower.

7.1 Vertical structure comparison

To compare the vertical structure, we converted the dust surface densities from `DustPy` into 2D volume densities by calculating the diffusion-settling equilibrium in the vertical axis for each dust grain (see Takeuchi & Lin 2002), as might be done when taking a `DustPy` output and calculating a radiative transfer simulation. Fig. 16 shows the grain size distributions at 1, 2 and 3 gas scale heights, where the gas scale height is calculated from the mid-plane temperatures. This means the scale heights are the same for the isothermal and `DustPy` runs, but lower for the non-isothermal run given the cooler interior temperature. It can now be seen that the upper cut-off grain size for the isothermal and `DustPy` runs are very similar at the mid-plane but differ at increasing height. This arises because the fragmentation limit sets the maximum grain size and the fragmentation limit is proportional to gas density. The density decreases as we move away from the mid-plane in the 2D `cuDisc` runs, leading to larger Stokes numbers and, therefore, larger turbulent velocities. This lowers the fragmentation threshold on grain size. After vertical integration, this leads to the lower cut-off grain size we see in Fig. 14. For the non-isothermal simulation, we see a similar result but with the cut-off at larger sizes due to the cooler interior. After vertical integration, this leads to the larger maximum grain size seen in Fig. 14. There are also fewer particles at high altitudes for the same reason: the cooler temperature lowers the scale height of the disc, bringing the dust closer to the mid-plane.

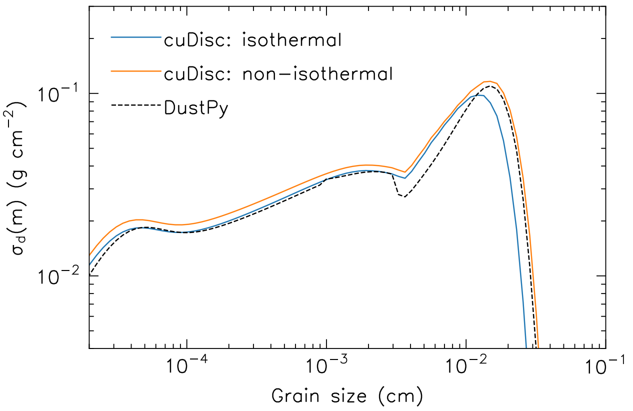

Figure 14. Vertically-integrated mass-grid independent dust densities at a radius of 6 AU for each $v_{\rm frag}$ = 100 cm s^{-1} simulation.

The trough in the grain size distribution arises at the particle size

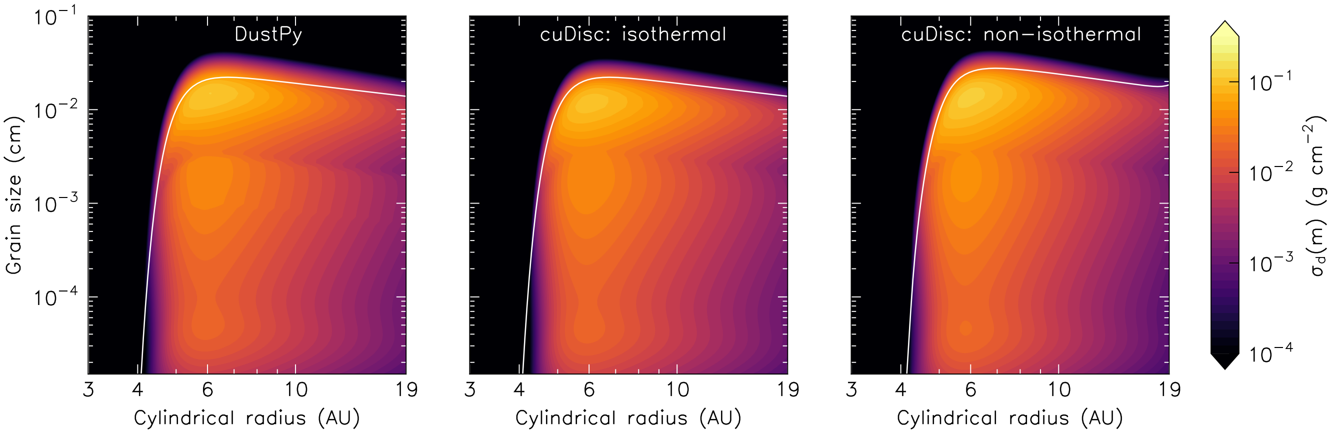

Figure 15. Dust grain size distributions as a function of disc radius for each $v_{\rm frag}$ = 100 cm s^{-1} simulation. The fragmentation limit (given by Birnstiel et al. 2012) is over-plotted on each distribution for reference.

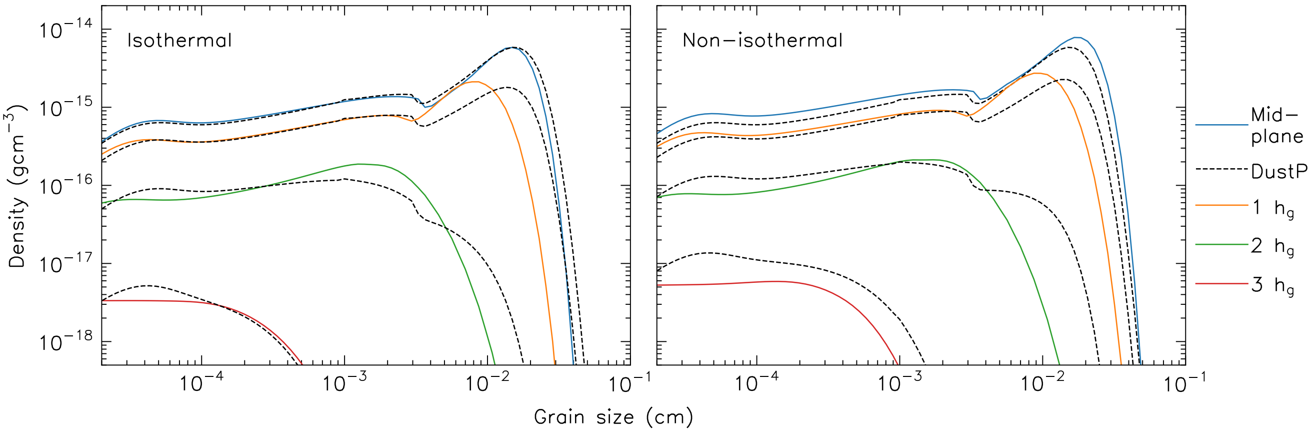

Figure 16. Dust grain size distributions at 1, 2 and 3 gas scale heights above the mid-plane for each $v_{\rm frag}$ = 100 cm s^{-1} simulation. The heights in Z of the plotted gas scale heights in each plot are different as they are calculated from the mid-plane temperatures of the cuDisc runs. For the isothermal simulation, this is the same as DustPy, but for the non-isothermal simulation, the gas scale heights are lower due to the cooler interior of the non-isothermal disc.

where turbulent motions start to become a strong source of relative velocities between particles. The location of the trough is dependent on the Reynolds number, which represents the strength of turbulent motions. In cuDisc, the Reynolds number varies as a function of height because it depends on the gas density and sound speed; however, in DustPy, it is assumed to be equal to the mid-plane value everywhere. This causes the trough location to vary as a function of height in cuDisc, leading to a smearing out of the feature in the vertically integrated density.

Fig. 17 shows the density of three different grain sizes, ∼ 0.5, ∼ 50 and ∼ 150 μm, as functions of height. The seemingly lower scale height of the 150 μm grains in the cuDisc runs when compared to DustPy is due to the decrease in the maximum grain size allowed by fragmentation with height in the cuDisc runs discussed earlier. This decrease in maximum allowable grain size can be seen by noting that 150 μm is around the peak of the grain size distribution at the mid-plane in Fig. 16 but that at 1 gas scale height, the peak has moved to ∼100 μm. This effect is less noticeable for the small grains (i.e. 0.5 μm in Fig. 17) as the small grain distribution is less affected by the change in the upper cut-off grain size. The non-isothermal vertical profiles are more condensed than DustPy, again due to the cooler interior temperatures leading to lower scale heights.

The vertical profiles found using cuDisc also show increases in the amount of intermediate size (10-100 μm) grains away from the mid-plane, with some grain sizes having a higher density at around a gas scale height than at the mid-plane. To investigate this, we need to look at how the collision rates vary with height. Fig. 18 shows the collision rate, Stokes numbers and relative turbulent velocity of grains with sizes 19 and 24 μm as functions of height for the non-isothermal run. Re is the Reynolds number that describes the strength of turbulent viscosity over molecular viscosity, i.e. Re= $\nu_t/\nu_{\rm mol}$, whilst St* is the Stokes number for the boundary between slow and fast eddies (class I and class II in the terminology used by Ormel & Cuzzi 2007) for a particular particle. Slow eddies have turn-over times longer than particle stopping times and induce large-scale systematic motion in the grains, whilst fast eddies have turn-over times smaller than the particle stopping times and, therefore, induce stochastic motions in the particles (Völk et al. 1980). Slow eddies do not drive

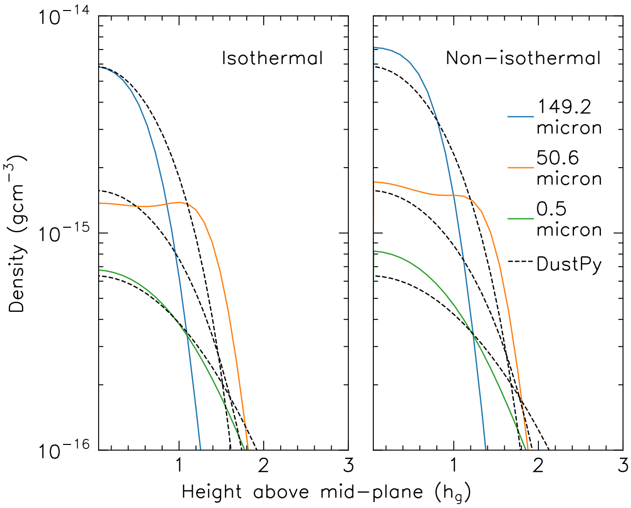

Figure 17. The vertical structure of three different grain sizes at a radius of 6 AU for the two $v_{\rm frag}$ = 100 cm s^{-1} cuDisc simulations over-plotted on the calculated DustPy profiles. The gas scale heights are calculated from the mid-plane temperatures of the respective cuDisc simulations; as the non-isothermal simulation has a cooler mid-plane, the gas scale height is lower in the right-hand panel.

large relative velocities between similarly sized grains, whereas fast eddies do (Ormel & Cuzzi 2007). Close to the mid-plane, the Stokes numbers of the grains are smaller than the Stokes number associated with the smallest turbulent eddies (Re$^{-1/2}$), meaning the particles only experience slow eddies; therefore, the relative velocities are low. As we move to higher regions of the disc, the particle Stokes numbers are in the intermediate regime, and particles feel the impact of fast eddies. These stochastic motions lead to larger relative velocities between similarly sized particles. The densities of the intermediate-sized grains are enhanced at these heights due to the increase in collision rates in combination with the decrease in maximum grain size with height discussed previously. Higher up in the disc, settling causes a decrease in dust density that leads to a sharp decrease in collisions regardless of large turbulent velocities, counteracting the enhancement seen at around one gas scale height.

7.2 Particle sweeping

These results differ from those seen by Krijt & Ciesla (2016), who found that small grains become trapped in the disc mid-plane due to colliding with, and sticking to, larger grains before vertical mixing can distribute them higher in the disc. This “sweeping-up” of small grains by large grains decreases the abundance of small grains at high altitudes in regions where particle collision rates at the mid-plane are high compared to the rate of diffusion due to turbulent processes. A reason why we do not see this effect may be that the largest grains in our simulations are ∼ 150 μm - these grains are less strongly settled than the cm-sized grains reached in the Krijt & Ciesla (2016) models, lessening the effect that leads to trapping in the mid-plane. To investigate this, we ran the same set of simulations with a higher fragmentation velocity of 1000 cm s^{-1}. Fig. 19 shows the vertically-integrated density as a function of radius and grain size for this set of simulations, now with the addition of the approximate drift limit. Both the cuDisc and the DustPy simulations show very similar dust distributions, with the majority of the disc exhibiting drift-limited growth and the largest grains in the dust trap reaching ∼ a few cm. Turning to the vertical structure, Fig. 20 shows the density profiles of three different dust species from each simulation at the peak of the dust trap (∼5.7 AU) on the top row and away from the trap (∼ 10 AU) on the bottom row. The cuDisc dust profiles in the trap clearly exhibit the trapping of smaller grains around the

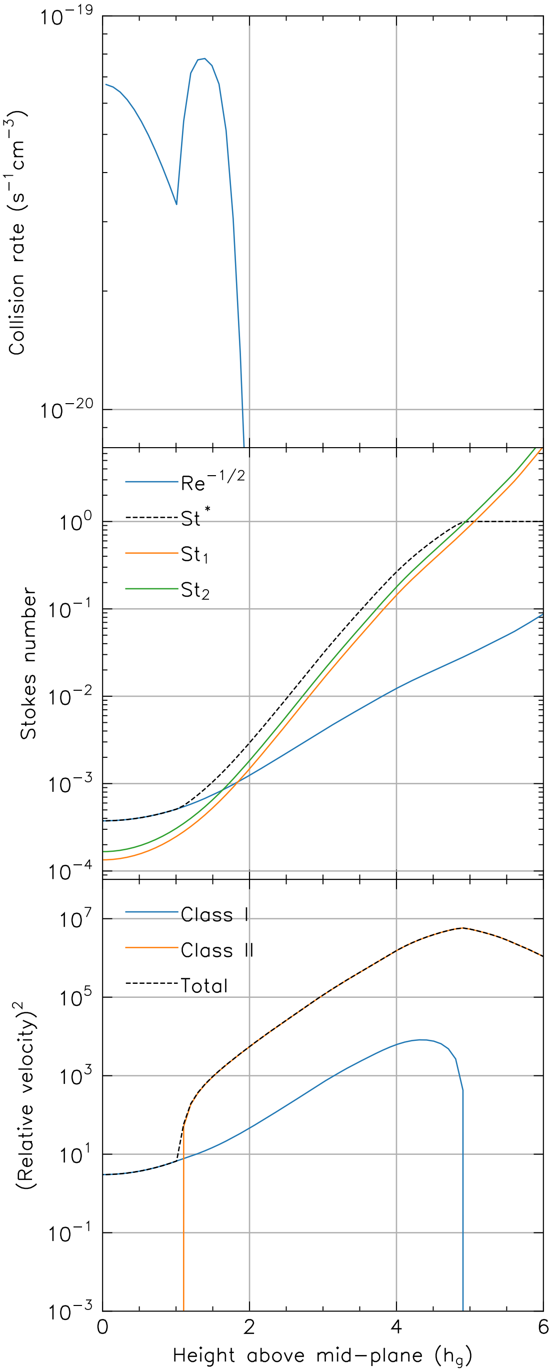

Figure 18. Collision rates, Stokes numbers and relative turbulent velocities of two different grain sizes, 19 and 24 μm, as a function of height. St* is the boundary between class I and class II eddies for a particular particle. Its minimum value is equal to the Stokes number of the smallest eddies, Re$^{-1/2}$, whilst its maximum value is equal to the Stokes number associated with the largest eddies (one orbit), 1.

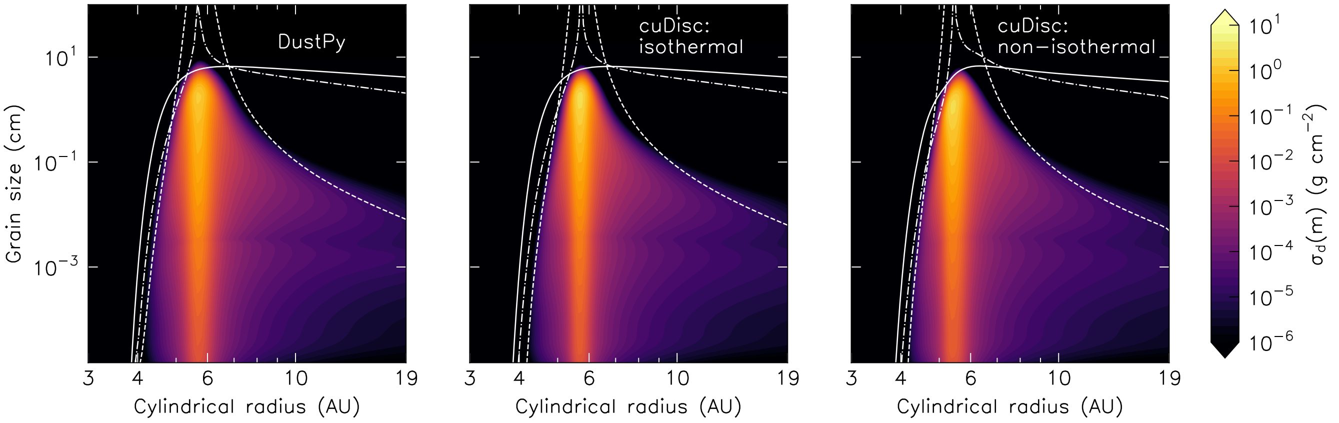

Figure 19. Dust grain size distributions as a function of disc radius for each $v_{\rm frag}$ = 1000 cm s^{-1} simulation. The fragmentation and drift limits (given by Birnstiel et al. 2012) are over-plotted for reference in solid white and dashed white respectively.

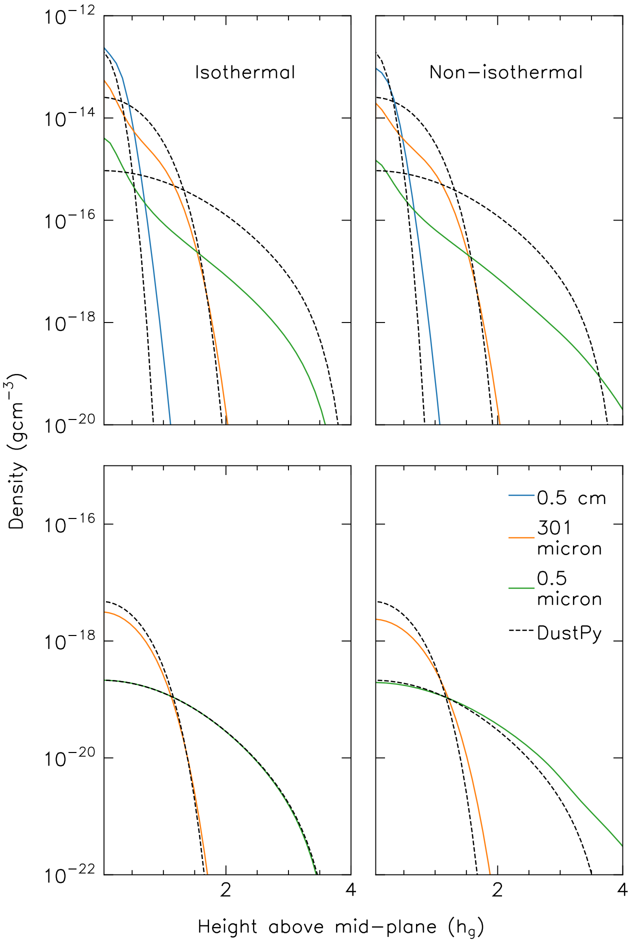

Figure 20. The vertical structure of three different grain sizes for the $v_{\rm frag}$ = 1000 cm s^{-1} simulations. The top and bottom rows show the profiles within the dust trap (~5.7 AU), and away from the dust trap (~ 10 AU) respectively. The density of the 0.5 cm grain is at the floor value at 10 AU in all of the simulations due to being above the drift limit. The gas scale heights are calculated from the mid-plane temperatures of the respective cuDisc simulations.

mid-plane, as found by Krijt & Ciesla (2016). However, the effect is not visible at 10 AU, where the isothermal profiles closely follow the Takeuchi & Lin (2002) analytic profiles calculated from the DustPy densities, and the non-isothermal profiles differ at a few scale heights due to the hot surface layers that increase the turbulent velocities, lofting the smaller grains to higher heights. The lack of sweeping at these radii is to be expected if the cause of the effect is the presence of large cm-sized grains, because here the the maximum grain size is lower due to the disc being in the drift-limited regime.

Krijt & Ciesla (2016) found that the effect of sweeping should be important in regions where the collisional timescale of grains is less than the timescale associated with diffusion to higher regions of the disc. In terms of disc quantities, this criterion can be written as $\alpha/(\Sigma_d/\Sigma_g) < 1$. Fig. 21 shows the radial regions of the discs in both the low and high fragmentation velocity simulations where this criterion is satisfied. In the high fragmentation velocity disc, the condition is only met around the peak of the dust trap - this concurs with our findings, as shown in Fig. 20. For the low fragmentation velocity case, the condition is satisfied for the entirety of the disc; but we find no evidence of trapping. We suspect this is due to the difference in scale height between the largest and smallest grains in the simulations as discussed above; for the low fragmentation velocity simulation, the difference between the scale heights of the 0.5 and 150 micron grains is a factor of ~ 2, whilst, for the high fragmentation velocity simulation, the difference in scale height between the 0.5 micron and 1 cm grains is a factor of ~ 80. In the low fragmentation velocity case, this means that any sweeping by the largest grains does not manifest itself as an appreciable change in the vertical density distribution of the smaller grains, as the sweeping grains and swept-up grains have similar scale heights. The effect is noticeable, however, in the high fragmentation velocity case, where the largest grains are much more settled than the smallest grains.

7.3 Diffusion-settling-coagulation equilibrium

These results indicate that the diffusion-settling equilibrium profiles (Takeuchi & Lin 2002) for the dust vertical structure do not fully describe the dust population in regions where collisions are important. In these regions, we suspect that a diffusion-settling-coagulation equilibrium is established; the form of which varies

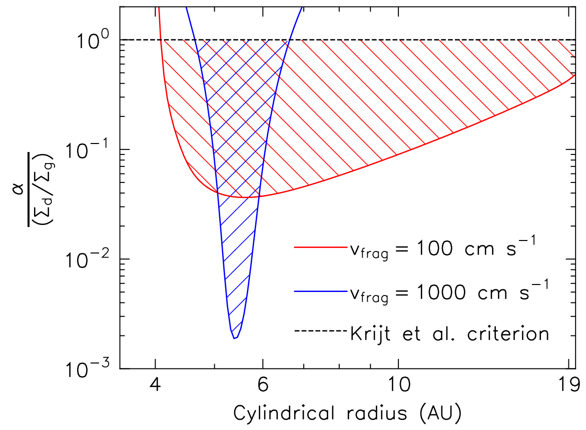

Figure 21. The regions of the disc within which the Krijt & Ciesla (2016) criterion for mid-plane trapping of small grains is satisfied, shown as hatched regions, for both the low and high fragmentation velocity `cuDisc` simulations.

depending on the fragmentation velocity. For a low fragmentation velocity of 100 cm s^{-1}, we find enhancements of intermediate-sized dust grains at ∼ one gas scale height, whilst for a high fragmentation velocity of 1000 cm s^{-1}, we find that large grains sweeping up smaller grains in the dust trap leads to the enhancement of small grains at the disc mid-plane. These findings may have implications for the analysis of observed disc spectral energy distributions (SEDs) as one must assume a disc structure to estimate the emission layers of different-sized dust grains (see e.g. D'Alessio et al. 2006). Scattered light images could also be affected, as mid-plane densities calculated from the observed small grain distribution at large disc heights are dependent on the assumed vertical structure of said grains. However, our results also demonstrate that if one only cares about overall, general, disc evolution and not specific problems that require a resolved vertical dimension (e.g. winds or temperature instabilities caused by vertical structure), the 1D results from `DustPy` are a good approximation to the full 2D problem.

8 SUMMARY

`cuDisc` is a new protoplanetary disc code that aims to allow long time-scale calculations of discs with self-consistently calculated dynamics, thermodynamics and dust grain growth and fragmentation. Modelling these physical processes alongside one another allows us to answer important problems in disc physics, such as structure formation due to instabilities and dust removal. With the use of GPU acceleration, `cuDisc` enables simulations to run for large fractions of the disc lifetime, removing the need for running simplified secular models. We have shown that 2D structure affects the dust spatial and grain size distributions even for simple systems, such as dust trapped in the pressure bump of a transition disc. This may be important when analysing SEDs and scattered light images, given the need to assume some model of grain vertical structure. We also find that for studying overall disc evolution, 1D models can quite accurately concur with 2D models. More features can and will be added to the code as we continue to develop it, such as more detailed dust microphysics and ice-vapour chemistry, making many other problems in disc physics able to be investigated through the use of `cuDisc`.

ACKNOWLEDGEMENTS

We would like to thank the referee for their detailed and helpful comments. RAB and JEO are supported by Royal Society University Research Fellowships. This work has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant agreement No. 853022, PEVAP) and a Royal Society Enhancement Award. Some of this work was performed using the Cambridge Service for Data Driven Discovery (CSD3), part of which is operated by the University of Cambridge Research Computing on behalf of the STFC DiRAC HPC Facility (`www.dirac.ac.uk`). The DiRAC component of CSD3 was funded by BEIS capital funding via STFC capital grants ST/P002307/1 and ST/R002452/1 and STFC operations grant ST/R00689X/1. DiRAC is part of the National e-Infrastructure. For the purpose of open access, the authors have applied a Creative Commons Attribution (CC-BY) licence to any Author Accepted Manuscript version arising.

DATA AVAILABILITY

The GitHub repository for `cuDisc` can be found at `https://github.com/cuDisc/cuDisc/`.

REFERENCES

Andrews S. M., 2020, Annual Review of Astronomy and Astrophysics, 58, 483
Bae J., Isella A., Zhu Z., Martin R., Okuzumi S., Suriano S., 2022, arXiv e-prints, p. arXiv:2210.13314
Balbus S. A., Papaloizou J. C. B., 1999, ApJ, 521, 650
Batalha N. M., et al., 2013, ApJS, 204, 24
Birnstiel T., Dullemond C. P., Brauer F., 2010, A&A, 513, A79
Birnstiel T., Klahr H., Ercolano B., 2012, A&A, 539, A148
Birnstiel T., et al., 2018, ApJ, 869, L45
Bitsch B., Crida A., Morbidelli A., Kley W., Dobbs-Dixon I., 2013, A&A, 549, A124
Bogacki P., Shampine L., 1989, Applied Mathematics Letters, 2, 321
Booth R. A., Clarke C. J., 2021, MNRAS, 502, 1569
Booth R. A., Clarke C. J., Madhusudhan N., Ilee J. D., 2017, MNRAS, 469, 3994
Brauer F., Dullemond C. P., Henning T., 2008, A&A, 480, 859
Chiang E. I., Goldreich P., 1997, ApJ, 490, 368
Clarke C. J., Pringle J. E., 1988, MNRAS, 235, 365
Commerçon B., Teyssier R., Audit E., Hennebelle P., Chabrier G., 2011, A&A, 529, A35
Courant R., Friedrichs K., Lewy H., 1928, Mathematische Annalen, 100, 32
Cuzzi J. N., Dobrovolskis A. R., Champney J. M., 1993, Icarus, 106, 102
D'Alessio P., Calvet N., Hartmann L., Franco-Hernández R., Servín H., 2006, ApJ, 638, 314
Dohnanyi J. S., 1969, J. Geophys. Res., 74, 2531
Draine B. T., 2003, ApJ, 598, 1026
Dubrulle B., Morfill G., Sterzik M., 1995, Icarus, 114, 237
Dullemond C. P., Juhasz A., Pohl A., Sereshti F., Shetty R., Peters T., Commercon B., Flock M., 2012, RADMC-3D: A multi-purpose radiative transfer tool, Astrophysics Source Code Library, record ascl:1202.015
Eistrup C., 2023, ACS Earth and Space Chemistry, 7, 260
Ercolano B., Pascucci I., 2017, Royal Society Open Science, 4, 170114
Franz R., Picogna G., Ercolano B., Birnstiel T., 2020, A&A, 635, A53
Fritsch F. N., Carlson R. E., 1980, SIAM Journal on Numerical Analysis, 17, 238
Fuksman J. D. M., Klahr H., 2022, ApJ, 936, 16
Garaud P., Meru F., Galvagni M., Olczak C., 2013, ApJ, 764, 146
Huang P., Bai X.-N., 2022, ApJS, 262, 11

Hutchison M. A., Clarke C. J., 2021, MNRAS, 501, 1127
Jankovic M. R., Owen J. E., Mohanty S., Tan J. C., 2021, MNRAS, 504, 280
Kley W., 1989, A&A, 208, 98
Kley W., 1998, A&A, 338, L37
Krijt S., Ciesla F. J., 2016, ApJ, 822, 111
Krumholz M. R., Ireland M. J., Kratter K. M., 2020, MNRAS, 498, 3023
Kuiper R., Klahr H., Dullemond C., Kley W., Henning T., 2010, A&A, 511, A81
Leveque R. J., 2004, Journal of Hyperbolic Differential Equations, 01, 315
Levermore C. D., Pomraning G. C., 1981, ApJ, 248, 321
Madhusudhan N., 2019, Annual Review of Astronomy and Astrophysics, 57, 617
Mathis J. S., Rumpl W., Nordsieck K. H., 1977, ApJ, 217, 425
Mayor M., et al., 2011, arXiv e-prints, p. arXiv:1109.2497
Mignone A., 2014, Journal of Computational Physics, 270, 784
Naumov M., 2011, NVIDIA White Paper
Öberg K. I., Murray-Clay R., Bergin E. A., 2011, ApJ, 743, L16
Ohtsuki K., Nakagawa Y., Nakazawa K., 1990, Icarus, 83, 205
Ormel C. W., Cuzzi J. N., 2007, A&A, 466, 413
Owen J. E., 2016, Publ. Astron. Soc. Australia, 33, e005
Owen J. E., 2020, MNRAS, 495, 3160
Owen J. E., Kollmeier J. A., 2019, MNRAS, 487, 3702
Pascucci I., Wolf S., Steinacker J., Dullemond C. P., Henning T., Niccolini G., Woitke P., Lopez B., 2004, A&A, 417, 793
Philippov A. A., Rafikov R. R., 2017, ApJ, 837, 101
Pinte C., Harries T. J., Min M., Watson A. M., Dullemond C. P., Woitke P., Ménard F., Durán-Rojas M. C., 2009, A&A, 498, 967
Pringle J. E., 1981, ARA&A, 19, 137
Rafikov R. R., Silsbee K., Booth R. A., 2020, ApJS, 247, 65
Ramsey J. P., Dullemond C. P., 2015, A&A, 574, A81
Rodenkirch P. J., Dullemond C. P., 2022, A&A, 659, A42
Roe P., 1981, Journal of Computational Physics, 43, 357
Shakura N. I., Sunyaev R. A., 1973, A&A, 24, 337
Smoluchowski M. V., 1916, Zeitschrift fur Physik, 17, 557
Stammler S. M., Birnstiel T., 2022, ApJ, 935, 35
Stone J. M., Gardiner T., 2009, New Astronomy, 14, 139
Takeuchi T., Lin D. N. C., 2002, ApJ, 581, 1344
Tanaka H., Inaba S., Nakazawa K., 1996, Icarus, 123, 450
Urpin V. A., 1984, Soviet Ast., 28, 50
Völk H. J., Jones F. C., Morfill G. E., Roeser S., 1980, A&A, 85, 316
Watanabe S.-i., Lin D. N. C., 2008, ApJ, 672, 1183
Winn J. N., Fabrycky D. C., 2015, Annual Review of Astronomy and Astrophysics, 53, 409
Woitke P., Kamp I., Thi W. F., 2009, A&A, 501, 383
Wootten A., Thompson A. R., 2009, IEEE Proceedings, 97, 1463
Wu Y., Lithwick Y., 2021, ApJ, 923, 123
Wu J., Gao Z., Dai Z., 2012, Journal of Computational Physics, 231, 7152
Zhu W., Dong S., 2021, Annual Review of Astronomy and Astrophysics, 59, 291

APPENDIX A: DIFFUSION MATRIX

The full form of $\tilde{D}_{kl}$ is found by writing out $\tilde{F}_\sigma$ (Eqn. 21) for each interface σ about cell i, j and summing all terms that apply to cell k, l in the 3×3 stencil of cells centred on cell i, j. For this exercise, we will change cell references from the form i, j to the single letter form k, for ease of reading. As an example, referring to Fig. 3, when calculating the fluxes over the interfaces σ and λ, $\tilde{F}_\sigma$ and $\tilde{F}_\lambda$ respectively, according to Eqn. 25 we find

$$\tilde{F}_\sigma = -D_k w_{k\sigma}\left[w_{l\sigma} c_{kl}(u_l - u_k) + w_{m\tau} c_{km}(u_m - u_k)\right] + \\ D_l w_{l\sigma}\left[w_{k\sigma} c_{lk}(u_k - u_l) + w_{p\nu} c_{lp}(u_p - u_l)\right], \quad \text{(A1)}$$

$$\tilde{F}_\lambda = -D_a w_{a\lambda}\left[w_{l\lambda} c_{al}(u_l - u_a) + w_{m\beta} c_{am}(u_m - u_k)\right] + \\ D_l w_{l\lambda}\left[w_{a\lambda} c_l a(u_l - u_a) + w_{b\epsilon} c_{lb}(u_b - u_l)\right], \quad \text{(A2)}$$

where ν, β and ϵ are the interfaces between cells l & p, a & m and l & b respectively.

Taking for example cell m, the value in the diffusion matrix $\tilde{D}_m$ is therefore given by the sum of all terms that are multiplied by u_m, multiplied by their respective interface areas A, i.e.

$$\tilde{D}_m = -D_k w_{k\sigma} w_{m\tau} c_{km} A_\sigma - D_a w_{a\lambda} w_{m\beta} c_{am} A_\lambda. \quad \text{(A3)}$$