Computational Cost Optimization for Numerical Integration of the Smoluchovski Equation via Stochastic Sampling of the Coagulation Kernel

Vincent C. Mader

May 24, 2023

Contents

List of Figures

List of Tables

1 Preface

1.1 Acknowledgements

1.2 Abstract

What is the topic of this thesis?

- We want to test whether the numerical integration of the Smoluchovski coagulation equation can be sped up via stochastic sampling of the kernel.
- Depending on how accurate one wants to model the involved processes, the integration of the coagulation equation can become quite costly (from a computational standpoint).
- This is due to the "high dimensionality" of the problem.
    - 3 dimensions of space (in principle)
    - 3 dimensions of mass (explain why -> fragmentation)
    - X dimensions of porosity
    - + maybe other attributes of the dust particles
- Therefore, the idea is that a big gain in computational performance could be gained by only including the "relevant" parts of the kernel into the integration. ("relevant" according to what criterion?)
- Finding out whether this works (and by how much the computational cost can be reduced) is the big goal of this thesis.

2 Introduction

2.1 "As of yet unsorted Things to put into the Introduction"

2.1.1 QUES What even is coagulation?

Coagulation is

- the process of particles sticking together and forming larger bodies
- the most-likely origin of planetary cores [citation needed]
- a complex process
    - coagulation + fragmentation
    - ...

How can you model coagulation?

- Smoluchovski equation
- ... [other ansatz?]

2.1.2 QUES How to model the gas inside the disk?

What are the most relevant properties of the gas?

- density
- viscosity
- velocity
- ... [QUES]

`QUES` Write this section

- Give broad overview
    - More detail can be given later.

2.1.3 QUES What characterizes a dust particle?

A dust particle has the following attributes:

Particle position:

- This can be given either
    - in cartesian coordinates (x, y, z), or
    - in cylindrical coordinates (r, φ, z).
- Due to the (approximately) circular shape of the disk, cylindrical coordinates are the more fitting choice here.

Particle velocity:

- Analogously to the particle position, the preferred system of coordinates is a cylindrical one here.
- Thus, we write the particle velocity as (v_r, v_φ, v_z).

Particle mass:

- In a very naive/simple model, the particles are assumed to
  - be perfectly spherical (with radius a), and
  - share a common solid density ρ_s.
- This is of course not the case in reality, but we will make this assumption throughout the thesis (if not otherwise stated.)
- Thus, the particle mass m can be calculated directly from the particle radius. The two quantities are related by the equation

$$m = \frac{4}{3}\pi a^3 \cdot \rho_s$$

Time:

- With the exception of the particle mass, all of the quantities discussed above can not assumed to be constant with time (in this model -> in reality, none of them are constant).
- Thus, the temporal parameter t is included into the model as well.

Porosity:

- (not included in the model)
  - Particles have "fractal-like" shape, not spherical.

QUES:

- Think re:
  - Only list attributes & their labels/symbols here.
  - Make model assumptions elsewhere?

2.1.4 QUES What is a particle distribution function?

- Consider a distribution of particles.
- Let the particles have various different properties, e.g. position, velocity, and mass.
- Assume these properties to *not* be distributed homogenously among the particles.
- The distribution can then be described by a fitting *particle distribution function*

$$f(...) = ...$$

QUES: Continue writing this section.

- What is a particle distribution function *generally*?

2.1.5 QUES What is the dust particle distribution function?

QUES: Apply the definition of a general particle distribution function (from above) to the (special) case of a protoplanetary disk.

- We assumed the relevant properties to be
  - radial position in the disk (radial distance from star)
  - height above the midplane

- particles mass

- These properties (in general) depend on time.

Therefore, we can write the distribution function for dust particles inside the modeled PPD as

$$f(r, z, m, t)$$

- NOTE:
  - After this section, focus (for now) on naive/simple particle *mass distribution function.*

2.1.6 QUES Assumptions needed for the (naive/simple) particle mass distribution.

Consider a distribution of dust particles with different masses m.

- Let them all have the same material density ρ_s (and other properties).
- Also, (for now) assume that their shape is always spherical.

Then, each dust particle is characterized solely by its mass, and hence its radius a:

$$m = \frac{4\pi}{3}\rho_s a^3$$

2.1.7 QUES What is the dust particle mass distribution function?

The particle mass distribution function $n(m)$ is defined in such a way that the integral

$$N = \int_0^\infty n(m)\ \mathrm{d}m$$

is the total number of dust particles per unit volume.

Following from that, the integral

$$\rho = \int_0^\infty m \cdot n(m)\ \mathrm{d}m$$

gives the total dust mass per unit volume, i.e. the dust density.

2.2 "Proto-Planetary Disks"

2.2.1 QUES What is the interstellar medium composed of?

The interstellar medium consists mainly of low-mass particles, which can be categorized into

1. atomic gas (mostly atomic hydrogen & helium), and (to a lesser extent)
2. larger dust particles (characterized mainly by their mass & porosity) [citations needed]

2.2.2 QUES How do protoplanetary disks form?

- Why does it even happen?
  - A cloud of gas/dust collapses under its own gravity.
- What happens?
  - The cloud flattens out into a disk.
- Why does the cloud flatten out?
  - Angular momentum is conserved.
  - There is a most likely a non-zero *total* angular momentum. When an object is hit, and it's not hit *exactly* in the center, it starts to spin.
  - Gravity pulls the particles inwards, but rotation (centrifugal/centripetal forces) opposes gravity.
  - Thus, a disk is formed.

2.2.3 QUES Why does a star form at the center of the disk?

- gravity $\rightarrow$ high pressure $\rightarrow$ nuclear fusion

2.2.4 QUES What does a protoplanetary disk look like?

`QUES`: Talk about

- approximately (first-order) circular shape of a PPD,
- disk flaring, midplane, ...
- ...

`QUES`: Include pictures: observations.

`QUES`: Include pictures: diagrams/graphs of simplified disk.

2.2.5 QUES What is a protoplanetary disk made up of?

- It's made of the same material that made up the original cloud.
- This is mostly gas, and a little bit of dust.

2.2.6 QUES How is mass distributed inside the disk?

Most of the mass inside PPDs is concentrated inside the star at its center. [citation needed]

- We can take a (kind of random) look at our own solar system, where [how much? I believe about 99] percent of the mass is concentrated inside the Sun.

`QUES`: Find data/sources about mass distribution inside PPDs.

The model used in this thesis will be based on the assumption that this is the case. Let M_* label the stellar mass, and M_{disk} label the disk mass.

Then, we can define a "disk-to-star" mass ratio, which we label q_{dts}. If not otherwise stated, the value of this ratio will be assumed to be...

`QUES`: Choose value of "disk-to-star" mass ratio.

Let us now turn our attention to the mass not concentrated inside the star, but spread across the protoplanetary disk. It is comprised mostly [citation needed] of atomic/molecular gas (mostly hydrogen & helium) [citation needed] and larger (but still very small [how small?]) dust particles [citation needed].

`QUES`: Talk about element composition of interstellar medium, and how that translates into the composition of the disk. `QUES`: Talk about approximate scale of dust particles.

Analogously to before, we can define a "dust-to-gas" mass ratio, which we label q_{dtg}.

If not stated otherwise, a value of ... will we used throughout this thesis.

`QUES`: Talk about how to choose the "dust-to-gas" ratio, cite others.

2.3 "Particle Kinematics inside the Disk"

2.3.1 QUES How does the central star influence the disk around it?

- The star exerts its gravitational influence onto the disk.
    - As a first order approxmiation, the matter in the disk moves around the star on Keplerian trajectories.
- The lighter particles in the disk move on sub-Keplerian trajectories though.
    - "This is due to the radiation pressure".

2.3.2 QUES How does the gas affect the dust?

- The sub-Keplerian motion of the gas leads to a non-zero relative velocity between gas and dust particles.
    - This results in a drag force acting on the dust particles.
    - Therefore, the dust particles are "slowed down" and spiral inwards.

2.3.3 QUES What other effects are there?

TODO: Talk about (relative) velocity components.

2.4 "Particle Collisions"

2.4.1 QUES Why do particles collide?

In a PPD, dust particles move around at

- partly systematic, and
- partly random

velocities.

This includes [QUES: sort this out]

- Keplerian motion
- drag from interaction with gas
- Brownian motion
- radial drift
- turbulence
- ...

This causes the dust particles to occasionally collide with each other.

[...coagulation...] [...fragmentation...]

2.4.2 QUES What happens when particles collide? (from standpoint of particle mass distr.)

Collisions can be regarded as two-body collisions:

- The occurrence of three-body interactions is so tiny that it can be neglected entirely (negligible).

Outcomes:

1. bouncing
2. merging
3. fragmentation

Bouncing:

- Here: The only property we trace is the particle mass.
- Therefore, "nothing happens".

Merging:

- The two original particles with masses m_1 and m_2 "disappear".
- A single new particle with mass $m_{\text{tot}} = m_1 + m_2$ "appears".

Fragmentation:

- The two original particles with masses m_1 and m_2 also "disappear".
- But: A whole series/range of particles can appear, with a total mass of $m_{\text{tot}} = m_1 + m_2$

The input to a collision event are therefore always two bodies, but the output can be any number of particles.

2.4.3 QUES How do collisions lead to the formation of planetary cores?

2.4.4 QUES How can particle collisions be modeled?

["ensemble approach"]

[...]

- Particle collisions can be modeled using the Smoluchovski equation.

[...]

2.5 The Smoluchovski Coagulation Equation

2.5.1 QUES What is the Smoluchovski coagulation equation?

The general time evolution of the particle mass distribution function due to collision can (by virtue of the two-body nature) be expressed using the Smoluchovski coagulation equation.

... background ... [1916 Smoluchovski]

The coagulation equation reads

$$\frac{\mathrm{d}n}{\mathrm{d}t}(m,t) = \int_0^\infty \int_0^\infty K(m,m',m'') \cdot n(m',t) \cdot n(m'',t)\ \mathrm{d}m'\ \mathrm{d}m''$$

Here, $K(m,m',m'')$ labels the "so-called" *coagulation kernel*.

This general form of the (temporal) evolution equation of the particle mass distribution function encompasses all possible collision outcomes (bouncing, merging, fragmentation).

2.5.2 QUES What is the coagulation kernel?

- NOTE:
  - Stay very general/"vague" here.
  - Details will follow in sections on stick-and-hit / fragmentation.
- First problem to look at:
  - simple case of Stick-and-Hit coagulation
- But before we do that, we need to talk about discretization...
  - discretization of mass axis
  - discretization of dust particle (mass) distribution function
  - discretization of kernel
  - discretization of time axis
- ...and about numerical integration of the Smoluchovski equation
  - for that, we need to talk about discretization of the time axis

2.6 "Coagulation"

2.7 Notes on the Structure of this Thesis

2.7.1 QUES What is the structure of this thesis?

- ...

1. Introduction
2. Attempt A
   (a) "Stick-and-Hit Coagulation"
   (b) "Initialization of the Dust Particle Mass Distribution Function"
   (c) "Implementation of Physical Reaction Rates"
      i. "Implementation of Physical Collision Rates"
         A. "Implementation of Physical Collision Cross Section"
         B. "Implementation of Physical Relative Velocities"
      ii. "..."
3. Attempt B
   - Give context/introduction.
     (a) What is the goal?
     (b) (?) How do planets form?
     (c) How do PPDs form?
     (d) What makes up a disk?
     (e) What does a disk "look" like?
         - radially
     (f) .....
   - Implement a first hands-on approach of what we're even talking about:
     (a) How do we `discretize` the particle mass distribution function?
         - Here: Do it on a `linear` grid.
     (b) How do we discretize the Smoluchovski equation?
     (c) How do we define a (simple/naive) coagulation kernel?
         - Here: Include only `stick-and-hit` coagulation.
         - Here: Set reaction rates to 1.
     (d) How do we define the initial particle mass distribution?
         - Here: Use Dirac-delta distribution.
     (e) How to we `integrate` the Smoluchovski equation?
         - Here: Do it `explicitly`.
   - Move to a `logarithmic` representation.

(a) How do you define a logarithmic (discrete) mass axis?

- Why even do this? -> Talk about involved scales.
- What are the problems here?
  - Talk about resulting masses not lying on grid points.
  - Talk about near-zero cancellation.

(b) How do we fix the grid-point problem?

- Implement `Kovetz-Olund` algortihm.

(c) How do we fix the cancellation problem?

- Implement `cancellation handling`.

- Move away from explicit integration, use `implicit integration`.

(a) Implement `logarithmic time axis`.

(b) ...

- Include `fragmentation` into the model.

(a) What is a first attempt at doing this?

- Move mass into the smallest-mass bin.

(b) How can one do this more accurately?

- Implement MRN distribution.

(c) (?) Implement cancellation handling.

- (?) Implement initial distribution using MRN

(a) ...

- Implement `physical reaction rates`.

(a) What are the reaction rates made up of?

- ...

(b) What are the collision rates made up of?

- Define collision rate from cross section & rel. velocity.

(c) How is the collision cross section defined?

- Define cross section.

(d) How are the relative velocities defined?

- Define total rel. velocity.

- (?) Implement `physical gas model` (radial). <- Do this here?

(a) ...

- Implement `physical relative velocites`.

(a) ... (RD, AZ, DS, TU)

- Implement `stochastic sampling` of the kernel.

  (a) ...

3 Definition of a simplified Radial Model for the Proto-Planetary Disk

3.1 Introduction, Overview, & Goals for this Section

3.1.1 QUES How to formulate a model for the PPD, its contents, & the planet formation processes?

The choice of a model is not at all trivial, and a lot of assumptions will have to be made (in order to not complicate things too much).

This can not be assumed to be a comprehensive and/or physically accurate model. The attempt is to formulate a model that is *good enough* to capture the "behavior [...]" qualitatively, "and even quantitatively in an approximate way".

We assume:

A protoplanetary disk is essentially made up of... [citations needed]

- the star at its center, (We assume it to already have formed.)
- the atomic/molecular gas, and
- the dust.

The disk can be assumed to be (approximately) symmetrical. [citation needed]

- This simplifies the treatment by "not including" the azimuthal direction. [except for rel. vel., there it *is* included!]
- The [later introduced] particle distribution function does not depend on φ.

We have to think about...

1. how the mass is distributed inside the disk, i.e.
   - the ratio between stellar mass & gas/dust mass,
   - the ratio between gas & dust mass.
2. how to model the gas, i.e.
   - its relevant properties (and how their are distributed radially)
     - densities, viscosities, sound speed, [...]
     - kinematics (velocities, relative velocities -> collisions -> coagulation)
     - ...
3. how to model the dust.
   - interaction with the gas
   - interaction with other dust [-> coagulation]

QUES:

- Talk more about the central star. If it has already formed...
  - then *when* are we?
  - how old is the disk already?

3.1.2 QUES How to model the central star?

What do we need to define?

- Stellar mass
  - Default:

$$M_{\text{star}} = 1 \cdot M_{\odot}$$

- Stellar luminosity
  - Default:

$$L_{\text{star}} = 1 \cdot L_{\odot}$$

3.1.3 QUES How to model the gas?

3.1.4 QUES How to model the dust?

3.2 Properties of the Star at the Center of the PPD

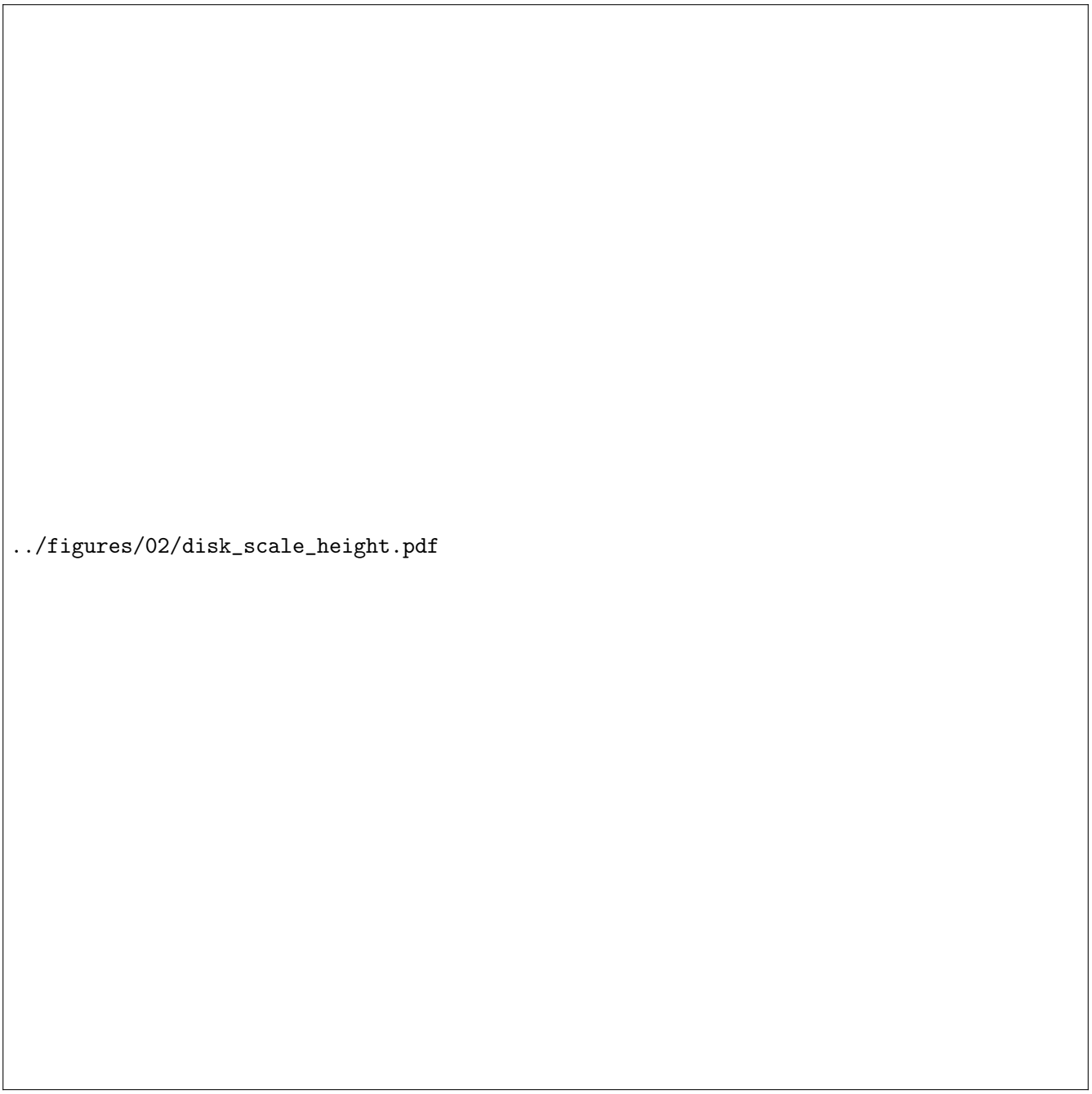
3.2.1 Age

3.2.2 Mass

3.3 Properties of the PPD

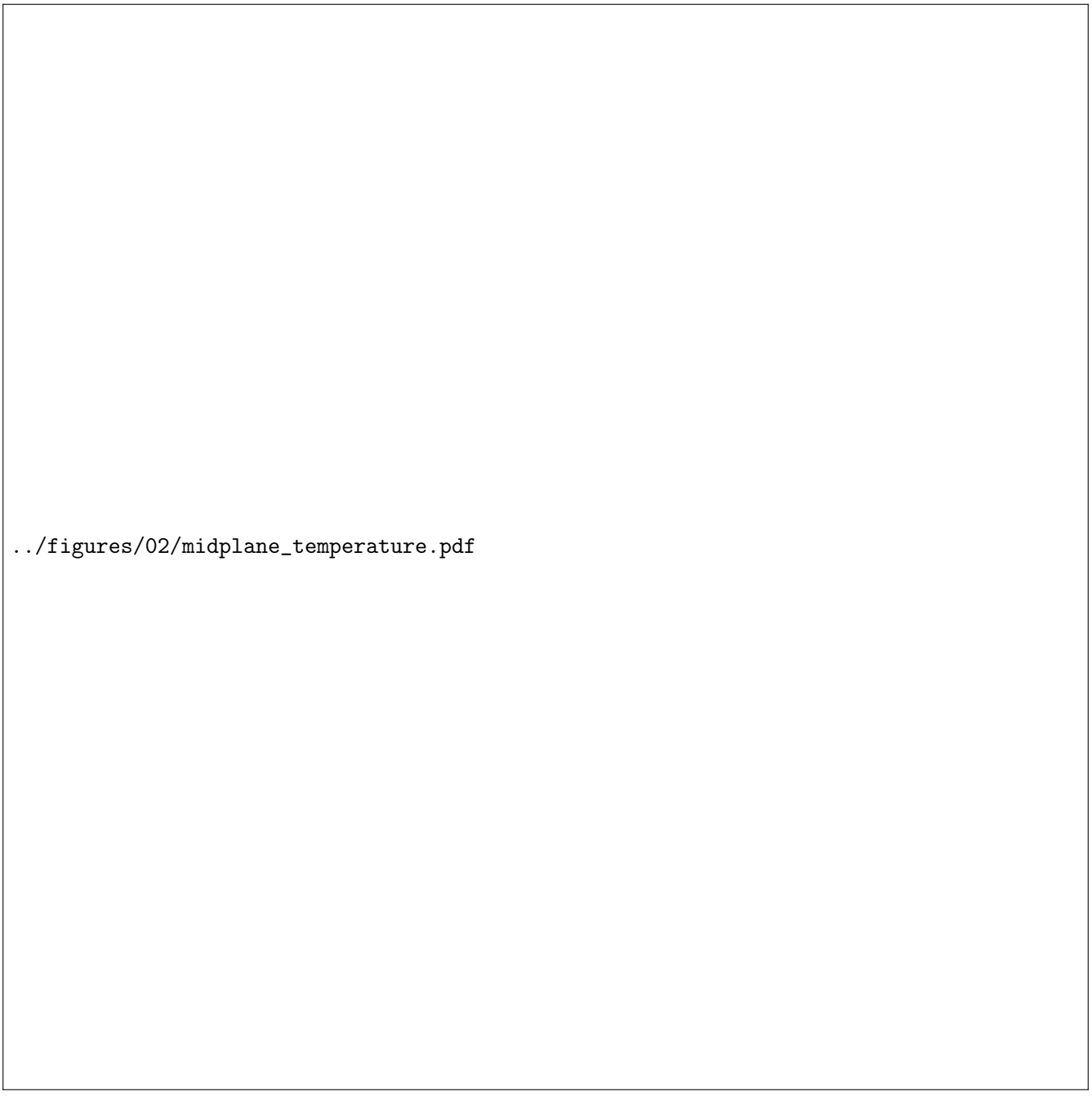
3.3.1 Disk Scale Height

$$H_p = \frac{c_s}{\Omega_K}$$



3.3.2 Midplane Temperature

$$\sigma_{\mathrm{SB}} \cdot T^4 = \frac{\alpha_{\mathrm{fl}}}{2} \cdot L_{\mathrm{star}}$$



1. Derivation Flux at distance r from star:

$$B(r) = \frac{\alpha_{\mathrm{fl}}}{2} \frac{L_{\mathrm{star}}}{4\pi r^2}$$

Now use Stefan-Boltzmann law:

$$B(T) = \sigma_{\mathrm{SB}} \cdot T^4$$

Midplane temperature at distance r from star:

$$\begin{aligned} T_{\mathrm{mid}} &= \left[\frac{B(r)}{\sigma_{SB}} \right]^{1/4} \\ &= \left[\frac{\alpha_{\mathrm{fl}}}{2} \cdot \frac{L_{\mathrm{star}}}{4\pi r^2 \cdot \sigma_{\mathrm{SB}}} \right]^{1/4} \end{aligned}$$

Rearrange for T_{mid} to arrive at the above definition.

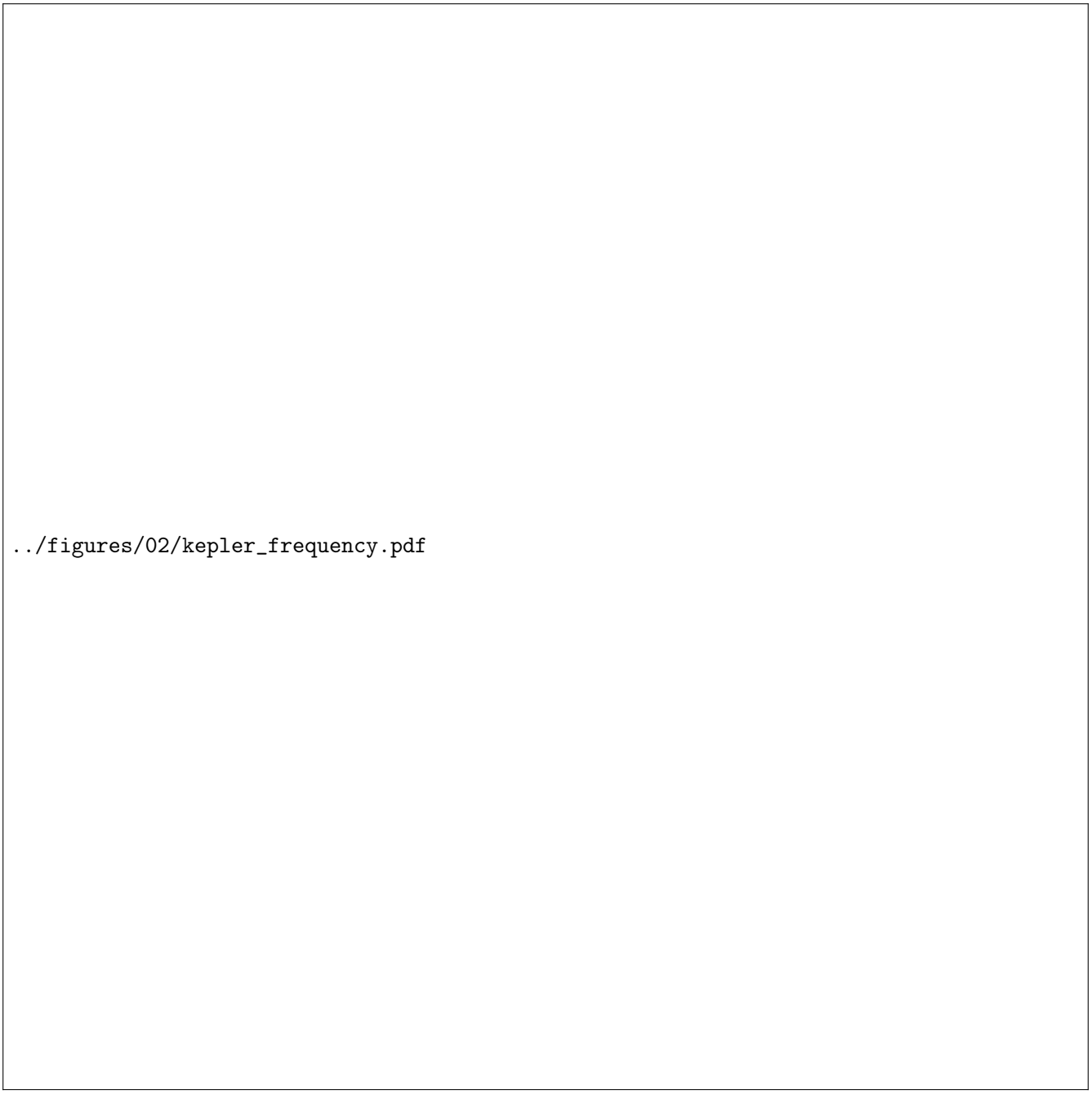
3.3.3 Kepler Frequency

Kepler velocity:

$$v_K = \sqrt{\frac{GM_*}{r}}$$

Kepler frequency:

$$\Omega_K = \sqrt{\frac{GM_*}{r^3}}$$



3.4 Properties of the Gas inside the PPD

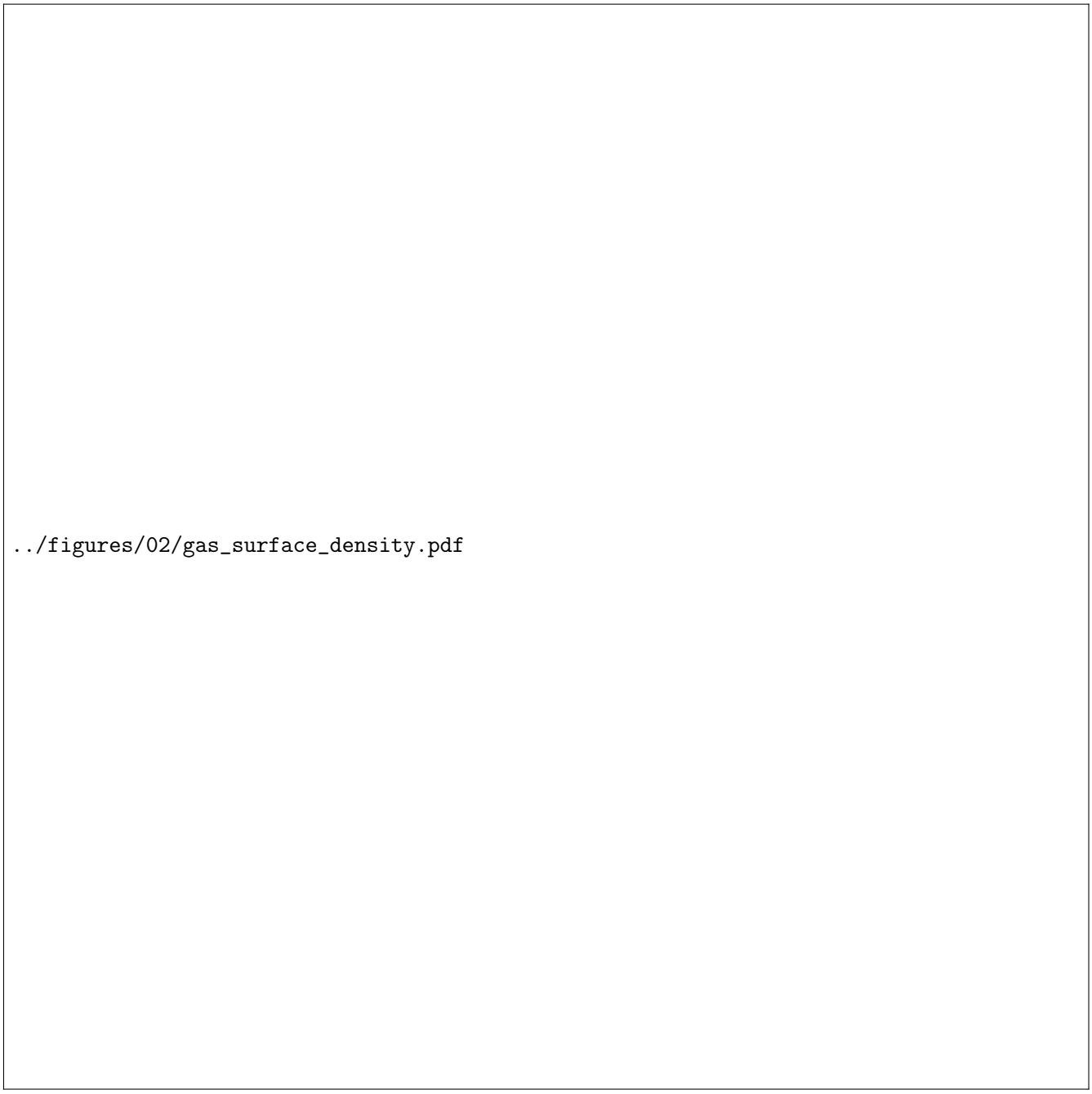
3.4.1 Radial Profile of Gas Density

1. Gas Surface Density How is it defined?

$$\Sigma_g = \int_{-\infty}^{\infty} \rho_g(z) \, \mathrm{d}z$$

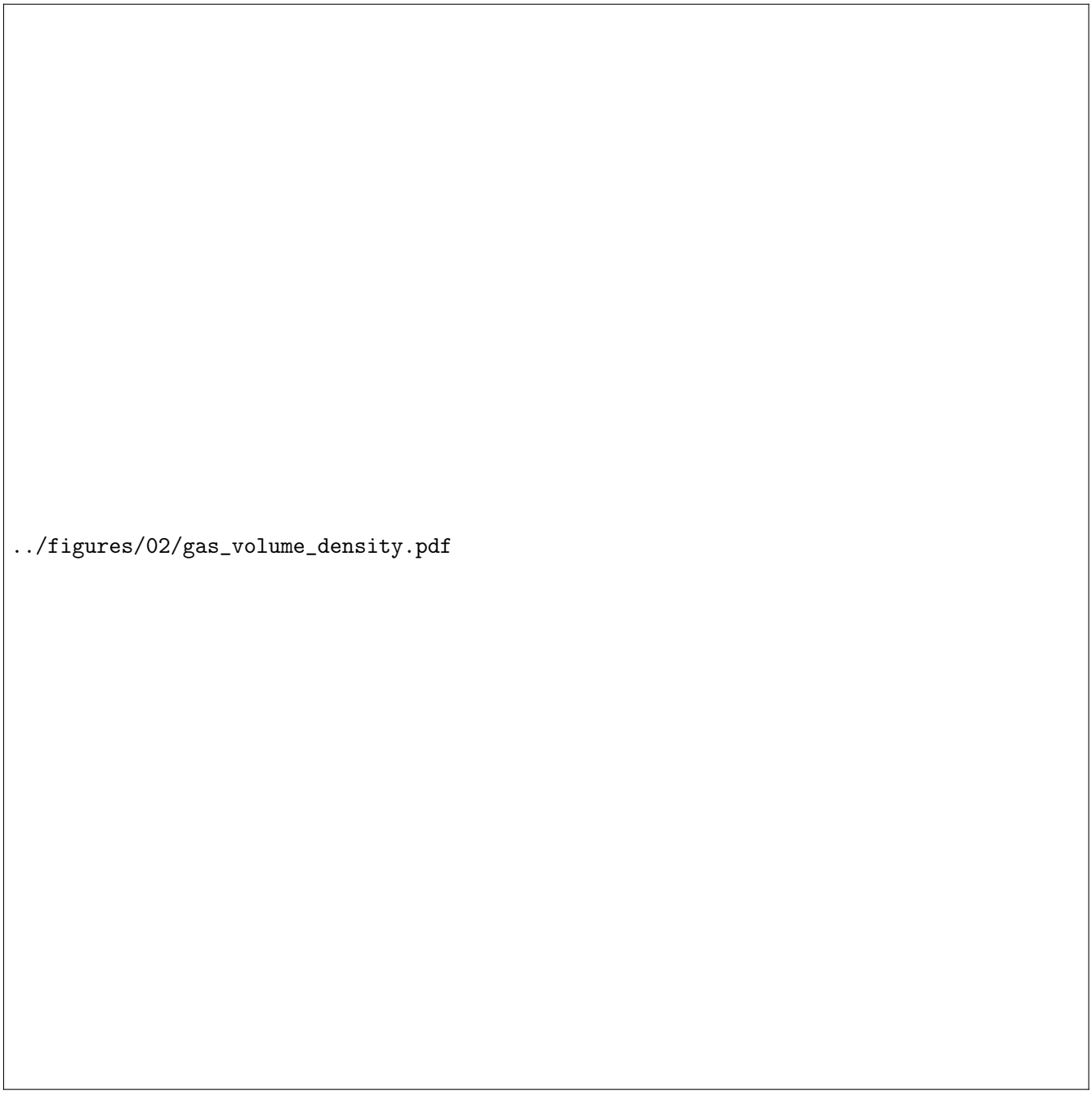
How can it be written as a function of the distance to the star? Ansatz: Use a power law, such that

$$\Sigma_g \sim \frac{1}{r^k}$$



2. Gas Volume Density

$$\rho_g = \frac{1}{\sqrt{2\pi}} \frac{\Sigma_g}{H_p}$$



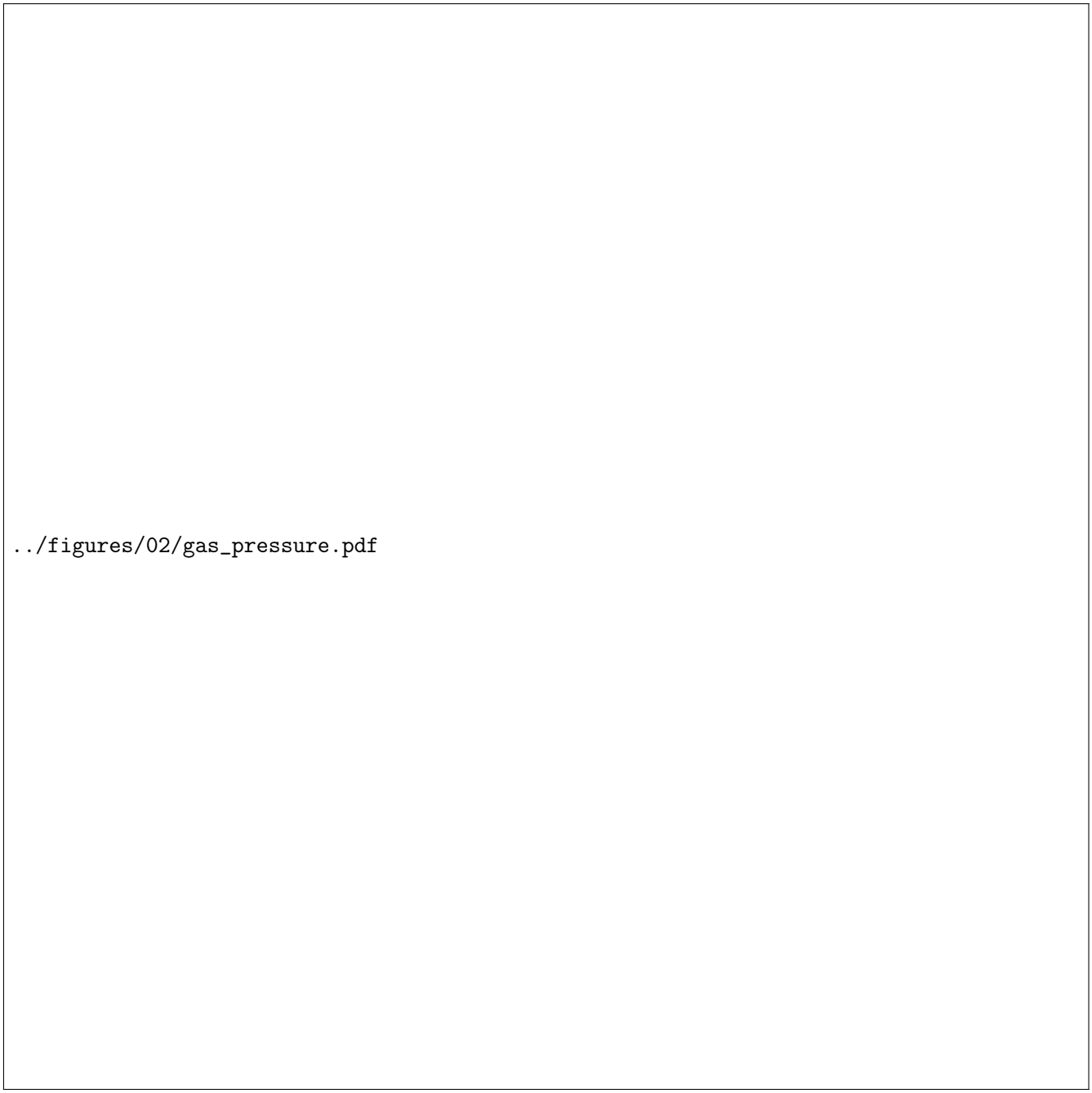
3. Gas Volume Number Density

$$n_g = \frac{\rho_g}{2.3 \cdot m_{\rm p}}$$

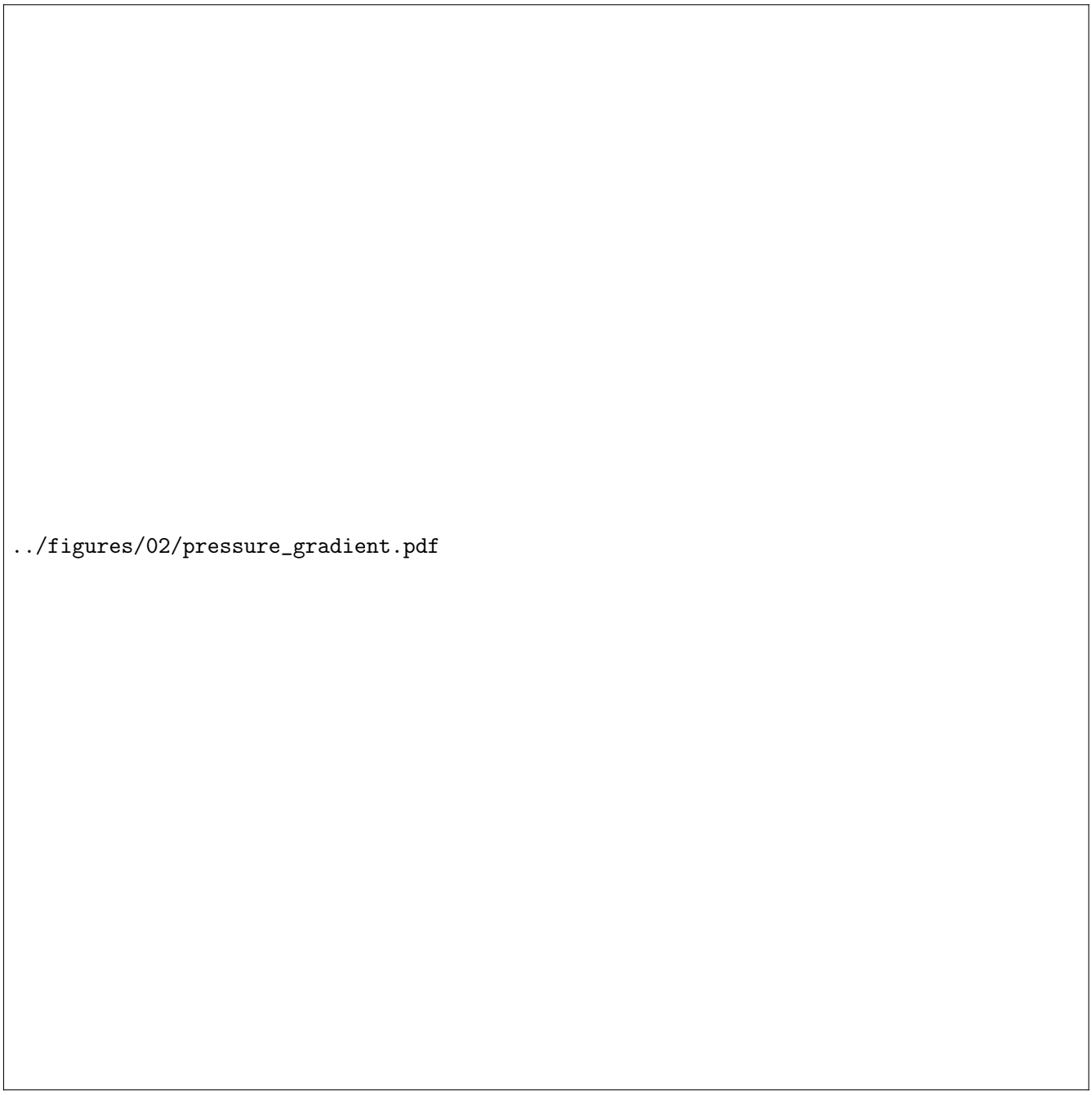
3.4.2 Radial Profile of Gas Pressure

1. Midplane Gas Pressure

$$P_g = \rho_g \cdot c_s^2$$



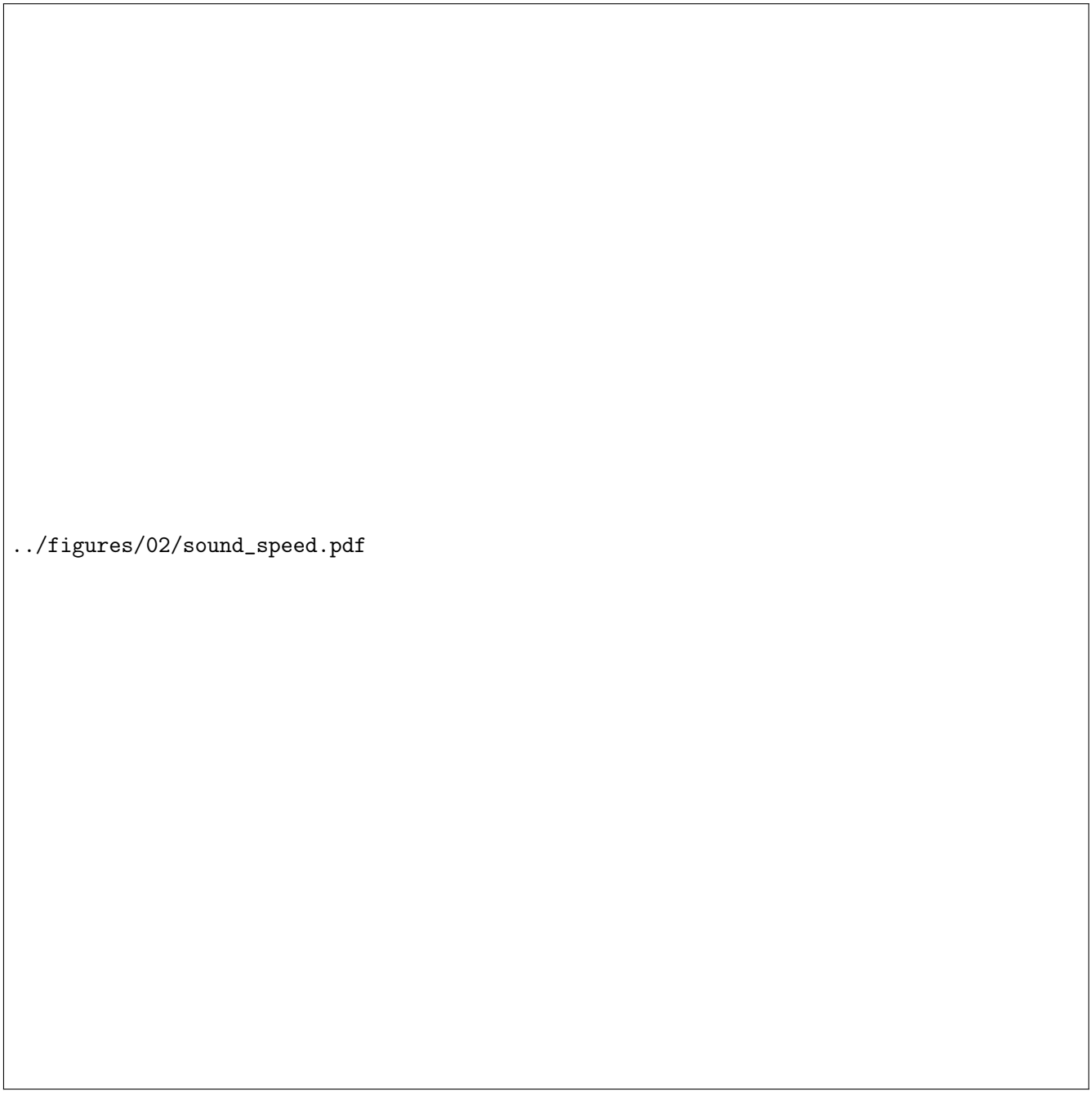
2. Gas Pressure Gradient



3.4.3 Thermal Speed & Sound Speed

1. Sound Speed

$$c_s = \sqrt{\frac{k_B \cdot T_{\rm mid}}{2.3 \cdot m_{\rm p}}}$$



(a) Derivation

$$2.3 \cdot m_{\mathrm{p}} \cdot c_s^2 = k_B \cdot T_{\mathrm{mid}}$$

Why though?

2. Mean Thermal Velocity of Gas Particles

$$u_{\mathrm{th}}\bar{v} = \sqrt{\frac{8}{\pi}} \cdot c_s$$

3.4.4 Gas Molecular Viscosity

$$\nu_{\text{mol}} = \frac{\bar{u} \cdot \lambda_{\text{mfp}}}{2}$$

3.4.5 Mean Free Path

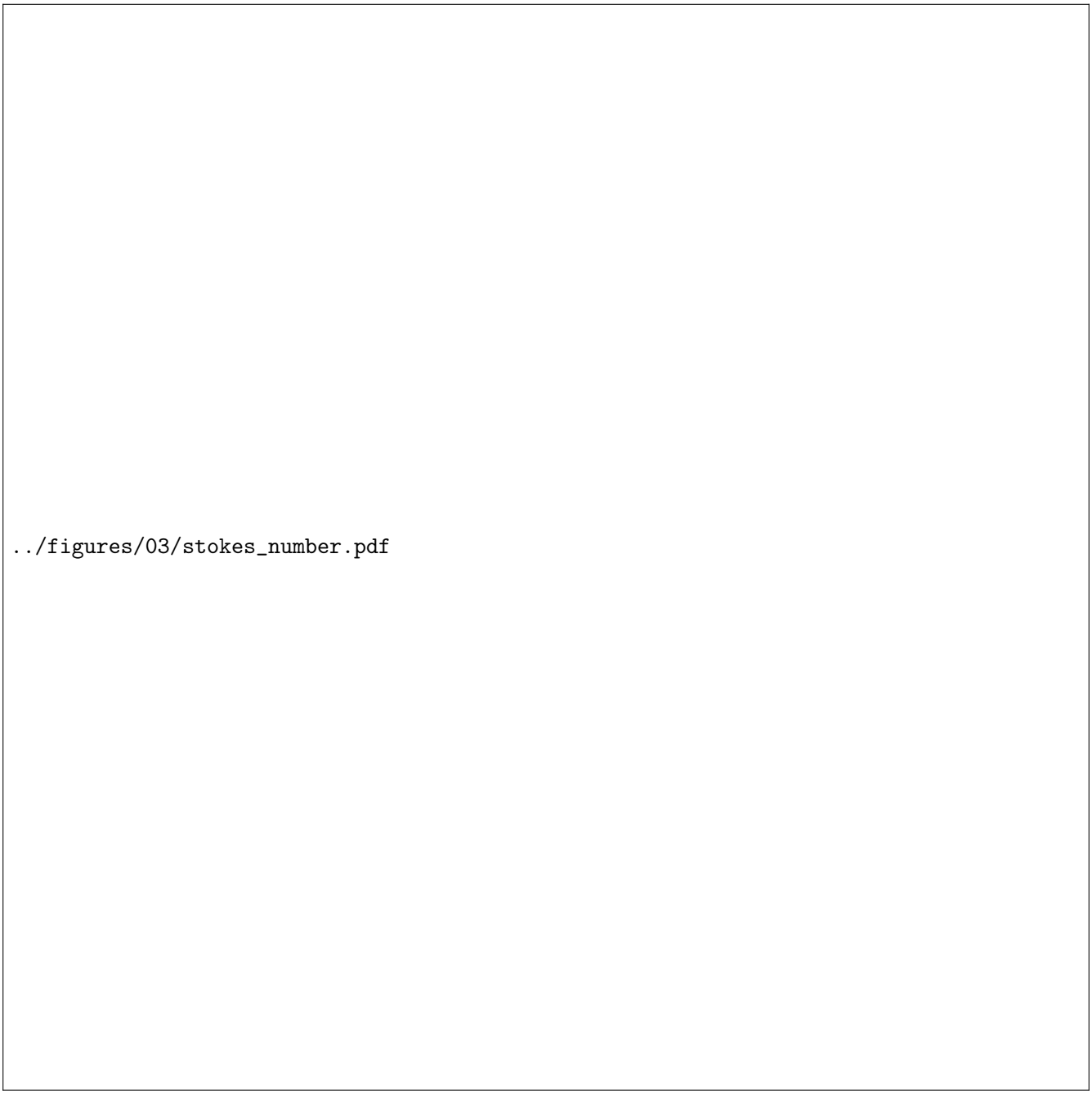
$$\lambda_{\mathrm{mfp}} = \frac{1}{n_g \cdot \sigma_{\mathrm{H_2}}}$$

with

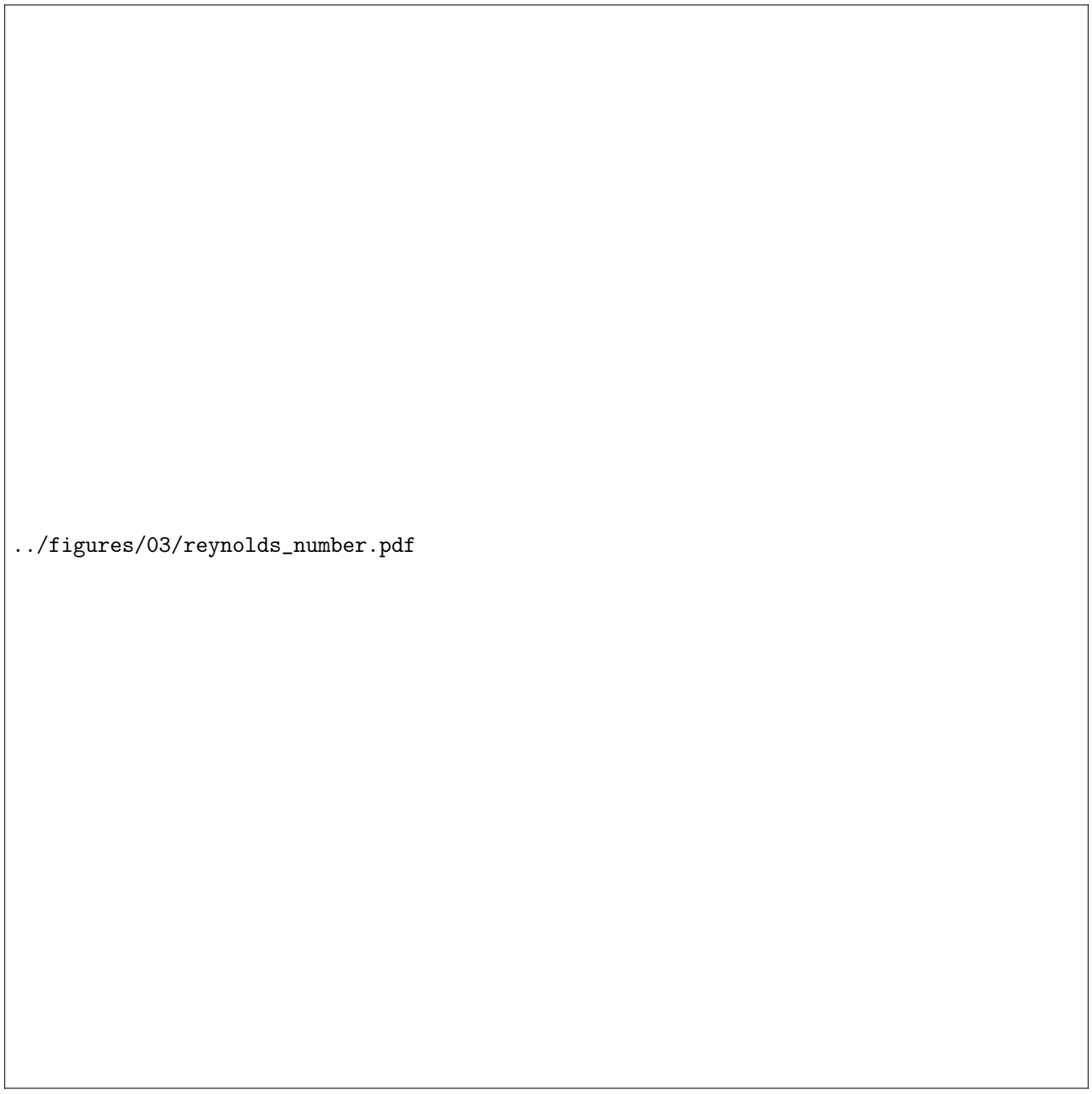
- (volume) particle number density

3.5 Properties of the Dust inside the PPD

3.5.1 Stokes Number



3.5.2 Reynold's Number

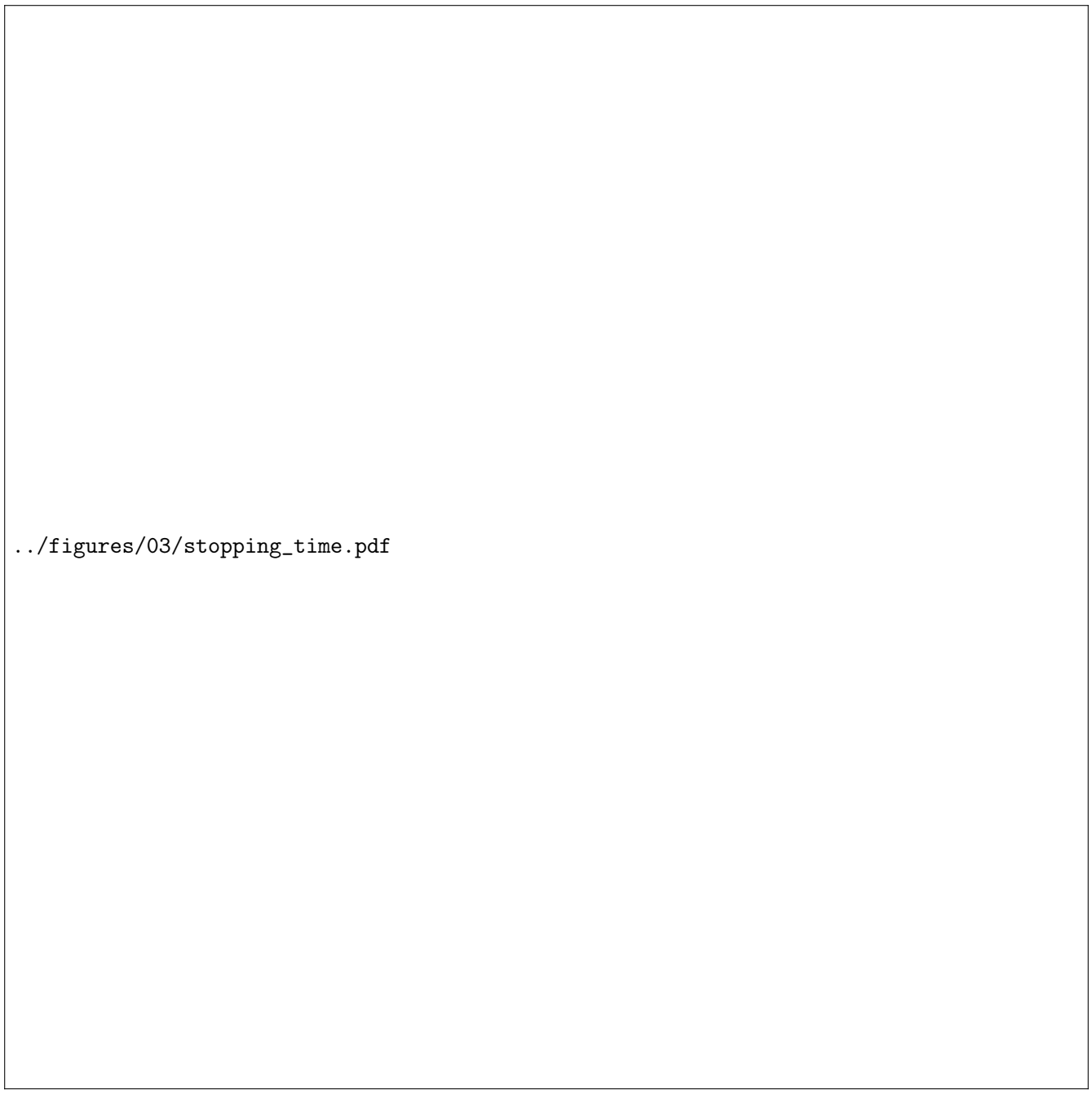


3.5.3 Stopping Time

= the ratio of the momentum of a particle divided by the drag force acting on it.

$$\tau_{\mathrm{s}} = \frac{mv}{F_d}$$

$$\tau_s = \begin{cases} \frac{\rho_s a}{\rho_g \bar{u}} & \text{for } \lambda_{\mathrm{mfp}}/a \gtrsim \frac{4}{9} \\ \frac{2\rho_s a^2}{9\nu_{\mathrm{mol}}\rho_g} & \text{for Re} < 1 \\ \frac{2^{0.6}\rho_s a^{1.6}}{9\nu_{\mathrm{mol}}^{0.6}\rho_g^{1.4}u^{0.4}} & \text{for } 1 < \text{Re} < 800 \\ \frac{6\rho_s a}{\rho_g u} & \text{for Re} > 800 \end{cases}$$


../figures/03/stopping_time.pdf

4 Formulation of a Discretized Model for Dust Coagulation

4.1 Introduction, Overview, & Goals for this Section

4.2 Discretization of the Mass Axis

4.2.1 QUES Why do we even need to discretize at all?

Due to the discrete nature of number representation on computers, for numerical treatment it is necessary to move away from the continuous representation of the mass grid, the evolution equations, and the coagulation kernel. Instead, a discrete analogon has to be constructed.

4.2.2 QUES How to discretize the mass axis in a linear fashion?

- Here: The relevant dimension is that of *mass*.
    - We look at the distribution of differently-sized particles in the disk, and how that distribution (defined on the mass axis) changes with time.
- In reality, the involved scales for the masses of particles in the disk range from those of individual atoms up to those of planets.
    - I.e., approximately from 1 proton mass to 1 Jupiter mass. That's quite the range!
    - In this thesis, we will limit this range, since we're specifically interested in the evolution of the distribution of *dust particles*.
    - Let $m_{\min}$ and $m_{\max}$ label the minimum and maximum masses "relevant to our discussion", respectively. `TODO`: Define minimum & maximum mass relevant for dust.
    - Therefore it will not make sense to use a linear representation of the mass axis, since "small masses will be under-, and large masses overrepresented". `TODO`: Is the sentence about overrepresentation correct?
- Let's not get ahead of ourselves & start with a very simple/basic model:
    - We will start off by discretizing the mass axis in a linear fashion.
    - Therefore, we will "split" the axis into a number of "grid points" / "bins".
- Let $\mathcal{N}_m$ label the number of grid points.
    - `TODO`: Decide: Is $\mathrm{N_m}$ the number of grid points, or the number of bins? Be consistent here!
- `TODO`: Define the grid spacing.

4.3 Discretization of the Particle Mass Distribution Function

4.3.1 QUES How to discretize the dust particle mass distribution function?

$$n_i := n(m_i)$$

To be more specific, the current time value should be included into the notation as well. We define:

$$n_{i,\xi} := n(m_i, t_\xi)$$

4.4 Discretization of the Smoluchovski Coagulation Equation

4.4.1 QUES How to discretize the Smoluchovski equation?

With $i, j \in \mathbb{N}$ "being indices":

$$\frac{\mathrm{d}n_k}{\mathrm{d}t} = \sum_{i=1}^{\mathcal{N}_m} \sum_{j=1}^{\mathcal{N}_m} K_{kij} \cdot n_i \cdot n_j$$

Equivalent formulation: [which one to use? I like the lower one <3]

$$\frac{\mathrm{d}n_k}{\mathrm{d}t} = \sum_{i=0}^{\mathcal{N}_m - 1} \sum_{j=0}^{\mathcal{N}_m - 1} K_{kij} \cdot n_i \cdot n_j$$

4.5 Integration of the Smoluchovski Equation

4.5.1 QUES How to discretize the time axis in a linear fashion?

Let the temporal domain be divided into $\mathcal{N}_t$ discrete "bins". Then, let ξ be the index used to uniquely identify each time step, i.e.

$$\xi \in \mathbb{N} \cap [0, \mathcal{N}_m - 1]$$

When discretizing the time axis in a linear way, this leads to

$$t_\xi = t_0 + \xi \cdot \Delta t$$

Here, t_0 labels the time value at the start of the simulation (can be set to $t_0 = 0$ without loss of generality), and Δt labels the size of each time step.

The step size needs to be chosen sufficiently small as to assure stability of the integration scheme.

[...CFL criterion...] [...implicit integration...]

4.5.2 QUES What is the difference between additive/multiplicative incrementation of time

- TODO: Move?

1. Additive time-steps
   - straightforward approach
   - incrementation of time via

$$t_{n+1} = t_n + \Delta t$$

2. Multiplicative time-steps
   - more appropriate here [explain why]

$$t_{n+1} = t_n \cdot q_t$$

4.5.3 QUES How to integrate the Smoluchovski equation using an explicit Euler scheme?

As above, we use the notation $n_{i,\xi}$ to label the dust particle mass distribution function evaluated for the mass m_i at a point in time t_ξ.

The goal now is to compute the distribution function's value after one additional step in time, i.e. at the time

$$t_{\xi+1} = t_\xi + \Delta t$$

This can be done by making use of an explicit Euler integration scheme [when?]. For a sufficiently small time step Δt, we can thus arrive at an approximate expression for the distribution function's updated value via the transformation

$$
\begin{aligned}
n_{k,\xi} &\to n_{k,\xi} + \Delta t \cdot \frac{\mathrm{d}n_{k,\xi}}{\mathrm{d}t} \\
&= n_{k,\xi} + \Delta t \cdot \sum_{i=0}^{\mathcal{N}_m - 1} \sum_{j=0}^{\mathcal{N}_m - 1} K_{kij} \cdot n_i \cdot n_j
\end{aligned}
$$

If the step size is chosen too large, the method will break down due to the insufficient stability properties of the explicit Euler integration scheme.

Instead, [. . . implicit Euler/Radau. . .] [see above]

4.5.4 QUES How to discretize the time axis in a logarithmic fashion?

As the temporal domain spans over multiple orders of magnitude for the processes involved in planet formation, it makes sense to discretize the time axis in a logarithmic fashion. [write this differently?]

[. . .]

4.5.5 QUES How to integrate the Smoluchovski equation using implicit schemes?

5 Construction of a Simple Model for Hit-and-Stick Coagulation

5.1 Introduction, Overview, & Goals for this Section

- Here, we only take a look at the simple case of...
  - ...simple *linear* mass axis discretization.
  - ...simple *Dirac-delta* initialization.
  - ...simple *explicit Euler* integration.

5.2 Definition of the Kernel for simple Stick-and-Hit Coagulation

5.2.1 QUES How to implement stick-and-hit coagulation on a linear mass axis?

Let us first turn our attention to the simple case of stick-and-hit coagulation: Here, two particles with masses m_1 and m_2 collide, and subsequently merge into a single particle with combined mass $m_{\rm tot} = m_1 + m_2$.

Let $C(m_1, m_2)$ be the rate at which collisions occur between such a pair of particles. Then, let

$$R(m_1, m_2$$

label the rate of collision events that lead to a merging of the two particles into a single new one.

Of course, in actuality not every collision leads to a merge event. For simplicity, we will for now will assume that it does. As such, we can write

$$R(m_1, m_2) = C(m_1, m_2)$$

Later on, we will more carefully examine how the collision+sticking rate can be calculated.

[... also talk about definition of collision rate...] [... also talk about "reaction rates"...] [... difference between "collision", "collision+merge", "reaction" rates (fragmentation?)...]

[... collision+sticking rate, sticking probability, "Dust evolution w/ binning methods" eq.1.6-1.11...]

[... move the above elsewhere? -> "collisions" section?] [...]

A kernel describing the stick-and-hit coagulation process can be written as

$$K(m, m_1, m_2) = \frac{1}{2}\Big[\delta_D(m - m_1 - m_2) - \delta_D(m - m_1) - \delta_D(m - m_2)\Big] \cdot R(m_1, m_2)$$

Here, δ_D labels the Dirac delta function.

[...]

5.3 Initialization of the Particle Mass Distribution Function

5.3.1 QUES How to initialize the particle mass distribution function?

Now, let us turn our attention to the initialization of the particle mass distribution function.

Options:

1. Dirac-delta distribution
2. MRN distribution

What would be best?

- MRN is more physical than Dirac-delta. (see later chapter)
- It does not *really* make that much of a difference though.
- The information about the initialization is quickly lost anyways.
- Both 1. & 2. qualitatively lead to the same results (even if not quantitatively).

Observation:

- Using the MRN distribution leads to a significant slowdown of the integration. (minutes, instead of seconds)
- This might be because I chose $m_{\max}$ to be the largest mass grid bin.
  - Do not do this!
  - This is unphysical!
  - Instead use $\approx 0.5\mu$m
- Test whether this problem persists!

5.4 Visualization of Numerical Integration Results

5.4.1 QUES Why does the mass error increase over time? -> "Mass Outflow"

6 Implementation of a Logarithmic Representation for the Mass Axis

6.1 Introduction, Overview, & Goals for this Section

6.2 Reasons for Making Use of a Logarithmic Representation

6.3 Challenges Associated with a Logarithmic Representation

6.4 Definition of the Logarithmic Mass Axis

6.4.1 QUES How to discretize the mass axis in a logarithmic fashion?

6.5 Implementation of the Kovetz-Olund algortihm

6.5.1 Reasons for the Necessity of the Kovetz-Olund Algorithm

Even in a highly simplified scenario, where only hit-and-stick coagulation is included, the definition of the kernel K_{kij} is not at all trivial. To assure both the consistency and accuracy of the algorithm, one has to take care of two separate problems, namely:

1. The conservation of mass *has* to be assured, otherwise the numerical solution can not be assumed to remain stable for long. In the case of stick-and-hit coagulation, this means that for every pair of colliding particles, a single new particle has to be created. At the same time, the two initial particles have to be removed from the distribution. During this process, the total mass should remain unaffected down to machine precision.

2. When using a logarithmically spaced grid for the discretized mass axis, it can not be assumed that after a collision of two dust particles with masses m_i and m_j the resulting particle will carry a mass $m_k = m_i + m_j$ whose value can be mapped trivially onto the grid. In general, the corresponding index will not be an integer, and instead lie between somewhere between the two neighboring grid points with indices k and $k+1$. Therefore, the result of the merging of m_i and m_j has to be divided in some sensible way between these two neighboring bins.

6.5.2 Implementation of the Kovetz-Olund Algorithm

An elegant way for solving the two problems listed above is given in the paper by [Kovetz & Olund, 1969], where they used the following procedure:

1. The stick-and-hit coagulation kernel is split into two parts. The first is the *gain* of particles in bin k due to the collision of particles from the bins i and j. The second is the *loss* of particles from bin k due to collisions of particles in bin k with particles from any other bin j.

   Using this separation into gain & loss, the dust particle mass distribution's temporal derivative can be expressed in the following form:

   $$\frac{\mathrm{d}n_k}{\mathrm{d}t} = \sum_{i,j} K^{\mathrm{gain}}_{kij} n_i n_j - \sum_j K^{\mathrm{loss}}_{kj} n_k n_j$$

   In other words, the total kernel [from above, cite eq.] can be written as

   $$K_{kij} = K^{\mathrm{gain}}_{kij} - K^{\mathrm{loss}}_{ij} \delta_{ki}$$

   Splitting the kernel like this into a gain & a loss term is a quite general approach, and can be used in more complex scenarios as well (including e.g. particle fragmentation processes).

2. For the scenario of pure hit-and-stick coagulation, a unique discretization of the kernel can be defined such that both the number of particles and the conservation of total mass are handled correctly. To do this, consider a pair of colliding particles with indices i and j. Then, let the index $\bar{k}$ be chosen in such a way that the condition

   $$m_{\bar{k}} \leq m_i + m_j < m_{\bar{k}+1}$$

   is satisfied.

3. As stated before, in hit-and-stick coagulation, a single new particle emerges for each pair of colliding particles. Using the definitions from above, this condition can be expressed as follows:

   $$K^{\mathrm{gain}}_{\bar{k},ij} + K^{\mathrm{gain}}_{\bar{k}+1,ij} \stackrel{!}{=} K^{\mathrm{loss}}_{ij}$$

4. The second condition is that of mass conservation, which can be written as:

$$m_{\bar{k}} K^{\text{gain}}_{\bar{k},ij} + m_{\bar{k}+1} K^{\text{gain}}_{\bar{k}+1,ij} \stackrel{!}{=} (m_i + m_j) K^{\text{loss}}_{ij}$$

5. Now, for the mapping of the resulting particle's mass onto two neighboring bins, let us define a parameter ε such that

$$K^{\text{gain}}_{\bar{k},ij} = K^{\text{loss}}_{ij} \cdot (1 - \varepsilon), \text{ and}$$
$$K^{\text{gain}}_{\bar{k}+1,ij} = K^{\text{loss}}_{ij} \cdot \varepsilon$$

This assures that [equation from pt 3] is satisfied. If we now plug this definition into [equation from pt 4] and solve for ε, we arrive at

$$\varepsilon := \frac{m_i + m_j - m_{\bar{k}}}{m_{\bar{k}+1} - m_{\bar{k}}}$$

This is the algorithm of [Kovetz & Olund (1969)], and it was also used in subsequent papers like [Brauer et al. (2008)] and [Birnstiel et al. (2010)].

6.6 Implementation of Near-Zero-Cancellation Handling for Coagulation

6.6.1 QUES What is Near-Zero-Cancellation and where can it occur in general?

When using floating-point numbers following the representation defined by the IEEE-754 standard, it can occur that

$$a + b = a \qquad \text{for} \qquad b \neq 0$$

Typically, this happens when

$$|b| < \varepsilon_m \cdot |a|$$

Here, ε_m labels the *machine precision*.

[Add: How big is ε_m for an f32, how big for an f64?]

6.6.2 QUES Where can Near-Zero-Cancellation occur in our model?

Let i and j once again be the indices used to label two colliding particles. Additionally, assume now that particle i is *much smaller* than particle j.

The detailed balance approach from above requires the removal of both the big and the small particle from the mass distribution, followed by the re-insertion of a new particle carrying the initial pair's combined mass. This new particle would then have a mass which is nearly identical to that of the bigger one of the original two particles, it would be only a tiny bit heavier.

In the approach defined above this would mean that $\bar{k} = j$, i.e. the resulting particle will reside in the same bin as the larger original one. Also, it would follow that $\varepsilon \ll 1$.

Let us now take a look at the particle mass distribution in the bin $\bar{k}$ and, more specifically, by how much it changes from one timestep to the next. For this particular pair of i and $\bar{k} = j$, we can write:

$$\frac{\mathrm{d}n_{\bar{k}}}{\mathrm{d}t} = K^{\text{gain}}_{\bar{k},i\bar{k}} n_i n_{\bar{k}} - K^{\text{loss}}_{\bar{k}i} n_i n_{\bar{k}}$$

Plugging in [equation from above] leads to

$$\frac{\mathrm{d}n_{\bar{k}}}{\mathrm{d}t} = (1-\varepsilon) K^{\text{loss}}_{\bar{k}i} n_i n_{\bar{k}} - K^{\text{loss}}_{\bar{k}i} n_i n_{\bar{k}}$$

Here, the two terms alsmost cancel each other out. What remains is a contribution which is proportional to ε.

If ε is small enough, the double-precision accuracy of the floating point representation will lead to breakdown of the method. [rewrite this sentence, copied almost exactly from Kees]

6.6.3 Handling Near-Zero-Cancellation for Stick-and-Hit Coagulation

It is relatively easy to identify the particle pairs (i, j) for which the scenario detailed above will occur. Let i (without loss of generality) be the index of the larger one of the two colliding masses. Cancellation may then occur when the resulting k is equal to j.

In that case, we carry out the subtraction in [previous equation] analytically, and write:

$$\frac{\mathrm{d}n_{\bar{k}}}{\mathrm{d}t} = -\varepsilon K^{\text{loss}}_{\bar{k}i} n_i n_{\bar{k}}$$

[Elaborate on this, see "Dust Evolution with Binning Methods"]

7 Inclusion of Fragmentation Processes into the Model

7.1 "First/Simple/Naive Attempt"

7.1.1 QUES How to implement naive fragmentation?

- "Pulverization"

7.2 The MRN Distribution

- introduced by Mathis-Rumpl-Nordsieck, 1977

7.3 "Redistribution of Mass onto the Bins using MRN Distribution"

7.3.1 QUES How to implement MRN distribution (for initialization)?

Particle mass distribution function is defined by the condition:

$$\rho = \int_{m_{\min}}^{m_{\max}} n(m) \cdot m \, \mathrm{d}m$$

Ansatz: power law with $q = -11/6$ (see 'sizedistributions.pdf')

$$n(m) \sim m^q$$

Choose constant A such that

$$n(m) = A \cdot m^q$$

Plug into the integral from above

$$\begin{aligned}
\rho &= \int A \cdot m^q \cdot m \, \mathrm{d}m \\
&= \int A \cdot m^{q+1} \, \mathrm{d}m \\
&= \frac{A}{q+2} \, m^{q+2} \Big|_{m_{\min}}^{m_{\max}} \\
&= \frac{A}{q+2} \, (m_{\max}^{q+2} - m_{\min}^{q+2})
\end{aligned}$$

Rearanging for A leads to

$$A = \frac{(q+2) \cdot \rho}{m_{\max}^{q+2} - m_{\min}^{q+2}}$$

Now we only need to choose values for $m_{\min}$ and $m_{\max}$.

- For the minimum mass, no lower limit exists. For it, the value corresponding to the smallest bin can be chosen [citation needed].
- For the maximum mass, an upper limit has to be chosen. It can be left as a free parameter to be set in 'config.toml', or set to something approximating 0.5 μm. [see 2004 Dullemond & Dominik] ^ max. in interstellar medium (use for $\mathrm{m}_{\min}$?)

7.3.2 QUES How to implement MRN distribution for fragmentation?

Let m_i and m_j be the masses of two particles that are involved in a collision leading to fragmentation.

How do we model this?

- Distribute the mass $m_{\mathrm{tot}} = m_i + m_j$ onto bins with $m < m_{\mathrm{tot}}$.
- For this, use the MRN distribution $n(m) = Am^q$.
- For this, find the value of A.

- For this, use the formula from [Summarize: How to implement MRN distri].
- For this, find a value for $m_{\max}$.

How do we choose a value for $m_{\max}$?

- Can it be larger than $\max(m_i,\ m_j)$?
  - "Some merging, some fragmenting"
  - "1 + 1 → 1.5 + 0.1 + 0.1 + 0.1 + 0.1 + 0.1"
- If...
  - ...yes: It can at most be equal to $m_i + m_j$
  - ...no: It can at most be equal to $\max(m_i, m_j)$
- In either case, it can't really be *equal* to either of the two values.
  - That would be stick-and-hit in case "yes", and bouncing in case "no".
  - It can at most be equal to the mass corresponding to the next-lower bin.

Answer:

- We don't really know.
- It doesn't *really* matter that much anyhow.
- One could specify the method in 'config.toml', or just use one of the methods.

7.3.3 QUES Test integration speed with MRN distribution enabled. Is there a slow-down?

7.4 "Implementation of Near-Zero-Cancellation Handling for Fragmentation" (?)

8 Implementation of a more Realistic Model for the Dust Particle Reaction Rates

8.1 Definition of the Dust Particle Reaction Rates

8.2 Definition of the Dust Particle Collision Rates

8.2.1 QUES How to define the particle collision rate?

The collision rate for a pair of masses m_1 and m_2 can be written as

$$C(m_1, m_2, \Delta v) = \sigma(m_1, m_2) \cdot \langle |\vec{v}_1 - \vec{v}_2 \rangle$$

Here, $\sigma(m_1, m_2)$ labels the cross section for a collision event. The relative velocity between such two particles is given by the difference $\vec{v}_1 - \vec{v}_2$, its absolute value is $|\vec{v}_1 - \vec{v}_2|$, and $\langle |\vec{v}_1 - \vec{v}_2| \rangle$ is the expectation value of that quantity.

[Simplification was done here] [Elaborate: Give more precise formula]

In our notation:

$$C_{ij} = \sigma_{ij} \cdot \langle \Delta v_{ij} \rangle$$

8.3 Definition of the Collision Cross Section

8.3.1 QUES How to implement realistic collision cross sections?

The cross section for a collision of two particles i and j with radii a_i and a_j can be written as

$$\sigma_{ij} = \pi \cdot (a_i + a_j)^2$$

Here, the radii can be computed from the corresponding masses m_i and m_j from equation [Summarize: Assumptions needed for the (naive/simple) particle mass distribution.].

[Add plot here]

8.4 Definition of the Relative Particle Velocity

8.4.1 QUES How to implement realistic relative velocities between dust particles?

9 Implementation of a more Physical Model for the Relative Particle Velocities

9.1 Definition of the Relative Velocity due to Radial Drift

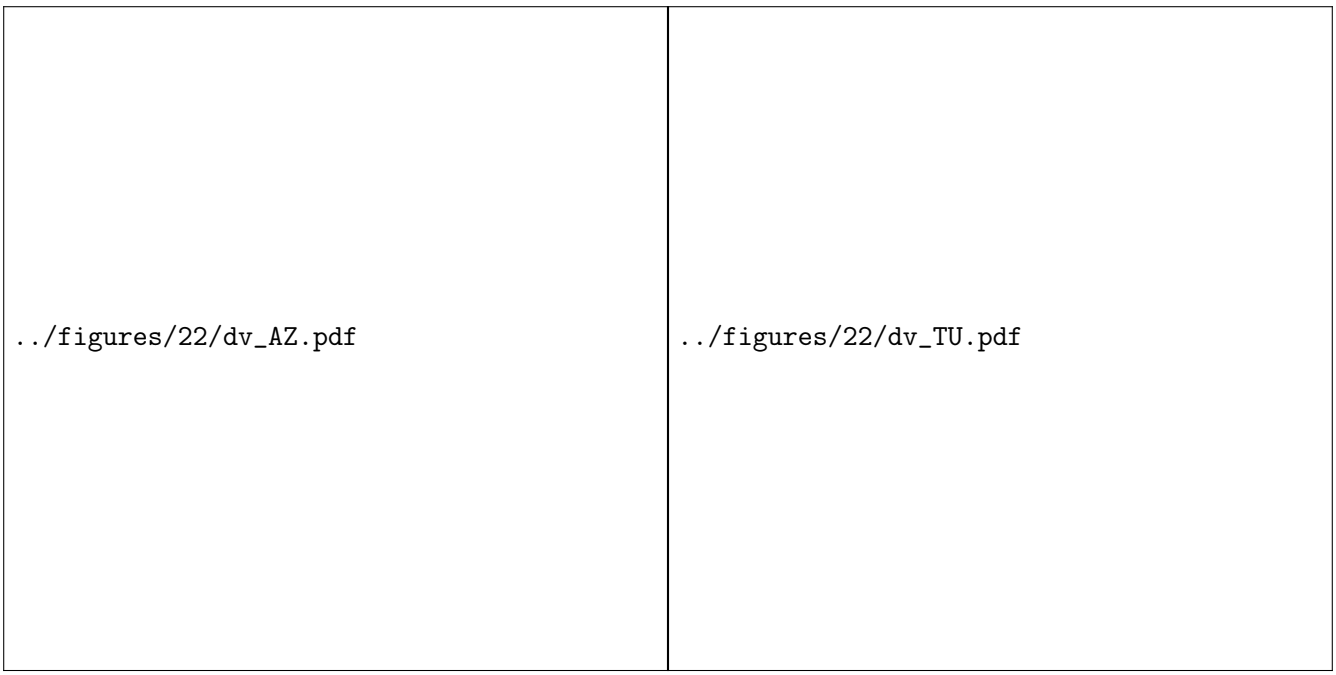
9.2 Definition of the Relative Velocity due to Differential Settling

9.3 Definition of the Relative Velocity due to Turbulence

9.4 Definition of the Azimuthal Relative Velocity

10 "Recap of what we have so far"

11 Implemention of Stochastic Kernel Sampling



(a) 1a (b) 1b

Figure 1: plots of....

12 Appendix

12.1 Glossary

12.2 Testing: Can I use *subfigures* for "multicol" LaTeX plots in Emacs org-mode?

- Yes! See below:

12.3 TODO Sort

12.3.1 Necessary Simplifications

- PPD is a 3D object.
  - This is too complex to model fully.
  - We will take advantage of symmetries.
  - This is of course a big simplification.
  - It is assumed that the model is still sophisticated enough to arrive at realistic results. (TODO: Argue why this is possible.)
- In reality, disk properties are functions
  - $f(x, y, z)$ of the three cartesian coordinates, or
  - $f(r, \phi, z)$ in cylindrical coordinates. (TODO: Elaborate on meaning of variables.)
- Simplifications
  1. Assume radial symmetry.

$$f(r, \phi, z) \;\rightarrow\; f(r, z)$$

(TODO: Explain why one can do this.)

2. Integrate vertically.

$$f(r, z) \;\rightarrow\; f(r)$$

(TODO: Explain why one can do this.)

- As such, the disk model used in this thesis depends only on the radial coordinate r. This is of course a very simplified model making use of a rather big assumption. (TODO: Explain for what this is good enough, and where it may fall short.)

12.3.2 Discretization of the Mass Axis

1. Introduction

   - The continuous range of masses inside the disk is partitioned into $\mathcal{N}_m \in \mathbb{N}$ "bins".
   - Change in notation: Use indices to label the mass grid bins:

$$m \to m_i \quad \text{with} \quad i \in \mathbb{N} \cap [0,\ \mathcal{N}_m)$$

   - For the lower & upper boundaries of the resulting mass grid, an appropriate choice of a minimum mass value $m_{\min}$ and maximum mass value $m_{\max}$ has to be made.

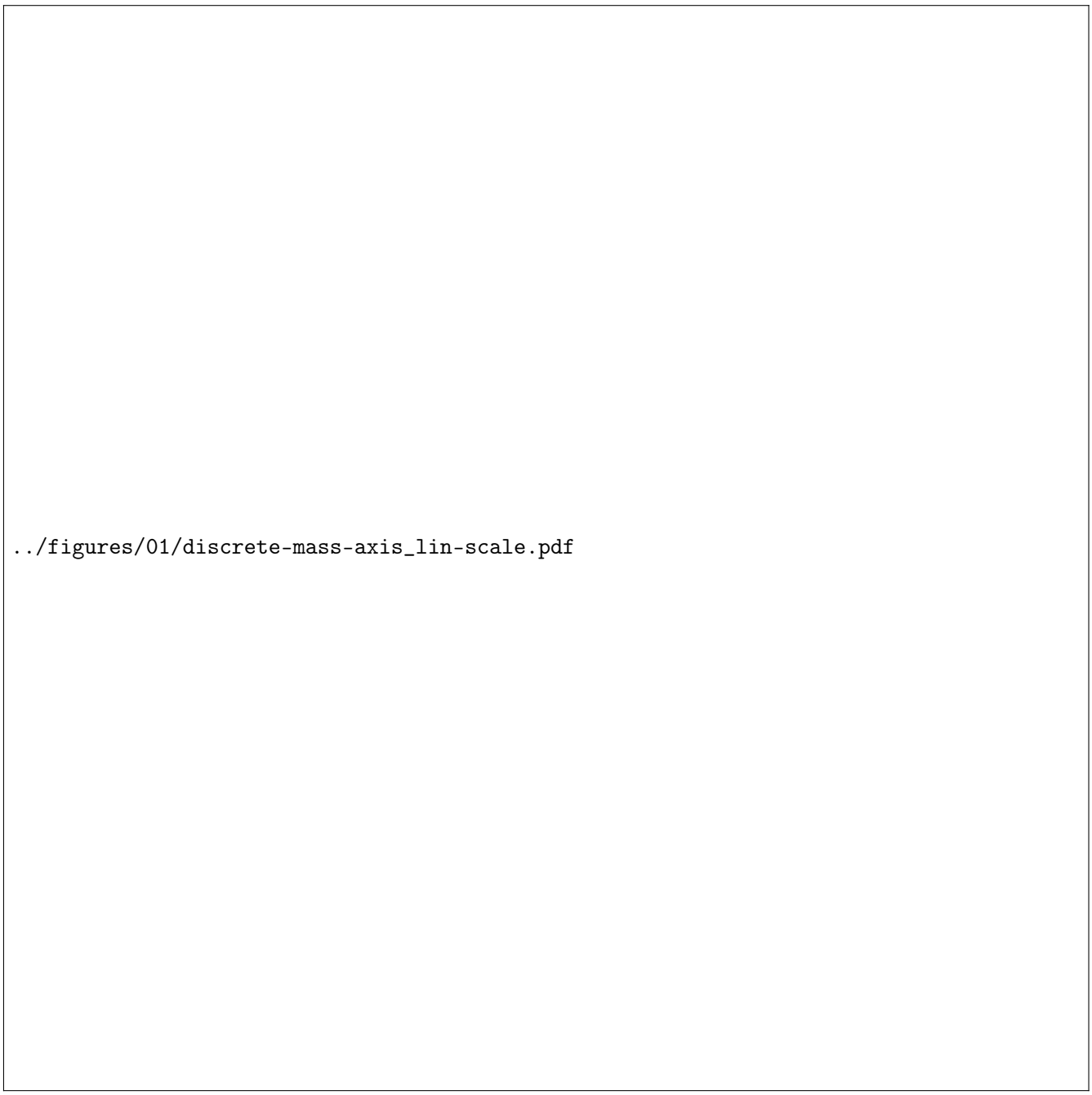
2. Discretization of the Mass Axis on a Linear Scale

   - Pro: Simpler to implement than logarithmic grid.
   - Con: Under-representation of small, over-representation of large masses.
   - Later: Logarithmic, better representation of all orders of magnitude.
   - The bins are defined in such a way that the boundaries are centered around the mass m_i. The lower and upper boundaries of a bin with index i are then given by the masses $m_{i-1/2}$ and $m_{i+1/2}$.
   - In the linear case, the "width" of each bin is given by

$$m_{i+1/2} - m_{i-1/2} = \frac{m_{\max} - m_{\min}}{\mathcal{N}_m}$$

   - Note that this width is constant across all bins, independent of their labeling indices. This will change once the switch to a logarithmic grid is made. [...]

3. Mass-Index Conversion on Linear Scale


../figures/01/discrete-mass-axis_lin-scale.pdf

4. Discretization of the Mass Axis on a Logarithmic Scale

   - As above, $\mathcal{N}_m$ is used to label the number of bins, and the grid boundaries are labeled $m_{\min}$ and $m_{\max}$ to signify the lower and upper bound of the grid, respectively,
   - In contrast to the linear grid, the "width", i.e. additive offset from one bin to the next is not the same for all bins in the logarithmic grid.
   - Instead, the *relative* mass increase from one mass grid point to the next is what stays the same for all bins i.
   - The factor describing this relative mass increase is labeld q_m. It has to be chosen in such a way as to satisfy the conditions

$$m_{i+1} = q_m \cdot m_i$$

$$m_{\max} = m_{\min} \cdot \left(q_m\right)^{\mathcal{N}_m}$$

   - This leads to the expression

$$q_m = \sqrt[N_{\text{grid}}]{\frac{m_{\max}}{m_{\min}}}$$

5. Mass-Index Conversion on Logarithmic Scale

   (a) Index to Mass With the above definition of the relative mass increase q_m from one grid point to the next, the conversion from index to mass becomes trivial. It can be described by

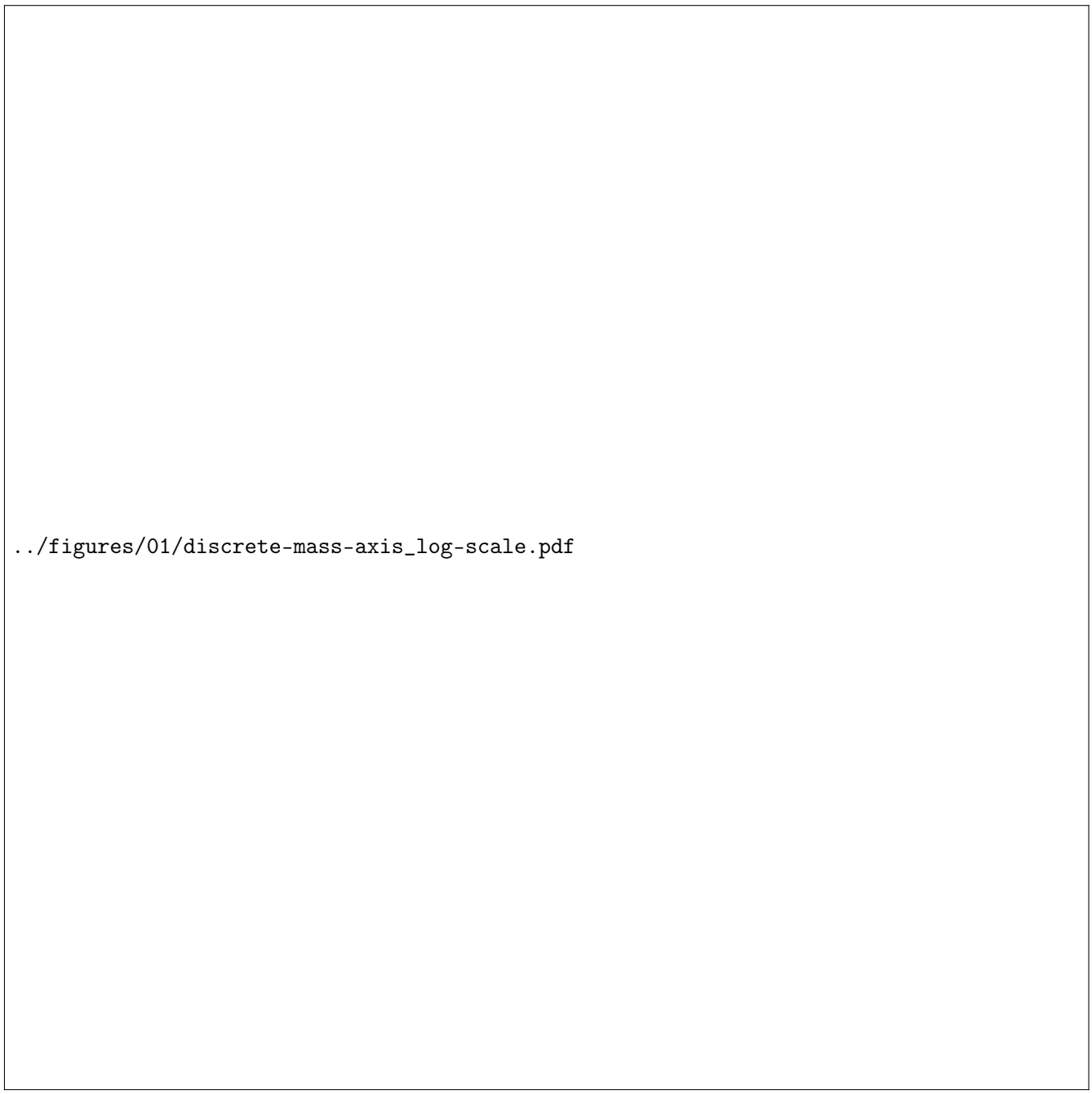
$$m(i) = m_{\min} \cdot q_m^i$$

   (b) Mass to Index To arrive at a description for the inverse conversion from mass to index, we simply rearange the above for i. The resulting law can then be written as

$$i(m) = \frac{1}{\log(q_m)} \cdot \log\left(\frac{m_i}{m_{\min}}\right)$$

   Here, attention needs to be given to one additional step: Since the resulting i is not necessarily an integer, the right-hand side of the above equation needs to be "floored" (integer division).

   [explain: why nearest-lower index, not upper? just convention?]


../figures/01/discrete-mass-axis_log-scale.pdf

6. "Things to Watch out for" When switching from a linear representation of the mass grid to a logarithmic representation one, several points need to be adressed to guarantee both stability and accuracy of the algorithm.

   For hit-and-stick collisions:

   (a) Mass conservation [rename section?] In a merge event, the masses corresponding to the two colliding particles have to be removed from the mass distribution, and exactly one new particle has to be added. It is important that this particle's mass is exactly equal to the sum of the two colliding particles' masses, to ensure conservation of mass.

   (b) "Non-integer indices of resulting mass" [rename section] In a merge event between two masses m_i and m_j, the resulting mass

   $$m_{\text{res}} = m_i + m_j$$

   does not necessarily correspond exactly to any of the mass grid points.

   Instead, m_{res} typically lies somewhere in between two grid points. In this case, the mass thus has to be divided between the nearest neighboring grid points in some sensible way.

   Let the nearest neighbors in the grid be labeled by the indices k and $k+1$, with m_k being the mass point just below the mass resulting from the merge event, and m_{k+1} the mass point just above, such that

   $$m_{\bar{k}} < m_{\text{res}} < m_{\bar{k}+1}$$

   i.e.

   $$m_{\bar{k}} \leq m_i + m_j < m_{\bar{k}+1}$$

12.3.3 Total Disk Mass Calculation

12.3.4 Particle Mass Distribution Function

- Continuous
- Let $n(m)$ be the particle mass distribution function, i.e. the number of particles per unit volume that possess a mass m.
- Total number of particles per unit volume:

$$N = \int_0^\infty n(m)\ \mathrm{d}m$$

- Total mass per unit volume: (mass density)

$$\rho = \int_0^\infty m \cdot n(m)\ \mathrm{d}m$$

- Total disk mass: (correct?)

$$m_D = \int_0^{2\pi} \int_0^\infty \int_{-\infty}^\infty \rho\ \mathrm{d}\phi\ \mathrm{d}r\ \mathrm{d}z$$

12.3.5 Discretization

Just as we discretized the range of masses, so we also have to discretize the particle mass distribution function. For this, we define:

$$n_i := n(m_i)$$

The number of particles per unit unit volume in bin i is then calculated by integration over the particle mass distribution function from the lower to the upper bin boundary:

$$N_i = \int_{m_{i-1/2}}^{m_{1+1/2}} n(m)\ \mathrm{d}m$$

The mass density ρ_i in the bin i is given by

$$\rho_i = \int_{m_{i-1/2}}^{m_{1+1/2}} m \cdot n(m)\ \mathrm{d}m$$

To arrive at the total mass density independent of particle mass, one has to sum over all bins:

$$\rho = \sum_{i=0}^{\mathcal{N}_m - 1} \rho_i$$

12.3.6 Normalization

All of the following distributions (here labeled f) are normalized in such a way that the condition

$$\int_{-\infty}^{\infty} f(x)\ \mathrm{d}x = 1$$

holds. This then assures that the total disk mass

$$\sum_{i=0}^{\mathcal{N}_m - 1} N_i = 1$$

is equal to unity (bounded, not too large, easy conservation-checking). [Is this correct?]

1. NOTE (delete later)

   - x labels the values m_i of the discretized mass grid.
   - $f(x)$ labels the discretized particle mass distribution function

$$N_i = \int_{i-\frac{1}{2}}^{i+\frac{1}{2}} n(m)\ \mathrm{d}m$$

12.3.7 Initialization

1. Dirac-delta Together with the normalization criterion from above, the Dirac-delta distribution δ_D is defined by the condition

$$\delta_D(x) = 0 \; \forall x \neq 0$$

Assuming all particles have a starting mass m_0, we can write the initial discrete mass distribution function as

$$N_i = \delta_D(m_i - m_0)$$

2. Gaussian The Gaussian distribution f_{Gauss} is given by

$$f_{\text{Gauss}}(x) = \frac{1}{\sigma\sqrt{2\pi}} \cdot \exp\left(-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}\right)$$

Here, μ labels the "expectation value" around which the distribution is centered, and σ^2 is the variance distribution's variance ("width").

Assuming a Gaussian state in the beginning of the simulation, the initial discrete mass distribution function can then be written as

$$N_i = f_{\text{Gauss}}(m_i)$$

3. Flat (homogenous) Choose N_i in such a way that

$$N_i = C \quad \forall i$$

with a constant C.

The normalization condition leads to

$$C = \frac{1}{\mathcal{N}_m}$$

and thus also

$$N_i = \frac{1}{\mathcal{N}_m}$$

4. "True Flat" Choose N_i in such a way that

$$m_i \cdot N_i = C \quad \forall i \in [0, \mathcal{N}_m - 1) \cap \mathbb{N}$$

with a constant $C \in \mathbb{R}$.

This leads to

$$N_i = \frac{C}{m_i}$$

Before we can construct the discretized initial particle density distribution function though, we need to find out what C is. A constraint for this is given by the normalization condition

$$1 = \sum_{i=0}^{\mathcal{N}_m - 1} N_i$$

Plugging in and rearranging for C leads to

$$C = \frac{1}{\sum_{i=0}^{\mathcal{N}_m - 1} \frac{1}{m_i}}$$

12.3.8 Particle Kinematics in the Disk

1. Brownian Motion

2. Radial Drift

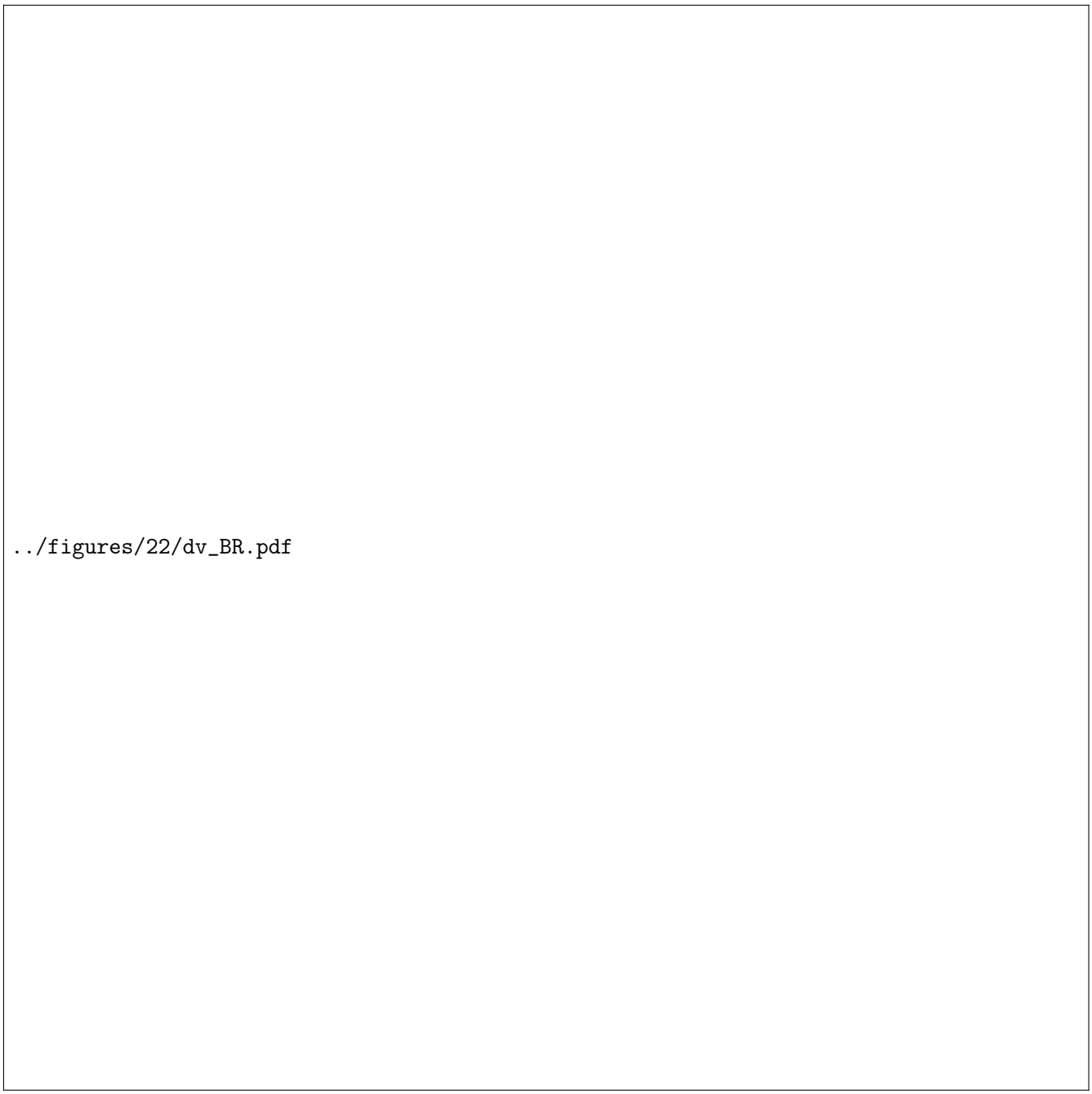
3. Azimuthal “Drift?”

5. Settling

12.3.9 Relative Velocities

1. Brownian Motion

$$\begin{aligned}\Delta u_{ij}^{\mathrm{BR}} &= \sqrt{\frac{8k_BT}{\pi} \cdot \frac{m_i + m_j}{m_i \cdot m_j}} \\ &= \sqrt{\frac{8}{\pi\beta\mu_{ij}}}\end{aligned}$$



2. Radial Drift

$$\Delta u_{\mathrm{RD}} = \left| u_r(m_1) - u_r(m_2) \right|$$

dust radial velocity:

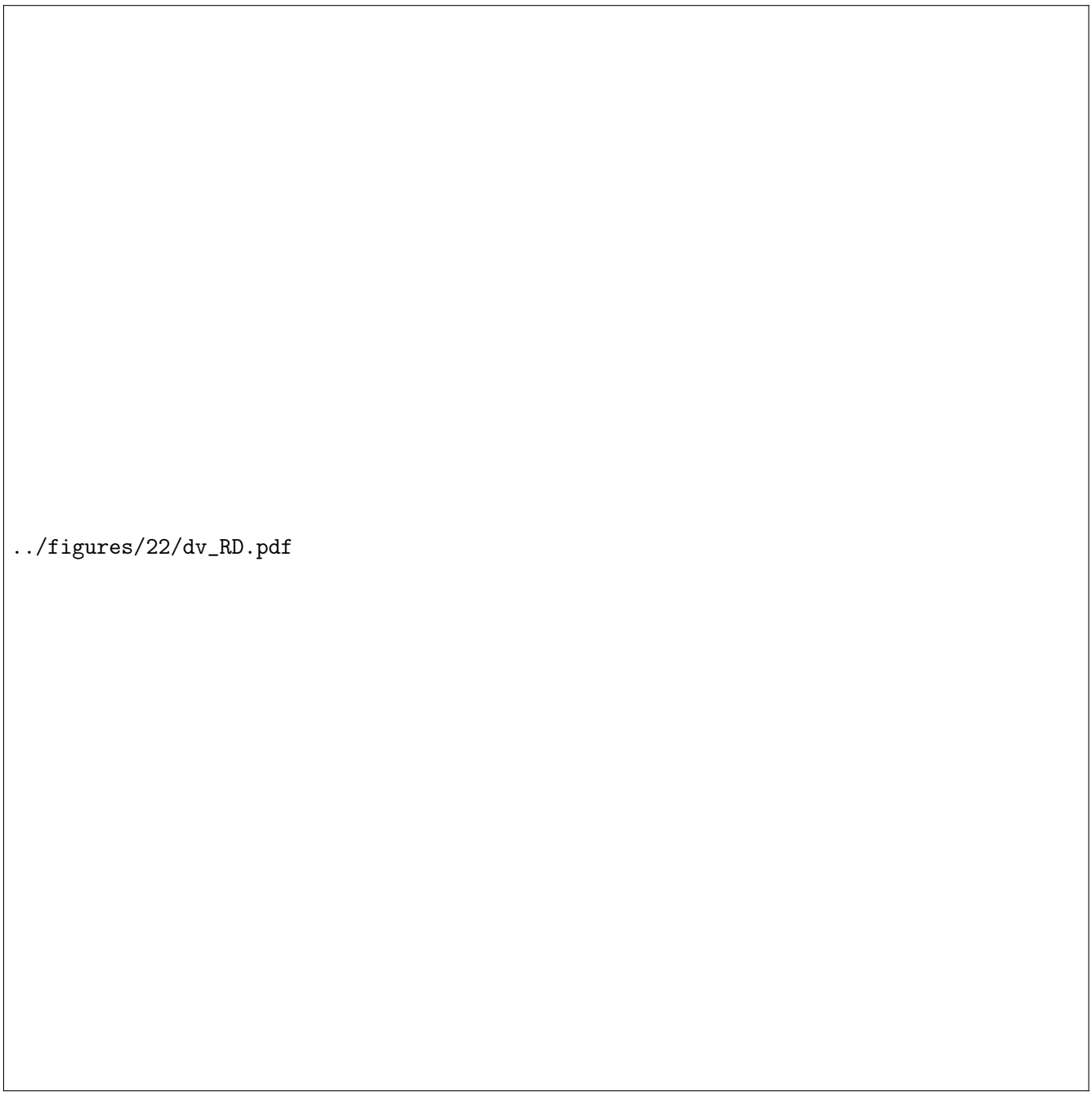
$$u_r = \frac{u_g}{1 + \mathrm{St}^2} - \frac{2u_n}{\mathrm{St} + \mathrm{St}^{-1}}$$

gas radial velocity:

$$u_g = -\frac{3}{\Sigma_g \sqrt{r}} \cdot \partial_r \left(\Sigma_g \nu_g \sqrt{r} \right)$$

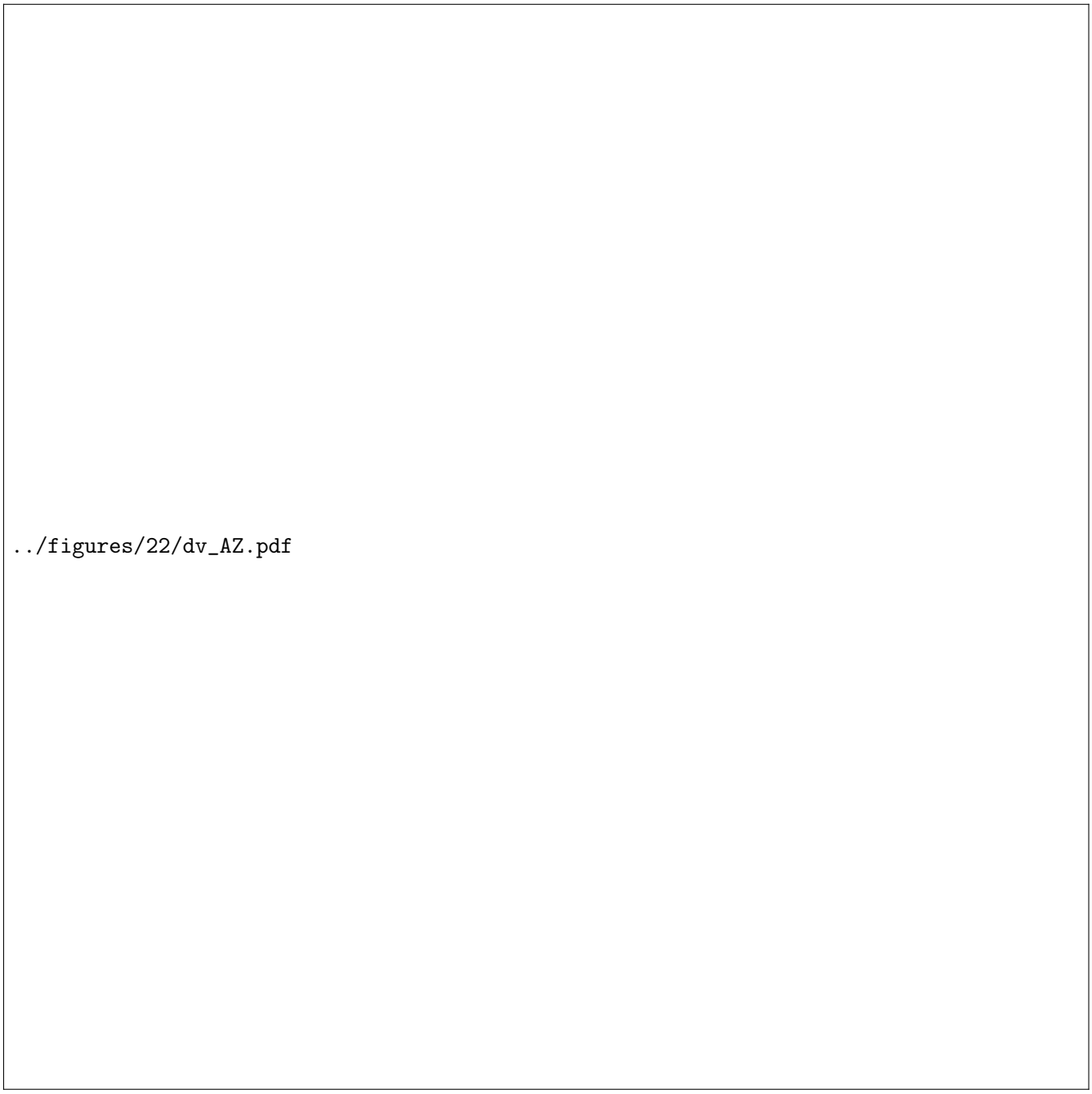
particle maximum drift velocity:

$$u_n = -\frac{E_d}{2\rho_g \, \Omega_K} \cdot \frac{\partial P_g}{\partial r}$$

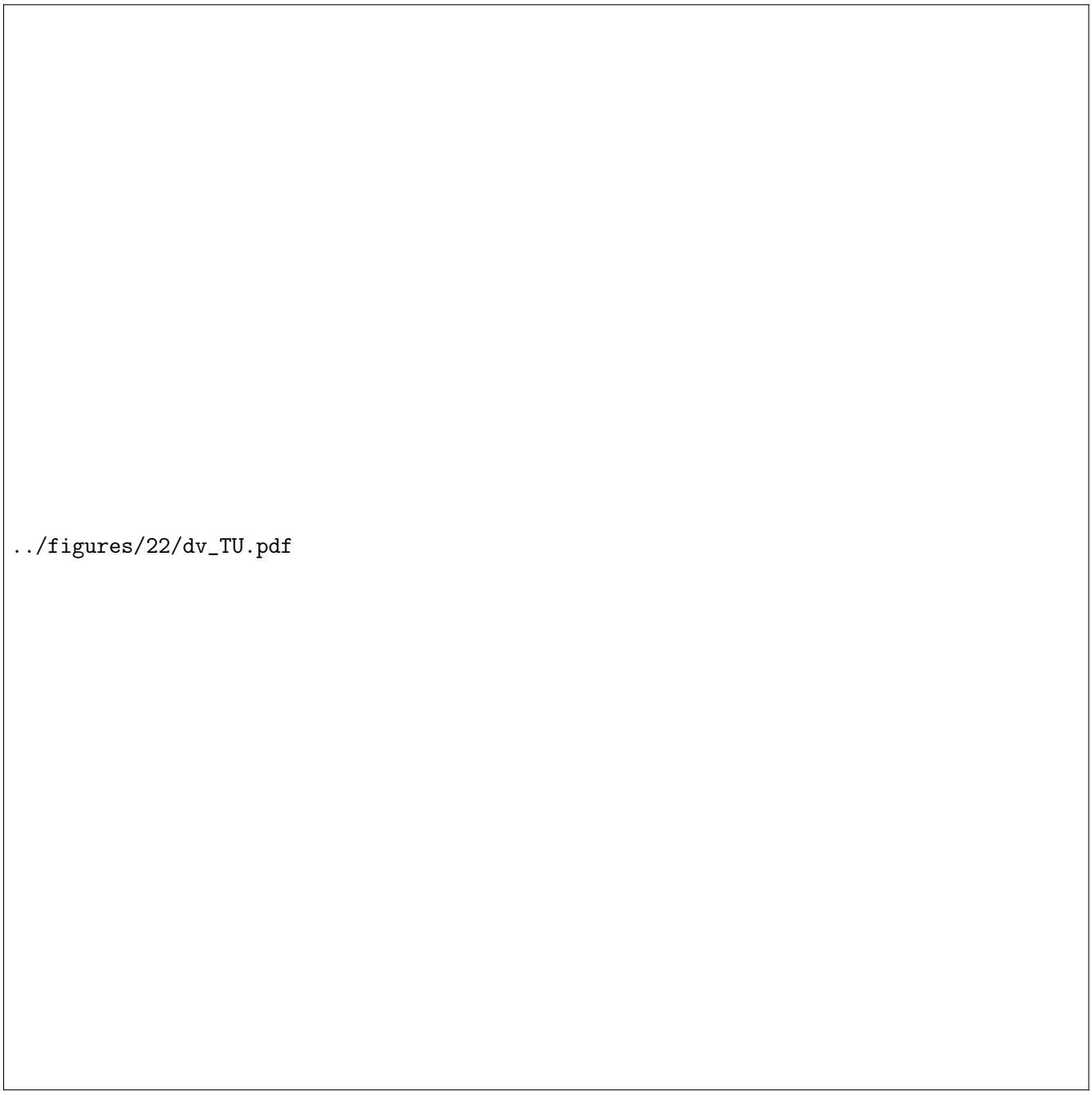

../figures/22/dv_RD.pdf

3. Azimuthal Relative Velocity

$$\Delta u_{ij} = \left| u_n \cdot \left(\frac{1}{1 + \mathrm{St}_i^2} - \frac{1}{1 + \mathrm{St}_j^2} \right) \right|$$



4. Turbulent Motion

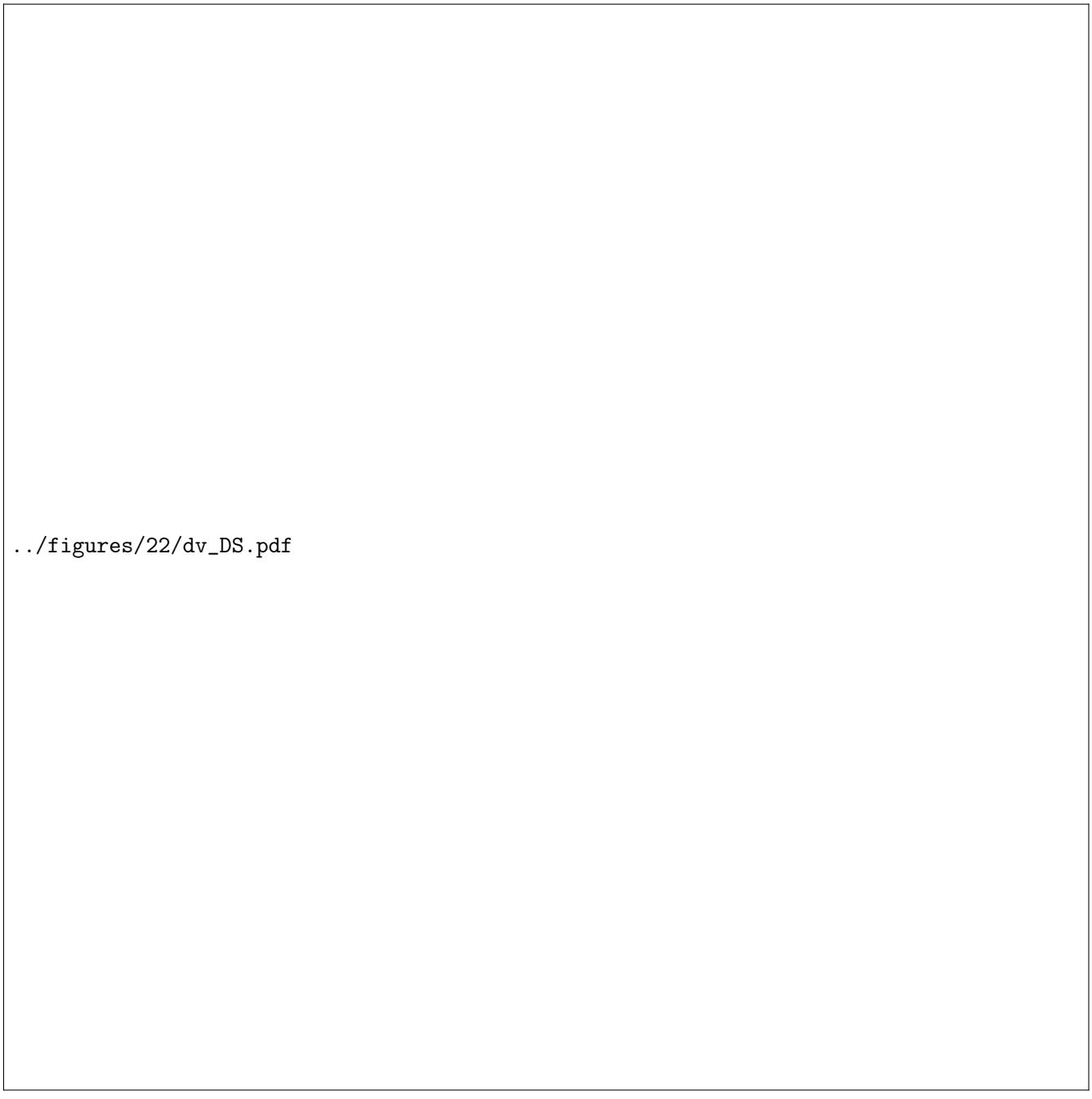


5. Differential Settling

$$\Delta u_{\mathrm{DS}} = \left| u_i - u_j \right|$$

$$u_i = ts \cdot \Omega_K^2 \cdot z$$

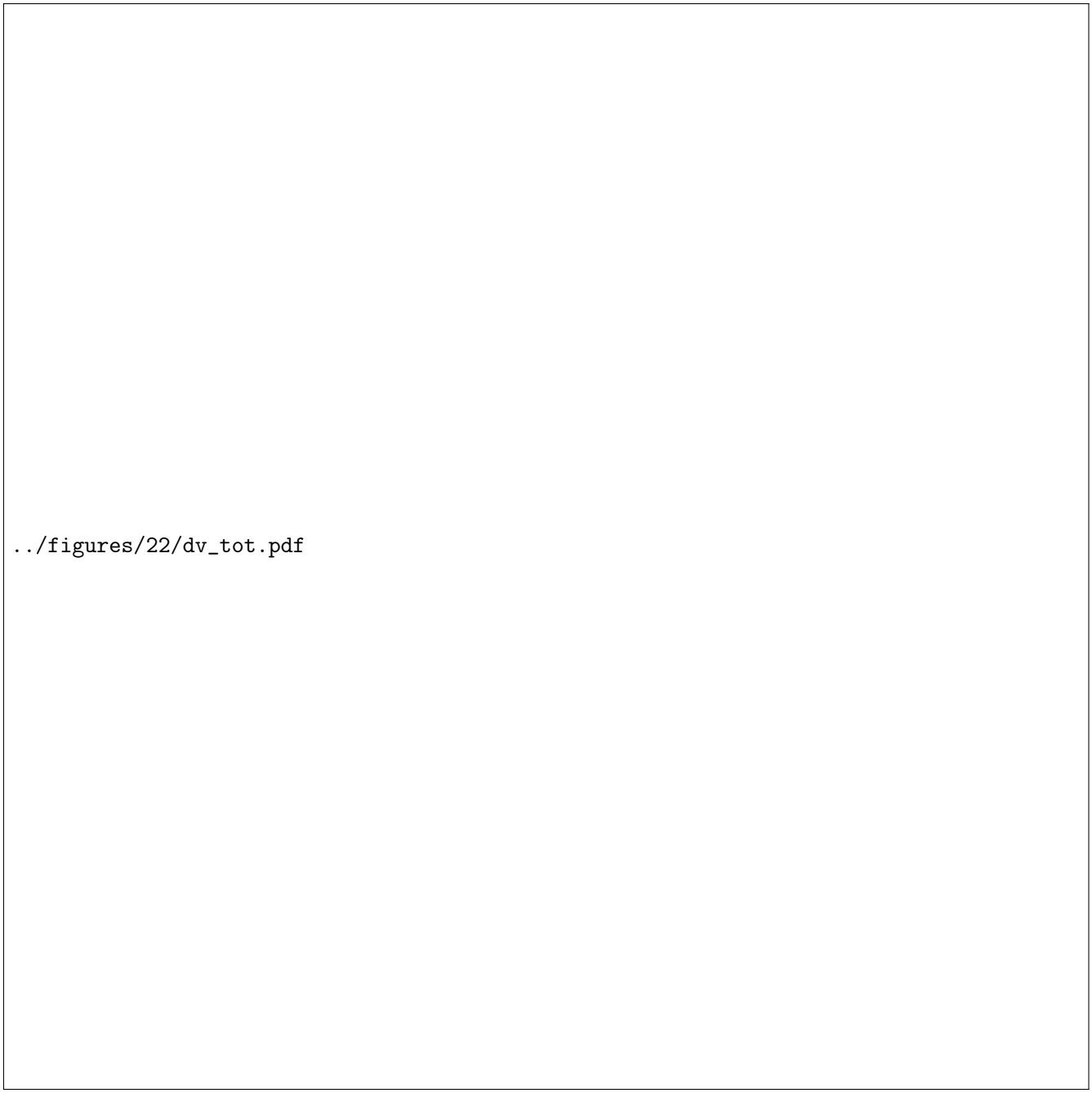
$$\mathrm{St} = \Omega_K \cdot t_s$$



6. Total Relative Velocity

$$\Delta u_{ij} = \Delta u_{\mathrm{BR}} + \Delta u_{\mathrm{RD}} + \Delta u_{\mathrm{AZ}} + \Delta u_{\mathrm{TU}} + \Delta u_{\mathrm{DS}}$$

$$\Delta u_{ij} = \sqrt{\Delta u_{\mathrm{BR}}^2 + \Delta u_{\mathrm{RD}}^2 + \Delta u_{\mathrm{AZ}}^2 + \Delta u_{\mathrm{TU}}^2 + \Delta u_{\mathrm{DS}}^2}$$



12.3.10 Particle-Particle Collisions

1. Introduction In a two-body collision, we label the particles' masses m_1 and m_2, and their velocities $\vec{v}_1$ and $\vec{v}_2$. The relative velocity is then given by the difference $\Delta\vec{v} := \vec{v}_2 - \vec{v}_1$. For its absolute value we define the shorthand notation $\Delta v := |\Delta\vec{v}| = |\vec{v}_2 - \vec{v}_1|$.

   The probability of more than two particles colliding at the same time is so tiny that three-body-collisions can be neglected entirely [citation needed].

2. Collision Outcomes

   - What outcomes are there?

     (a) Bouncing

     (b) Merging

     (c) Fragmentation

   - What happens in bouncing?

     - ...

   - What happens in merging?

     - ...

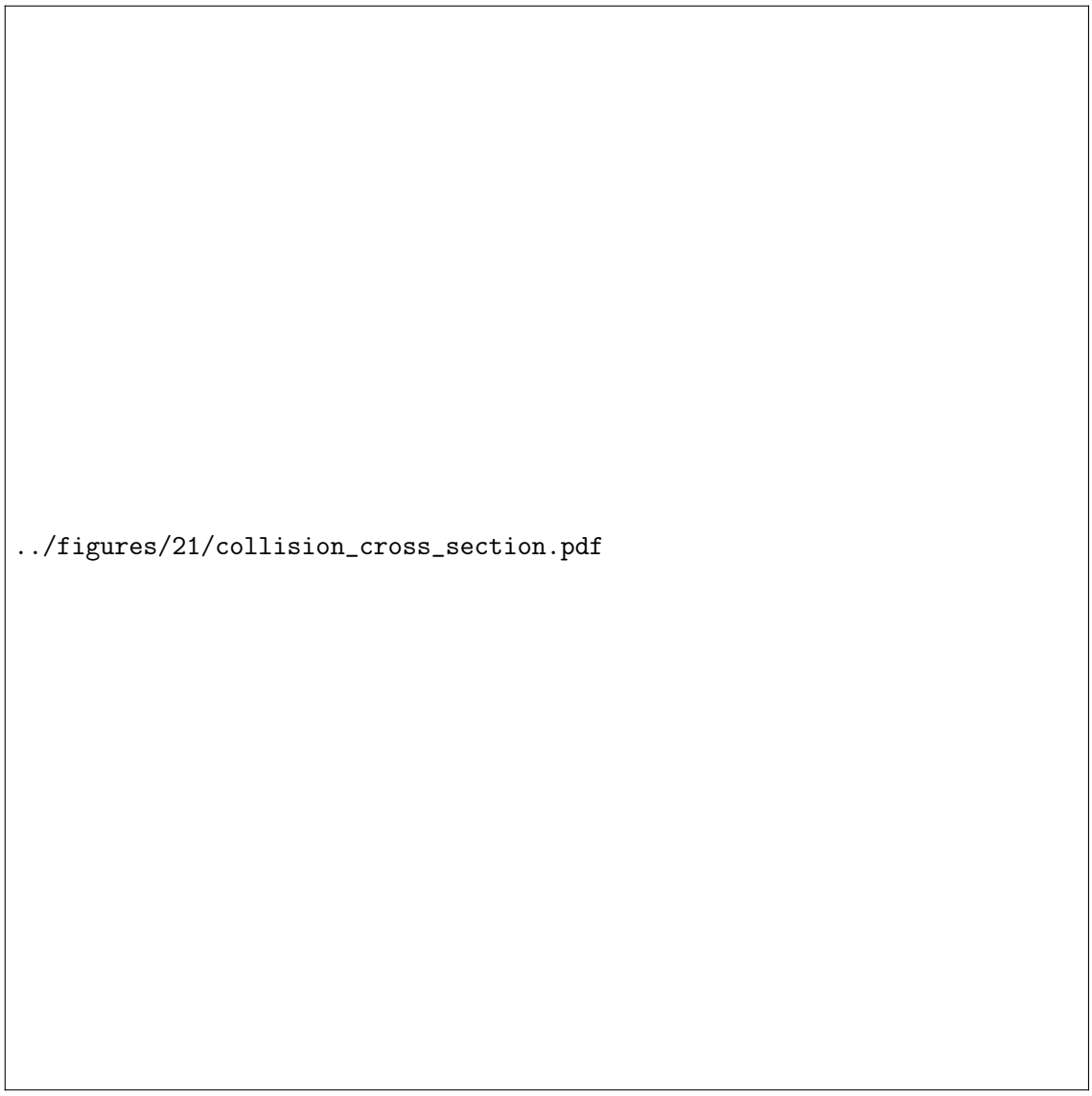
   - What happens in fragmentation?

     - ...

3. Collision Cross Section

$$\sigma_{ij} = \pi \cdot (a_i + a_j)^2$$

$$a_i \sim m_i^{1/3}$$



4. Collision Rate

$$C_{ij} = \sigma_{ij} \cdot \langle \Delta v \rangle$$

../figures/23/collision_rate.pdf

12.3.11 Smoluchovski Coagulation Equation

A commonly-used method of formulating a general description for the temporal evolution of a protoplanetary disk's particle mass distribution is given by the Smoluchovski coagulation equation:

$$\frac{\partial n(m,t)}{\partial t} = \int_0^\infty \int_0^\infty K(m, m_1, m_2) \cdot n(m_1, t) \cdot n(m_2, t) \; \mathrm{d}m_1 \; \mathrm{d}m_2$$

Here, $K(m, m_1, m_2)$ labels the *coagulation kernel*, into which all information on the collisional dynamics is encoded. It summarizes all aspects of the collisions.

- integro-differential equation, population balance equation
- introduced by M.V. Smoluchovski in 1916
- can be used to determine the temporal evolution of the number density per mass $n(m)$ of particles as they coagulate.

12.3.12 Introduction

The kernel K_{kij} is approximately equal to the right-hand side of the following statement:

$$K_{kij} \approx (m_{k+1/2} - m_{k-1/2}) \cdot K(m_k, m_i, m_j)$$

[explain why]

Here, $m_i = (m_{i+1/2} + m_{i-1/2})/2$ (same for j and k).

The discrete kernel is a 3D matrix of dimension $\mathcal{N}_m \times \mathcal{N}_m \times \mathcal{N}_m$. [why]

- $\mathrm{K} = \mathrm{K}_{\mathrm{coag}} + \mathrm{K}_{\mathrm{frag}}$

12.3.13 Gain & Loss Term

Inserting the kernel into the Smoluchovski equation

- we see: there is a positive term and a negative term (gain & loss)
- why loss term? → when two particles merge, the original particles are no longer there
- why gain term? → when two particles merge, a new particle emerges [...] [reformulate] [generalize to include fragmentation] [add equation 1.11 from "Dust Evolution with Binning Methods"]

12.3.14 The Kovetz-Olund algorithm

The two points outlined above are addressed by an algorithm outlined by [cite: Kovetz & Olund, 1969].

To solve the issues at hand by separating the kernel into two parts:

$$\frac{\partial N_k}{\partial t} = \sum_{i,j} K_{kij}^{\text{gain}} N_i N_j - \sum_j K_{kj}^{\text{loss}} N_k N_j$$

This is equivalent to the above-stated formulation of the discrete Smoluchovski equations, if

$$K_{kij} = K_{kij}^{\text{gain}} - K_{kij}^{\text{loss}} \cdot \delta_{ki}$$

Here, δ_{ki} labels the Kronecker-delta.

This approach is suitable for both hit-and-stick coagulation as well as when fragmentation processes are included into the model.

For the simpler hit-and-stick case:

- Let (i, j) be a pair of colliding (& merging) particles.
- Each collision of two particles results in exactly one new particle. This condition can be expressed as

$$K_{\bar{k},ij}^{\text{gain}} + K_{\bar{k}+1,ij}^{\text{gain}} = K_{\bar{ij}}^{\text{loss}}$$

- Mass conservation

$$m_{\bar{k}} K_{\bar{k},ij}^{\text{gain}} + m_{\bar{k}+1} K_{\bar{k}+1,ij}^{\text{gain}} = (m_i + m_j) \cdot K_i^{\text{loss}} j$$

- Linear ansatz for distributing the m_{res} onto the two bins $\bar{k}$ and $\bar{k}+1$:

$$K_{\bar{k},ij}^{\text{gain}} = (1 - \varepsilon) \cdot K_{ij}^{\text{loss}}$$
$$K_{\bar{k}+1,ij}^{\text{gain}} = \varepsilon \cdot K_{ij}^{\text{loss}}$$

- This leads to

$$\varepsilon = \frac{m_i + m_j - m_{\bar{k}}}{m_{\bar{k}+1} - m_{\bar{k}}}$$

12.3.15 Near-Zero-Cancellation Handling

1. "Cases where near-zero cancellation occurs" [rename section]

   - Collision events between particles of very different sizes/masses, e.g. very small mass m_i, very big mass m_j fuse.
   - Resulting mass $m_{\bar{k}}$ differs only by a tiny bit from m_j

   $$m_{\bar{k}} = m_i + m_j \approx m_j$$

   - The new particle will be assigned to "the same bin"

   $$\bar{k} = j$$

   - Here, $\varepsilon \ll 1$. [explain why]
   - Smoluchovski eq.:

   $$\frac{\partial N_{\mathrm{k}}}{\partial t} = K^{\mathrm{gain}}_{\bar{k},i\bar{k}} N_i N_{\bar{k}} - K^{\mathrm{loss}} N_i N_{\bar{k}}$$
   $$\frac{\partial N_{\mathrm{k+1}}}{\partial t} = K^{\mathrm{gain}}_{\bar{k}+1,i\bar{k}} N_i N_{\bar{k}}$$

   - Loss term and gain term related by: [link to eq. from K.O. algorithm]

   $$K^{\mathrm{gain}}_{\bar{k},ij} = (1-\varepsilon) \cdot K^{\mathrm{loss}}_{ij}$$

   - Plugging it in:

   $$\frac{\partial N_{\mathrm{k}}}{\partial t} = (1-\varepsilon) \cdot K^{\mathrm{loss}}_{\bar{k},i\bar{k}} N_i N_{\bar{k}} - K^{\mathrm{loss}} N_i N_{\bar{k}}$$

   - For $\varepsilon < \varepsilon_m$ (machine precision), this will not give accurate results.

2. Solution

   - Identify cases where method breaks down,
   - then carry out the cancellation analytically.

   "Undangerous" cases: All pairs $i,\ j$ for which $\bar{k} > \max(i,\ j)$

   - Here, simply use $K^{\mathrm{gain}}_{\bar{k},ij}$ and $K^{\mathrm{loss}}_{\bar{k},ij}$ (no risk of near-zero cancellation).

   "Dangerous" cases: All pairs $i,\ j$ for which $\bar{k} = max(i,\ j)$

   - Here, use $-\varepsilon K^{\mathrm{loss}}_{\bar{k},\min(i,j)}$

12.3.16 QUES Where to talk about discretization of the radial axis?

12.3.17 QUES Where to talk about Stokes?

1. Stokes' Law The force of viscosity on a small sphere moving through a viscous fluid is given by

$$F_d = 6\pi\mu R v$$

2. Stokes' Number = the ratio of the characteristic time of a particle to the characteristic time of the flow (or of an obstacle).

$$\mathrm{St} = \frac{t_0\ u_0}{l_0} = \frac{\tau_\mathrm{s}}{\tau_\mathrm{ed}}$$

   where τ_ed is the eddy turn-over time and τ_s is the stopping time

3. Stokes' Regime = the regime where the drag force on a particle is described by the Stokes law.
   - This is not directly related to the Stokes number.

12.3.18 QUES Where to talk about the Toomre parameter?

1. Toomre parameter

12.3.19 General

TODOs:

- Auf Fehler/Tuecken eingehen, Tipps geben.
- Logarithmische Zeit-Skala einfuehren.
- Beispiel mit linearem Mass Grid machen.
- Stability Analysis machen, Limits testen.
- Do local & global disk simulations.
- Importance kernel sampling machen.
- Vergleiche
  - linear/logarithmic
  - near-zero cancellation handling on/off
  - cross section on/off
  - relative velocities on/off (each)
  - Kovetz-Olund on/off
- Compare performance at the end.
  - sampling vs. no sampling
  - ? parallelization vs. no parallelization
  - ? rust vs. python
- Talk about kernel variants:
  - constant

- linear

- quadratic

12.3.20 PPD Formation

Processes:

1. Newtonian gravitational n-body interaction
   ⇒ Keplerian motion (+gas pressure)
2. Particle Collisions
   ⇒ Angular momentum averages out.
   ⇒ Cloud flattens out into a disk (oriented perpendicular to total angular momentum).
3. Merge events ("coagulation").
   ⇒ Larger bodies form (slowly, over many orders of magnitude).
   ⇒ This eventually leads to the formation of planetary cores.

12.3.21 Disk Model

1. Introduction What do we need to define?
   - Properties of central star
   - Properties of disk
   - Location inside the disk (loc. of "measurement")
2. Disk around Star What do we need to define?
   - Disk mass
     - Default:
     $$M_{\text{disk}} = q_{m,\text{disk}} \cdot M_{\text{star}}$$
   - Ratio of disk mass to stellar mass
     - Default:
     $$q_{m,\text{disk-to-star}} = 0.01$$
   - Ratio of dust to gas
     - Default:
     $$q_{m,\text{dust-to-gas}} = 0.01$$
   - Flaring Angle of Disk
     - Default:
     $$\alpha_{\text{fl}} = 0.05 \text{ rad}$$
3. Location in Disk What do we need to define?
   - Coordinate system: Cylindrical.
   - Radial distance to star r
     - Default:
     $$r = 10 \text{ au}$$
   - Vertical distance above midplane
     - TODO
   - Azimuthal angle
     - Assume radial symmetry $\rightarrow$ no dependence on azimuthal angle.

12.3.22 Particle Kinematics

1. Introduction Two competing motions:
   (a) Settling towards the mid-plane.
   (b) Turbulent stirring back up again. [?]

   Velocity profiles:
   - For pairs of masses, the relative velocities follow a probability density function $\pi(\Delta v)$.

- Relative speed differences stem from:
  - Brownian motion,
  - differential settling,
  - relative motions caused by turbulence,
  - etc. (above are most important).
- If the relative velocities are due to
  - turbulent stirring: $\rightarrow$ Maxwell-Boltzmann distribution.
  - systematic differential drift: $\rightarrow$ Dirac δ_D function.

In absence of coagulation:

- Eventually, equilibrium is reached.
- Then, the upper layers of disk are devoid of dust of particular size.
- Below a certain Z, the grains are fully mixed with the gas.
  - This leads to a more or less constant abundance.
    (e.g. Dubrulle et al. 1995; Takeuchi & Lin 2002)

Thus, the motion of particles in the disk is partly systematic (driven by e.g. Keplerian forces), and partly random (e.g. thermal motion). [. . .]

This leads to occasional particle collision events.

2. Particle Collisions

(a) Merging Probability In reality, not every collision leads to a merge event. The probability that it does, i.e. the merging probability $P(m_1, m_2, \Delta v)$, depends not only on the particle masses m_1 and m_2, but also on the absolute value of their relative velocity Δv.

If the relative velocity of two colliding particles is low, this usually leads to a merge event [citation needed]. For high values of Δv on the other hand, this assumption can not be made [citation needed]. Here, the particles either bounce, which does not change the particle mass distribution at all, or they undergo fragmentation. [...]

The merging probability, also called "sticking coefficient" has to be measured empirically. [...]

(b) Collision-and-Merging Rate

The rate of collision events that leading to a merging of the two colliding particles depends on both the collision rate $C(m_1, m_2)$, as well as on the merging probability $P(m_1, m_2, \Delta v)$. It can be written as

$$R(m_1, m_2) = \int_0^\infty C(m_1, m_2, \Delta v) \cdot P(m_1, m_2, \Delta v) \, \mathrm{d}\Delta v$$

If every collision leads to merging, then $R = C$ (hit-and-stick case). the collision-and-merging rate R is equal to the collision rate.

Since the inclusion of velocity dispersion into the calculation of R leads to an increased computational cost, a simplification can be made by using the mean value of the relative velocity $\langle \Delta v \rangle$ instead. This leads to the (mostly sufficiently [citation needed]) approximate expression

$$R(m_1, m_2) \approx C(m_1, m_2, \langle \Delta v \rangle) \cdot P(m_1, m_2, \langle \Delta v \rangle)$$

12.3.23 Results

1. First Runs
   (a) Hit-and-Stick Collision on Linear Mass Grid
   (b) Hit-and-Stick Collision on Logarithmic Mass Grid

12.3.24 Construction of a Kernel for Hit-and-Stick Coagulation

Assuming simple hit-and-stick coagulation, the kernel can be written as

$$K(m, m_1, m_2) = \frac{1}{2}\big[\delta(m - m_1 - m_2) - \delta(m - m_1) - \delta(m - m_2)\big] \cdot R(m_1, m_2)$$

Here, the coefficient $R(m_1, m_2)$ labels the collision-and-merging rate.

12.4 Testing Mass Conservation

13 References