PHYSICS LAB: RESEARCH PROJECT

Heat transfer simulations using discrete random processes

Authors:

Stanislavs Dubrovskis	S5661919
Toms Ozoliņš	S5485363
Eduard Mrug	S5749956
Markuss G. Ķēniņš	S5342988

Supervisor:
Ena Salihovic

Course Coordinator
Dr. J.S. Blijleven

Faculty of Science and Engineering
University of Groningen
Groningen, the Netherlands
Spring 2024

Abstract

The study described in this report is concerned with simulating the heat transfer process by macroscopic random movements of bouncing ball-like particles, thus proving the fundamental similarity of the probabilistic processes with rising entropy. In this study, authors use the vibrating table divided by the walls (energy barriers) into different compartments as a material through which the heat energy (small bouncing particles) propagate. The two main research questions of this work are: firstly, is this macroscopic approximation accurate, and secondly, how is the effective conductivity of such setup in the steady state case dependent on the amplitude of oscillations? In pursuit of the answers, the authors have not only conducted physical experiments but also written probabilistic as well as physical computer simulations of randomly moving particles. The results acquired in both real and virtual experiments have clearly indicated that the heat flow can indeed be simulated by the limited number of randomly bouncing particles and that the effective heat conductivity of such a setup tends to increase with an increase in the amplitude of stimulated random motion, however, the system fundamentally introduces a homogeneous dispersion (heating) term that does not follow from the heat equation. This was determined to be due to the energy barrier posed by the first wall rather than experimental error.

Contents

1 Introduction

> "Ludwig Boltzmann, who spent much of his life studying statistical mechanics, died in 1906, by his own hand. Paul Ehrenfest, carrying on the work, died similarly in 1933. Now it is our turn to study statistical mechanics."
>
> — David L. Goldstein, *States of Matter* [1]

The authors of this document report no fatalities.

Statistical mechanics dates back to the 18th century and has seen the largest developments in the late 19th to 20th century by the likes of Boltzmann, Maxwell, and Gibbs. It is of utmost importance in describing the behaviour of gases, phase transitions, heat transfer, and material science. The core idea is to use probability and statistics to describe the macroscopic properties of a large number of interacting microscopic particles. Closely tied to this is the concept of equilibrium, where the macroscopic properties remain stable while the microscopic processes are far from static, and merely equalize on average [1].

In this experiment, the authors tried to simulate continuous heat flow using macroscopic processes. Small particles and a vibrating box with several compartments were used. The particles represented the warm chaotic-moving particles in matter and the lack of them represented the cold parts, the more particles a compartment has, the more heat it has.

An approximation was made that the heat transfer is linear and also computer simulations with 50 compartments were made. These computer methods simulated the main experiment of this study but they allowed more freedom in varying different parameters.

1.1 Description of the field

Random processes have been studied and simulated a lot. Especially due to the processing power of computers, computer simulations have become more than a norm in the field of studying random processes, such as heat transfer. One of the methods is stochastic finite element modelling (SFEM) [2] which is proposed as an improvement of the widely used finite element model (FEM). The finite element model cannot deal with random parameters, unlike SFEM. The method has proved to be more accurate than statistical techniques, such as Monte Carlo simulations.

Another method that is used is a random walk [3]. The random walk's basic principle is that at every step it is randomly selected in which direction a particle moves one step. This technique is also a very good method of modelling heat transfer and has a wide range of applications, for example, in porous media. This method is based on the fact that particles, that start from the centre will distribute themselves according to the Gaussian distribution [4]. The Gaussian distribution can, of course, be cut in half resulting in a distribution along a line if all the particles were on one side in the beginning.

Not only can computer simulations be used, but the loudspeaker method has been used by many before [5]. While some of the works only look at a single ball bouncing on a loudspeaker, its movements and collisions with the speaker, others also consider interactions of a large number of smaller ball-like particles. The theoretical approaches are widely different to this phenomenon as well.

The concept of effective temperature has been explored in the so-called Granular media. Here the standard is to use glass beads and drive the oscillations by speakers, even in two or three dimensions. The data is collected using a high-speed camera or positron emission particle tracking. Notably, the particles used are monitored continuously and are of later sizes and weights (0.5 mm, 2.2 g). Multiple experimental techniques have been devised to define the effective temperature of such media consistently [6, 7].

1.2 Motivation for research

The inspiration for this experiment was taken from the second experimental exercise (E-2) of the 47th International Physics Olympiad in Liechtenstein [8]. In the experiment, the participants had to use a speaker to which a transparent pipe was attached with a wall separating two halves of the pipe. In this experiment, the authors set out to simulate continuous heat flow rather than phase transition.

1.3 The aim of this research

The aim is to answer the following **research questions**:

- How well do the discrete dynamics—distribution of the particles on the vibrating platform—describe the continuous model of heat conductivity, namely, the entropy distribution and heat flow?
- Assuming that the setup describes steady-state heat flow in a heat conductor, what is the relation between the equivalent thermal conductivity (arbitrary units) and the amplitude of the oscillating table?

The following are the **hypotheses** about the results:

- There will be an analogy between the statistical mechanical model of heat transfer and the spread of particles used in the experiment.
- Increasing either the frequency or the amplitude of the vibrating table will increase the rate at which the particles spread from the initial configuration. This is analogous to increasing the conductivity constant and observing quicker spread of heat quanta.
- Using the number of particle states to find entropy, it will increase in time as predicted by the second law of thermodynamics.

Furthermore, the goal is to deepen the knowledge about heat transfer and heat processes in general. Moreover, to determine what further research shall be most illuminating.

2 Theory

2.1 The heat equation in a continuous setting

2.1.1 Description of conduction

Consider a thin one-dimensional rod of length L and mass m with a coefficient of thermal conductivity k, and specific heat σ; the mass density of the rod is defined as $\delta = m/L$. Let the rod be described by a coordinate x along the length and time t and endow each point of the rod with a temperature value T. The theoretical interest lies in finding a physically sensible solution T, namely, a continuously second-order differentiable (class C^2) function $T : \mathbb{R} \times \mathbb{R} \to \mathbb{R}_{\geq 0}$ that maps each point along the rod x at any time t to a non-negative temperature value $(x, t) \mapsto T(x, t)$; the change of this function will describe the change of the temperature along the rod.

The existence of such a function is known, and it may be derived (for the derivation see Appendix B.3.1). The resulting partial differential equation (PDE) given by Eq. (1), namely,

$$\frac{\partial T}{\partial t}(x,t) = \frac{k}{\sigma\delta}\frac{\partial^2 T}{\partial x^2}(x,t), \tag{1}$$

is called the heat equation in one dimension. Mathematically, the constant $k/\sigma\delta$ stretches the time axis. Let $\tilde{T}(x,t) = T(x, t/(k/\sigma\delta))$, then

$$\frac{\partial \tilde{T}}{\partial t}(x,t) = \frac{\partial}{\partial t}[T(x,t/(k/\sigma\delta)] = \frac{\sigma\delta}{k}\frac{\partial T}{\partial t}(x,t/(k/\sigma\delta)) = \frac{\sigma\delta}{k}\cdot\frac{k}{\sigma\delta}\frac{\partial^2 T}{\partial x^2}(x,t/(k/\sigma\delta) = \frac{\partial^2 \tilde{T}}{\partial x^2}(x,t),$$

proving the above statement.

The above derivation may be performed in a three-dimensional space using $\mathbf{x} = (x, y, z)$ and $\Delta\mathbf{x} = (\Delta x, \Delta y, \Delta z)$, obtaining the three-dimensional heat equation given below

$$\frac{\partial T}{\partial t}(\mathbf{x},t) = \frac{k}{\sigma\delta}\left[\frac{\partial^2 T}{\partial x^2}(\mathbf{x},t) + \frac{\partial^2 T}{\partial x^2}(\mathbf{y},t) + \frac{\partial^2 T}{\partial y^2}(\mathbf{x},t)\right] = \frac{k}{\sigma\delta}\nabla^2 T(\mathbf{x},t), \tag{2}$$

where the Laplacian operator ∇^2 was used to rewrite the equation. In this project, however, only the one-dimensional case is relevant.

2.1.2 Thermal diffusivity and characteristic time

The value $D = k/\sigma\delta$ is defined as the thermal diffusivity of Eq. (1). Then within time $\tau = \Delta t$ the behaviour of the system follows $\Delta T/\Delta t = D\Delta T/(\Delta x)^2$, rearranging and setting $\Delta x = L$, $\tau := \Delta t = L^2/D$ the characteristic time τ for the system in Eq. (1) to change substantially is obtained.

Physically, thermal diffusivity D describes the rate of heat transfer within a material [9]. The dimension of this value is length squared over second, and it is often expressed as $\mathrm{cm^2\,s^{-1}}$ or $\mathrm{Hz\,cm^2}$. A larger D value implies faster diffusion of temperature within a material; whereas a large thermal conductivity means that large power may be conducted, diffusivity instead refers only to temperature. Large (order at least 0.1 $\mathrm{Hz\,cm^2}$) diffusivity is associated with materials that cool quickly, e.g. graphite ($D = 12.2\,\mathrm{Hz\,cm^2}$), gold ($D = 1.27\,\mathrm{Hz\,cm^2}$), and iron ($D = 0.2\,\mathrm{Hz\,cm^2}$), whereas small (order no more than 0.01 $\mathrm{Hz\,cm^2}$) with materials that do not disperse temperature well, e.g. alcohol ($D = 7 \times 10^{-4}\,\mathrm{Hz\,cm^2}$), brick ($D = 0.004\,\mathrm{Hz\,cm^2}$), and ice ($D = 0.01\,\mathrm{Hz\,cm^2}$) [9].

It is expected that a larger amplitude of oscillations in the experimental setup will lead to faster diffusion of particles across the compartments, which is described by a larger thermal diffusivity. It is difficult to predict the relationship between the amplitude and diffusivity, however.

2.2 The steady state solution

Some specific boundary conditions need to be introduced, which will create an easily measurable system, which will be useful during the simulations and the practical parts of the report. Consider a one-dimensional rod of length L parameterised by a coordinate $x \in [0, L]$, where the end of the rod at $x = 0$ is endowed with temperature T_1, in further analysis this point will be called the hot end, whereas the other end of the rod with coordinates $x = L$ and temperature T_2 will be called the cold end (see Fig. 1), where $T_1 > T_2$. This section concerns studying this system after time t has passed, which is much greater than the characteristic time τ of the solutions to the one-dimensional heat transfer equation given by Eq. (1), namely, $t \gg \tau$, since the solution should be stationary it means that the distribution is time-independent which significantly reduces the complexity of the problem. The claim is that the solution to this equation is of the following form:

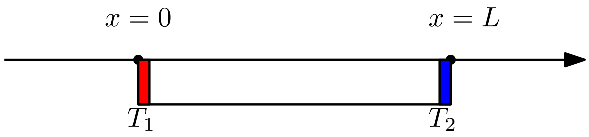


Figure 1: The one-dimensional rod with the relevant boundary conditions

$$T(x) = Ax + B \tag{3}$$

Substituting Eq. (3) into Eq. (1), it is obtained that

$$\frac{\partial^2 T(x)}{\partial x^2} = \frac{\partial^2}{\partial x^2}(Ax + B) = 0 \qquad \text{and} \qquad \frac{\partial T(x)}{\partial t} = 0,$$

proving the claim above.

Due to the fact that $T(x)$ is stationary and time-independence of temperature $T(x)$, it is evident that derivative of $T(x)$ with respect to time t is zero, which also equals the second-order derivative of the linear function. Now all that is left to do is to determine the parameters A, B as in the Boundary Value Problem. It is found that $T_1 = T(0) = B$ and $T_2 = T(L) = AL + B$, hence $B = T_1$ and $A = (T_2 - T_1)/L$. Thus obtaining the final solution of the steady state

$$T(x) = \left(\frac{T_2 - T_1}{L}\right)x + T_1. \tag{4}$$

It is now possible to compute a useful quantity—the heat leaving the system via the cold end, namely, the heat flux $\dot{q}$. The Fourier Law of Conduction [10] states that the heat flux is proportional to the negative local temperature gradient $\dot{q} = -k\nabla T$ with the coefficient of proportionality being the thermal conductivity, as before. In the case of the one-dimensional steady state solution it is obtained that

$$\dot{q} = -k\frac{dT(x)}{dx} = -k\frac{T_2 - T_1}{L}. \tag{5}$$

It is evident that the heat flux in a steady state is proportional to the constant k which depends on the parameters of the rod (both geometric and physical).

2.3 The pre-steady state solution

In the previous section the asymptotic ($t \to \infty$) behaviour was obtained; what is missing is the behaviour of the system in reaching this state, which is developed in this section.

Section 2.3.1 is adapted from [10, 11], whereas Section 2.3.2 from [11]. Consider a system that begins in an initial state with all particles heat at $x = 0$, that is, not already in the steady-state solution. In order to describe the system before it reaches this state, the heat equation with the boundary conditions and initial value

$$\begin{cases} \frac{\partial^2 T}{\partial x^2}(x,t) = D\frac{\partial T}{\partial t}(x,t), & 0 < x < L \\ T(0,t) = T_1, \quad T(L,t) = T_2, & \forall t > 0 \\ T(x,0) = \Lambda(x), & 0 < x < L \end{cases}, \tag{6}$$

where $\Lambda : \mathbb{R} \to \mathbb{R}$ is some function, must be solved.

2.3.1 Solving a homogeneous system

This system is more manageable with the following transformation: Let $h(x,t) = T(x,t) - s(x)$, where $T(x,t)$ is the solution of the system in Eq. (6), $s(x)$ is the solution given by Eq. (4) of the steady-state solution, which is to be solved for $h : \mathbb{R} \to \mathbb{R}$. Taking the derivatives of $h(x,t)$, then with appropriate simplification the following system is obtained

$$\begin{cases} \frac{\partial^2 h}{\partial x^2}(x,t) = D\frac{\partial h}{\partial t}(x,t), & 0 < x < L \\ h(0,t) = 0, \quad h(L,t) = 0, & \forall t > 0 \\ h(x,0) = \Lambda(x) - s(x), & 0 < x < L \end{cases}, \tag{7}$$

which is seen to have homogeneous boundary conditions, namely, $h(0,t) = h(L,t) = 0$ the endpoints are zero. The following procedure attempts to find separable solutions $h(x,t) = \rho(x)\theta(t)$ to Eq. (7), which are not trivial, that is, $h(x,t)$ is not zero-everywhere for all time.

Taking the derivatives, $\partial^2 h/\partial x^2 = \rho''\theta$ and $\partial h/\partial t = \rho\theta'$, where prime $[\cdot]'$ denotes the derivative with respect to the applicable variable. It is thus obtained that $\rho''\theta = D\rho\theta'$, which must be true for all time; this is possible if and only if both sides of the equation are constant for all time, namely $\exists\lambda \in \mathbb{R}$ such that $\rho''\theta = D\rho\theta' = \lambda$, which gives the system of two ordinary differential equations (ODEs)

$$\begin{cases} \rho'' - \lambda\rho = 0, \quad \text{with} \quad \rho(0) = \rho(L) = 0 \\ \theta' - \lambda D\theta = 0 \end{cases}. \tag{8}$$

To solve this system, the sign of λ must be known. It may be shown that $\lambda < 0$, which is the statement of Lemma 1 given with proof in Appendix B.3.2.

Based on the results from the proof of Lemma 1, the ρ-solutions defined on integers[1] $n \in \mathbb{Z}_{\geq 1}$ to Eq. (8) are obtained to be

$$\rho_n(x) = B_n \sin\left(\frac{n\pi x}{L}\right), \tag{9}$$

where $B_n \in \mathbb{R}$ is some constant. Now on to the time-dependent θ-solution of Eq. (8). Inserting the value $\lambda = -n^2\pi^2/L^2$, it is obtained that $\theta' + \big(n^2\pi^2 D/L^2\big)\theta = 0$, which may be solved as any first-order linear ODE using the trial function $\theta(t) = e^{rt}$, $r \in \mathbb{R}$, obtaining

$$\theta_n(t) = \lambda_n e^{-(n^2\pi^2 D/L^2)t}. \tag{10}$$

It is seen that $\theta_n(t) \to 0$ as $t \to \infty$, as is expected—the system converges to the steady state.

[1] The symbol $\mathbb{Z}_{\geq 1} = 1, 2, 3, ...$ is used instead of $\mathbb{N}$ representing the natural numbers to avoid ambiguity whether $0 \in \mathbb{N}$ or $0 \notin \mathbb{N}$, which are both commonly used conventions in mathematics.

2.3.2 Series expansion of the non-homogeneous solution

Note that the solutions in Eqs. (9) and (10) are for a fixed n, where to obtain the general solution a linear combination must be made over all n, which in this instance is a countably infinite sum with the constants combined into $\gamma_n = B_n \lambda_n$ to obtain the general solution of Eq. (7) given by

$$h(x,t) = \sum_{n=1}^{\infty} \gamma_n \sin\left(\frac{n\pi x}{L}\right) e^{-(n^2\pi^2 D/L^2)t}. \tag{11}$$

Recall that $h(x,t)$ was used to denote the difference $h(x,t) = T(x,t) - s(x)$, where a solution $T(x,t)$ must be found to the pre-steady state $T(x,t) = s(x) + h(x,t)$ given by the system in Eq. (6), thus

$$T(x,t) = T_1 + \left(\frac{T_2 - T_1}{L}\right)x + \sum_{n=1}^{\infty} \gamma_n \sin\left(\frac{n\pi x}{L}\right) e^{-(n^2\pi^2 D/L^2)t} \tag{12}$$

is the general solution of the pre-steady state. For the solution Eq. (12) to satisfy the boundary conditions, the coefficients γ_n must be found $\forall n \in \mathbb{Z}_{\geq 1}$. This is done by following a procedure alike that to finding the Fourier series coefficients; the details are exposed in Appendix B.3.3; the resulting coefficients γ_n (with $S(x) = \Lambda(x) - s(x)$) are given by

$$\gamma_n = \frac{2}{L}\left[\int_0^L \Lambda(x) \sin\left(\frac{n\pi x}{L}\right)\, dx - \int_0^L s(x) \sin\left(\frac{n\pi x}{L}\right)\, dx\right]. \tag{13}$$

The first integral is denoted $I_\Lambda(n)$ and the second $I_s(n)$. Clearly, the initial conditions $\Lambda(x)$ and $s(s)$ must be specified to find a closed-form expression of Eq. (13), which is the content of the next section.

2.3.3 Specifying a constant initial condition

In order to find the coefficients γ_n explicitly, the initial condition function $\Lambda(x)$ must be known. Based on the configuration of the experimental setup, the following parameter choices are sensible: $L = 1$, $T_1 = 1$, and $T_2 = 0$. Hence $s(x) = 1 - x$. The rest of the theory will only be concerned with an initial condition $\Lambda(x)$ non-negative in the first compartment $[0, 1/6]$ and zero everywhere else.

A sensible measure for the heat $Q_k(t)$ in compartment $k = 1, 2, ..., 6$ is the integral over this compartment $Q_k(t) = \int_{(k-1)/6}^{k/6} T(x,t)\, dx$; when a choice for $\Lambda(x)$ is made, it is thus required that the integral over the first compartment $\int_0^{1/6} \Lambda(x)\, dx$ matches the integral of $s(x)$ over this compartment, namely, $\int_0^{1/6} s(x)\, dx = 11/72$. A choice can be made to set the value of $\Lambda(x)$ in first compartment $x \in [0, 1/6]$ to be a positive constant $a \in \mathbb{R}$, hence the parameter a satisfies $11/72 = \int_0^{1/6} a\, dx = a/6$, giving $a = 11/12$, therefore

$$\Lambda(x) = \begin{cases} 11/12 & \text{if} \quad 0 \leq x \leq 1/6 \\ 0 & \text{otherwise} \end{cases}. \tag{14}$$

The integrals over the box $I_\Lambda(n) = (11/12n\pi)[1 - \cos(n\pi/6)]$ and $I_s(n) = 1/n\pi$ are computed in Appendix B.3.4, namely, Eqs. (33) and (35). The γ_n coefficients are thus obtained to be $\gamma_n = -(1/6n\pi)[1 + 11\cos(n\pi/6)]$. Therefore obtaining the pre-steady-state solution

$$T(x,t) = 1 - x - \sum_{n=1}^{\infty} \frac{1}{6n\pi}\left[1 + 11\cos\frac{n\pi}{6}\right] \sin(n\pi x) e^{-n^2\pi^2 Dt} \tag{15}$$

to Eq. (6). The result expanded to the first 1000 terms is given in Fig. 2.

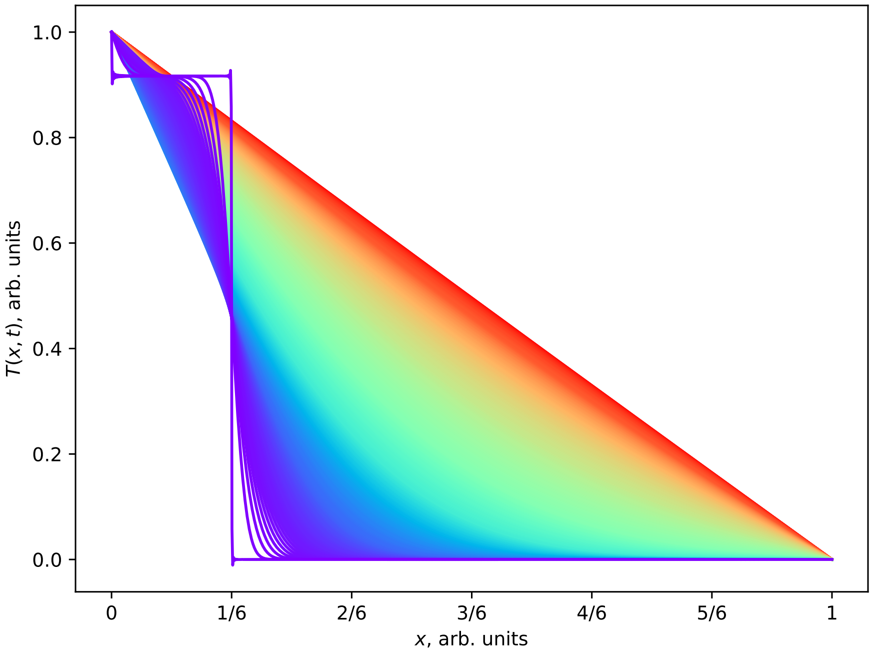


Figure 2: The theoretical solution to the pre-steady state equation given in Eq. (15) expanded up to $n = 1000$ terms. The colours correspond to different time values with purple being close to 0 and red close to 100. All units are arbitrary and thus not explicitly given; the dimension of the thermal diffusivity D is length squared over time (see Section 2.1 and Eq. (12)).

2.3.4 The distribution of the particles

The number of balls in each bin may be found by integration of the solution Eq. (15) over the corresponding bin. The details are given in Appendix B.3.4, from which it is obtained that the non-renormalised distribution of heat $Q_k(t)$ over each of the bins $k = 1, ..., 6$ is given by

$$Q_k(t) = \frac{1}{72}(13-2k) - \frac{1}{3}\sum_{n=1}^{\infty}\frac{1}{n^2\pi^2}\left[1 + 11\cos\frac{n\pi}{6}\right]\sin\left[\frac{n\pi}{12}(2k-1)\right]\sin\frac{n\pi}{12}e^{-n^2\pi^2 Dt}. \quad (16)$$

Certain choices and assumptions are now described: (1) it is assumed that the particles in the first well always behave as a gas (constant variable), (2) the convention is adopted that Eq. (16) is multiplied by some constant such that the first well is set to the value 1, namely, $Q_1(t) = 1 - f(t) = 1$, where $f(t)$ denotes the time-dependent sum in Eq. (16), and which should be zero—this process will be called *normalisation* henceforth. Hence it follows that the exact masses of the particles are irrelevant and the experimental data may be normalised to unitless values with $Q_1^{\text{exp}} = 1 \ \forall t \geq 0$ and $Q_k^{\text{exp}} < 1 \ \forall k > 1$. Notice that in the first box at steady state $(13-2k)/72 = 11/72$, hence, this may be easily achieved by multiplying Eq. (16) by $72/11$, obtaining

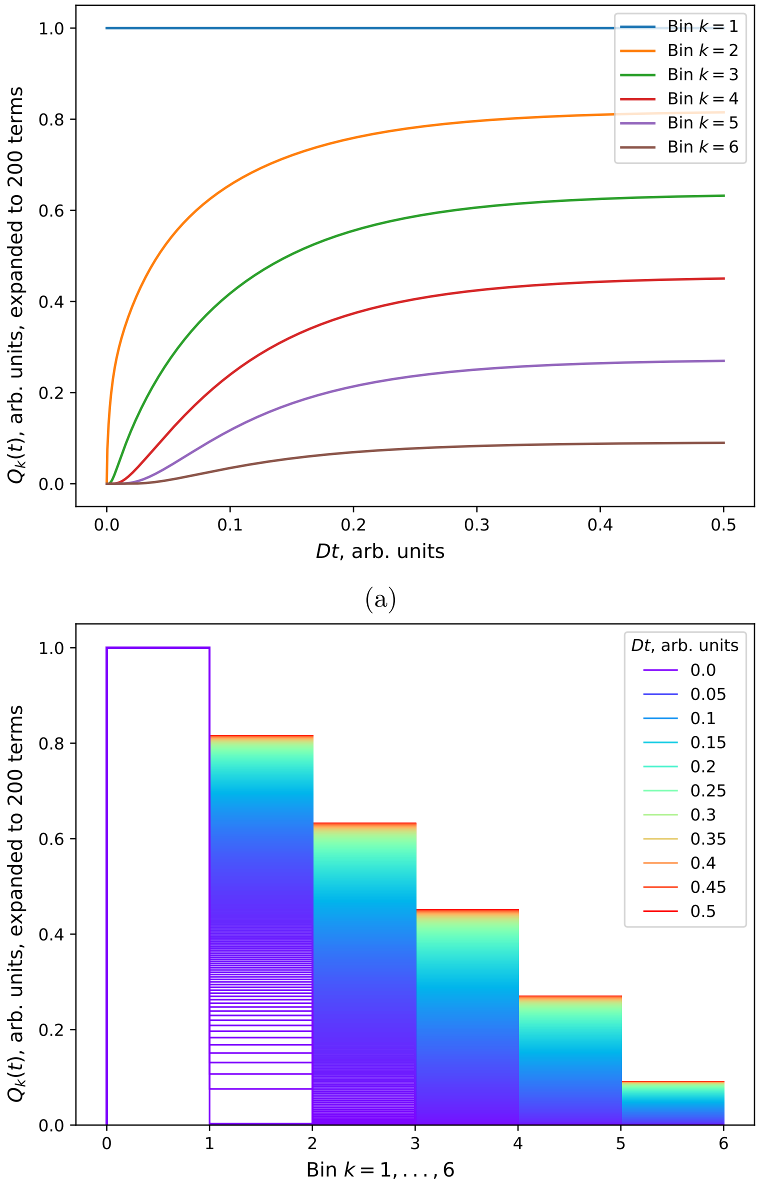
$$Q_k(t) = \frac{1}{11}(13-2k) - \frac{24}{11}\sum_{n=1}^{\infty}\frac{1}{n^2\pi^2}\left[1 + 11\cos\frac{n\pi}{6}\right]\sin\left[\frac{n\pi}{12}(2k-1)\right]\sin\frac{n\pi}{12}e^{-n^2\pi^2 Dt} \quad (17)$$

as the predicted result of the pre-steady state distribution. This result given by Eq. (17) is the main result of this theory section that will be fitted to normalised experimental data.

2.4 Computed distributions

In this subsection the predicted results according to Eq. (17) are given. The calculation were done in Python, expanding the sum in Eq. (17) to 200 terms; it was found that considering more terms does not change the results significantly. The computation and plotting script is given in Appendix A.5, and the resulting plots displayed in Figs. 3a and 3b. It is evident that the theoretical distribution $Q_1(t), Q_2(t), ..., Q_6(t)$ reaches a linear distribution as $t \to \infty$, which is approximately achieved already for $Dt \geq 0.5$. In the experiments, the time values (in seconds) are specified by the experimental procedure, and thus only the D value must be found by fitting (see Section 4.4.2).

It is evident that the first bin stays constant (as was specified), whereas the amount of heat in the second bin increases rapidly, whereas the amount in the other bins increases slowly. This will become important later when the prediction given by Eq. (17) as in Fig. 3b is fitted against data (see Fig. 11 in Appendix B.2).



(a)

(b)

Figure 3: Computed distributions of the theoretical $Q_k(t)$ values according to Eq. (17) with different Dt values (a) depending on the Dt value with each line corresponding to a different bin, (b) with respect to each bin.

3 Simulations

In order to test the mathematical theory with a large number of randomly moving particles, it was first decided to create some computer simulations, which although quasi-random, still allow the group to at least roughly determine whether their theoretical conclusions are right and if is there a point in a real-world study.

3.1 Probabilistic simulation

The first simulation that was created is entirely probabilistic, it works with N particles and Q compartments, on each of the simulation steps, each particle can go either to the right, or to the left(relative to the compartment that it is in) with a probability of p, or stay in the same compartment with a probability of $\bar{p}$, where $\bar{p} = 1 - p$. The full code of the simulation script, which is free to copy/use can be found in Appendix A.1.

Using the script below it was decided to first check weather groups understanding of the Gaussian nature of such systems is correct, so the first part of the simulation script has been launched with ($N = 10^4$ particles which initially are situated in the middle compartment and $Q = 50$ compartments after 100 steps the following particle(or heat quanta) distribution was achieved Fig. 4a:

As can be seen, this is a Gaussian which of course was expected because of the central limit theorem. If the script is launched for a longer period of time, the heat(particles) is expected to spread out and make a homogeneous distribution. So the same settings were used to make a simulation of 750 steps, getting the following distributionFig. 4b.

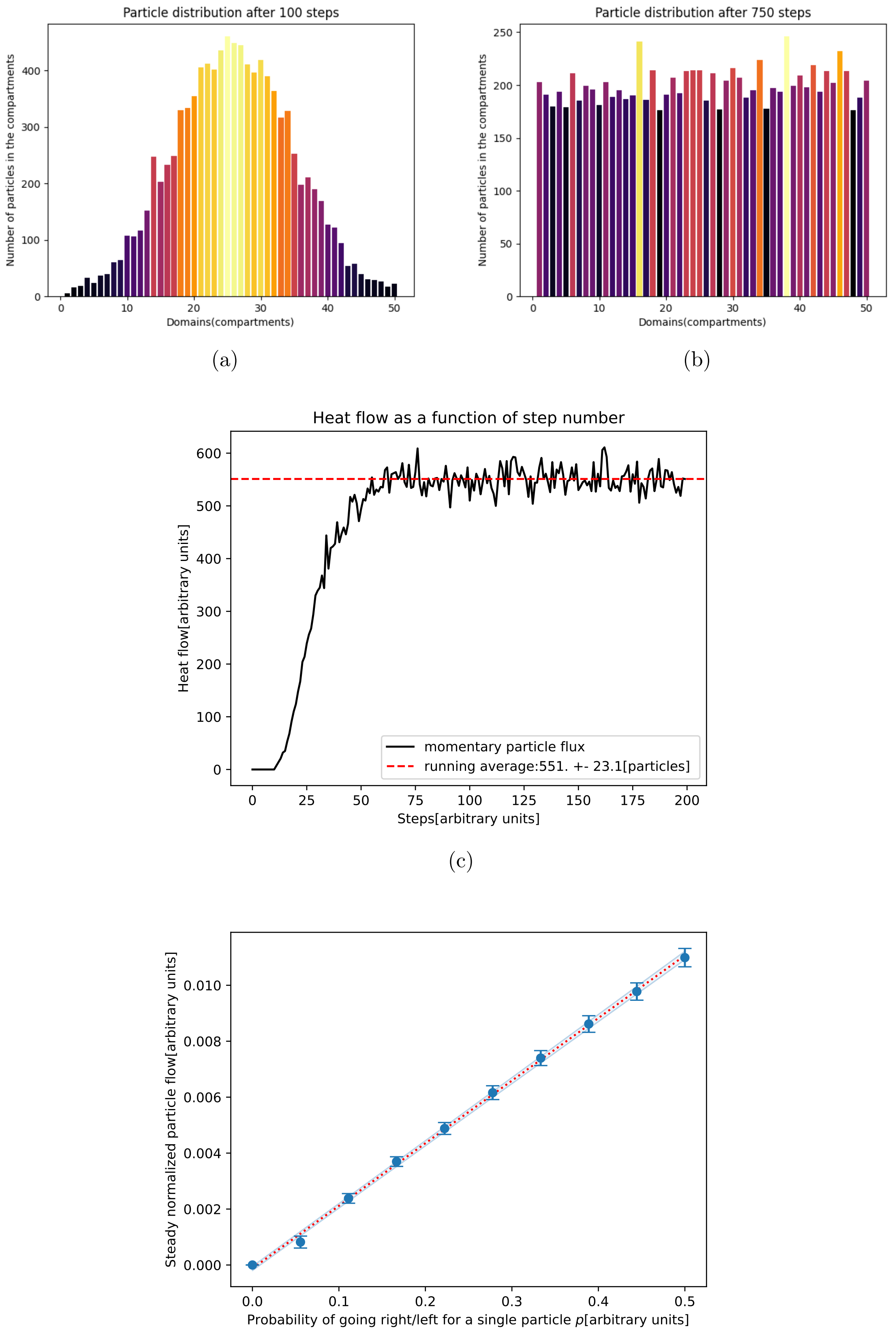
As visible from Fig. 4b the distribution is nearly homogeneous just with slight fluctuations which are inevitable when dealing with random processes.

Now, that it has been seen that a simple probabilistic model can approximate the behaviour of the heat flow, it has been decided to test whether it can simulate the stationary heat flow. For that, it is important to make sure that the simulation satisfies the boundary conditions, namely that the first compartment always contains the same number of particles(N_1) and that the last compartment is always kept at zero temperature, which means that it is always kept empty($N_{\text{last}} = 0$). The script checks these conditions on each step of the simulations, removing all the particles from the last compartment and adding the necessary amount of particles to the first compartment so that the sum thereafter in each step is still equal to the N_1.

In this setup, the number of particles that are being removed from the last compartment on each step is a precise analogue to the heat flow that goes through the system. To track this heat flow data as a function of time the simulation was launched with the following set of parameters: (10 compartments, 10^5 particles, 200 steps and $p = 0.25$, $\bar{p} = 0.5$ (unitless)). Running this simulation resulted in Fig. 4c. As can be seen from the Fig. 4c, the heat flux initially grows but then stabilises as the distribution reaches the stationary solution, which is exactly the theoretical prediction for the stationary heat flow.

Finally, it is important to know how the stationary heat flow(the quantity that is measured in the real experiment) depends on the p(probability of the single particle jumping left or right on each step of the simulation). To measure this the previous simulation was launched ten times with different values of p linearly distributed from 0 to 0.5, the N_1 and N_{last} were constant and equal for all the simulations, the stationary heat flow was recorded and its standard deviation σ, and composed the following graph:

As can be seen from the Fig. 4d, the stationary heat flow appears to be increasing linearly with probability p, with $a = (0.022 \pm 0.008)$ arb. units. Since the "temperature" difference $N_1 - N_{last}$ is constant for all of the simulations so is the "length" of the setup(number of compartments), from the Eq. (5) it can be concluded that the only way that this can be



(a)

(b)

(c)

(d)

Figure 4: (a) Particle distribution after 100 seconds. (b) Particle distribution after 750 seconds(Homogeneous). (c) Graph of the heat flow as a function of the step. (d) Graph of the stationary heat flow as a function of p.

happening is when the effective "conductivity" of the setup k is proportional to p, namely, $k \propto p$, which is the effect that the is expected in the real-life experiment as well.

3.2 Physical simulation

The previous section details a simulation that considers particle movement as purely probabilistic, whereas the experiment deals with particles in physical boxes; hence this section deals with simulating particles bouncing around in a box like that used in the setup (see Section 4.1). The script given in Appendix A.2 creates all non-interacting particles in the first compartment, where upon hitting the floor their velocity is updated from a random distribution with probability density $f(x)$ given by

$$f(x) = \frac{1}{s} e^{-|x|/s}, \tag{18}$$

where s is the scaling parameter, dimension 1/length, arb. units. Upon hitting a wall, the script checks whether the ball may bounce through it.

Taking infinitely many bins in order to observe decay from the first bin, the results are shown in Fig. 5a. It is evident that the amount in the first bin decreases rapidly until the amount is comparable to that in the second bin, and then slow decay of the first six bins is achieved as the particles move onto compartments beyond the 6th. In Fig. 5b, a steady state condition is imposed at each time step, namely, if the amount in the first bin decreases below 300, enough particles are added such that the amount is 300.

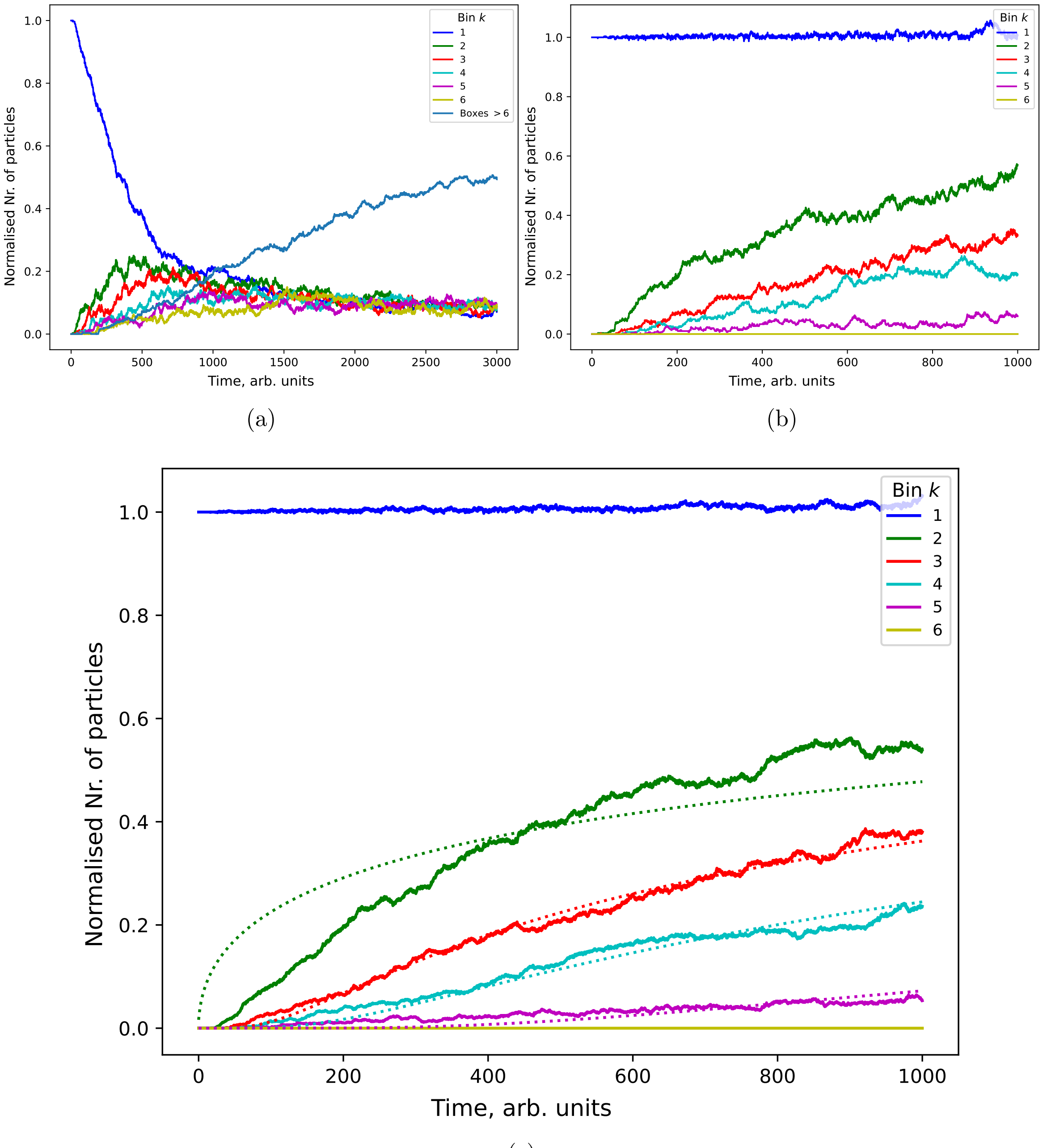
It is evident that a graph visually similar to Fig. 3a, except for the second bin, is obtained. To reduce noise, averaging over five such runs was done and the end result was normalised by dividing by 300 (the controlled number of balls in the first compartment), obtaining Fig. 5c. Fitting was done to Eq. (17), whereby the best-fitting thermal diffusivity values were obtained in arbitrary units for each box; see Appendix A.2 and Section 4.4.2 for `curve_fit`, where the `SciPy` function could be used on the value arrays themselves without rearrangement. The resulting D values are given in Table 1.

k	D, arb. units
2	$(0.204 \pm 0.002) \times 10^{-5}$
3	$(3.532 \pm 0.004) \times 10^{-5}$
4	$(6.926 \pm 0.009) \times 10^{-5}$
5	$(6.25 \pm 0.02) \times 10^{-5}$

Table 1: Thermal diffusivity D fitted to the physically simulated data (Fig. 5c) with error to each of the boxes k.

An attempt was be made to transform the thermal diffusivity values to SI units, using the fact that the length of the box was 30 arb. units, which corresponds to 20 cm in the setup (see Fig. 6a in Section 4.1), and time could be taken as seconds. This, however, simply made the values smaller and did not give much insight. It is clear that all except for $k = 2$ thermal diffusivity values lie within $(5 \pm 2) \times 10^{-5}$ arb. units.

The fact that the bin $k = 2$ does not fit well to Eq. (17) is evident by the rapid increase of $Q_2(t)$ in Fig. 3a for small time t values, which is the not the case in the simulation. Nevertheless, bins $k = 3, 4, 5$ fit the expected result reasonably well. Intriguingly, the distribution for the largest time simulated is not stepwise-linear (same difference between bins at the steady state) as in Fig. 3b, rather it is stepwise-linear for bins $k = 2, 3, 4, 5$, but not between 5–6 and 1–2; it is possible, however, that the steady state has not yet been reached.



(a)

(b)

(c)

Figure 5: The physical simulation of the number of particles normalised by dividing by 300 particles (thus unitless) in each of the bins with respect to time (a) with infinitely many bins, (b) in 6 bins by updating the amount in the first bin to be constant (c) averaging over 5 runs as in b. The dashed lines indicate the best theoretical fit, as described in the text.

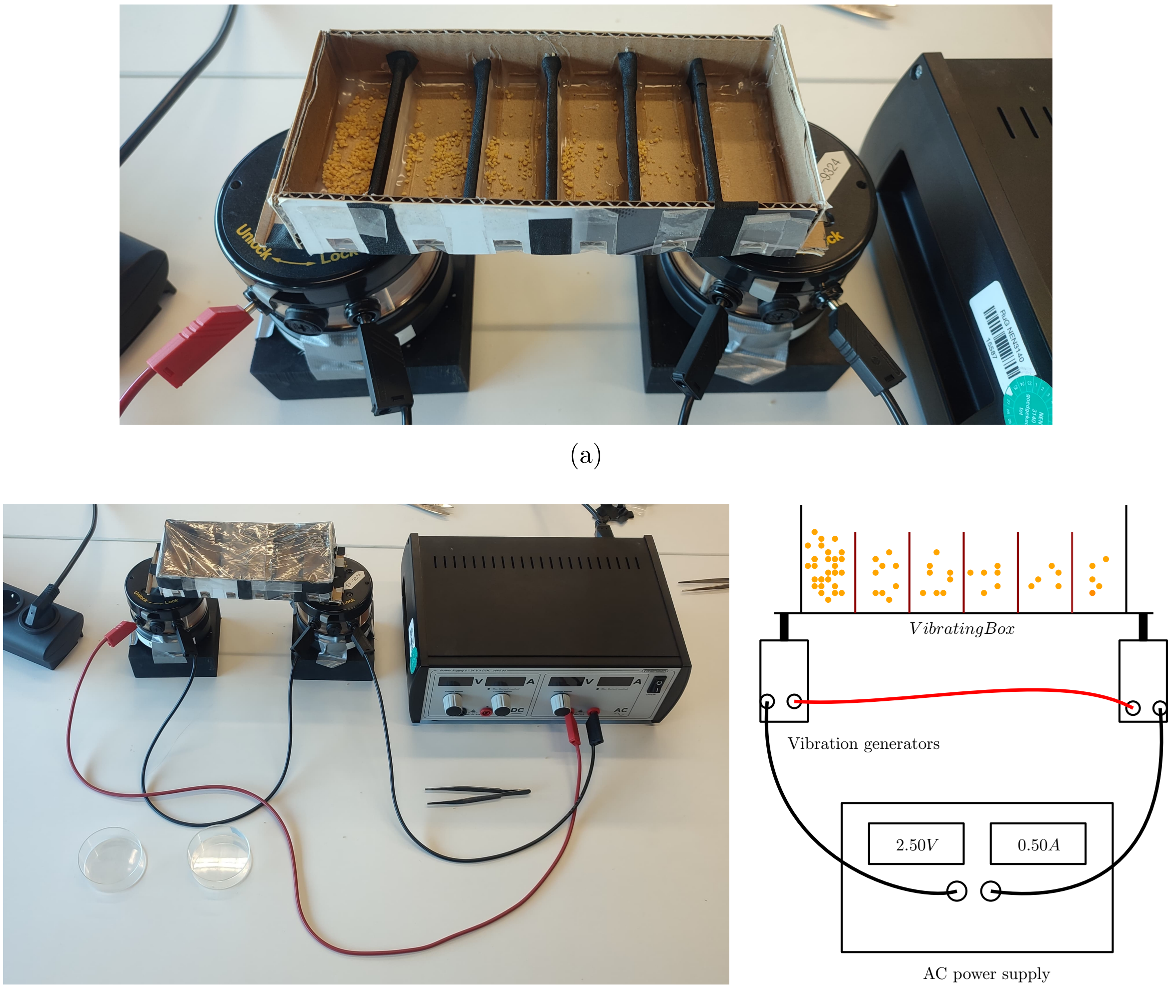
4 Methods

4.1 Setup

The setup rests on two vibration-generating motors available in the laboratory (see Fig. 6a). Each was placed on a rubber slab to absorb the vibrations of the table; the two motors were connected in series to an alternating current source; finally, a homemade carton box was fixed on each end to one motor in a level and stable position, as shown in Figs. 6b and 6c. The dimensions of the box are (20.0 ± 0.1) cm length, (10.0 ± 0.1) cm width, (3.0 ± 0.1) cm height.

The box was supported on the bottom face by an extra layer of stiff cardboard to minimise energy losses due to deformation. Within the box, six compartments were created using four evenly spaced inner walls of the same height of (2.5 ± 0.1) cm. Each compartment also had a hole on a side, sealed with duct tape to act as doors. From above, the box was sealed using a transparent plastic bag fixed on the box walls from the outside with duct tape, in a removable manner.

Aside from the main device, a weighing device with a precision of ± 0.01 g was used. Other materials included couscous as observed particles, containers for its weighing, and a brush with a spoon for easier manipulation of the fine particles.



(a)

(b)

(c)

Figure 6: (a) The compartmentalised box with particles inside. The doors are on the front wall. (b) The entire setup with cable connections. The box is shown with a lid on. (c) The schematic picture of the setup.

4.2 Safety

As for the safety precautions, the motors indicate a maximum current of one ampere, and they include a lock to fix the motor while manipulating the setup. Other than that, only general laboratory rules apply.

4.3 Measurements

The measured relation is between the amplitude of the alternating current as the independent variable and the steady flow of particles between compartments depending on it. This is meant to represent the dependence of the heat flux on the heat conductivity of the material.

The measurement procedure is described next. In every run of the experiment, an amplitude of the current was chosen. Into the first compartment of the box, 0.40–0.80 $\pm 0.01\,\mathrm{g}$ of particles were loaded and evenly spread, based on the current amplitude. This end is referenced as the *hot end* and the other one is the *cold end.* The box was covered with the lid and the motors were unlocked to ensure their synchronisation. The current was turned on for different time periods in the range of 5–30 $\pm 0.1\,\mathrm{s}$, letting the particles chaotically disperse. After this time period, the current was turned off and the motors locked, the lid was taken off.

Then the particles in each compartment were measured, by counting where possible and by weighting if their number was too big. For weighing, the particles were brushed out of the compartments through the doors, always one compartment at a time, weighted and then returned to the same compartment. A conversion factor between weight and number of average-sized particles was determined to combine both types of measurements.

Lastly, the particles in the hot end were restocked back to the initial value, and all particles from the cold end were removed. This was to act as an analogy to the rod in Section 2.2. This procedure continued until the particle flow through the system was relatively stable, which means that for the minimum of three runs, the sum of the first (hot) and the last (cold) end was equal to the initial hot end value (within the error margins). That meant that the stable distribution was reached and the number of particles that left the system was equal to the number of particles added, which means (by the definition) the stationary flow of material through the setup.

In later measurements, the hot compartment weight was decreased to prevent unwanted effects of the liquid phase transition of the particles. This is further explained in Section 5.

4.4 Analysis

4.4.1 Scripts used

To analyse the large amounts of data, the scripts Appendix A.3 and Appendix A.7 were used. Note that for the qualitative results, Appendix A.3 uses a list of manually chosen stopping points as the steady states. This is because of the inconsistent criteria for terminating the experiment, meaning some runs continued to become homogeneous instead of linear. This is due to the discrete nature of emptying and refilling, effectively allowing the distribution to level after too much time has passed.

4.4.2 Fitting the pre-steady state solution

Let the experimental values in the bins be denoted $Q_1^{\exp}(t), Q_1^{\exp}(t), ..., Q_1^{\exp}(t)$, whereas the theoretical values predicted by Eq. (17) be $Q_1(t), Q_1(t), ..., Q_1(t)$, where the experimental values are functions only of time, whereas the theoretical values also depend on time, but also the parameter D. This parameter is the only parameter to be fitted. Since the prediction describes normalised data with $Q_1(t) = 1$ for all time $t \geq 0$, the experimental data must be normalised by dividing all values by the value in the first compartment (hence all values will lie between 0 and 1).

The fitting script is given in Appendix A.6. The goal is to utilise the `SciPy` package `optimise` function `curve_fit`, which finds the best fitting parameter D. It is, however, not so simple, as the data and prediction is a function that takes as input the time t (and parameter D) and outputs six values within the interval $[0, 1]$, hence is a function $Q : \mathbb{R} \to [0, 1]^6$, which is not immediately suitable for `curve_fit` as it only takes functions that map to the reals. Hence the data and function were transformed; some notation is introduced. Fix the amplitude value (fitting is independent for each). Let ℓ_t be the length of the list of time values $t_1, t_2, ..., t_{\ell_t}$ for which measurements were made; e.g. if measurements were made at 0, 5, 10, 15 s, then $\ell_t = 4$. Then the data is stored as a two-dimensional array with ℓ_t rows and 6 columns corresponding to $Q_1^{\text{exp}}(t), Q_1^{\text{exp}}(t), ..., Q_1^{\text{exp}}(t)$ for the t value specified by the row.

For the fitting, the aforementioned array was reshaped to a one-dimensional array as

$$Q_1^{\text{exp}}(t_1), ..., Q_1^{\text{exp}}(t_{\ell_t}), \quad Q_2^{\text{exp}}(t_1), ..., Q_2^{\text{exp}}(t_{\ell_t}), \quad ..., \quad Q_6^{\text{exp}}(t_1), .., Q_6^{\text{exp}}(t_{\ell_t}),$$

and similarly for the function that generates $Q_k(t)$ values from Eq. (17). Then the `curve_fit` algorithm was applied, as is written in the script in Appendix A.6. Several modifications had to be made to achieve a correct (by visual inspection) fit; the results are given in Section 5.2.1, where the modifications are explained.

In the remainder of this section, errors are discussed. The `curve_fit` algorithm on $f(x; a_1, a_2, ..., a_n)$ with input x and parameters $a_1, a_2, ..., a_n$ returns the best fitting parameter $a_1, a_2, ..., a_n$ values and their covariance $\text{Cov}(a_i, a_j) = \mathbb{E}[(a_i - \mathbb{E}\, a_i)(a_j - \mathbb{E}\, a_j)]$ for all pairs $i, j \in \{1, 2, ...n\}$, where $\mathbb{E}$ denotes the expectation value of a random variable. For the diagonal elements this simplifies to $\text{Cov}(a_i, a_i) = \mathbb{E}[(a_i - \mathbb{E}\, a_i)^2] = \text{Var}[a_i]$, namely, the variance of the parameter, and thus the square root will correspond to the standard deviation $\sigma_{a_i} = \sqrt{\text{Cov}(a_i, a_i)}$. In the Python script, this corresponds to taking the diagonal of the covariance matrix and then the square root of it (see Appendix A.6). The error in the D parameter is thus obtained as the standard deviation of this fit.

The D values obtained from a fit of $Q_k(t)$ values for each current amplitude value were plotted against the current values and a linear fit was taken (see Section 5.2.1 in results), namely, $D(I) = aI + b$, where the error of this function was computed by error propagation from the fit (see Appendix C). The error region of this fit was plotted as the value of the fit plus or minus the error value for each I value.

5 Results

5.1 Comment on the data

The raw data is listed in Table 3. The main method of determining the number of particles was by weighing, but due to the scale's minimum registered weight threshold of around 0.08 g, often the particles had to be counted. These numbers are in the brackets and were converted to weight using a conversion factor determined by counting (4240 ± 20) particles in an (8.22 ± 0.01) g sample to be (5.16 ± 0.03) particles per 0.01 g. The error propagation Eq. (43) in Appendix C was used.

As for the columns $m_{1,\text{real}}$ and m_{flux}—the $m_{1,\text{real}}$ indicates the number of particles after stopping the current and before refilling the hot end to 0.80 g or the later smaller values. This corresponds to the value of m_1 comes from. A similar case is also m_{flux}—the mass in the cold end compartment after stopping the run. Before the next run, this compartment was emptied, making m_6 always 0.

An important note is that for Figs. 7a to 7c, the final steady state and stabilisation time range were declared subjectively, consulting the weight distribution time evolution. This is an unfortunate but necessary step, because of the discrete nature of refilling and emptying the end compartments, preventing reaching an ideal state where the refilling would have to be continuous. Nevertheless, the graphs show qualitative behaviours and are a reminder of the limitations of this experimental method.

5.2 Analysis of the particle distribution

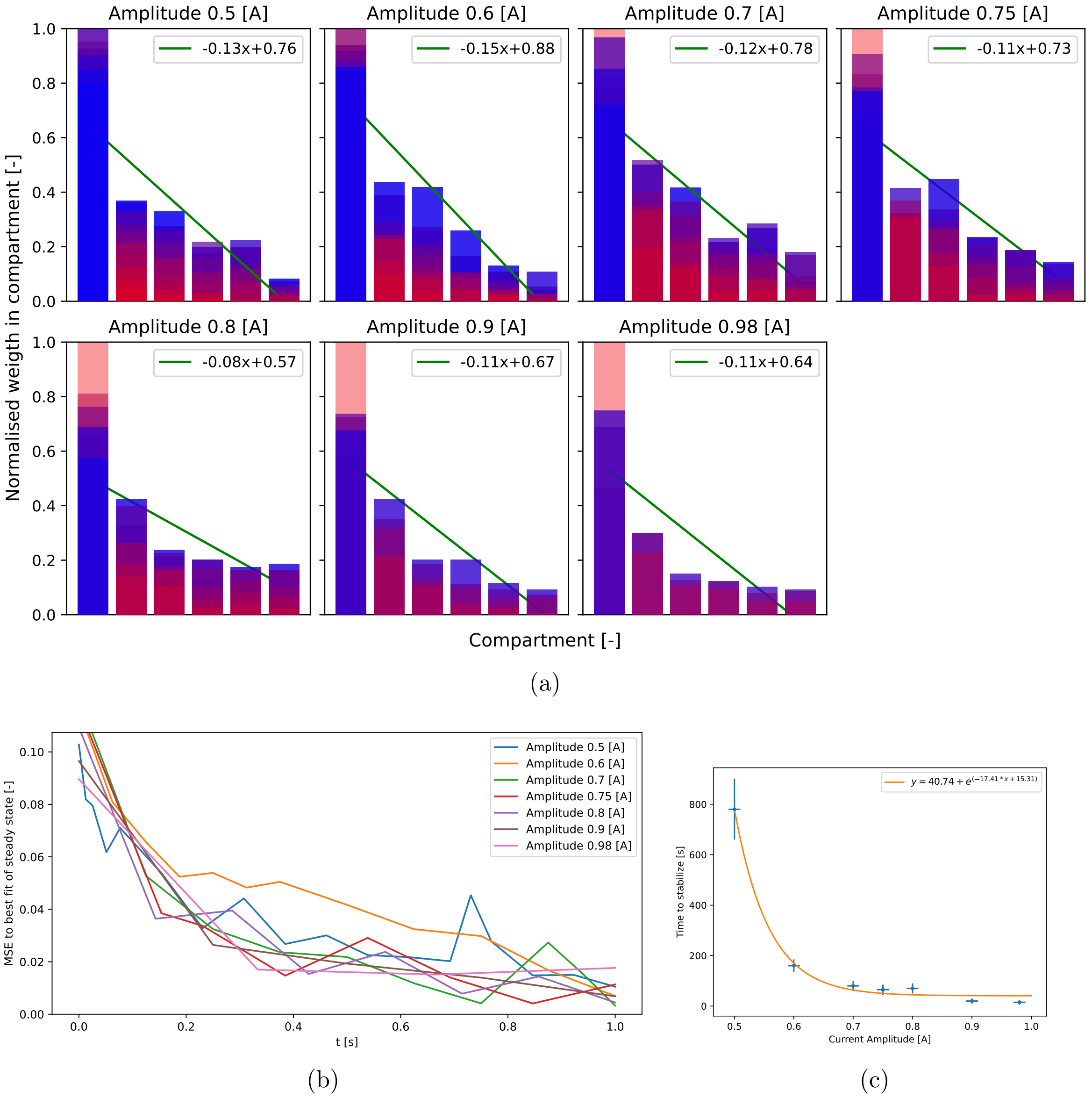
One of the aims of this research is to determine whether the Steady-state solution to the heat equation in one dimension, is good at describing the bouncing particle distribution (with the given boundary condition described above) as $t \to \infty$. To do that the data on all of the compartments was gathered during the experiment(which can be seen in Table 3). Combining the distribution data from these data tables Fig. 7a was created.

In Fig. 7a, the bar plot depicting the evolution of the particle distribution in the system can be seen, where the height of the bar represents the weight of the particles in the respective compartment, the time evolution is represented by the colour changing from red to blue. Note that the weight is normalised by the weight refilled in the hot compartment, to better qualitatively compare behaviours for different amplitudes. Each bar plot shows a separate run with different alternating current amplitude through the vibration generators.

5.2.1 Fitting the distribution to the pre-steady state solution

The data in Fig. 7a is expected to fit to the pre-steady state solution given by Eq. (17). The data must first, however, be renormalised to have value 1 in the first compartment; the experimental values in the bins are denoted $Q_1^{\text{exp}}, Q_1^{\text{exp}}, ..., Q_1^{\text{exp}}$, whereas the theoretical values $Q_1, Q_1, ..., Q_1$, hence $Q_1^{\text{exp}} = 1$. The fitting was done at first according to the procedure (algorithm in Appendix A.6) as described in the methods Section 4.4.2; this, however, did not produce a reasonable fit if the algorithm converged to a parameter D (thermal diffusivity) value. The issue was that the flux value was too high, whereas the value in the 2nd to 5th compartments too low to be accurately fitted by Eq. (17). It was observed by visual inspection that the decrease from the 4th to the 6th compartment was quite gradual, which is indicative of the pre-steady state solution for low D values.

This prompted a choice between fitting the pre-steady state plus some offset value, or subtracting the flux value from all compartments and ignoring the 6th compartment when fitting; it could be argued which is the better choice for this purpose, however, it was decided to use the latter in this report. The experimental values and the associated best fit are given



(a)

(b)

(c)

Figure 7: (a) The weight of the particles in each compartment after a given time, normalized by the initial weight of the hot compartment. The colour changes from red to blue as time progresses. The green line is the best fit for the steady state. Unitless quantity is denotrd [-]. (b) The Mean Squared Error evolution in time for all amplitudes. The linear model used was the best fit to the declared steady state. The time is normalised so all amplitude are displayed to end at time 1.(c) The time it takes to reach the steady state as a function of the current amplitude.

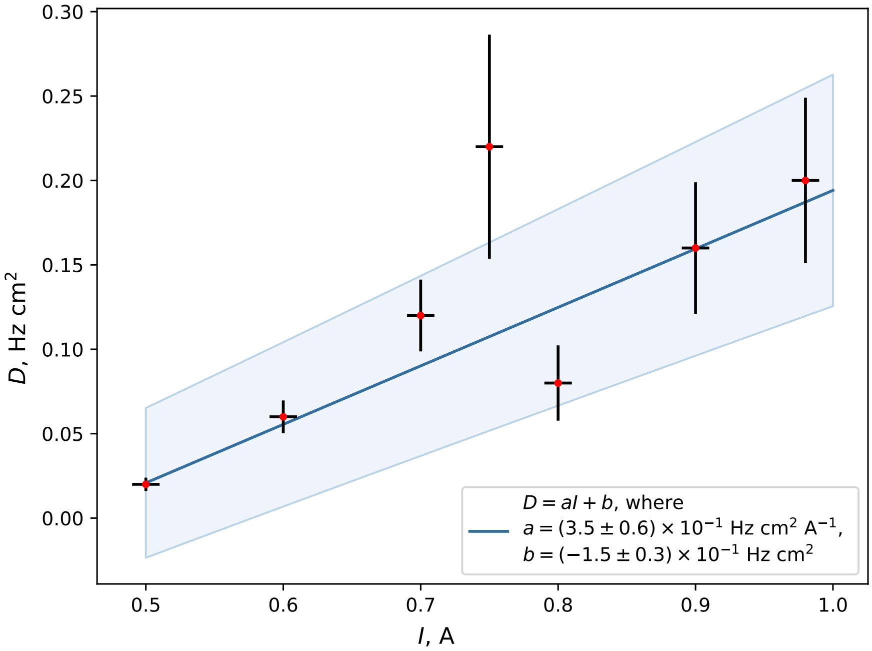


Figure 8: The best-fitted thermal diffusivity D depending on the current amplitude I for which the parameter was fitted. The linear fit with error (see Section 4.4.2) is given in the legend.

in Fig. 11 in Appendix B.2 with a different current value in each figure, and the fitted D parameters given. The fitting algorithm did not, however, converge either when using this subtracted data. It was up to the authors to determine a suitable value by attempting several values and performing adjustments guided by visual inspection of the difference between the experimental values and the values obtained; here lies a possible statistical error of this experiment. However, substantial attention was given to choosing a value that best represents the fit over all compartments *regardless* of what this value is (i.e. imposing no pre-defined relation between D and I). Independent refitting of the data by visual inspection or a more sophisticated algorithm is expected to return D values within the error intervals reported in Fig. 11 in Appendix B.2.

The resulting thermal diffusivity D values in arbitrary length units per second were recomputed to SI units by multiplying by the length of the box value squared, which was 20 cm, obtaining the D values in Hz cm^2. The best-fitted thermal diffusivity values to the data $Q_1^{\text{exp}}, Q_1^{\text{exp}}, ..., Q_1^{\text{exp}}$ for a fixed current I value were compared against these I values, obtaining the plot in Fig. 8. It is evident that with higher values, the thermal diffusivity increases, meaning that the heat quanta in the box may diffuse better (see Section 2.1.2). Note that the I-axis intercept is not at zero current, but at $I = -b/a = (0.15 \pm 0.03)\,\text{Hz}\,\text{cm}^2/(0.35 \pm 0.06)\,\text{Hz}\,\text{cm}^2\,\text{A}^{-1} = (0.428 \pm 0.159)\,\text{A} = (0.4 \pm 0.2)\,\text{A}$.

5.2.2 Linearity of the distribution

To determine how the linearity of the distributions changes in time, the mean square error of the distribution against the best linear fit to the declared steady state was plotted against time for all amplitudes in Fig. 7b.

$$MSE = \frac{1}{6}\sum_{n=1}^{6}[m_n - (k \cdot n + b)]^2 \tag{19}$$

Where m_n is the measurement and $k \cdot n + b$ is the prediction of the linear fit of the normalised weight in the n-th compartment. This means that the sum of the differences squared between measured and fit-predicted weights was plotted against time for each amplitude. The time for each value was normalised by its stabilisation time, such that the whole evolution played out on the interval from 0 to 1. The resultant curves for each of the amplitudes can be observed in Fig. 7b. While this attributes a small value to the steady state by construction, the focus is on the qualitative time decrease of this value.

Another parameter to consider in the pre-steady behaviour is the time it took to reach the steady state. In Fig. 7c, the range of times declared steady is plotted against the current amplitude of the oscillators. The best-fit curve is an exponential decay:

$$t_{\text{steady}} = a + Ce^{b \cdot x} \tag{20}$$

where x denotes the current amplitude and the parameters are $a = (40.7 \pm 12.6)\,\text{s} = (40 \pm 10)\,\text{s}$, $b = (-17.4 \pm 2.2)\,\text{A}^{-1} = (-18 \pm 3)\,\text{A}^{-1}$, and $C = [1\,\text{s}]e^c$, where $c = (15.3 \pm 1.1) = (15 \pm 2)$ (strictly unitless), where the former values are given with higher expansion, whereas the latter with correct error formatting. These results and errors were determined using Appendix A.3, where the least-square method was used by the `lmfit` module.

5.3 Effective conductivity measurements

As stated in Section 1.3, one of the main objectives of this research was to find the dependence between the effective conductivity of the setup k and the amplitude of the alternating current that runs in the vibration generators. Remembering Eq. (3), it is clear that in the steady state of such a system, the flux through it will be proportional to the conductivity constant k, namely, $\dot{q}_{steady} \propto k$. To determine the relation between k and I the steady state particle flux $\dot{q} = \#\text{particles/time}$ was used. But first, the steady state flux and its uncertainty had to be determined. The flux was determined using the following formula

$$\dot{q} = \frac{m_6}{\Delta t} \tag{21}$$

where the m_6 was the weight or the number of particles(using the known conversion coefficient) and Δt is the time interval in which this number accumulated. In order to determine the mean value of the flux the last three measurements were taken and averaged:

$$\dot{q}_{mean} = \frac{1}{3}(\dot{q}_1 + \dot{q}_2 + \dot{q}_3) \tag{22}$$

In order to find the uncertainty of this value, the standard deviation formula was used Eq. (38). But in order to get the maximum available error margins the uncertainty of the measurement device: $\Delta m = 0.01\,\text{g}$ was used in order to calculate the uncertainty of the flux per second value which was found using the following formula Eq. (42) if it was larger than the value that was found by Eq. (38). For the uncertainty of the amplitude of the current ΔI the instrument error was taken $\Delta I = 0.01\,\text{A}$. Using this data and the calculated uncertainties Table 2a was created.

It was then used to construct the graph in Fig. 9a. Although, as can be seen in Fig. 9a, the linear approximation does not exactly describe the trend, the best-fit coefficient a_{best} was still calculated using the Eq. (40) to be $a_{\text{best}} = 0.05\,\text{g/sA}$, and the error of this quantity was found using Eq. (39) to be $\Delta a_{\text{best}} = (0.08 - 0.02)/2 = 0.03\,\text{g/sA}$. Therefore the final solution for the linear coefficient that relates I and $\dot{q}_{steady}$ is $a = (0.05 \pm 0.03)\,\text{g/sA} \approx (0.05 \pm 0.03)\,\text{g/sA}$.

As seen in Fig. 9a, the steady state flux and thus the effective conductivity of the system did indeed grow with the amplitude of the oscillations. However the group was not satisfied

I, A	$\dot{q}$, g/s	$\Delta\dot{q}$, g/s	ΔI, A
0.50	0.0006	0.0003	0.01
0.60	0.0022	0.0005	0.01
0.70	0.0026	0.0003	0.01
0.75	0.005	0.002	0.01
0.80	0.183	0.001	0.01
0.90	0.200	0.002	0.01
0.98	0.207	0.002	0.01

(a)

I, A	$\dot{q}_{norm}$ 1/s	$\Delta\dot{q}_{norm}$ 1/s	ΔI, A
0.50	0.0011	0.0003	0.01
0.60	0.0054	0.0004	0.01
0.70	0.008	0.002	0.01
0.75	0.015	0.002	0.01
0.80	0.023	0.002	0.01
0.90	0.025	0.002	0.01
0.98	0.025	0.002	0.01

(b)

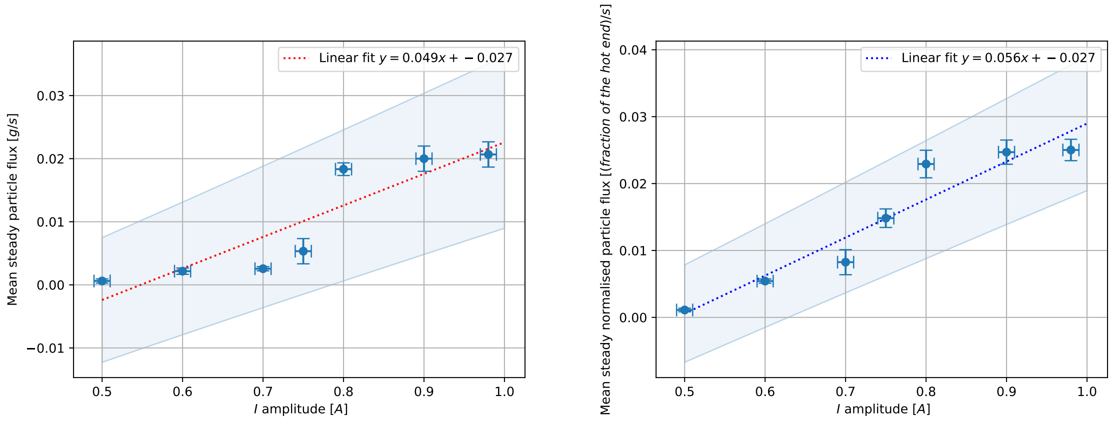
Table 2: The tables of the calculated values of (a) steady state fluxes and (b) normalised steady state fluxes for different amplitudes and the respective uncertainties

with these results, the first graph(Fig. 9a) that was obtained does not present the clear relation between the amplitude and the mean steady particle flux, it can be viewed as both the linear relation or like a sigmoid shape. That indicated that there was another physical effect involved that contributed to this shape. After further data and literature analysis, it was decided that the effect disturbing the data is the phase transition. Namely, because of the self-interaction between the particles(which was not considered in the theoretical model, where all particles are independent of each other), at the lower amplitudes the “hottest” compartments were behaving like a liquid, which caused an extremely low flux and a long stabilising period. At some value of I the phase transition occurred which made the particles behave like a gas and that caused the sigmoid shape on the graph.

In order to avoid the phase transition, the following method was developed: the number of particles in the first compartment was varied as amplitude changed. To get relevant and comparable results it was decided to measure the so-called “normalised flux”:

$$\dot{q}_{norm} = \frac{\dot{q}}{N_{hot}} \tag{23}$$

This quantity indicates the proportion of the hot end that leaves the system in a unit of time and can be compared across all the amplitudes and weights. Furthermore, it was decided to



(a)

(b)

Figure 9: (a) The steady state particle flow as a function of the I amplitude. (b) Updated normalised flux as a function of I.

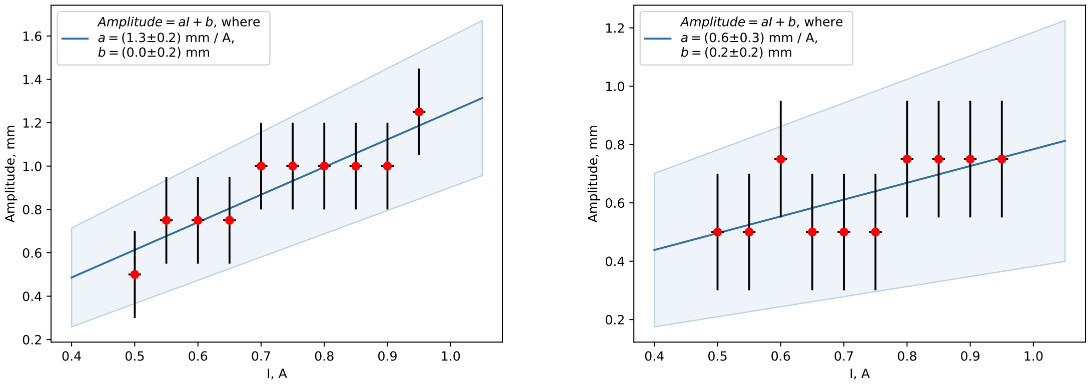
make an additional measurement in the region where the phase transition occurred namely $I = 0.75\,\mathrm{A}$.

For the uncertainties of the new measured variable, it was decided to use the standard deviation of the normalised flux as the $\Delta\dot{q}_{norm}$ (in order to simplify the error analysis process). Thus the new table of results was acquired Table 2b. Using data from Table 2b the second graph was constructed: Fig. 9b. Using the Fig. 9b, the new slope and its uncertainties were acquired(using Eq. (41)). And so the normalised a value was found to be $a_{norm} = (0.06 \pm 0.01)\,\mathrm{s}^{-1}\,\mathrm{A}^{-1}$.

5.4 Amplitude dependence on the current

It is worth noting that the oscillation amplitude's dependence on the applied current was also determined at the end of all the experiments (Fig. 10). However, by the end, the right side was not attached too well to the vibration generators, which caused it to vibrate less, and also, the tools used to measure the amplitude were not precise enough (the uncertainty of the ruler was 0.25 mm) and that is why mostly Current was used instead of this amplitude.

This amplitude was determined using a smartphone camera at 240 frames per second and a ruler that was attached next to the vibrating platform. The linear dependence was found using the least-squares method.



(a)

(b)

Figure 10: The dependence on the oscillation amplitude of (a) the left, (b) the right side of the box on the applied current.

The dependencies can be expressed linearly with functions as $A_l = a_l I + b_l$ for the left side and $A_r = a_r I + b_r$ for the right side respectively. A is the amplitude in millimetres and I is the applied current in Amperes. The values for the coefficients were calculated using the least-squares method using Python's package `SciPy` function. The coefficients were determined to be: $a_l = (1.3 \pm 0.2)\,\mathrm{mm\,A}^{-1}$, $b_l = (0.0 \pm 0.2)\,\mathrm{mm}$, $a_r = (0.6 \pm 0.3)\,\mathrm{mm\,A}^{-1}$, $b_r = (0.2 \pm 0.2)\,\mathrm{mm}$. It should also be mentioned that the b values should theoretically be 0 as there should be no amplitude when there is no current.

6 Discussion

6.1 Fit and linearity development

6.1.1 Ideal and continuous case predictions

In this experiment data describing particle flow through the box was obtained; the core research question is whether this flow may be described by continuous conduction in a thermal conductor. In the theory section the continuous case was considered via the famous heat equation (Eq. (2)) that describes the time-evolution of temperature within a heat-conducting material; an analogous setup to the one used was made for the continuous case, namely, that at the ends (infinitesimally thin), the temperature is kept at a constant value, namely, T_1 and T_2 for both ends, respectively. From a choice of values, a prediction for the distribution was made, given by Eq. (17) and visualised in Figs. 2, 3a and 3b.

Another prediction was made using probabilistic simulations that with an increasing probability of a particle jumping over a wall, the flux will increase linearly with the probability (see Fig. 4d). In both cases, with long enough time, a steady state is to be reached, where the particle distribution does not change significantly with time as seen in Figs. 2, 3a and 4c. In addition, it was predicted in Section 2.1.2 that the thermal diffusivity will increase with respect to the amplitude of the oscillations (and thus also current, see Fig. 10).

6.1.2 Homogeneous heating of the system

It must be noted that the predictions are for a particle distribution normalised with respect to the first bin, which is kept constant. Therefore all data had to be re-normalised in order for fitting to be possible. Due to complications with the `SciPy` `curve_fit` algorithm, fitting had to be done on a concatenated array, as described in Section 4.4.2. It was attempted to find a fit between the data given in Table 3, and shown in Fig. 7a, and the predicted distribution in Fig. 3b given by Eq. (17). As is described in the results Section 5.2.1, this fit was fatally unsuccessful.

The major difference between the predicted and observed distributions was that the predicted distribution was concentrated in the first and second bins, and assumed very small values in the other bins, for the time-and-thermal diffusivity-product Dt values much smaller than 0.5 arb. units or $200\,\mathrm{cm}^2$. The predicted values in the final wells were only comparable to the values in the first bins once near $Dt = 200\,\mathrm{cm}^2$ was reached, which corresponds to the system being in a steady-state or close to it. For small experimental time (in seconds) values, the distribution of particles was indeed comparable to the prediction, namely, there were hardly any particles in the final compartments.

With increasing time, however, this fit no longer held true. There were a lot of particles in the final compartments and this amount was similar to the amount in the second compartment. From this regard, it is evident that the flow of particles in the experimental setup does not behave as is expected for an ideal conductor. It is unclear, however, whether this is due to experimental error by only sporadically filling up the first compartment and removing the flux, or a fundamental difference between the discrete and continuous case. In order to resolve this question, an ideal simulation was performed, where refilling of the compartments was instantaneous. The results of it are given in Section 3.2, in particular, in Fig. 5c and Table 1. In the simulation, the particles do not interact, and thus there can be no effects of liquefaction, which are discussed later on as a source of error in the experiment.

It was found that fitting of the simulated data is adequate for bins except the second bin. This shows that the misfit between data and prediction is not due only to the experimental error; this can be characterised by the particles from the second bin quickly moving on to the remaining bins, yet the explanation is unclear. It is evident from Fig. 3a that the number of particles in the second bin rises rapidly in the beginning of the experiment; the continuous case is special by the fact that there is no significant barrier between the first and second compartments, namely, the particles move though a fluid with some resistance (as in a uniform material). In the experimental (discrete) case, however, the first wall is the most significant as the other compartments are completely inaccessible before a particle has made it across the first wall.

After this point, most compartments are near-identical in their particle density, and thus may increase near-homogeneously. It is, nevertheless, expected that particle flow will follow temperature diffusion, as described by Eq. (15). Fitting to experimental data can thus be done either by introducing another free variable that describes the homogeneous increase in distribution—this corresponds to homogeneous heating—or by subtracting the flux. For arbitrary reasons, the latter was adopted, as described in Sections 4.4.2 and 5.2.1. The resulting fits are given in Fig. 11.

In the figures, it may be observed that the fit between the flux-subtracted data and the predicted distribution by the best-fitting thermal diffusivity D parameter is reasonable. In some plots, the fit overestimates the amount in the second compartment, whereas, in others the end-distribution. For reasons of the algorithm, namely, lack of convergence, the fits had to be done by visual inspection, which may be a source of error (discussed later). Overall, one parameter D value may reasonably explain the time-evolution of the distribution of flux-subtracted data. The fits are deemed suitable enough to proceed with further analysis.

6.1.3 Thermal diffusion cutoff

Recall that it was predicted in Section 2.1.2 that the thermal diffusivity would increase with amplitude of the vibrations. A plot was made relating the current applied and the best-fitting thermal diffusivity D value, given in Fig. 8; a linear trend (albeit noisy) may be observed. An important note, however, is that the I-axis intercept is not close to $0\,\mathrm{A}$; this implies that there is some current below which the diffusivity is zero. When the current applied to the vibration generators is zero, there are, of course, no vibrations visible to the naked eye, and the particles would never move between compartments; with very small current, the vibrations were observed to be not strong enough for the particles to jump over the wall of the first compartment.

This is a fundamental difference between the discrete and continuous cases. In the continuous case, even very small non-zero diffusivity would imply that the temperature reaches the steady state for time going to infinity $t \to \infty$; in the discrete case, the diffusivity must be high enough to overcome the energy barrier posed by the first wall. This current value is calculated to be $I = (0.4 \pm 0.2)\,\mathrm{A}$, which is close to the lowest-measured current of $I = 0.5\,\mathrm{A}$ for which hardly any diffusion was observed, validating the value to be close to this estimate. Note that thermal diffusion cannot be negative, both for reasons of physical nonsense (temperature increasing with its gradient), and due to the fact that the sum in Eq. (15) would diverge to infinity for long-enough time t.

In summary, homogeneous heating is observed of the system, upon which the distribution of particles reasonably well follows behaviour described by the continuous heat equation Eq. (2). The core physical reason is the energy barrier posed by the first wall in the setup, for which there is no analogue in the continuous case. There exists an amplitude value below which (the effective) thermal diffusivity is zero, and above which thermal diffusivity is proportional to the current applied. This answers the first research question: *the discrete particle distribution describes conduction in a material with additional homogeneous heating.*

The best measure for the amplitude used was the current applied to the vibration generators. An attempt was made to relate it to the true amplitude in mm, however, due to poor resolution with a ruler, the errors were too great, as evident in Figs. 10a and 10b. Nevertheless, linear behaviour is observed, with near-ideal proportionality between the applied current and the amplitude of the vibration generators, validating the choice of amplitude in the rest of the analysis.

6.1.4 Asymptotic linearity of the system

The predicted steady-state solution Eq. (4) is linear (see also Fig. 3b). By first approximation, it may be stated that with increasing time the data is expected to approach the linear solution. In Section 5.2.2 a measure for the linearity of data was developed, as given by Eq. (19).

As shown in Fig. 7b, the mean square error values for the linear approximation decreased as time increased, meaning that the linear approximation described the situation better over time. It is also evident that the convergence of Eq. (19) to zero follows a fast decay for small time values, and then slowly decays to a near-zero value for larger time; this is reminiscent of decay. Evidently from Figs. 2 and 3b, for small time values, the system is very far from linear, and only begins to resemble linearity for long time.

From this, it may be concluded that on a long timescale, the heat distribution can be approximated linearly. This can also be observed in Fig. 7a. The time required to reach this linear equilibrium decreases exponentially with the amplitude of oscillations as seen in Fig. 7c. The behaviour is well described by Eq. (20) and the parameters in Section 5.2.2. Note that the exponential fit is offset by $a = (40 \pm 20)\,\mathrm{s}$, which implies that even for a current much bigger than 1 A the time required to reach the approximate steady state will be at least this value, namely, not arbitrarily close to zero. This is sensible by considering the physical amplitude: even for very large (much larger than tested here) amplitudes, the balls will hit the ceiling and will need to first bounce to the second compartment, then to the third, and so on—they will *not* almost immediately go from the first to the last compartment.

This separate verification is rather useful given the poor direct fit to the theoretical prediction Eq. (15) as described above. Namely, to indicate that steady-state flow is being approached.

6.2 Effective conductivity

The second research question concerns measurement of the effective coefficient of thermal conductivity k of the experimental system. This was done in Section 5.3.

The results indicate dependence between increasing the amplitude and observing larger flux as visible in Fig. 9a. The dependency between the current I and the flux $\dot{q}$ is observed to resemble a sigmoid shape with inflection near $I = 0.75\,\mathrm{A}$; unfortunately, a fit was not made—the reasoning being that not much information may be derived from it. Given the rather small spread of data, however, noise is difficult to rule out, and thus this relation may be said to be linear as it matches the data well enough, although with a large uncertainty. The effective conductivity was determined by varying the number of particles used for the amplitudes and then normalised as using the same amount for each amplitude; using the same amount of particles resulted in particles strongly interacting with each other for lower amplitudes, which introduces experimental error, and thus normalised values were used. Using the normalised data as given in Fig. 9b, the value of the linear coefficient that relates the steady flux to the amplitude (the current applied in this case) was obtained to be $a_{\mathrm{norm}} = (0.06 \pm 0.01)\,\mathrm{s}^{-1}\,\mathrm{A}^{-1}$. It is evident, however, that this value has a large uncertainty, therefore, more measurements should be taken in order to confirm the relation.

More measurements would also serve to give confidence in determining whether the sigmoid shape arises from noise, or whether it is truly the experimental outcome. That is, should the points lie near points next to it, as expected from a sigmoid shape, then the linear fit is faulty, and a sigmoid must be fitted instead; on the other hand, more noise, namely, points of higher flux for lower amplitudes or lower flux for higher amplitudes would reduce confidence in the sigmoid, and a noisy linear fit is to be obtained instead. In any case, given the noise, the linearity is only an approximation to reality.

Given the relation of proportionality between the coefficient of thermal conductivity k and the steady-state flux, it is evident that the effective conductivity increased with amplitude; the linearity of the relation (given by a_{norm}) implies that the coefficient of conductivity is proportional to the current applied $k \propto I$. Unfortunately, however, the coefficient of proportionality was not determined in this report. Examining the relation between thermal diffusivity and the coefficient of thermal conductivity $D = k/\sigma\delta$, should $\sigma\delta$, the volumetric heat capacity in three-dimensions—analogously in one dimension in this case—, be constant, then the coefficient thermal diffusivity would be proportional to coefficient of thermal conductivity. Since both $D \propto I$ and $k \propto I$, it follows that indeed $D \propto k$ and thus $\sigma\delta$ is constant for the system examined in this report. This answers the second research question: *the coefficient of thermal conductivity is proportional to the amplitude of the oscillations.*

Any objection that could be raised about fitting the thermal diffusivity values before without any regard for the mass (particle) density of the rod δ—or the specific heat capacity σ, although it may be assumed to be constant more clearly—could thus be rejected. The proportionality between $D \propto k$ implies a constant density.

6.3 Experimental errors

During experimentation, many issues and possible sources of errors were encountered. One problem was the loss of particles, either leaking out or getting stuck in the cardboard walls. The particles were not the same size or shape either, possibly having an effect as the bigger ones usually spread further than the smaller ones. This could be fixed by picking a more suitable particle and sealing the box more thoroughly. Given the extremely large amount of particles used to determine the ratio between mass and number of particles (5.16 ± 0.03) particles per 0.01 g (see Section 5.1), quite high certainty may be placed on the mass measurements. However, counting measurements involved particles of various sizes, where the count may not be large enough for proper averaging.

Another major issue was preventing particles from clumping up. This happened if the box was not level, the amplitude was too small or the number of particles was too large; namely, errors arising from the quality of the experimental setup, which was improvised at best. This played a large role, as another statistical process was observed with too high of a particle density—liquefaction; this involves particles interacting such that gas-like behaviour where particles jump high into the box is not observed, instead, there is movement near the floor of the box with large horizontal component; a lot of effort was placed into minimising this source of error. This is reminiscent intuitively of the evaporation of the upper layer of a liquid in a gas. This was done by lowering the mass used until by visual inspection gas-like behaviour was predominantly observed.

For lower current amplitudes, it can be noticed that despite a large gradient between the hot end and the second compartment, it hardly smoothness out. At higher amplitudes, this problem disappeared, hinting at a phase transition depending on the particle density and vibration amplitude. The platform being slanted is also something that could be easily solved by just attaching the platform in a better way to the vibration generators; the authors have reached particular experimental skill at using tape during this project—intuitive methods of applying tension via tape to tilt the box back to stability had to be employed.

Furthermore, by the end of the experiment, the set-up was not as sturdy as in the beginning and one of the sides, vibrated with lower amplitude as the attachment loosened. This difference can be seen in Fig. 10 as the plot of the left side Fig. 10a is steeper than the right side's plot Fig. 10b. This was partly taken care of by upgrading the attachment of the faulty side and as well by choosing the left side as the hot end. A more optimal solution would be to fix the system entirely and consistently across measurements.

While the mentioned factors contribute to random errors, there were some sources of systematic errors too. One of them was only having 6 compartments, which led to an extreme approximation of the "infinitely many compartments" in real materials. Of course, the number of particles was lower by orders of magnitude compared to real materials. This is in fact the main question of this report (relates to the first research question): is the amount of particles used enough? The answer to the first research question above indicates that the quality of the fit (Fig. 11) is not likely due to the small number of particles.

Furthermore, the scale utilised only had an accuracy of 0.01 g, and the emptier compartments were often registered incorrectly. The imprecise definition of "steady flow" produced discrepancies between the final states of different runs, decreasing the confidence in the findings. An attempt was made to subjectively declare which states were steady without bias for nice results, but this has limitations.

It also took some time for the power supply to reach the current needed and to drop back to 0. While for the 20 s intervals, this was not a too large problem, for the 5 s intervals this may have been a crucial source of error. The time was luckily not quantitatively relevant to this experiment, only serving to keep consistency in stabilising the end compartments. Attention was made to consistently start the stopwatch at the same moment, which was when the current had almost reached the required value; before this, the vibrations were often minute.

The problem of the first compartment not having a constant weight of particles can be solved by running and refilling in multiple shorter time intervals. However, doing so would increase the problem of the AC generator not jumping instantly to the desired current forcing a compromise or more innovative approach. The physical simulation in Section 3.2, however, indicates that there is a fundamental discrepancy between continuous conduction and the experiment, such that the untimely refilling was not the main source of the issue.

Lastly, despite the liquefaction being an error source in this study, it is an interesting phenomenon worth exploring. It was, in fact, explored in the problem that was the first motivation of this study. Here it was indirectly shown that the phase transition described by statistical mechanics using probabilistic methods on a large number of indistinguishable small particles can indeed be simulated with a vibrating table and couscous in a container. A process reminiscent of evaporation of the top layer of liquid was qualitatively observed too, but determining details is perhaps a task of a separate study.

7 Conclusions

It can be concluded that the set-up can be used to describe and simulate the heat flow according to conduction in a material described by the heat equation together with a homogeneous heating term. The linear approximation can be applied to the equilibrium of a system with the small particles in a vibrating box with compartments.

The time it takes for the system to reach this linear equilibrium was shown to depend exponentially on the amplitude of the current through oscillators, decreasing with higher amplitudes. The existence of a bound for the time required to reach a steady state regardless of the amplitude applied was found. The amplitude of vibrations was found to be proportional to the current applied.

The relation between the normalised flux and the amplitude can be described as linear. Similarly, the coefficient of thermal diffusivity increased linearly with current. From their relation, it follows that the volumetric heat capacity analogue in one dimension is constant for the system.

The experiment can be improved by taking more data points, fixing the box, increasing the number of compartments, and taking more precise scales. Ideally, a system would be devised, that continuously keeps the weight at the ends constant and registers the amount of particles flowing in and out.

Overall, the two research questions were answered. 1. The system describes heat conductivity with a homogeneous heating term. 2. The equivalent coefficient of thermal conductivity is proportional to the amplitude of the vibrations. Clearly, the first hypothesis is confirmed—the analogy is not direct since an extra mode of heating is introduced. Similarly, for the second hypothesis, it was confirmed that the particles spread out faster with higher amplitudes.

An aspect of future study could also be the phase transition based on oscillation amplitude, which was treated merely as an error source in this experiment.

Bibliography

[1] D. Goodstein, *States of Matter*, ser. Dover Books on Physics. Dover Publications, 1985.

[2] T. D. Hien and M. Kleiber, “Stochastic finite element modelling in linear transient heat transfer,” *Computer Methods in Applied Mechanics and Engineering*, vol. 144, pp. 111–124, 1997.

[3] S. Chevalier and O. Banton, “Modelling of heat transfer with the random walk method. part 1. application to thermal energy storage in porous aquifers,” *Journal of Hydrology*, vol. 222, pp. 129–139, 1999.

[4] J. Blijleven and R. Klein-Douwel, “Data and error analysis for physics laboratory: Skills,” 2023.

[5] A. Mehta and J. M. Luck, “The bouncing ball revisited,” *Modern Physics Letters*, vol. 4, pp. 1245–1248, 1990.

[6] A. R. Abate and D. J. Durian, “Effective temperatures and activated dynamics for a two-dimensional air-driven granular system on two approaches to jamming,” *Physical Review Letters*, vol. 101, 2008.

[7] H. Viswanathan, R. D. Wildman, J. M. Huntley, and T. W. Martin, “Comparison of kinetic theory predictions with experimental results for a vibrated three-dimensional granular bed,” *Physics of Fluids*, vol. 18, 2006.

[8] C. Aegerter and A. Kish, “47th International Physics Olympiad E-2: Jumping beads—a model for phase transitions and instabilities,” 2016.

[9] D. R. Lide, Ed., *CRC Handbook of Chemistry and Physics*, 88th ed. CRC Press, 2007.

[10] R. Daileda, “Partial differential equations,” 2017, lecture notes, Trinity University.

[11] K. R. Hiremath, “Partial differential equations,” 2021, lecture notes, Indian Institute of Technology, Jodhpur.

[12] S. Robinson, “Derivation of the heat equation,” 2009, lecture notes MATH/PHYS 4530, University of Michigan.

A Appendix: programming and calculation scripts

The scripts were developed in the GitHub repository available under https://github.com/MikaHvostenberg/RandomWalk.git (7.06.2024).

A.1 Probabilistic simulation script

```python
import matplotlib.pyplot as plt
from matplotlib import cm
import random
import numpy as np

#parameters of the simulation
domains=50
particles=10**4


def statistics(N:int,particles:int,steps:int):
    #here the key parameters are defined
    walks=[-1,0,1]
    weights=[0.4,0.2,0.4]
    #here we define the initial destribution
    current=np.zeros(N)
    current[0]=particles
    for step in range(0,steps,1):
        tmp=np.zeros_like(current)
        for j in range(0,len(tmp),1):
            for k in range(0,int(current[j])):
                if j==0:
                    walk=random.choices([0,1],[weights[0]+weights[1],weights[2]])
                elif j==len(current)-1:
                    walk=random.choices([0,-1],[weights[0]+weights[1],weights[2]])
                else:
                    walk=random.choices(walks,weights)
                tmp[j+int(walk[0])]+=1
        current=tmp
    final=current
    print(np.sum(final))
    print(final)
    my_cmap = plt.get_cmap("inferno")
    rescale = lambda final: (final - np.min(final)) / (np.max(final) - np.min(final))
    # Create bar chart
    plt.bar(np.arange(1,domains+1,1),final, color=my_cmap(rescale(final)))
    plt.show()
    return current

#statistics(domains,particles,750)
#here is the function for the stationary flow of the particles

def stationary_flow(N:int,particles:int,steps:int,weights:list,graph=True):
    '''''This function simulates the stationary flow through the probabilistic setup and calculates
    the average heat flow (over the last 100 seconds)'''''
     #here the key parameters are defined
    walks=[-1,0,1]
    flow=np.zeros(steps)
    #here we define the initial destribution
    current=np.zeros(N)
    current[0]=particles
    for step in range(0,steps,1):
        tmp=np.zeros_like(current)
        for j in range(0,len(tmp),1):
            if j==len(current)-1:
                flow[step]=current[j]
            for k in range(0,int(current[j])):
                if j==0:
                    walk=random.choices([0,1],[weights[0]+weights[1],weights[2]])
                elif j==len(current)-1:
                    walk=[1-len(current)]
                else:
                    walk=random.choices(walks,weights)
                tmp[j+int(walk[0])]+=1
        current=tmp
    avg_flow=np.average(flow[-100:])
    print(avg_flow)
    #printing block(makes a graph of a single measurement)
    if graph==True:
        plt.plot(flow,color="black",label="momentary particle flux")
        plt.grid()
        plt.axhline(y=avg_flow,color='red',linestyle='dashed',label=f"running average: {avg_flow} ")
        plt.legend()
        plt.show()
    return avg_flow
#draws a single thing
#stationary_flow(10,10**5,200,[0.25,0.5,0.25])

#this is another experimental function
def measure_flow(N=10):
    """"Fucntion that measures the stable heat flow as a function of the probabilities of
    particles going left or right"""
    #target array
    steady_flow=[]
    probabilities=np.linspace(0,0.5,N)
    for prob in probabilities:
        steady_flow.append(stationary_flow(10,10**5,200,[prob,1-2*prob,prob],graph=False))
    plt.plot(probabilities,steady_flow)
    plt.show()
    return
#lets make a graph of the flow as a function of probabilities
measure_flow()
```

A.2 Physical simulation script

```python
from os import ttyname
import numpy as np
import scipy as sp
import random
import matplotlib.pyplot as plt
from datafit import compute_qk


def update_values(ballveloc:np.ndarray, scaling:float):
    """
    Updates the x,y,z values of the particle with the exp(-x) probability distribution
    when it hits the box floor (z==0).
    """

    plusminus = np.random.randint(0,2,2)*2-1
    plusminus = np.append(plusminus, 1)
    newvi = ballveloc + plusminus*sp.stats.expon.rvs(loc=0, scale=scaling, size=3, random_state=None)

    return newvi


def make_simulation() -> tuple[np.ndarray, np.ndarray]:
    downacc = 0.01
    ttotal = 1000
    tstep = 0.2
    floorheight = 0.0025
    intensity= 0.1
    damping= 0.5
    boxlength = 5
    boxlengtherror = 0.1
    boxwidth = 5
    wallheight = 3

    n = 300
    velfactor = 10 # how much to shrink the initial normal dist values by
    pos = np.zeros((n,3), dtype=float)
    vel = np.random.randn(n,3)/velfactor

    # out_xlist = np.array([])
    # out_ylist = np.array([])
    # out_zlist = np.array([])

    i = 0
    counterlist = []
    print("Beginning simulation.")
    while i <= ttotal:

        vel[:,2] += -downacc*tstep
        # print(vel)
        # exit()

        pos += vel*tstep
        # print(pos)
        # exit()

        counter0 = 0
        counter1 = 0
        counter2 = 0
        counter3 = 0
        counter4 = 0
        counter5 = 0
        counter6 = 0
        for j, partipos in enumerate(pos):

            if partipos[2] <= floorheight:
                pos[j,2] = 0
                vel[j] = update_values(damping*np.abs(vel[j]), intensity)
            if abs(partipos[0]) >= boxwidth:
                vel[j][0] *= -1
            if partipos[1] < 0:
                pos[j,1]   = 0
                vel[j][1] *= -1
            if partipos[1]%boxlength <= boxlengtherror and partipos[2] < wallheight:
                vel[j][1] *= -1

            match partipos[1]//boxlength:
                case 0: counter0 += 1
                case 1: counter1 += 1
                case 2: counter2 += 1
                case 3: counter3 += 1
                case 4: counter4 += 1
                case 5: pos[j] = [0.0,0.0,0.0]; counter0 += 1
                case _: pos[j] = [0.0,0.0,0.0]; counter0 += 1

        # steady_state
        if counter0 < n:
            pos = np.append(pos,[[0.0,0.0,0.0]],axis=0)
            vel = np.append(vel,np.random.randn(1,3)/velfactor,axis=0)


        counterlist.append([counter0,counter1,counter2,counter3,counter4,counter5,counter6])

        # out_xlist = np.append(out_xlist, pos[:,0])
        # out_ylist = np.append(out_ylist, pos[:,1])
        # out_zlist = np.append(out_zlist, pos[:,2])

        if int(1/tstep*round(i,1))%int(10/tstep**2)==0:
            print(f"Step \t{int(round(i,0))}/{ttotal} completed.")

            print(f"Particles: \tn[0|1|2|>2]=[{counter0} \t{counter1} \t{counter2} \t{counter3} \t{counter4} \t{counter5} \t{counter6}]")

        i += tstep

    print("Simulation completed.")
    return np.array(counterlist), np.linspace(0,ttotal,int(ttotal/tstep))




if __name__ == "__main__":

    avgnum = 5
    counterarray, tvals = make_simulation()
    for inst in range(avgnum-1):
        counterarray += make_simulation()[0]
    counterarray = counterarray/avgnum/300

```

```python
    colors = [’b’,’g’,’r’,’c’,’m’,’y’]
    fig, ax = plt.subplots(1,1,sharex=True,sharey=True,constrained_layout=True)
    # fig.suptitle(f"Physical simulation of {n}\nparticles in infinitely many boxes")

    ax.plot(tvals,counterarray[:,0], label="$1$", c=colors[0])
    ax.plot(tvals,counterarray[:,1], label="$2$", c=colors[1])
    ax.plot(tvals,counterarray[:,2], label="$3$", c=colors[2])
    ax.plot(tvals,counterarray[:,3], label="$4$", c=colors[3])
    ax.plot(tvals,counterarray[:,4], label="$5$", c=colors[4])
    ax.plot(tvals,counterarray[:,5], label="$6$", c=colors[5])
    # ax.plot(tvals,counterarray[:,6], label="Boxes $>6$")

    ax.legend(loc="upper right", fontsize="small", title="Bin $k$")

    fig.supxlabel("Time")
    fig.supylabel("Normalised Nr. of particles")

    for k in range(2,6):
        paramd, pcovd = sp.optimize.curve_fit(lambda t,D: compute_qk(t,D,k), tvals,counterarray[:,k-1], bounds=[0,0.0001])
        perrod = np.sqrt(np.diag(pcovd))
        print(f"For k={k}, D = {paramd} +- {perrod}")

        ax.plot(tvals, compute_qk(tvals,paramd,k),c=colors[k-1],linestyle="dotted")

    fig.savefig("plots/ballsinbins-steady-avg.pdf")
    np.savetxt("data/counter.csv",counterarray,delimiter=’,’)
    exit()
```

A.3 Data processing script

```python
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
from scipy.optimize import curve_fit
from lmfit import Model

#Helper functions
def f(x,a,b):
    'Linear function'
    return a*x+b
def MSE(dataset,model):
    'Mean square error'
    return np.sum(((dataset-model)**2))/len(dataset)
def exi(x,a,b,c):
    return a+np.exp(b*x+c)

#Predefined variables and data
data = pd.read_csv('/home/edo/Downloads/CousPrelim.csv',delimiter = ',')
chis=[[],[],[],[],[],[],[]]
amplitudes=np.array([0.5,0.6,0.7,0.75,0.80,0.90,0.98])     #not 0.5
lens=[17,11,9,7,7,4,3]       #last index i want
time_errs=np.array([120,25,20,20,20,10,10])
stable_time=np.zeros(7)

#Time evolution of all compartments, Main cycle
fig,axs=plt.subplots(2,4,figsize=(10,6),constrained_layout=True, sharex=True, sharey=True)
for id, j in enumerate (amplitudes):
    u=data.loc[data['Amplitude']==j]
    l=lens[id]
    z=10
    c=np.array([0.95,0,0.05,0.4])
    del_c=np.array([-0.95/(l+1),0,0.95/(l+1),0.5/(l+1)])
    pars,_=curve_fit(f,np.array(range(1,7)),np.array(u.iloc[l,[9,3,4,5,6,8]])/u.iloc[l,2])  #compare to final fit
    t=np.array(u['Time'])
    stable_time[id]=t[l]
    for i in range(0,l+1):
        #varying color, plotting bars, finding MSE
        x=np.array(range(1,7))
        y=np.array(u.iloc[i,[9,3,4,5,6,8]])/u.iloc[i,2]         #9,3,4,5,6,8 are the six compartments
        axs[id//4,id%4].bar(x[0],y[0], color=c,zorder=10-z)
        axs[id//4,id%4].bar(x[1:],y[1:], color=c,zorder=z)
        z-=1
        c+=del_c
        chis[id].append(MSE(y,f(x,*pars)))
    #Visual figure aspects
    ppars=[round(pars[i],2) for i in (0,1)]
    axs[id//4,id%4].plot(x,f(x,*pars),color='g',label=f'{ppars[0]}x+{ppars[1]}')
    axs[id//4,id%4].set_ylim(0,1)
    axs[id//4,id%4].tick_params(right = False , labelbottom = False, bottom=False)
    axs[id//4,id%4].set_title(f't={t[i]}s')
    axs[id//4,id%4].legend()
    axs[id//4,id%4].set_title(f'Amplitude {j} A')
fig.supylabel(r'${\mu}$ [-]')
axs[1,3].remove()
fig.savefig('CousTimedev.pdf')
```

```python


#Stabilization time decay Figure
fugus,axus=plt.subplots()
print(stable_time)
axus.errorbar(amplitudes,stable_time,xerr=0.01,yerr=time_errs,fmt='.')
axus.set_xlabel('Current Amplitude')
axus.set_ylabel('Time to stabilize')
#Fitting
gmodel = Model(exi)
result = gmodel.fit(stable_time, x=amplitudes, a=10, b=-1, c=0, method='least_squares')
#print(result.fit_report())
xxx=np.linspace(0.5,1,100,)
axus.plot(xxx,exi(xxx,40.74,-17.41,15.31), label=r'$y=40.74+e^{(-17.41*x+15.31)}$') #read from fit_report
axus.legend()

fugus.savefig('CousStabTime.pdf')

#Linearity over time Figure
figa,axsa=plt.subplots(figsize=(8,4),constrained_layout=True)
for k,a in enumerate (amplitudes):
    #Normalising time, plotting MSE
    u=data.loc[data['Amplitude']==a]
    t=np.array(u['Time'][:lens[k]+1])
    t=t/(t[lens[k]])
    axsa.plot(t,chis[k],label=f'Amplitude {a} A')
    axsa.set_xlabel('t [s]')
    axsa.set_ylabel('MSE to best fit of steady state')
    axsa.set_ylim(0)
    axsa.legend()
figa.savefig('CousMSE.pdf')
plt.show()
```

A.4 Computation script for the pre-steady state solutions

```python
import numpy as np
import matplotlib.pyplot as plt
import math


if __name__ == "__main__":
    coeffs = []
    numterms = 1000

    for i in range(numterms):
        n = i+1
        # initial condition LAMBDA(x) = 11/12
        I_lam = 11/(12*n*math.pi)*(1-math.cos(n*math.pi/6))

        I_s = 1/(n*math.pi)
        coeffs.append(2*(I_lam - I_s))

    #small list! not the axis
    timetotal = np.concatenate(
        (
            np.linspace(0,1,40),
            np.linspace(1,5,80),
            np.linspace(5,10,40),
            np.linspace(10,20,40),
            np.linspace(20,50,80),
            np.linspace(50,100,40),
        ),
        axis=None
    )

    xrange = np.linspace(0,1,1000)
    param_D = 0.005

    color = plt.cm.rainbow(np.linspace(0,1,len(timetotal))[::-1])

    fig, ax = plt.subplots(1,1,constrained_layout=True)
    # ax.set_ylim(bottom=0,top=6)
    # ax.set_xlim(left=0,right=1)
    ax.set_xticks([0,1/6,2/6,3/6,4/6,5/6,1],labels=["0","1/6","2/6","3/6","4/6","5/6","1"])
    ax.set_title(f"$T(x,t)$ for $0<t<100$, $D= {param_D}$, expanded to {numterms} terms\nwith $\Lambda(x)=11/12$ over $[0,1/6]$")
    ax.set_xlabel("$x$")
    ax.set_ylabel("$T(x,t)$")


    for j,t in enumerate(timetotal[::-1]):
        solvals = 1-xrange
        for i, cf in enumerate(coeffs):
            solvals += cf*np.sin((i+1)*np.pi*xrange)*np.exp(-(i+1)**2*np.pi**2*param_D*t)

        ax.plot(xrange,solvals,c=color[j])

    fig.savefig("initialplot-constant.pdf")
    plt.show()
```

A.5 Computation and plotting scripts of the heat distribution

```
import numpy as np
import matplotlib.pyplot as plt
import math


# computation of Q_k(t)


def compute_qvals(
        timetotal:np.ndarray,
        param_D:float=0.005,
        numterms:int=200,
        nbins:int=6,
        norm_first:bool=True,
        print_progress:bool=True
    ):
    """
    Computes the expected Q values over numbins boxes.
    """
    pi = math.pi
    # iterate over each of the bins and then for each time value compute the series
    qvals = np.zeros((len(timetotal), nbins))
    for i,k in enumerate(range(1,nbins+1)):
        # bin index i and k in formula
        qvals[:,i] += (13-2*k)/11

        for j,t in enumerate(timetotal):
            term = 0
            for n in range(1,numterms+1):
                term += 1/(n**2*pi**2) * (1+11*math.cos(n*pi/6)) * math.sin(n*pi/12*(2*k-1)) * math.sin(n*pi/12) * math.exp(-n**2 * pi**2 * param_D * t)

            qvals[j,i] -= (24/11)*term

            if norm_first:
                qvals[j,i] = qvals[j,i]/qvals[j,0] # fix the first bin to be 1

        if print_progress:
            print(f"Computed bin {k}")

    return qvals


def main(
        timetotal:np.ndarray,
        display:int,
        param_D:float=0.005,
        numterms:int=200,
        nbins:int=6,
        norm_first:bool=True,
        time_fixed:float=100
    ):
    """
    Main part of the program. Choose which plot to make.
    """
    qvals = compute_qvals(timetotal, param_D, numterms, nbins, norm_first)
    color = plt.cm.rainbow(np.linspace(0,1,len(timetotal))[::-1])

    if display == 0:
        return qvals

    if display == 1:
        ncols = 3
        fig, ax = plt.subplots(math.ceil(len(timetotal)/ncols),ncols, constrained_layout=True,sharex=True,sharey=True)
        for i, axis in enumerate(ax.flatten()):
            if i >= len(timetotal):
                continue
            axis.stairs(qvals[i])
            axis.set_title(f"$t={timetotal[i]}$")

            axis.set_xticks([0,1,2,3,4,5,6])
            axis.set_ylim(0,1)
            axis.set_xlim(0,6)

        fig.suptitle(f"The distribution of particles in bins w.r.to time, $D={param_D}$,\nexpanded to $n={numterms}$ terms")
        fig.supxlabel("Bins $k=1,...,6$")
        fig.supylabel("$Q_k(t)$")

        fig.savefig("plots/qvalsplot.pdf")

        return None

    if display == 2:
```

```python
        fig, ax = plt.subplots(1,1,constrained_layout=True, sharex=True, sharey=True)
        for i in range(nbins):
            ax.plot(timetotal*param_D,qvals[:,i],label=f"Bin $k={i+1}$")
            # ax.set_title(f"Bin $k={i+1}$")

        ax.legend(loc="upper right")
        fig.supxlabel("$Dt$, arb. units")
        fig.supylabel(f"$Q_k(t)$, arb. units, expanded to {numterms} terms")
        # fig.suptitle(f"The distribution of particles in bins w.r.to time, $D={param_D}$,\nexpanded to $n={numterms}$ terms")


        fig.savefig("plots/qvals-per-bin-plot.pdf")

        return None

    if display == 3:
        fig, ax = plt.subplots(1,1,constrained_layout=True, sharex=True, sharey=True)
        for i,qk in enumerate(qvals[::-1]):
            if i%100 == 0:
                ax.stairs(qk,color=color[i], label=f"{round(timetotal[::-1][i]*param_D,2)}")
                continue
            ax.stairs(qk,color=color[i])
        handles, labels = ax.get_legend_handles_labels()
        ax.legend(handles[::-1], labels[::-1], loc="upper right", title="$Dt$, arb. units")
        fig.supxlabel("Bin $k=1,...,6$")
        fig.supylabel(f"$Q_k(t)$, arb. units, expanded to {numterms} terms")
        # fig.suptitle(f"The distribution of particles in bins w.r.to time, $D={param_D}$,\nexpanded to $n={numterms}$ terms")


        fig.savefig("plots/qvals-oneplot.pdf")

        return None

    if display == 4:
        fig, ax = plt.subplots(1,1,constrained_layout=True, sharex=True, sharey=True)
        ax.plot(timetotal*param_D/time_fixed, qvals[:,-1])

        fig.supylabel(f"$Q_6(t={time_fixed})$")
        fig.supxlabel("$D$")
        fig.suptitle(f"$t={time_fixed}$")
        fig.savefig("plots/fluxplot.pdf")

        return None

    else:
        raise ValueError("Incorrectly specified display.")


if __name__ == "__main__":

    # timetotal = np.concatenate(
    #     (
    #         np.linspace(0,1,40),
    #         np.linspace(1,5,80),
    #         np.linspace(5,10,40),
    #          np.linspace(10,20,40),
    #          np.linspace(20,50,80),
    #          np.linspace(50,100,40),
    #      ),
    #      axis=None
    # )
    # timetotal = np.array([0,1,2,3,4,5,6,7,8,9,10,11,12,24,48,96,192,384])

    # make sure the t values is hit!
    timetotal = np.linspace(0,100,1001)

    main(timetotal,display=3,param_D=0.005,numterms=200,nbins=6,norm_first=True,time_fixed=100)
    exit()

```

A.6 Pre-steady state solution fitting script

```
import numpy as np
import math
import scipy as sp
import pandas as pd
import matplotlib.pyplot as plt
from typing import Callable
from distribution import compute_qvals


def qk_term_formula(
        timevalue:np.ndarray,
        param_D:float,
        bin_k:int,
        n:int
    ):
    """
    Computes the Q value term in box k for a fixed n.
    """
    pi = math.pi
    return 1/(n**2*pi**2) * (1+11*math.cos(n*pi/6)) * math.sin(n*pi/12*(2*bin_k-1)) * math.sin(n*pi/12) * np.exp(-n**2 * pi**2 * param_D * timevalue)


def compute_qk(
        timevalue:np.ndarray,
        param_D:float,
        bin_k:int,
        numterms:int=200,
        norm_first:bool=True,
    ):
    """
    Computes the expected Q value in box k.
    """

    # iterate over each of the bins and then for each time value compute the series
    qvalue = 0
    qvalue += (13-2*bin_k)/11

    term = 0
    for n in range(1,numterms+1):
        term += qk_term_formula(timevalue,param_D,bin_k,n)

    qvalue -= (24/11)*term

    # perform normalisation by the first value
    if norm_first:
        qfirst = 1
        term = 0
        for n in range(1,numterms+1):
            term += qk_term_formula(timevalue,param_D,bin_k,n)

        qfirst -= (24/11)*term

        qvalue = qvalue/qfirst # fix the first bin to be 1

    return qvalue


def function_to_fit(time_combined:np.ndarray, param_d:float):
    """
    Computes the function for fitting.
    """

    t1, t2, t3, t4, t5, t6 = np.hsplit(time_combined,6)

    result_k1 = compute_qk(t1,param_D=param_d,bin_k=1)
    result_k2 = compute_qk(t2,param_D=param_d,bin_k=2)
    result_k3 = compute_qk(t3,param_D=param_d,bin_k=3)
    result_k4 = compute_qk(t4,param_D=param_d,bin_k=4)
    result_k5 = compute_qk(t5,param_D=param_d,bin_k=5)
    result_k6 = compute_qk(t6,param_D=param_d,bin_k=6)

    return np.concatenate([result_k1,result_k2,result_k3,result_k4,result_k5,result_k6])


def plot_comparison(
        tvals:np.ndarray,
        qexperlist:np.ndarray,
        qtheorlist:np.ndarray,
        ampl:float,
        dval:float,
        derr:float,
        savetitle:str,
        ncols:int=5
    ):
    """
    Plots the experimental q values against the fitted q values.
    """

    nrows = int(math.ceil(len(tvals)/ncols))
    fig, ax = plt.subplots(nrows, ncols, figsize=(8,1+(7/5)*nrows), constrained_layout=True, sharex=True, sharey=True)

    for i, axis in enumerate(ax.flatten()):
        if i >= len(tvals):
            axis.remove()
            continue

        qexper = qexperlist[i]
        qtheor = qtheorlist[i]

        axis.stairs(qexper, color="r", label="Exp.")
```

```python
        axis.stairs(qtheor, color="b", label="Thr.")
        axis.set_title(f"$t={tvals[i]}$ s")
        axis.set_xticks([0,1,2,3,4,5,6])
        axis.set_aspect(6)

    # fig.legend(loc="lower right")
    fig.supxlabel("Bin")
    fig.supylabel("Normalised amount, arb. units")
    # fig.suptitle(f"Experimental and theoretical $Q_k$ values with $D={dval} \\pm {derr}$ Hz length at $I={ampl}$ A")
    fig.savefig("plots/" + f"ampl{ampl}" + savetitle + ".pdf")

    return None


def rnd_fl(num):
    """
    Round the float to the first non-zero digit.
    """
    if not num:
        return num, 0
    current_num = abs(num) * 10
    round_value = 1

    while not (current_num//1):
        current_num *= 10
        round_value +=1

    return round(num, round_value), round_value


def make_normal_qvals(data:np.ndarray):
    """
    Makes the normalised concatenated qvals for fitting
    and qvals for plotting.
    """
    # make normalised q values for amplitude
    normalisation = data[:,2]
    qvals = np.concatenate(
        [
        data[:,2]/normalisation,
        data[:,3]/normalisation,
        data[:,4]/normalisation,
        data[:,5]/normalisation,
        data[:,6]/normalisation,
        data[:,8]/normalisation #flux 8, zero 7
        ]
    )

    # make nonconcatenated q values as for plotting
    databefore = data[:, [2,3,4,5,6,8]]
    qvals_forplotting = databefore/np.transpose(np.tile(normalisation, (6,1)))

    return qvals, qvals_forplotting



def compute_fit(
        funct:Callable,
        tvals:np.ndarray,
        qvalsexp:np.ndarray,
        p0:float=0.0003,
        bounds:list[float]=[0.0001,0.0005],
        compfitname:str=""
        ) -> tuple[float, float, float, float]:

    # compute fitting
    tvals_concat = np.tile(tvals,6)
    params, pcovs = sp.optimize.curve_fit(funct, tvals_concat, qvalsexp, p0=p0, bounds=bounds)
    perr = np.sqrt(np.diag(pcovs))[0]
    dparam = params[0]

    print(f"Executing compute_fit {compfitname}")
    print(f"D={params} +- {perr} Hz length^2")

    rnd_perr, dig_p = rnd_fl(perr)
    rnd_p = round(params[0], dig_p)

    return dparam, perr, rnd_p, rnd_perr


def funcexp(x, a, c):
    return a*np.exp(c*x)


def round_to_error(f:float, ef:float) -> tuple[float, float]:
    """
    Given a number f and its error ef, returns the number and error
    rounded to the first significant digit of the error.
    """

    rnd_ef, dig_ef = rnd_fl(ef)
    rnd_f = round(f, dig_ef)

    return rnd_f, rnd_ef


def xpdecomp(f:float):
    """
    Decomposes a number into a base 10 exponent and scalar.
    Example 0.00056 -> 5.6, -4

    Credits: jpm (StackOverflow).
    """
    fexp = int(math.floor(math.log10(abs(f)))) if f != 0 else 0
    return f/10**fexp, fexp


def num_decomp_error(f:float, ef:float) -> tuple[float, float, int]:
    """
    Given a number and its error, returns the number, error
    and common exponent.
    """
    f, ef = round_to_error(f, ef)

    f_man, f_exp = xpdecomp(f)
    ef_man, ef_exp = xpdecomp(ef)

    if f_exp*ef_exp <= 0:
        print(f"No valid exp decomposition for {f} and {ef}")
        return f, ef, 0

    common_exp = max(f_exp, ef_exp) if f_exp<0 else min(f_exp, ef_exp)
    rnd_f, rnd_ef = round_to_error(f/10**common_exp, ef/10**common_exp)

    return rnd_f, rnd_ef, common_exp




if __name__ == "__main__":
    # initialising data
    df = pd.read_csv("data/datafile.csv")

    data = df.to_numpy()
    data = data[8:]

    ampl5  = data[np.where(data == 0.5)[0]]
    ampl6  = data[np.where(data == 0.6)[0]]
    ampl7  = data[np.where(data == 0.7)[0]]
    ampl75 = data[np.where(data == 0.75)[0]]
    ampl8  = data[np.where(data == 0.8)[0]]
    ampl9  = data[np.where(data == 0.9)[0]]
    ampl98 = data[np.where(data == 0.98)[0]]

    tvallist = []
    datalist = [ampl5, ampl6, ampl7, ampl75, ampl8, ampl9, ampl98]


    # make a list of all t values to try, seconds
    for i in range(len(datalist)):
        tvallist.append(datalist[i][:,1])


    # subtract the flux
    for i in range(len(datalist)):
        ti = datalist[i][:,1]
        q60 = datalist[i][:,7]
        qi = datalist[i]
        x = datalist[i][:,8][:, np.newaxis]
        datalist[i] = np.abs(qi - x)
        datalist[i][:,1] = ti
        datalist[i][:,7] = q60


    # execute code for each data
    cont = [1,1,1,1,1,1,1] # select which to fit (1=fit, 0=skip)
    ampls = [0.5, 0.6, 0.7, 0.75, 0.8, 0.9, 0.98]
    p0s = [0.0001,0.0003,0.0015,0.02,0.03,0.04,0.04]
    boundvals = [
        [0.00005, 0.0002],
        [0.00015, 0.001],
        [0.0003, 1],
        [0.00055, 1],
        [0.0002, 1],
        [0.0004, 1],
        [0.0005, 1],
    ]
    dfitted = []
    dfittederr = []
    for i in range(len(tvallist)):
        if cont[i] == 0:
            print(f"Skipping amplitude {ampls[i]}")
            continue
        tvals = tvallist[i]
        qvals, qvals_forplotting = make_normal_qvals(datalist[i])
        dpar, perr, rnd_d, rnd_perr = compute_fit(function_to_fit, tvals, qvals, p0=p0s[i], bounds=boundvals[i], compfitname=str(ampls[i]))
        qtheor = compute_qvals(tvals, param_D=dpar, print_progress=False)
        dfitted.append(dpar)
        dfittederr.append(perr)


        # toggle plotting (very time consuming)
        plot_comparison(tvals, qvals_forplotting, qtheor, ampls[i], rnd_d, rnd_perr, "")


    # begin making the final plot of D in terms of I
    dfitted = np.array(dfitted)
    dfittederr = np.array(dfittederr)

    length_unit = 20 # centimeters
    dfitted *= length_unit**2
    dfittederr *= length_unit**2
    arr_ampls = np.array(ampls)

    for i in range(len(dfitted)):
        dr, drerr = round_to_error(dfitted[i], dfittederr[i])
        print(f"I = {ampls[i]} A, D = {dr} +- {drerr} Hz cm^2")


    fig, ax = plt.subplots(1,1, constrained_layout=True, sharex=True, sharey=True)

    eparams, epcovs = sp.optimize.curve_fit(lambda x,a,c: a*x+c, arr_ampls, dfitted, sigma=dfittederr)
    eerrs = np.sqrt(np.diag(epcovs))

    xd_pa, xd_erra, xd_expa = num_decomp_error(eparams[0], eerrs[0])
    xd_pc, xd_errc, xd_expc = num_decomp_error(eparams[1], eerrs[1])

    xvals = np.linspace(np.min(arr_ampls), 1, 1000)
    yvals = xvals*eparams[0]+eparams[1]
    yerror = np.sqrt((xvals*eerrs[0])**2 + eerrs[1]**2)

    ax.plot(
        xvals, yvals, c="#336ea0",
        label="$D=aI+b$, where \n" +
              "$a=(%g"%(xd_pa) + "\\pm %g"%(xd_erra) + ")\\times 10^{%g"%(xd_expa) + "}$ Hz cm$^2$ " + "A$^{-1}$, \n" +
             f"$c=(%g"%(xd_pc) + "\\pm %g"%(xd_errc) + ")\\times 10^{%g"%(xd_expc) + "}$ Hz cm$^2$"
    )
    ax.fill_between(xvals, yvals-yerror, yvals+yerror, alpha=0.5, facecolor="#dde9f4", edgecolor="#74a7d2")

    ax.errorbar(arr_ampls, dfitted, yerr=dfittederr, xerr=0.01*np.ones_like(arr_ampls), ecolor="black", color='r', linestyle='', marker='.')
    # ax.set_yscale("log")
    ax.legend(loc="lower right")
    fig.supxlabel("$I$, A")
    fig.supylabel("$D$, Hz cm$^2$")
    fig.savefig("plots/diplot.pdf")

    print("Created D/I plot")

    exit()
```

A.7 Stationary flow analysis script

```
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit
import numpy as np
import pandas as pd

amps=[0.50,0.60,0.70,0.80,0.75,0.90,0.98]
#weights=[206,258,309,413,335,413,413]

data = pd.read_csv('C:/Users/Cyber/Documents/lab_data.2.csv')
fluxes=[]
deviations=[]
for count,amp in enumerate(amps):
    run = data[data['Amplitude'] == amp]
    column=np.array(run["Sixth"].values)[-3:]
    #print(column)
    times=np.array(run["Time"].values)[-4:]
    print(times)
    print(column)
    deltas=np.array([times[i+1] - times[i] for i in range(len(times) - 1)])
    #column=(1/deltas)*column*(1/weights[count])
    column=(1/deltas)*column
    fluxes.append(np.mean(column))
    deviations.append(np.sqrt((0.01/deltas[-1])**2+(np.mean(column)*0.1/
    deltas[-1]**2)**2))
    #deviations.append(np.std(column))
print(fluxes,deviations)

def linear(x,k,b):
    """Linear function"""
    f=k*x+b
    return f
popt,pcov=curve_fit(linear,amps,fluxes)
perror = np.sqrt(np.diag(pcov))
print(np.sqrt(perror),"this is the error")
print(popt)

#print(pconv)
x1=np.linspace(0.5,1,100)
y1=linear(x1, popt[0], popt[1])
yerr = np.sqrt((x1*perror[0])**2 + perror[1]**2)

plt.plot(x1,linear(x1,popt[0],popt[1]),color="r",linestyle="dotted",
label=f"Linear fit $y={str(popt[0])[:5]}x+{str(popt[1])[:6]}$")
plt.legend()
plt.errorbar(amps,fluxes,xerr=0.01,yerr=deviations, fmt='o', capsize=5)
plt.fill_between(x1, y1-yerr, y1+yerr, alpha=0.5,facecolor="#dde9f4",
    edgecolor="#74a7d2")
plt.xlabel(f' $I$ amplitude $[A]$')
plt.ylabel('Mean steady particle flux $[g/s]$')

plt.grid()
plt.savefig("original_flux.pdf")
plt.show()
```

A.8 Wolfram Mathematica prompts

Here are given the relevant Wolfram Mathematica prompts used in the project, exported to a `*.tex` file and pasted here. To save space, some of the answers are given at the end of the line rather than below it.

Integrate[$1-x$,{x, 0, 1/6}] $\frac{11}{72}$

Integrate[$1-x$,{x, $(k-1)/6$, $k/6$}] $\frac{1-k}{6}+\frac{1}{72}(-1+k)^2+\frac{k}{6}-\frac{k^2}{72}$

FullSimplify[$\frac{1-k}{6}+\frac{1}{72}(-1+k)^2+\frac{k}{6}-\frac{k^2}{72}$] $\frac{1}{72}(13-2k)$

Sum $\left[\frac{1}{72}(13-2k),\ \{k,1,6\}\right]$ $\frac{1}{2}$

Integrate[Sin[nPix],{x, $(k-1)/6$, $k/6$}] $\frac{\text{Cos}\left[\frac{1}{6}(-1+k)n\pi\right]-\text{Cos}\left[\frac{kn\pi}{6}\right]}{n\pi}$

Sum[Refine[Sin[nPi/12$*$ $(2k-1)$],Element[n, Integers]],{k, 1, 6}]

$\text{Sin}\left[\frac{n\pi}{12}\right]+\text{Sin}\left[\frac{n\pi}{4}\right]+\text{Sin}\left[\frac{5n\pi}{12}\right]+\text{Sin}\left[\frac{7n\pi}{12}\right]+\text{Sin}\left[\frac{3n\pi}{4}\right]+\text{Sin}\left[\frac{11n\pi}{12}\right]$

Simplify[Sin $\left[\frac{n\pi}{12}\right]$ + Sin $\left[\frac{n\pi}{4}\right]$ + Sin $\left[\frac{5n\pi}{12}\right]$ + Sin $\left[\frac{7n\pi}{12}\right]$ +Sin $\left[\frac{3n\pi}{4}\right]$ + Sin $\left[\frac{11n\pi}{12}\right]$]

$4\left(\text{Cos}\left[\frac{n\pi}{12}\right]+\text{Cos}\left[\frac{n\pi}{4}\right]+\text{Cos}\left[\frac{5n\pi}{12}\right]\right)^2\text{Sin}\left[\frac{n\pi}{12}\right]$

B Appendix: supplementary information

B.1 Raw data tables

Here are all of the raw data tables (Table 3). On their own, they might be confusing therefore a short explanation was added in the results section (see Section 5). See the caption at the end of all tables.

t, ±0.01 s	m_1(g)	m_2(g)	m_3(g)	m_4(g)	m_5(g)	m_6(g)	m_{flux} (g)	$m_{1,\text{real}}$(g)
0.00	0.80	0.00	0.00	0.00	0.00	0.00	0.00	0.80
20.00	0.80	0.02(10)	0.01(3)	0.01(3)	0(1)	0.00	0.00	0.65
40.00	0.80	0.04(23)	0.02(9)	0.01(4)	0.01(3)	0.00	0.00	0.75
60.00	0.80	0.05(24)	0.03(14)	0.02(9)	0.01(3)	0.00	0.01(3)	0.75
80.00	0.80	0.05(27)	0.04(20)	0.02(9)	0.01(5)	0.00	0.01(3)	0.78
100.00	0.80	0.06(30)	0.04(20)	0.03(14)	0.01(7)	0.00	0.01(3)	0.74
120.00	0.80	0.07(35)	0.08(39)	0.03(13)	0.01(5)	0.00	0.01(6)	0.74
140.00	0.80	0.05(28)	0.07(38)	0.05(28)	0.02(12)	0.00	0.01(7)	0.76
170.00	0.80	0.05(25)	0.07(34)	0.04(23)	0.03(18)	0.00	0.02(8)	0.77
200.00	0.80	0.05(28)	0.07(35)	0.05(26)	0.03(16)	0.00	0.02(11)	0.71

(a) $I = (0.50 \pm 0.01)$ A.

t ±0.01s	m_1(g)	m_2(g)	m_3(g)	m_4(g)	m_5(g)	m_6(g)	m_{flux} (g)	$m_{1,\text{real}}$(g)
0.00	0.80	0.00	0.00	0.00	0.00	0.00	0.00	0.80
20.00	0.80	0.10(54)	0.06(31)	0.03(18)	0.02(9)	0.00	0.01(3)	0.58
30.00	0.80	0.17	0.09(45)	0.04(21)	0.03(17)	0.00	0.02(9)	0.70
40.00	0.80	0.18	0.14	0.05(26)	0.05(27)	0.00	0.03(14)	0.72
50.00	0.80	0.17	0.14	0.06(33)	0.06(33)	0.00	0.04(23)	0.63
60.00	0.80	0.13	0.11	0.07(35)	0.07(35)	0.00	0.04(22)	0.79
80.00	0.80	0.11	0.15	0.08(41)	0.06(30)	0.00	0.06(29)	0.68
100.00	0.80	0.20	0.13	0.07(35)	0.06(31)	0.00	0.07(36)	0.70
120.00	0.80	0.16(85)	0.09(47)	0.06(32)	0.07(34)	0.00	0.09(45)	0.52
140.00	0.80	0.15(79)	0.07(37)	0.08(39)	0.06(29)	0.00	0.04(19)	0.76
160.00	0.80	0.13(66)	0.12(60)	0.06(32)	0.07(36)	0.00	0.05(25)	0.66
180.00	0.80	0.13(66)	0.10(50)	0.05(22)	0.04(22)	0.00	0.04(21)	0.78

(b) $I = (0.60 \pm 0.01)$ A.

t ±0.01s	m_1(g)	m_2(g)	m_3(g)	m_4(g)	m_5(g)	m_6(g)	m_{flux} (g)	$m_{1,\text{real}}$(g)
0.00	0.80	0.00	0.00	0.00	0.00	0.00	0.00	0.80
20.00	0.80	0.16(96)	0.05(26)	0.03(14)	0.01(5)	0.00	0.01(6)	0.56
40.00	0.80	0.10(52)	0.10(54)	0.05(28)	0.02(9)	0.00	0.01(6)	0.71
60.00	0.80	0.15(79)	0.07(34)	0.07(24)	0.03(18)	0.00	0.03(18)	0.67
80.00	0.80	0.15(79)	0.09(46)	0.07(34)	0.04(22)	0.00	0.05(24)	0.63
100.00	0.80	0.14(72)	0.10(50)	0.06(29)	0.05(24)	0.00	0.04(21)	0.78
120.00	0.80	0.17(90)	0.08(39)	0.06(29)	0.06(30)	0.00	0.06(30)	0.63
150.00	0.80	0.14(74)	0.12(61)	0.07(35)	0.06(31)	0.00	0.08(39)	0.66

(c) $I = (0.70 \pm 0.01)$ A.

t ±0.01s	m_1(g)	m_2(g)	m_3(g)	m_4(g)	m_5(g)	m_6(g)	m_{flux} (g)	$m_{1,\text{real}}$(g)
0.00	0.80	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10.00	0.80	0.11	0.09(44)	0.02(10)	0.03(15)	0.00	0.02(9)	0.44
20.00	0.80	0.15	0.13(68)	0.05(24)	0.06(33)	0.00	0.05(25)	0.65
30.00	0.80	0.21	0.14	0.08	0.10(50)	0.00	0.08(40)	0.48
40.00	0.80	0.21	0.12	0.14	0.13	0.00	0.13	0.61
50.00	0.80	0.32	0.18	0.16	0.13	0.00	0.12	0.53
60.00	0.80	0.26	0.17	0.10	0.10	0.00	0.13	0.55
70.00	0.80	0.34	0.19	0.16	0.14	0.00	0.15	0.46
80.00	0.80	0.32	0.21	0.17	0.13	0.00	0.17	0.64
90.00	0.80	0.30	0.14	0.16	0.13	0.00	0.16	0.66
100.00	0.80	0.27	0.20	0.22	0.17	0.00	0.20	0.64
110.00	0.80	0.31	0.27	0.21	0.17	0.00	0.19	0.56

(d) $I = (0.80 \pm 0.01)$ A.

t ±0.01s	m_1(g)	m_2(g)	m_3(g)	m_4(g)	m_5(g)	m_6(g)	m_{flux} (g)	$m_{1,\text{real}}$(g)
0.00	0.80	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5.00	0.80	0.17	0.08(43)	0.03(15)	0.02(11)	0.00	0.00(2)	0.47
10.00	0.80	0.25	0.10	0.08(41)	0.05(23)	0.00	0.05(26)	0.58
15.00	0.80	0.28	0.15	0.09	0.07(38)	0.00	0.06(30)	0.59
20.00	0.80	0.34	0.16	0.16	0.09(48)	0.00	0.07(38)	0.54
25.00	0.80	0.34	0.19	0.15	0.11	0.00	0.09(48)	0.65
30.00	0.80	0.29	0.21	0.15	0.13	0.00	0.10(51)	0.81
35.00	0.80	0.23	0.16	0.12	0.14	0.00	0.14(72)	0.77
40.00	0.80	0.28	0.12	0.12	0.11	0.00	0.09	0.71
45.00	0.80	0.23	0.15	0.13	0.11	0.00	0.11(55)	0.68
50.00	0.80	0.27	0.17	0.11	0.09	0.00	0.10	0.68
55.00	0.80	0.25	0.18	0.14	0.09	0.00	0.09	0.68

(e) $I = (0.90 \pm 0.01)$ A.

t ±0.01s	m_1(g)	m_2(g)	m_3(g)	m_4(g)	m_5(g)	m_6(g)	m_{flux} (g)	$m_{1,\text{real}}$(g)
0.00	0.80	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5.00	0.80	0.18	0.08(43)	0.07(39)	0.04(20)	0.00	0.04(21)	0.37
10.00	0.80	0.24	0.10	0.10(50)	0.06(32)	0.00	0.07(36)	0.55
15.00	0.80	0.24	0.12	0.10(51)	0.08(42)	0.00	0.08(38)	0.60
20.00	0.80	0.27	0.10	0.11(56)	0.08(39)	0.00	0.08(43)	0.66
25.00	0.80	0.32	0.10	0.12(59)	0.08(40)	0.00	0.08(40)	0.67
30.00	0.80	0.34	0.11	0.09	0.08(38)	0.00	0.11(56)	0.60
35.00	0.80	0.30	0.18	0.10(48)	0.10(52)	0.00	0.10(51)	0.65
40.00	0.80	0.35	0.10	0.10(53)	0.09(45)	0.00	0.11(56)	0.72
45.00	0.80	0.33	0.16	0.11(56)	0.10(50)	0.00	0.10(48)	0.60

(f) $I = (0.98 \pm 0.01)$ A.

t ±0.01s	m_1(N)	m_2(N)	m_3(N)	m_4(N)	m_5(N)	m_6(N)	m_{flux} (N)	$m_{1,\text{real}}$(N)
0.00	413	0	0	0	0	0	0	413
5.00	413	50	16	8	1	0	0	335
10.00	413	51	29	13	5	0	4	382
15.00	413	84	38	20	12	0	5	351
20.00	413	88	56	23	18	0	7	351
25.00	413	77	73	26	27	0	16	377
30.00	413	108	56	32	27	0	7	371
35.00	413	83	67	40	35	0	12	382
40.00	413	77	83	44	36	0	13	377
45.00	413	103	77	62	51	0	15	340

(g) $I = (0.75 \pm 0.01)$ A.

t ±0.01s	m_1(N)	m_2(N)	m_3(N)	m_4(N)	m_5(N)	m_6(N)	m_{flux} (N)	$m_{1,\text{real}}$(N)
0.00	206	0	0	0	0	0	0	206
10.00	206	11	2	2	0	0	0	186
20.00	206	16	5	0	1	0	0	191
40.00	206	26	9	6	1	0	1	175
60.00	206	23	12	6	2	0	4	196
120.00	206	34	19	4	8	0	2	181
180.00	206	44	32	13	15	0	3	160
240.00	206	44	25	22	15	0	6	191
300.00	206	45	32	20	24	0	5	139
360.00	206	52	36	20	24	0	9	175
420.00	206	54	44	32	21	0	4	170
480.00	206	58	44	45	29	0	13	175
540.00	206	68	53	34	40	0	10	181
570.00	206	51	52	36	41	0	5	217
600.00	206	56	56	26	46	0	7	186
660.00	206	67	57	33	38	0	15	165

(h) $I = (0.50 \pm 0.01)$ A.

t ±0.01s	m_1(N)	m_2(N)	m_3(N)	m_4(N)	m_5(N)	m_6(N)	m_{flux} (N)	$m_{1,\text{real}}$(N)
0.00	206	0	0	0	0	0	0	206
10.00	206	11	2	2	0	0	0	186
20.00	206	16	5	0	1	0	0	191
40.00	206	26	9	6	1	0	1	175
60.00	206	23	12	6	2	0	4	196
120.00	206	34	19	4	8	0	2	181
180.00	206	44	32	13	15	0	3	160
240.00	206	44	25	22	15	0	6	191
300.00	206	45	32	20	24	0	5	139
360.00	206	52	36	20	24	0	9	175
420.00	206	54	44	32	21	0	4	170
480.00	206	58	44	45	29	0	13	175
540.00	206	68	53	34	40	0	10	181
570.00	206	51	52	36	41	0	5	217
600.00	206	56	56	26	46	0	7	186
660.00	206	67	57	33	38	0	15	165
720.00	206	75	53	41	37	0	9	175
780.00	206	76	68	36	30	0	17	165

(i) $I = (0.60 \pm 0.01)$ A.

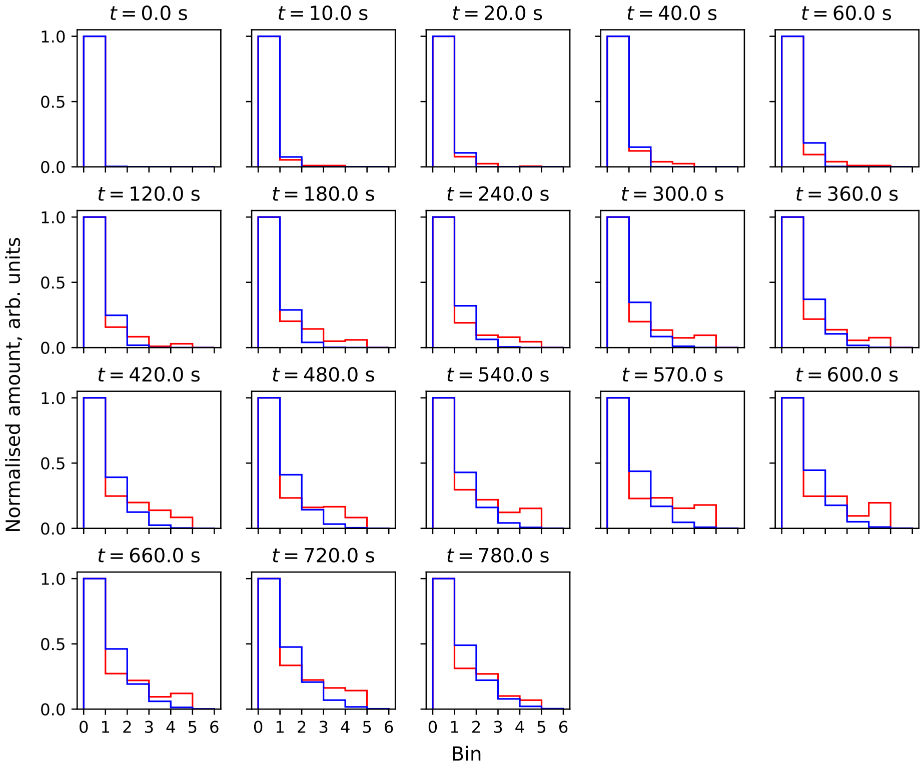
t ±0.01s	m_1(N)	m_2(N)	m_3(N)	m_4(N)	m_5(N)	m_6(N)	m_{flux} (N)	$m_{1,\text{real}}$(N)
0.00	206	0	0	0	0	0	0	206
10.00	206	11	2	2	0	0	0	186
20.00	206	16	5	0	1	0	0	191
40.00	206	26	9	6	1	0	1	175
60.00	206	23	12	6	2	0	4	196
120.00	206	34	19	4	8	0	2	181
180.00	206	44	32	13	15	0	3	160
240.00	206	44	25	22	15	0	6	191
300.00	206	45	32	20	24	0	5	139
360.00	206	52	36	20	24	0	9	175
420.00	206	54	44	32	21	0	4	170
480.00	206	58	44	45	29	0	13	175
540.00	206	68	53	34	40	0	10	181
570.00	206	51	52	36	41	0	5	217
600.00	206	56	56	26	46	0	7	186
660.00	206	67	57	33	38	0	15	165
720.00	206	75	53	41	37	0	9	175
780.00	206	76	68	36	30	0	17	165

(j) $I = (0.70 \pm 0.01)$ A.

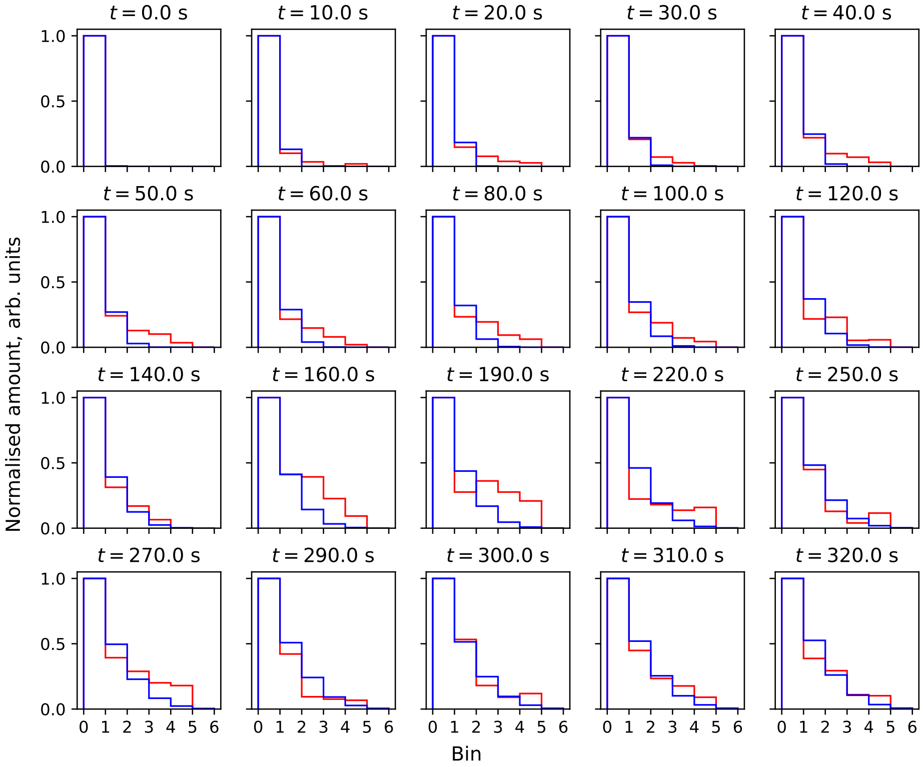
Table 3: The masses in each compartment depending on time for the current I value indicated under each table. The value in the brackets indicates the amount of particles if the mass was determined by counting. The error of all masses is $\Delta m(\text{g}) = \pm 0.01\,\text{g}$. The unit N denotes the number of particles. In the latest measurements, particles in nearly all of the compartments were counted.

B.2 Plots of the fitted pre-steady state solution

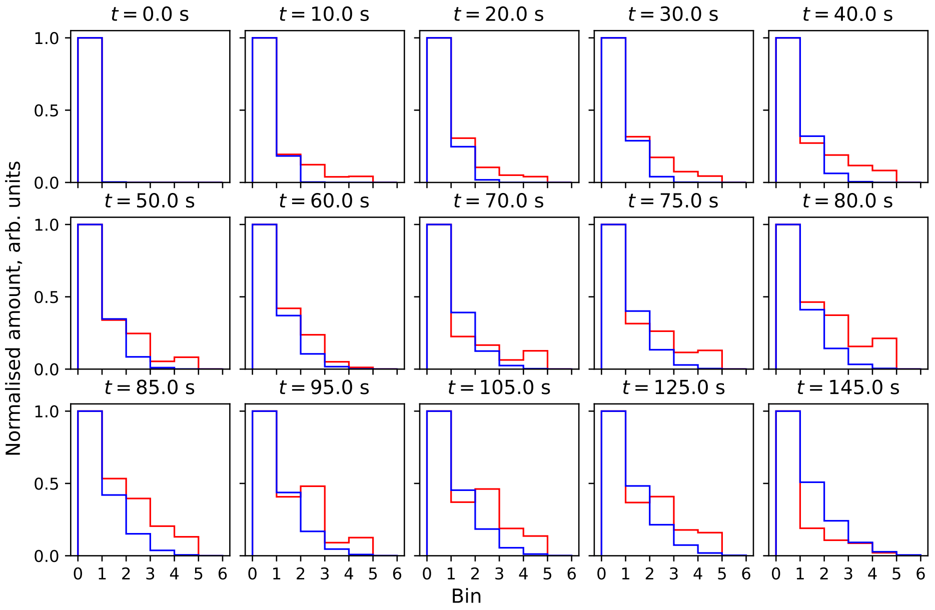
See the caption at the end of the figures.



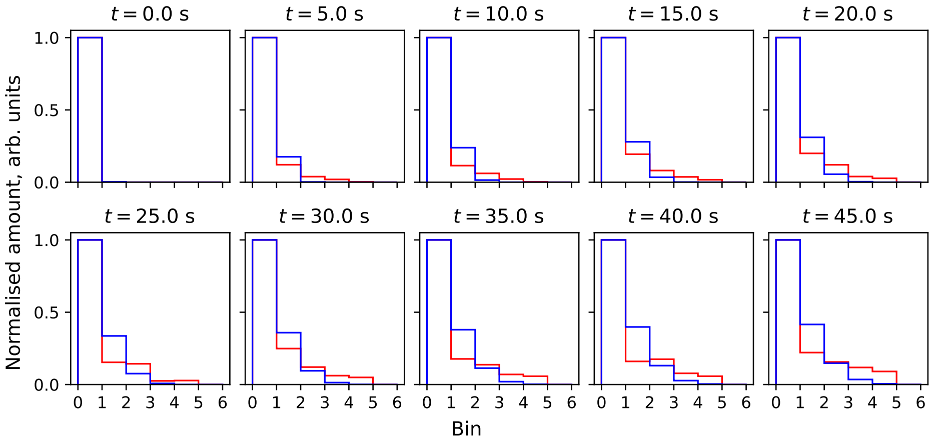
(a) $I = 0.5\,\mathrm{A},\ D = (0.020 \pm 0.004)\,\mathrm{Hz\,cm^2}$.



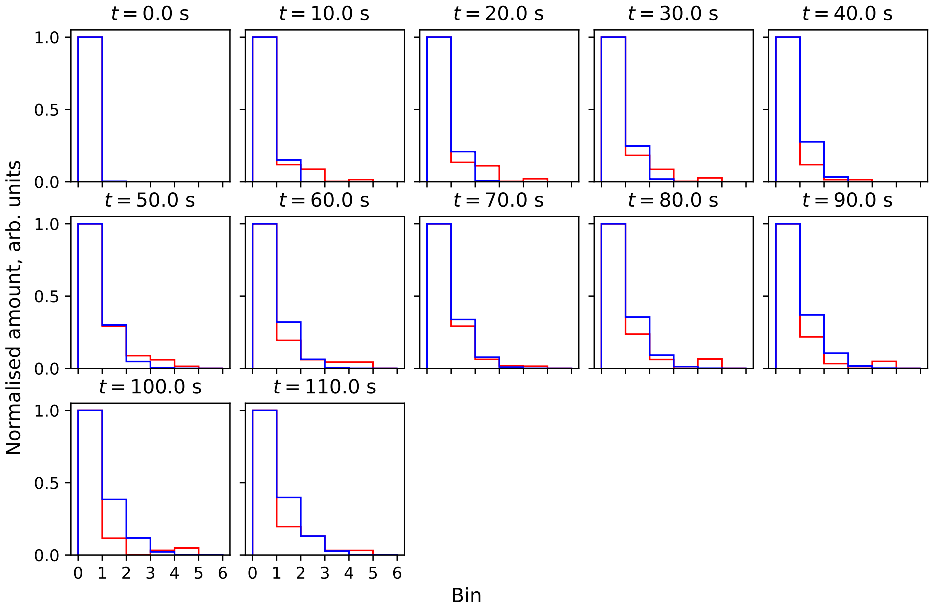
(b) $I = 0.6\,\mathrm{A},\ D = (0.06 \pm 0.01)\,\mathrm{Hz\,cm^2}$.



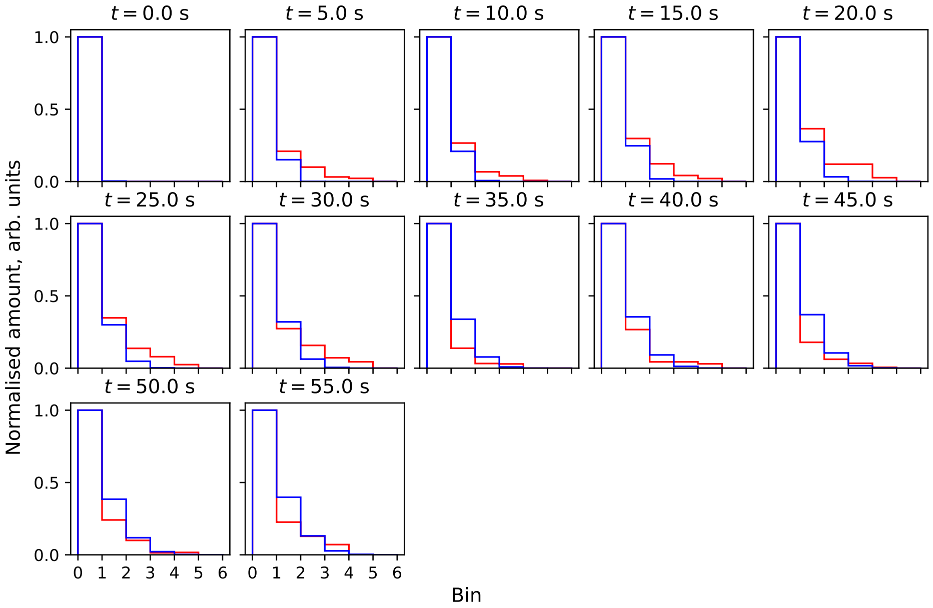
(c) $I = 0.7\,\mathrm{A},\ D = (0.12 \pm 0.02)\,\mathrm{Hz\,cm^2}$.



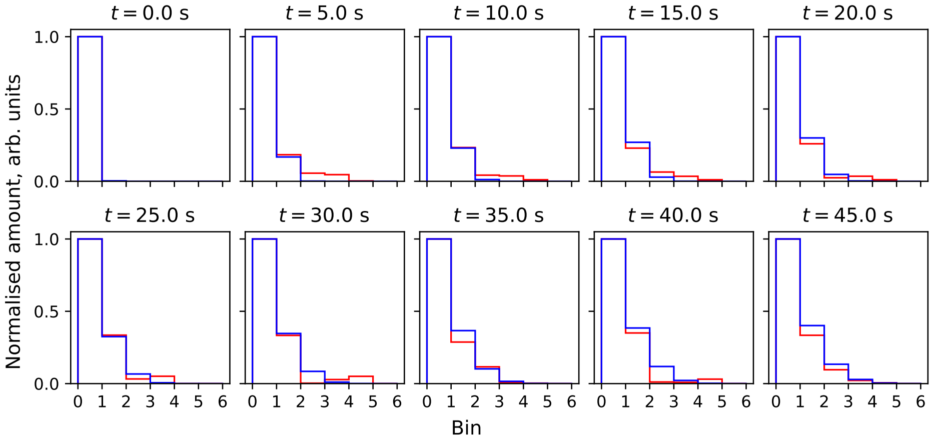
(d) $I = 0.75\,\mathrm{A},\ D = (0.22 \pm 0.07)\,\mathrm{Hz\,cm^2}$.



(e) $I = 0.8\,\mathrm{A},\ D = (0.08 \pm 0.02)\,\mathrm{Hz\,cm^2}$.



(f) $I = 0.9\,\mathrm{A}$, $D = (0.16 \pm 0.04)\,\mathrm{Hz\,cm^2}$.



(g) $I = 0.98\,\mathrm{A}$, $D = (0.20 \pm 0.05)\,\mathrm{Hz\,cm^2}$.

Figure 11: The experimental (red) $Q_k(t)$ values with the flux (i.e. $Q_6(t)$ value) subtracted, and the best fitting theoretical (blue) $Q_k(t)$ values, with the resulting D parameter displayed under each plot, depending on the current I applied and time t.

B.3 Supplement to the theory section

B.3.1 Derivation of the heat equation

This appendix refers to Section 2.1. The derivation is adapted from [12]. From elementary thermodynamics it is known that heat entering the system q upon a temperature increase of ΔT is given by:

$$q = \sigma m \Delta T = \sigma \Delta T \cdot \delta L. \tag{24}$$

Consider an interval Δx along the rod between x and $x + \Delta x$; for Δx small enough that the linear approximation may be made and the temperature of the segment considered to be given by the temperature at the midpoint given by $T(x + \Delta x/2, t)$.[2] If the temperature of the rod at this interval changes by ΔT during time Δt, then the heat entering the rod at this interval must be:

$$q = \sigma[T(x + \Delta x/2, t + \Delta t) - T(x + \Delta x/2, t)]\delta \Delta x. \tag{25}$$

It is also known from thermodynamics that the heat flux $\Phi(x, t)$, defined as the amount of heat per unit time crossing the point x in the positive direction, satisfies the Fourier heat law

$$\Phi(x, t) = -k \frac{\partial T}{\partial x}(x, t), \tag{26}$$

which is used to determine the heat q flowing into Δx during Δt as the heat flowing in from x minus the heat flowing out through $x + \Delta x$ given below

$$q = [\Phi(x, t) - \Phi q(x + \Delta x, t)]\Delta t = -k \left[\frac{\partial T}{\partial x}(x, t) - \frac{\partial T}{\partial x}(x + \Delta x, t)\right]. \tag{27}$$

Equating both equations for q, namely, Eqs. (25) and (27), it is obtained that

$$\begin{aligned} \sigma[T(x + \Delta x/2, t + \Delta t) - T(x + \Delta x/2, t)]\delta \Delta x &= -k \left[\frac{\partial T}{\partial x}(x, t) - \frac{\partial T}{\partial x}(x + \Delta x, t)\right] \\ \frac{T(x + \Delta x/2, t + \Delta t) - T(x + \Delta x/2, t)}{\Delta t} &= \frac{k}{\sigma \delta} \frac{\left[\frac{\partial T}{\partial x}(x + \Delta x, t) - \frac{\partial T}{\partial x}(x, t)\right]}{\Delta x}, \end{aligned}$$

whereupon taking the limit $\Delta t \to 0$ of both sides, from the definition it is obtained that the partial derivative for t

$$\frac{\partial T}{\partial t}(x + \Delta x/2) = \frac{k}{\sigma \delta} \frac{\frac{\partial T}{\partial x}(x + \Delta x, t) - \frac{\partial T}{\partial x}(x, t)}{\Delta x},$$

and similarly for $\Delta x \to 0$,

$$\begin{aligned} \frac{\partial T}{\partial t}(x, t) &= \frac{k}{\sigma \delta} \frac{\partial}{\partial x} \left[\frac{\partial \frac{\partial T}{\partial x}(x, t)}{\partial x}\right] \\ \frac{\partial T}{\partial t}(x, t) &= \frac{k}{\sigma \delta} \frac{\partial^2 T}{\partial x^2}(x, t), \end{aligned}$$

obtaining the heat equation in one spatial dimension given by Eq. (1).

[2] Alternatively, the limit as $\Delta x \to 0$ may be taken, which by the continuity of T gives one value of T.

B.3.2 Lemma for existence of solutions

This appendix refers to Section 2.3.1.

Lemma 1. *The boundary condition given in Eq.* (8) *may be satisfied by nontrivial functions* ρ *and* θ*, namely,* $\rho \neq 0$*,* $\theta \neq 0 \quad \forall t > 0$*, only if* $\lambda < 0$*.*

Proof. In *Eq.* (8) consider the equation in $\rho(x)$; clearly this is a homogeneous second-order ordinary differential equation for which solutions are known. The three cases of $\lambda > 0$, $\lambda = 0$, and $\lambda < 0$ are considered separately.

Case 1. Let $\lambda > 0$. To solve the equation for $\rho(x)$, a trial function $\rho(x) = e^{rx}$, where $r \in \mathbb{C}$ is a constant, is employed; then $\rho''(x) = r^2 e^{rx}$, thus $r^2 e^{rx} - \lambda e^{rx} = 0 \Leftrightarrow (r^2 - \lambda)e^{rx} = 0 \Leftrightarrow r^2 - \lambda = 0 \Leftrightarrow r = \pm\sqrt{\lambda}$ (note that $r \in \mathbb{R}$), obtaining that the solution is the linear combination

$$\rho(x) = \alpha e^{\sqrt{\lambda}x} + \beta e^{-\sqrt{\lambda}x},$$

where $\alpha, \beta \in \mathbb{R}$ are constants. Plugging in the boundary conditions, it is obtained that $0 = \rho(0) = \alpha + \beta$ and $0 = \rho(L) = \alpha e^{\sqrt{\lambda}L} + \beta e^{-\sqrt{\lambda}L}$. From the first equation, $\alpha = -\beta$, whereas from the second since $e^{[\cdots]}x$ is strictly-positive, by inserting α, it follows that $0 = -\beta e^{\sqrt{\lambda}L} + \beta e^{-\sqrt{\lambda}L} \Leftrightarrow \beta(e^{\sqrt{\lambda}L} - e^{-\sqrt{\lambda}L}) = 0$, therefore $\beta = 0$ and $\alpha = -\beta = 0$, obtaining for this case only the trivial solution $\rho(x) = 0$, which is not applicable. Hence $\lambda \leq 0$.

Case 2. Let $\lambda = 0$. Then $\rho'' = 0$, thus $\rho(x) = \alpha x + \beta$, $\alpha, \beta \in \mathbb{R}$. It is required that $0 = \rho(0) = \beta$, thus $\beta = 0$, and $0 = \rho(L) = \alpha L$, where since $L \neq 0$, it follows that $\alpha = 0$, obtaining only the trivial solution.

Case 3. Let $\lambda < 0$. Define $\sqrt{\lambda} = i\omega \in \mathbb{C}$, where $\omega \in \mathbb{R}$ and i is the imaginary unit. Then $\rho(x) = \alpha e^{\sqrt{\lambda}x} + \beta e^{-\sqrt{\lambda}x} = \alpha e^{i\omega x} + \beta e^{-i\omega x}$, which may be converted to $\rho(x) = A\cos(\omega x) + B\sin(\omega x)$. The boundary conditions give $0 = \rho(0) = A \cdot 1 + B \cdot 0 = A$, thus $A = 0$; and $0 = \rho(L) = B\sin(\omega x)$. There exists a solution for $B \neq 0$, which is found as follows: the sinus is zero for $\omega x = n\pi$, where $n \in \mathbb{Z}_{\geq 1}$ is a positive integer, thus $x = n\pi/\omega$. From the other boundary condition, it is obtained that $0 = \rho(L) = B\sin(\omega L)$, thus $\omega = n\pi/L$, which gives $\lambda = -\omega^2 = -n^2\pi^2/L^2$, obtaining a nontrivial solution, as was to be shown. □

B.3.3 Finding the Fourier expansion coefficients

This appendix concerns the procedure in Section 2.3.2. The coefficients γ_n of Eq. (12) of may be found following a procedure alike to finding the Fourier series coefficients. It is easier to find the coefficients for $h(x,t)$ with the initial state $S(x,0) = \Lambda(x) - s(x)$ (note that $e^{[\ldots]0} = 1$), that is

$$h(x,0) = \sum_{n=1}^{\infty} \gamma_n \sin\left(\frac{n\pi x}{L}\right) = S(x). \tag{28}$$

Assuming that the sum converges (a required criterion), a procedure is followed by multiplying both sides by $\sin(m\pi x/L)$ for some fixed $m \in \mathbb{Z}_{\geq 1}$ and integrating from 0 to L, obtaining

$$\int_0^L \sum_{n=1}^{\infty} \gamma_n \sin\left(\frac{n\pi x}{L}\right) \sin\left(\frac{m\pi x}{L}\right)\, dx = \int_0^L S(x) \sin\left(\frac{m\pi x}{L}\right)\, dx \tag{29}$$

$$\sum_{n=1}^{\infty} \gamma_n \int_0^L \sin\left(\frac{n\pi x}{L}\right) \sin\left(\frac{m\pi x}{L}\right)\, dx = \int_0^L S(x) \sin\left(\frac{m\pi x}{L}\right)\, dx, \tag{30}$$

where the convergence of the series to swap the integral was exploited by the Fubini/Tonelli theorems. It is known that the set of functions $\Omega = \{\sin(n\pi x/L) : n \in \mathbb{Z}_{\geq 1}\}$ is an orthogonal basis for the space of continuous real-valued functions $C_{\mathbb{R}}[-L, L]$, therefore $\forall n \neq m$ the extended integral from $-L$ to L evaluates to zero, whereas for $n = m$ the computation reveals that

$$\begin{aligned} I := & \int_{-L}^{L} \sin\left(\frac{n\pi x}{L}\right) \sin\left(\frac{n\pi x}{L}\right)\, dx = \int_{-L}^{L} \sin^2\left(\frac{n\pi x}{L}\right)\, dx \\ &= \frac{1}{2}\int_{-L}^{L}\left[1 - \cos\left(\frac{2n\pi x}{L}\right)\right] dx = \frac{1}{2}\left[2L - \int_{-L}^{L}\cos\left(\frac{2n\pi x}{L}\right)\right] dx \\ &= L - \frac{1}{2}\frac{L}{2n\pi}\sin\left(\frac{2n\pi x}{L}\right)\Bigg|_{-L}^{L} = L - \frac{L}{4n\pi}[\sin(2n\pi) - \sin(-2n\pi)] \\ &= L \qquad \forall n \in \mathbb{Z}_{\geq 1}, \end{aligned}$$

therefore

$$\int_{0}^{L} \sin\left(\frac{n\pi x}{L}\right) \sin\left(\frac{n\pi x}{L}\right)\, dx = \frac{L}{2}. \tag{31}$$

Combining the results above, the result for the general case is obtained

$$\int_{0}^{L} \sin\left(\frac{n\pi x}{L}\right) \sin\left(\frac{m\pi x}{L}\right)\, dx = \frac{L}{2}\delta_{nm}, \tag{32}$$

where $\delta_{[.][.]} : \mathbb{Z}_{\geq 1} \times \mathbb{Z}_{\geq 1} \to \{0, 1\}$ is the Kronecker delta function ($\delta_{nm} = 1$ if $n = m$, 0 otherwise). Inserting Eq. (32) into Eq. (30),

$$\int_{0}^{L} S(x)\sin\left(\frac{m\pi x}{L}\right)\, dx = \sum_{n=1}^{\infty}\gamma_n \int_{0}^{L} \sin\left(\frac{n\pi x}{L}\right) \sin\left(\frac{m\pi x}{L}\right)\, dx = \sum_{n=1}^{\infty}\gamma_n \frac{L}{2}\delta_{nm} = \frac{L}{2}\gamma_m,$$

therefore the coefficients γ_n are obtained to be (with $S(x) = \Lambda(x) - s(x)$)

$$\gamma_n = \frac{2}{L}\int_{0}^{L}[\Lambda(x) - s(x)]\sin\left(\frac{n\pi x}{L}\right)\, dx = \frac{2}{L}\left[\int_{0}^{L}\Lambda(x)\sin\left(\frac{n\pi x}{L}\right)\, dx - \int_{0}^{L} s(x)\sin\left(\frac{n\pi x}{L}\right)\, dx\right],$$

finishing the procedure.

B.3.4 Computations of integrals and simplifications

Series coefficients. Refer to Section 2.3.2. The following integrals are computed. Integrating for $\Lambda(x)$,

$$\begin{aligned} I_\Lambda(n) := & \int_{0}^{L}\Lambda(x)\sin\left(\frac{n\pi x}{L}\right)\, dx = \frac{11}{12}\int_{0}^{1/6}\sin(n\pi x)\, dx \qquad (33) \\ &= -\frac{11}{12n\pi}\cos(n\pi x)\Bigg|_{0}^{1/6} = \frac{11}{12n\pi}\left[1 - \cos\left(\frac{n\pi}{6}\right)\right]. \qquad (34) \end{aligned}$$

Similarly, a useful result is

$$I_s(n) := \int_{0}^{L} s(x)\sin\left(\frac{n\pi x}{L}\right)\, dx = \int_{0}^{1}(1 - x)\sin(n\pi x)\, dx = \frac{1}{n^2\pi^2}(n\pi - \sin(n\pi)) = \frac{1}{n\pi}. \tag{35}$$

Thus obtaining

$$\begin{aligned} \gamma_n &= 2\left[\frac{11}{12n\pi}\left(1 - \cos\frac{n\pi}{6}\right) - \frac{1}{n\pi}\right] = \frac{2}{n\pi}\left[\frac{11}{12} - \frac{11}{12}\cos\frac{n\pi}{6} - 1\right] \\ &= \frac{2}{n\pi}\left[-\frac{1}{12} - \frac{11}{12}\left(1 + \cos\frac{n\pi}{6}\right)\right] = -\frac{1}{6n\pi}\left[1 + 11\cos\frac{n\pi}{6}\right]. \end{aligned} \tag{36}$$

Distribution of particles. Refer to Section 2.3.4 Since $\forall n \geq 1$ the sequence of functions $f_n(x,t)$ under the summand is non-negative and the sum (series of function) converges, it follows by the Fubini/Tonelli theorems that the sum and integral may be exchanged. Therefore it follows that

$$\begin{aligned}
Q_k(t) &= \int_{(k-1)/6}^{k/6} T(x,t)\ dx \\
&= \int_{(k-1)/6}^{k/6} (1-x)\ dx - \int_{(k-1)/6}^{k/6} \sum_{n=1}^{\infty} \frac{1}{6n\pi}\left[1 + 11\cos\frac{n\pi}{6}\right] \sin(n\pi x) e^{-n^2\pi^2 Dt}\ dx \\
&= \int_{(k-1)/6}^{k/6} (1-x)\ dx - \sum_{n=1}^{\infty} \frac{1}{6n\pi}\left[1 + 11\cos\frac{n\pi}{6}\right] e^{-n^2\pi^2 Dt} \int_{(k-1)/6}^{k/6} \sin(n\pi x)\ dx \\
&= x - \frac{1}{2}x^2 \Big|_{(k-1)/6}^{k/6} + \sum_{n=1}^{\infty} \frac{1}{6n\pi}\left[1 + 11\cos\frac{n\pi}{6}\right] e^{-n^2\pi^2 Dt} \frac{1}{n\pi}\left[\cos(n\pi x)|_{(k-1)/6}^{k/6}\right] \\
&= \frac{1}{72}(13-2k) - 2\sum_{n=1}^{\infty} \frac{1}{6n\pi}\left[1 + 11\cos\frac{n\pi}{6}\right] \frac{1}{n\pi} \sin\left[\frac{n\pi}{12}(2k-1)\right] \sin\frac{n\pi}{12} e^{-n^2\pi^2 Dt} \\
&= \frac{1}{72}(13-2k) - \frac{1}{3}\sum_{n=1}^{\infty} \frac{1}{n^2\pi^2}\left[1 + 11\cos\frac{n\pi}{6}\right] \sin\left[\frac{n\pi}{12}(2k-1)\right] \sin\frac{n\pi}{12} e^{-n^2\pi^2 Dt},
\end{aligned}$$

where the trigonometric transformation $\cos\alpha - \cos\beta = -2\sin\frac{1}{2}(\alpha+\beta)\sin\frac{1}{2}(\alpha-\beta)$ was used to rearrange

$$\begin{aligned}
\cos(n\pi x)|_{(k-1)/6}^{k/6} &= \cos\frac{n\pi k}{6} - \cos\frac{n\pi(k-1)}{6} \\
&= -2\sin\frac{1}{2}\left[\frac{n\pi k}{6} + \frac{n\pi(k-1)}{6}\right] \sin\frac{1}{2}\left[\frac{n\pi k}{6} - \frac{n\pi(k-1)}{6}\right] \\
&= -2\sin\frac{n\pi}{12}[k+(k-1)] \sin\frac{n\pi}{12}[k-(k-1)] \\
&= -2\sin\left[\frac{n\pi}{12}(2k-1)\right] \sin\frac{n\pi}{12}.
\end{aligned}$$

C Appendix: error propagation formulas

The error propagation formula for $Q = f(x_1, x_2, ..., x_n)$ is

$$\Delta Q = \sqrt{\left(\frac{\partial f}{\partial x_1}\Delta x_1\right)^2 + \left(\frac{\partial f}{\partial x_2}\Delta x_1\right)^2 + \cdots + \left(\frac{\partial f}{\partial x_n}\Delta x_1\right)^2}. \tag{37}$$

The standard deviation of the mean:

$$\sigma = \sqrt{\frac{\sum(\dot{q}_n - \dot{q}_{mean})}{(n-1)}} \tag{38}$$

The uncertainty of the slope of the linear best-fit line:

$$\Delta a = \frac{a_{max} - a_{min}}{2} \tag{39}$$

Least-squares method for finding a as the coefficient of the linear function:

$$a = \frac{N\Sigma(xy) - \Sigma(x)\Sigma(y)}{N\Sigma(x^2) - (\Sigma(x))^2} = \frac{\bar{xy} - \bar{x}\bar{y}}{\bar{x^2} - \bar{x}^2} \tag{40}$$

Error propagation of the linear fit $y(x) = ax + b$, where standard deviations Δa and Δb are known. By Eq. (37), it is obtained that

$$\Delta y = \sqrt{(x\Delta a)^2 + (\Delta b)^2}. \tag{41}$$

The propagation formula for the regular flux:

$$\Delta \dot{q} = \sqrt{\left(\frac{\Delta m}{t}\right)^2 + \left(\frac{\Delta t m}{t^2}\right)^2} \tag{42}$$

The error of the weight conversion rate $c = N/m$ with mass m and number of particles N

$$\Delta c = \sqrt{\left(\frac{\Delta m}{m}\right)^2 + \left(\frac{\Delta N}{N}\right)^2} \tag{43}$$

D Appendix: digital logbook

During the experiment, notes were taken in a Digital lab notebook. The notebook mainly contains information about what was observed and what were the obstacles of the people conducting this experiment, as well as some general status updates.

Thu 18/4 11:00-14:00 (Tutorial 1)

Stanislavs

The first tutorial was held where the brainstorming session happened. The following ideas were discussed: The conductivity of fire. The possibility of measuring the conductivity of fire or other sorts of plasma doped with different amounts of metal ions(from the materials that are going to be burned). The idea was generally approved, but due to the fire safety rules the team will not be able to use any fire(plasma?)

Conductivity measurement. Idea status: Good idea, but very hard to execute due to regulations. Simulating phase transitions and heat distribution in heat-conducting materials using the Markov chains approach and random processes such as small plastic balls bouncing on dynamics.

Markov chanins IPhO problem (IPhO problem with a similar concept to our idea, in fact, we want to make a scaled-up version of the experimental setup in that olympiad problem). Idea status: Candidate for a plan A idea, interesting concept, cool theoretical part with Markov chains, theory of probability, and (possibly) simulations, the only problem is that we will need to buy all these plastic balls, the question is where? (Valid question) Coupled oscillations system with a magnet and a coil. Idea status: Potential candidate for a Plan B, it is possible to execute and create a theoretical part for this project, not very novel or exciting but definitely safe, which is exactly what we want for a plan B.

Fri 19/4 09:00-10:00 (Lab inspection)

Stanislavs

The Lab inspection was conducted by the following members of the team: Stanislavs, Eduard, Toms. All the safety rules were regarded. Observations: A pile of speakers was found which might come in very handy for the phase transition idea, the only question left is where to find the balls?

Eduard

The preliminary plans A and B were presented to and consulted with our Teaching Assistent. Feasibility and possible improvements were discussed. Two presenters were picked: Stanislavs and Toms. The workload for the plan presentation was divided between all four members. It was also decided that on the next day, available members would do another lab inspection to confirm details about the most recent plan A.

Wed 24/4 11:00-12:00 (Another Lab inspection)

Toms

Toms, Markuss and Stanislavs went to the lab again to test out some of the materials for the heat propagation experiment.

Observations: We tried to connect the vibration generators to the AC current supply. We attached a sturdy ruler to the vibration generators, held two laminated pages on the sides as walls, and dropped cut-up cotton swabs as particles (as that was all we had at hand). We decided that the vibration generators would be great, however, we will need to create a sturdy platform and walls, as well as how to properly attach the whole system to the vibration generators. As particles, we will probably use grains of rice or poppy seeds and will determine the amount by weighting them.

Thu 25/4 15:00-17:00 (Plan presentations)

Eduard

All members were present for the presentation of the plan. Stanislavs and Toms successfully presented the project and answered all questions. The idea was approved and it was suggested we finalize the set up as soon as possible due to its complexity. The feedback for the presentation was mostly positive, though we were missing a week-by-week plan of tasks for each member.

After the presentations, we discussed the possible materials (cardboard, plastic, wood) for the oscillating table and decided to look for them over the weekend to make a working setup prototype as soon as possible. t

Fri 26/4 19:10-21:10 (Writing the literature assignment)

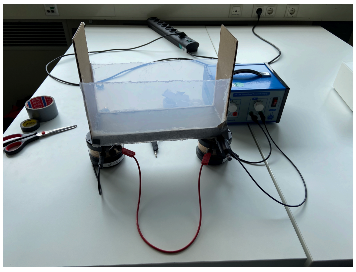
Markuss

We all joined a Google Meet call and agreed upon lab times for next week and shopping for materials. Created an overleaf file of the literature assignment, which was shared among the team. We then went on Google Scholar and other scientific databases in order to search for a review article that encapsulates the topic of our project; after finding various articles deemed reasonably suitable, we settled on Anomalous heat transport in one-dimensional systems: A description using non-local fractional-type diffusion equation. (Phys. Rev. Lett., 94:244301, 2005.) We then chose three articles from the list of references of the aforementioned article which seemed to the best of our knowledge most suitable for our project. Decided to meet up tomorrow to discuss the project and go to Bauhaus to purchase materials.

Sat 27/4 15:00-16:30 (Buying the required materials in Bauhaus)

Eduard

We all met in Bauhaus and looked for a material that would not resonate too much. Upon seeing the available options, the final purchase included a styrofoam box to use as a floor of the box that absorbs vibrations, a plastic box lid to cut up into transparent walls of our box to see the action going on on the inside, and glue with glue-gun to connect these. The cardboard that we had at home was also used.



Sun 28/4 (Building the first box)

Stanislavs

The first box was built using a styrofoam base and plastic(transparent) walls to enhance the visibility of the particles inside, unfortunately, since all of the team participants are theoreticians we forgot the fundamental principle of engineering: first, make a very simple prototype to just determine the range for the key parameters(like weight in our case) and only then start making the advanced prototype out of good materials. We neglected this principle (rookie mistake) and started making a very fancy box which turned out to be too heavy(500 grams). The lab visit demolished our hopes of using it in the practical setting.

Tue 30/4 13:00-15:00 (First lab visit)

Eduard

Stanislavs, Eduard, and Markuss were present. We connected the vibration generators to different AC sources and tested frequency and voltage ranges. Observations: The first box prototype was too heavy at around 600g with metal attachment and 350g without. Even after cutting off excess cardboard, the box was too heavy for the generators. A provisional setup was made using cardboard and duct tape to test the general behavior. Long-grain rice was used and deemed not suitable, while couscous was a great particle to observe but a smaller version would be preferable. The vibrations proved to be suitable for observing the entropic processes. Duct tape was used to attach the equipment firmly and big rubber blocks served to limit vibrations of the table and movement of the generators while reducing noise.

Wed 1/5 11:00-13:00 (Second lab visit)

Eduard

All members were present. A new box was built at home the day before, this time only using cardboard with glue, and reducing the size drastically to decrease the weight. Smaller couscous was purchased to compare which is best used for measurements.

Observations: The box worked very well, with the main problem being a reliable attachment to the vibration generators. Duct tape was used and will probably not be improved upon but this is the unfortunate nature of the generators. Another issue is actually observing the number of particles in each box compartment. Using slow-motion video and counting them manually would be very tedious, while artificial intelligence trained to recognize and count them on the video is unreliable and hard to train. The current solution options are: 1) cutting closable holes in the box to ease retrieving the particles from each compartment separately to weigh them. 2) Using removable "bags" in each compartment to hold all particles and enable removing them all at once while not disturbing other compartments.

Fri 3/5 13:00-14:00 (Second TA meeting)

Stanislavs

The second TA meeting was held, all of the Team members were present. The discussion went over the previous lab visits and the latest versions of the Lab setup. It became clear that the deadline for the midterm report is in two weeks, which means that the Team needs to start working on real data acquisition and the theoretical background of the report as soon as possible. Stanislavs and Markuss said that they are going to develop some simulation scripts to imitate the Lab setup with larger amounts of particles; these simulations can be considered a “theoretical” part of the midterm report(simulations were written).

Fri 17/5 (General status update)

Toms

For the past two weeks, nothing significant happened, but a lot of data was collected. We went to the lab for both of the two weeks and this week every day of the week someone was collecting data in the lab. We did it in pairs of two as that was the most efficient way. Today we have written a pre-report, which helped us to see what kind of data we need more, where we lack data, and where we have the largest errors and uncertainties. As well, Eduard has counted 8.22g of couscous grains (it is 4240 grains) which has greatly helped us to determine the weight in the range that the scale does not register.

Mon 20/5 (Further course of action)

Toms

During the writing of the pre-report, it was concluded that we need one more data point between 0.70A and 0.80A as there is a large jump in between. Furthermore, we realized that the data were not as qualitative as we thought they were. We realized that we needed to conduct the experiment for longer and not rush to the next amplitude so we need to take more measurements also for the amplitudes we already have. Furthermore, it was also understood that we cannot really use the same mass for every amplitude as the smaller amplitudes would lead to the particles interacting with each other quite a lot. And for the larger amplitudes, the small mass would lack precision. Therefore, it was decided that we vary the amount of mass but we normalize it after the mass in the first compartment.

Fri 24/5 (Broken set-up)

Toms

Unfortunately, it was understood that one of the sides was not too well attached to the system and therefore, it vibrates less. It was fixed as much as possible, but it still was faulty and that’s why this side was used as the “cold end” as it usually contained a lot fewer particles in any case. The platform itself also suffered some damage, the vibration generator rods had pierced the cardboard. This was successfully repaired.

Mon 3/6 (Amplitude)

Toms

All of the data was collected successfully (every one of us went to the lab regularly for this to happen) and we could finally move on to working with the data. Although one last thing was done - Toms went to the lab to record a video of liquefaction and also measured the amplitude of the oscillations depending on the applied current. Unfortunately, the vibrations were really small and the relative uncertainty was large. It was determined using a smartphone camera in the slow-motion regime (240FPS) and with a ruler next to the set-up. After that, the set-up was dismantled.