Generation and Analysis of Gas Particle Velocities
from the Maxwell-Boltzmann Distribution

Neema Rafizadeh[*]

Department of Physics and Astronomy,
University of Kansas, Lawrence, KS, 66045, USA

This experiment tested the ability for an apartment tenant to decide whether to trust their landlord about the furnace in their unit. The furnace temperature was set to be selected randomly from an inverse gamma distribution. To determine whether their apartment was actually a different temperature than the temperature outside, individual gas particles were sampled for their velocities and collected into a distribution. This was then compared to the null hypothesis and a statistical power analysis was performed. The number of particles needed before reaching an 80% confidence level for any randomly sampled temperature value of the furnace was determined through graphical analysis of the statistical power.

I. INTRODUCTION

Many physical systems studied in modern physics research follow the Maxwell-Boltzmann (M-B) distribution. From the study of electron diffusion in the semi-metal graphene to the modeling of photon interactions between a laser and a solid in photonics, the M-B distribution has proven to have great utility in explaining observed physical phenomena. The distribution has more fundamental roots in its derivation from the first principles of mass, temperature, and velocity of gas particles. However, the question becomes: how well can M-B curves truly be distinguished from one another?

If there exists a room that has an unknown temperature, the M-B distribution can be used to experimentally determine its temperature by sampling particles from the gas in equilibrium in the room. The process of removing individual gas particles from a room and measuring its velocity by hand turns out to be quite arduous and taxing, thus determining the fewest number of particles that need to be sampled before a temperature can be determined is useful to know.

Take, for example, an apartment resident is told that their unit has an furnace that is set to a random temperature sampled from an inverse gamma distribution while the temperature outside the building is 275 K. If the resident thinks their landlord is lying, they might want to determine if the furnace actually exists without access to it themselves. If they are able to measure the velocities of individual gas particles in the room, they can tell whether the conditioner is functioning by sampling the M-B distribution. The questions that they must consider become: *are the possible temperatures too close to be distinguished experimentally?* and *how many particles do we need to measure before we can tell them apart?*

---
[*] Email: rafizadehn@ku.edu

II. HYPOTHESES OF DIFFERENT TEMPERATURE GASES

The M-B distribution is derived from basic physics principles based on some assumptions, namely that the gas particles are non-interacting, non-relativstic, and in thermal equilibrium. If this is the case, then the probability distribution that you find a particle with a particular velocity is given by,

$$f(m, v, T) = \left(\frac{m}{2\pi kT}\right)^{1/2} 4\pi v^2 e^{-\frac{mv^2}{2kT}},$$

where m represents the molecular mass of the particles and T represents the equilibrium temperature of the gas. Two examples of this probability distribution are plotted in Fig. 1 with temperatures 100 K and 1000 K.
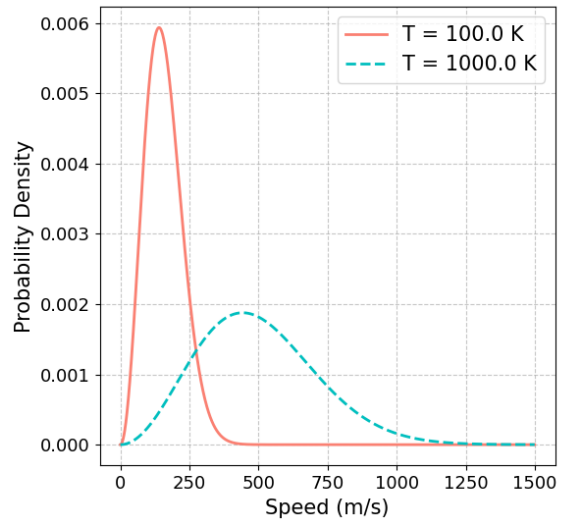

FIG. 1. Probability density for gas particle velocity at differing temperatures, 100 k and 1000 K. Values taken for a gas with an 85 amu molecular mass.

Note this is a probability *density*, not a direct probability. Thus, the probability can be calculated through

integration over a range of particle velocity. This means that the probability to find a particle at one particular velocity is 0 m/s as the integral will have the same two bounds of integration. Because of this, many statisticians plot cumulative density functions which represent the area under the curve of the probability density function up to a given value of velocity.

III. CODE AND EXPERIMENTAL SIMULATION

To simulate the furnace temperature being selected randomly from an inverse gamma distribution, a python script is written to generated this value efficiently. The inverse gamma function is given by,

$$f(y; \alpha, \beta) = \frac{1}{\Gamma(\alpha)} \frac{\beta^\alpha}{y^{\alpha+1}} e^{-\beta/y}$$

where α and β are predetermined prior values and y represents the data input which the inverse gamma distribution is describing. From the `PriorSampling.py`,

```python
def inv_gamma(x, alpha, beta, seed):
    np.random.seed(seed)
    y = (beta ** alpha) / (gamma(alpha)) * \
        (1/(x**alpha)) * np.exp(-beta/x)
    return
```

This listing uses inputs of α and β, which are predetermined prior values, to generate the inverse gamma distribution using the numpy gamma command. The distribution from the script is graphed and shown in Fig. below.
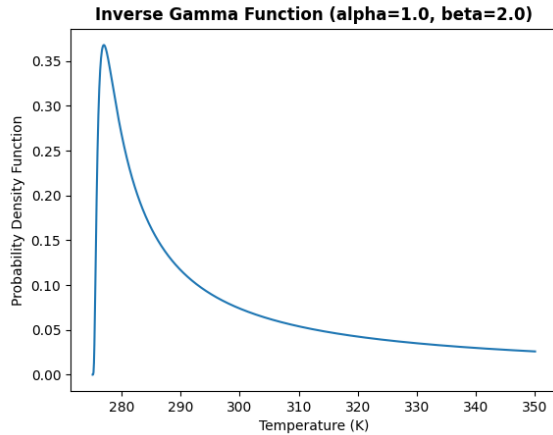

FIG. 2. Inverse gamma distribution.

The plots in Fig. 1 are idealized curves that are plotted directly from the M-B distribution equation. To simulate a real experiment sampling gas particles, a random number generator is used in conjunction with the M-B equation and plotted as a histogram shown in Fig. 2. This

is done by defining functions in python to return values of the M-B distribution, and also its inverse as shown in the listing below. From the `rng_MaxBoltz.py` file,

```python
def boltz(v,m,T):
    kB = 1.38e-23
    return (m/(2*np.pi*kB*T))**1.5 * 4*np.pi * v**2 \
        * np.exp(-m*v**2/(2*kB*T))


def inv_boltz(v,m,T):
    kB = 1.38e-23
    a = np.sqrt(kB*T/m)
    return erf(v/(np.sqrt(2)*a)) - \
        np.sqrt(2/np.pi)* v* \
        np.exp(-v**2/(2*a**2))/a
```

The M-B distribution takes physical parameters as an input and returns a probability density curve, while its inverse can give us the velocity values that we need to generate for the experiment. To get an entire distribution, another function can be defined to input arrays of randomly generated numbers to generate random velocities according to the M-B distribution.

```python
def velocities(n, seed):
    random = Random(seed)
    # rand_nums = np.random.rand(n)
    vals = []
    for i in range(0, n):
        vals.append(random.rand()
    )
    vel = inv_cdf(vals)
    return vel
```

This function requires use of the `Random` class generated by Dr. Christopher Rogan and is available in his GitHub listed in Appendix A.
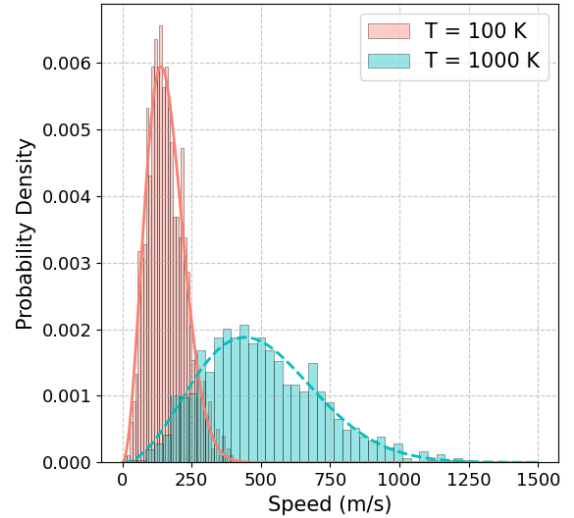

FIG. 3. Histogram of particle velocities taken at a N = 1000 sample size. Values taken for a gas with an 85 amu molecular mass.

As seen in Fig. 2, the randomly generated velocity val-

ues closely resemble the idealized curves from before. See Appendix B to reference how the randomly generated velocities fit the idealized curves significantly better at a much higher sample size. The temperatures shown in Figs. 1 and 2 were chosen to be 100 K and 1000 K for demonstration, but this does not help the tenant in the apartment on the hunt for their elusive furnace. The temperatures used in the experiment are 275 K, which represents the null hypothesis (H_0), and 320 K, which represents our hypothesis we are testing against the null (H_1).

With temperatures so close together in value, it may not be as obvious to resolve the curves as it is with the 100 K and 1000 K curves. See Appendix C for a graphical representation of the 275 K and 320 K histograms next to one another. Instead of resolving our curves graphically, we will use a statistical power analysis to differentiate between the data we collect and the data from the null hypothesis. Through analysis of our type II errors (referred to as the value β), we can determine how often our data *incorrectly* rejects the null hypothesis temperature, and then calculate our statistical power $(1-\beta)$ from that. The statistical power is essentially the reverse of the type II error as it tells us how often we *correctly* reject the null hypothesis. The higher the statistical power, the easier it is to distinguish data from the two temperatures, and thus the easier it will be for the apartment tenant to determine whether his landlord is juking him with his furnace. The calculation of this statistical power is given by the listing below.

```
for i in percentiles:
    vel1_temp = vel1[0:int(i*N)]
    vel2_temp = vel2[0:int(i*N)]
    sp_temp = PSD(vel1_temp, vel2_temp)
    cd_temp =
        (np.mean(vel2_temp)-np.mean(vel1_temp))/sp_temp
    statistical_power =
        tt_ind_solve_power(effect_size=cd_temp,
        nobs1=len(vel1_temp), alpha=aval, ratio=1.0,
        alternative='two-sided')
    power.append(statistical_power)``
```

This section of code uses a python package named `statsmodels` to calculate the log-likelihood ratios and uses it to calculate the power of the distributions. See Appendix A for more information regarding this package. It then appends it into an empty array named *power* for each iteration of the gas particles sampled.

IV. RESULTS AND ANALYSIS

The results of this method are shown in Fig. 3 for the randomly sampled temperature from the inverse gamma distribution.

The different temperatures in Fig. 3 show the difference in statistical power per sample size in comparison to the other temperatures. Statisticians generally regard 0.8 to
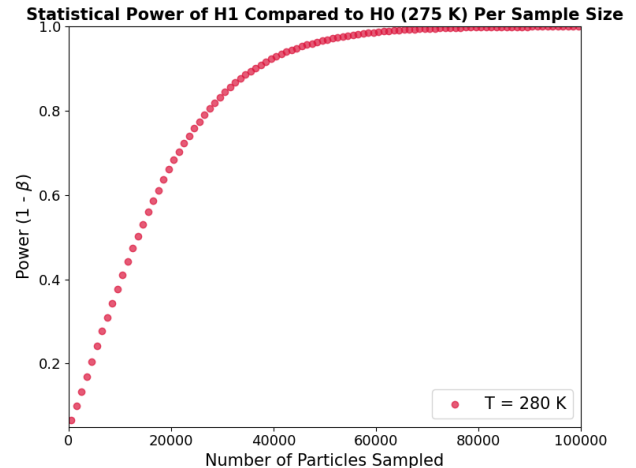

FIG. 4. Statistical power plotted as a function of sample size. Values calculated with an $\alpha = 0.05$ value, and a molecular mass of 85 amu.

be a statistical power significant enough to consider two distributions resolvable. Since the apartment tenant is trying to resolve a temperature of 320 K, they would need to sample roughly 250 particles before reaching the 80% confidence level.

Since the ability to distinguish between different scenarios of temperature depends heavily on the number of particles sampled, it is important for the apartment tenant to determine a temperature in which it will become unfeasable to distinguish between the distributions. They must then ensure the furnace is set outside of that range if they decide to pursue this experiment again.

This makes intuitive sense as if the distribution is farther away from the null hypothesis, the tenant is more likely to pick a particle that does not fall within the range of the null hypothesis and determine quickly that they are able to reject it. For a temperature very close to the temperature of the null hypothesis, 275 K, it will require many more sampled particles before the tenant can confidently say they have a working furnace. This is demonstrated by the trend shown in Fig. 3 by the difference between the 290 K curve and the 330 K curve.

V. CONCLUSION

A random number generator was used to sample velocity values from the Maxwell-Boltzmann distribution with a temperature parameter of 275 K and plotted into a histogram. Using this as the null hypothesis for the experiment, gas particles from the air were sampled from an apartment complex unit and compared against the null hypothesis to determine whether the complex had provided a furnace for the unit. The statistical power was calculated for different amount of sampled particles and a confidence level was found that could be used to

determine the state of the furnace in the complex.

The temperature of 320 K is far enough away from the temperature outside of the apartment (275 K) that the tenant of the apartment was be able to tell the landlord was not lying to them at a 80% confidence level from the 225 particles they measured. Through the use of statistical power analysis, the two distributions were able to be resolved from one another quite well despite the temperatures being so close in value. If the temperature values were closer, many more particles would need have been sampled before reaching the same 80% confidence level used by the tenant in their experiment.

In the future, this distribution can be studied further by changing other parameters of the M-B distribution. This may include things such as the mass or exploring similar distributions that do not have the same assumptions in their derivation from first princples.

Appendix A: Sources and Raw Data

All python scripts and data generated are contained in Neema Rafizadeh's GitHub repository for this project. Instructions for running the scripts and changing the inputs are listed on the README.md file listed in that repository.

The `MySort` and `Random` classes of functions can be located at Prof. Christopher Rogan's GitHub. The information regarding the Maxwell-Boltzmann distribution was sourced from the Wikipedia article discussing the subject. The calculation of the statistical power was done with the `statsmodels` package in python, with the docuentation provided here.