

THE MOTION OF THE SPHERICAL COW: NUMERICAL
SIMULATIONS OF THE DECELERATION OF A MULTI-ATOM
SYSTEM BY A MOTION THROUGH A GASEOUS
ENVIRONMENT

by

JULIA ABDOULLAEVA

A THESIS

Presented to the Department of Chemistry and Biochemistry
and the Robert D. Clark Honors College
in partial fulfillment of the requirements for the degree of
Bachelor of Science

May 2024

An Abstract of the Thesis of

Julia Abdoullaeva for the degree of Bachelor of Science
in the Department of Chemistry and Biochemistry to be taken May 2024

Title: The Motion of the Spherical Cow: Numerical Simulations of the Deceleration of a Multi-Atom System by a Motion Through a Gaseous Environment

Approved: Jeffrey Cina Ph.D.
Primary Thesis Advisor

Particle collision theory is applied in experimental settings such as mass spectroscopy and other particle fragmentation techniques to gain insight into observed behaviors. The energetic interactions that occur between large macromolecules and ions during a collision are complex and require several analytical tools and techniques to fully understand. The research described in this thesis raises the question of whether it is possible to develop a numerical simulation that simplifies this collision model and allows us to form a general understanding of particle interactions during a collision. Several computational methods are implemented in the development of this program including Monte Carlo, a binning procedure, and a double rolling procedure. The numerical simulation will aim to calculate an appropriate background gas particle distribution and relevant trajectories such as the monomer's center of mass velocity.

Acknowledgements

First I would like to thank my primary thesis advisor, Dr. Jeffrey Cina, for inviting me to work on this thought provoking research question and for guiding my learning. This research has played a huge role in sparking my passion for theoretical physical chemistry, and for inspiring me to pursue graduate studies. I am very thankful for all of the time and effort that was spent educating me. I would like to thank Dr. Rebecca Altman for being my CHC representative on this thesis. I would like to thank the Prell Lab at the University of Oregon for providing us with the experimental basis for the numerical simulation. I would like to thank the Pathway Oregon Program and the Summit Scholarship Program for funding my undergraduate education.

Table of Contents

Introduction	6
Computational Methods	11
Monte Carlo Method	11
Bins Procedure	12
Double-Roll Procedure	14
Results and Discussion	17
Conclusions and Future Directions	25
Appendix	26
Bibliography	38

List of Figures

Figure 1. Simulation setup	10
Figure 2. Distribution of particle velocities	17
Figure 3. Bin Distribution	18
Figure 4. Modified Bin Distribution	18
Figure 5. Distribution of Velocities in the n th Bin	19
Figure 6. Analytical Convergence to the Boltzmann Distribution at 1000 velocities	20
Figure 7. Analytical Convergence to the Boltzmann Distribution at 10,000 velocities	21
Figure 8. Analytical Convergence to the Boltzmann Distribution at 100,000 velocities	22
Figure 9. Analytical Convergence to the Boltzmann Distribution at 1,000,000 velocities.	23
Figure 10. Monomer center of mass velocity	24

Introduction

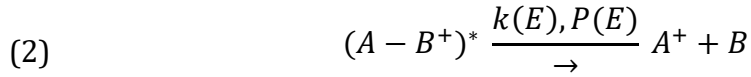
Numerical simulations in physical chemistry are essential tools for gaining a better understanding of experimental outcomes. The numerical simulation described in this thesis was developed in order to create a simple model of protein behavior in a gaseous environment. The original experiment was developed by the Prell Lab at the University of Oregon to determine mass fragmentation patterns of a protein structure.¹ Electrospray ionization (ESI) is an analytical technique in chemistry that aims to transfer a folded protein into the gas phase through ion mobility.¹ Ion mobility is a research technique that aims to promote separation of ionized molecules, as well as their identification. This technique is particularly relevant because of its ability to preserve elements of the initial structure along with the global size measurement.¹ The global size measurement is a way to describe the size of a complex using the mass to charge ratio. The application of these two techniques in conjuncture is referred to as native ion mobility-mass spectrometry (IM-MS).¹ IM-MS is relevant to experiments in bioanalytical chemistry, as it can be used as a technique to determine structural biology through a perturbation of the complex using gas-phase activation along with IM-MS.¹ Gas-phase activation is a technique used to fragment gas molecules through collisions with molecular ions. In the experiment of interest, IM-MS was implemented alongside gas-phase activation methods to determine the quaternary and tertiary structures of a protein complex.¹ IM-MS begins by introducing ions into an isothermal and isobaric gas chamber that is influenced by an electrical field.¹ As collisions occur, the technique promotes the degeneration of the protein complex to measure overall fragmentation patterns of the protein.¹ The numerical simulation takes foundational elements from the processes that occur within the IM-MS experiment to develop features of the

simulation. Although we aren't using the results from the simulation to make experimental insights, it is important to use realistic principles to develop a simple model.

Collision-induced unfolding (CIU) as well as collision-induced dissociation (CID) are combined in tandem to induce collisions between gas phase particles and the protein complex.¹ During the applications of CIU and CID, protein ions are accelerated to reach a high kinetic energy, and several thousand protein ions are forced to collide with the neutral gas atom. The collisions will eventually lead to the heating of the protein ions.¹ Once the protein ions surpass the maximum threshold for structure stability, the protein will begin dissociation or unfolding.¹ When a protein complex undergoes CID, it can provide information about the complex stoichiometry, as well as the identity of subunits.¹ Application of CIU allows experimenters to determine other important features such as the overall structure, binding patterns, and specificities of the ligand binding site.¹

In 1989, an article titled *Theory of Collisional Activation of Macromolecules. Impulsive Collisions of Organic Ions* was published by chemists at the University of Oslo describing computational methods for collisional activation.² The experiment that is initially described involves the guidance of ions towards a gaseous cell.¹ Energy transfers that occur between the ions and the gas cell result in ion fragmentation due to a technique referred to as collisional activation (CA).² Collisional activation is performed by positioning a collision cell directly between two mass analyzers.² The first mass analyzer (MS1) will transmit the selected ion, and the second mass analyzer (MS2) will scan and record the fragmented ions.² Collisional activation can be described using gas-phase kinetics, and the following model is used to show the time-dependent process of collisional activation and the resultant reaction that occurs because of the kinetic energy transfer.² This model denotes molecules *A* and *B* in various stages of

collisional activation. The ionic complex AB^+ undergoes a collision with a gas atom, G , to form the energized complex $(A-B^+)^*$.



Once the collisions have occurred, the energy distribution of the ions are referred to as the function $P(E)$.² The distribution of $P(E)$ is dependent on the transfer of energy between the ion beam and the internal degrees of freedom of the gas cell.² It is concluded that the rate of fragmentation during the experiment is dependent upon the distribution of $P(E)$ and the rate constants $k(E)$.² The ion fragmentation rate is determined by the unimolecular rate constants $k(E)$ as well as the internal energy distribution $P(E)$.² The theory of impulsive collision transfer (ICT) is a mathematical model used to describe energy transfers due to particle collisions in the experiment.² ICT defines the macromolecular ion as a sum of atoms that have masses $m_{a,j}$.² The summation of $m_{a,j}$ is defined as the total mass of the ion m_{ion} .² Prior to the collision and subsequent energy transfer, the macromolecular ion has a relative velocity in the lab frame of reference and is traveling on the defined x-axis within the mass spectrometer.² The lab frame of reference refers to the laboratory setting where the experiment of interest is performed. The kinetic energy and momentum of the ion are defined by equations 3 and 4.²

$$(3) \quad E = \frac{1}{2} \sum m_{a,j} v^2 = \frac{1}{2} m_{ion} v^2$$

$$(4) \quad p = \sum m_{a,i} v = m_{ion} v$$

In order to increase the quantity of fragmented ions, the amount of energy that is channeled into the system is increased by transferring a large amount of energy into the macromolecular ion.²

The center of mass energy of the macromolecular ion can be expressed using the following equation.² This equation indicates the maximum energy that can be used for the transfer to the internal degrees of freedom of the system.²

$$(5) \quad E_{cm} = \frac{m_g}{m_{ion} + m_g} E$$

The foundational theoretical concepts from the impulsive collision transfer (ICT) model are the basis for the development of the numerical simulation. The numerical simulation simplifies the treatment from a macromolecular ion to a unit referred to as the “spherical cow”. The spherical cow is a concept in theoretical physics that is used to describe highly simplified models. In the numerical simulation, the particle is modeled as a mono-unit spherical cow that participates in linear, one-dimensional motion through a gaseous environment. As described by the ICT model, the monomer will experience frequent collisions via gas atoms that result in changes of the total energy and center of mass velocity of the monomer. In my simulations, the focus is on the decrease over time of the monomer’s center of mass velocity. An important feature of the numerical simulation that is modeled after the gas-phase activation procedure in IM-MS, is a theoretical distribution of gas particles. The gas particle distribution is modeled after the infamous Boltzmann distribution which can be written in the following form where E is the energy of the system, k_B is the Boltzmann constant, and T is the temperature of the system.³

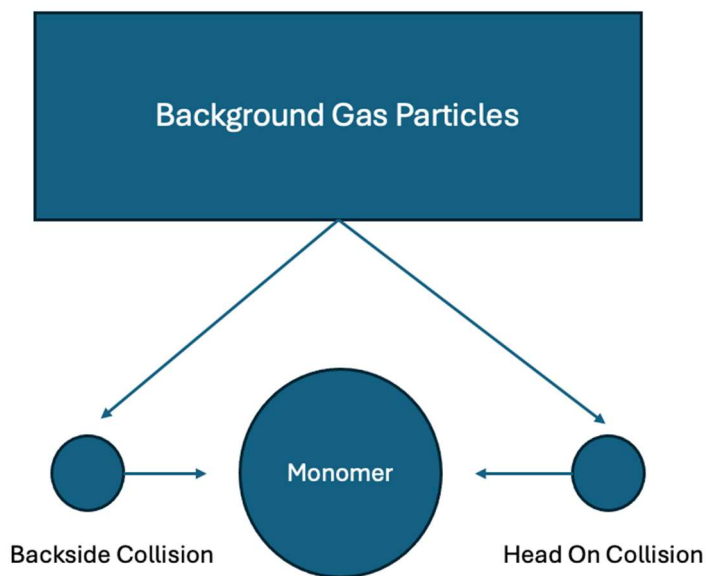
$$(6) \quad P \propto e^{-E/k_B T}$$

The Boltzmann distribution is applied in later calculations, along with factors accounting for the relative velocity of the monomer and a gas atom. Generally, this distribution is used to determine the behavior of the gas particles and how they interact with the monomer.

We can define two types of collisions that may occur during the simulation. A backside collision occurs when a gas particle hits the monomer from behind, this results in energy transfer

that alters the center of mass velocity. Similarly, a frontside collision is when the gas particle hits the monomer from the front. This will also result in an alteration of the center of mass velocity. Since the monomer is moving in one dimension, a backside collision will visually occur on the left side of the monomer, and a frontside collision will occur on the right side of the monomer. This can be visualized using the diagram modeled in Figure 1. These principles are used to develop the computational methods described below that can perform calculations and obtain results for further analysis.

Figure 1. Simulation setup



Computational Methods

Monte Carlo Method

A prevalent concept in computational chemistry that is often applied to large statistical ensembles is a method referred to as Monte Carlo.⁴ Monte Carlo procedures have a wide range of applications in research such as macromolecular structures and polymers. The method uses conditional probability to determine the appropriate statistical distributions for a system of study.

⁴ In the case of this simulation, a Monte Carlo method was developed in order to create a Boltzmann distribution of background gas particles that exist within an appropriate range of velocities. The appropriate range of velocities will be defined as any velocity between $-v_{max}$ and $+v_{max}$ where the variable v_{max} can be computed as follows where T_g and m_g are the temperatures of the background gas and the mass of the background gas particles, respectively.

$$(7) \quad v_{max} = \sqrt{\frac{2T_g}{m_g} \text{Log}[1000]}$$

My application of the Monte Carlo method was developed using the Mathematica software, and follows a procedure that is programmed to test each velocity. Randomized velocities are generated through a While/If loop structure where the initial step implements a RandomReal function. RandomReal is a Mathematica function that will generate a pseudorandom number between 0 and 1. The RandomReal function can be combined alongside other functions to generate an appropriate randomized number for the situation at hand. In the case of the Boltzmann distribution, we can combine the RandomReal function with a function that generates both negative and positive numbers, and multiply it by v_{max} to create an output that exists within the specified range. The program will then take the velocity, and generate a corresponding

probability using the Boltzmann distribution. The procedure will implement a condition for the calculated probability; if the probability is greater than a pseudorandom number generated by the RandomReal function, the condition is met and the gas particle velocity can be appended to a list of velocities. If the condition is not met, the program will restart the procedure until we reach the indicated number of gas particle velocities. The program is set to acquire one million gas particle velocities in order to have a complete understanding of the nature of this distribution.

Bins Procedure

Once the Monte Carlo method has generated the correct amount of gas particle velocities, we then implemented a procedure to categorize each velocity into a corresponding bin. The program will generate 200 bins, each with specific criteria for binned velocities. The width of each bin is defined as twice the value of v_{max} divided by the number of bins. A Do loop structure is utilized to test each velocity and determine the correct bin placement using the Floor function. The Floor function takes a number as an input and outputs the greatest integer less than or equal to the input. In this procedure, the Floor function is used to take each gas particle velocity and output the number of the corresponding bin. The function is written as follows where $v_{sb}[[nn]]$ is the specified gas particle velocity, v_{max} is the maximum gas particle velocity, and dv is the width of each bin.

$$(8) \quad bins = Floor \left[\frac{v_{sb}[[nn]] + v_{max}}{dv} \right] + 1$$

The Do loop will test each velocity using this procedure. The output of this function is the number of the corresponding bin that will be populated. It is important to note that the bin does not contain the numerical value of the velocity itself, instead it contains an integer which

represents the number of velocities that corresponds to that bin. Once every velocity is tested we obtain a rough representation of the Boltzmann distribution within the ranges of $-v_{max}$ and $+v_{max}$. We can then take the rough distribution and create a modified bin distribution by normalizing each value. This is accomplished by creating a new list of values that takes each bin and divides that value by the total number of elements in the original list of velocities. We can now implement a function that tests each modified bin for the middle value. We can define a function that calculates the middle of each bin using the following syntax where the variable n indicates the bin number.

$$(9) \quad mnb[n] := -v_{max} + \frac{dv}{2} + dv(n - 1)$$

This function is used in tandem with a normalized Boltzmann distribution function which is written as follows in equation 10;

$$(10) \quad P(mnb[n]) = \sqrt{\frac{m}{2\pi * Tg}} e^{-m(mnb[n])^2 / 2Tg} dv$$

The value for the middle of each bin is inputted into the normalized Boltzmann distribution, and the value obtained from the normalized bin distribution is placed into a list that when plotted is a representation of the distribution of velocities in each bin. We can then perform an analytical comparison between the original Boltzmann distribution and the normalized bin distribution to show a convergence to the Boltzmann distribution as we increasingly populate the velocity bins. This analytical convergence to the Boltzmann distribution is shown as we go from a distribution as small as 1000 velocities to the ideal range of 1,000,000 velocities. The convergence is an important factor to keep track of because we want to ensure the simulation is behaving properly. By performing the procedure at a wide range of velocities, we ensure that the simulation is actually converging to the ideal model.

Double-Roll Procedure

Now that we have a functioning distribution of gas particle velocities, we need to determine how the gas particles will interact with the monomer. The double roll procedure will determine when and what kind of collisions occur. There are two types of collisions that are relevant to the numerical simulation; a backside collision and a forward collision. This distinction is important to note because we will use slightly different probabilities and probability densities to describe the behavior of each type of collision. In the case of a backside collision we can define a probability for a collision between the monomer and a gas particle as follows where ρ is the linear density of the background gas particles, dt is the change in time, α is a variable defined as the mass of the gas particles divided by twice the temperature of the gas, and v_l is the monomer velocity.

$$(11) \quad p_1(v_1, \infty) = \delta t \rho \left\{ \frac{1}{2\sqrt{\pi\alpha}} e^{-\alpha v_1^2} - \frac{v_1}{2} \operatorname{erfc}[\sqrt{\alpha} v_1] \right\}$$

At the beginning of the timestep, the simulation uses the RandomReal function to select a number between 0 and 1. In the case where the randomized number is greater than the probability function, there is no collision that occurs during that timestep, and the procedure restarts. In the case where a collision does occur, we need to determine what type of collision occurs and ensure that the gas particle exists within an appropriate range of velocities. This part of the procedure is performed through a guess-and-select procedure using the probability density for backside collisions. The probability density is defined as follows where v_g is the gas particle velocity.

$$(12) \quad w_1(v_1, v_g) = \delta t (v_g - v_1) \operatorname{UnitStep}[v_g - v_1] \rho \sqrt{\frac{\alpha}{\pi}} e^{-\alpha v_g^2}$$

The probability density will take its maximum value when v_g is equal to $v_{g,bar}$. $v_{g,bar}$ is defined as follows in equation 13;

$$(13) \quad v_{g,bar} = \frac{1}{2} \left[v_1 + \sqrt{v_1^2 + \frac{2}{\alpha}} \right]$$

The following function will serve as a weighting function which accepts values between zero and unity. This can accept or reject values for v_g within the range of $-v_{g,max}$ and $v_{g,max}$.

$$(14) \quad \frac{w_1(v_1, v_g)}{w_1(v_1, v_{g,bar})} = UnitStep(v_g - v_1) \frac{v_g - v_1}{v_{g,bar} - v_1} e^{\alpha(v_{g,bar}^2 - v_g^2)}$$

After the weighting function determines the type of collision that occurs, we calculate what is referred to as the bump in velocity. This refers to the velocity changes that the monomer experiences as a result of frequent collisions, and is used to track the monomer's center of mass velocity. The velocity bump is calculated at the end of each successfully completed timestep, and can be expressed as follows where m_g is the mass of one gas particle, m is the mass of the monomer, and $vmf[[jj]]$ is the previous monomer velocity that is used for the updated calculation.

$$(15) \quad v_{bump} = \frac{2m_g}{m_g + m} (v_g - vmf[[jj]])$$

The velocity bump is added to the previous velocity to calculate the overall change in center of mass velocity. The probability for head on collisions is defined as follows.

$$(16) \quad p_3(v_1, -\infty) = \delta t \rho \left\{ \frac{1}{2\sqrt{\pi\alpha}} e^{-\alpha v_1^2} + \frac{v_1}{2} erf c[-\sqrt{\alpha} v_1] \right\}$$

Head on collisions will only occur if the rolled number between zero and unity is less than the probability for head on collisions to occur. The corresponding probability density is defined as follows.

$$(17) \quad w_3(v_1, v_g) = \delta t \rho (v_1 - v_g) \text{UnitStep}(v_1 - v_g) \sqrt{\frac{\alpha}{\pi}} e^{-\alpha v_g^2}$$

The probability density has a maximum value at the value $v_{g,dbar}$ which can be written using the following syntax.

$$(18) \quad v_{g,dbar} = \frac{1}{2} \left[v_1 - \sqrt{v_1^2 + \frac{2}{\alpha}} \right]$$

The weighting function for head on collisions, is written as follows, and is used to select gas particle velocities within the appropriate range for head on collisions.

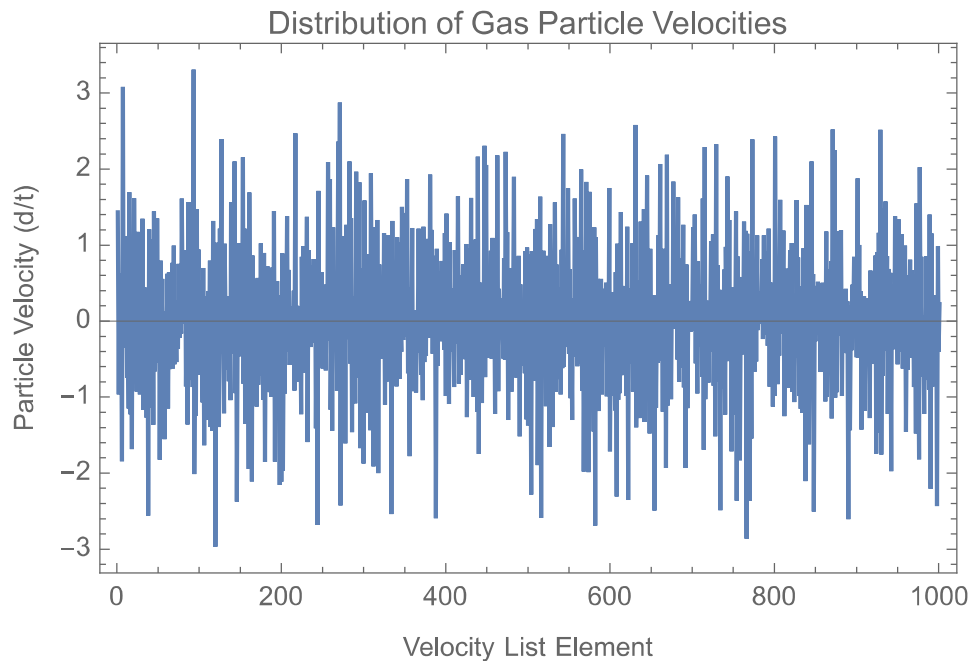
$$(19) \quad \frac{w_3(v_1, v_g)}{w_3(v_3, v_{g,dbar})} = \text{UnitStep}(v_1 - v_g) \frac{v_1 - v_g}{v_1 - v_{g,dbar}} e^{\alpha(v_{g,dbar}^2 - v_g^2)}$$

This part of the procedure will also calculate the bump of velocity after each successful timestep, and the updated velocity will be appended to the list of center of mass monomer velocities. Once the procedure has been performed the desired amount of times, we can plot the list of monomer velocities to view changes in the center of mass velocity throughout the simulation.

Results and Discussion

The Monte Carlo procedure allows us to generate a distribution of gas particle velocities, as shown below in Figure 2. The velocity list, where the gas particle velocities are stored, has 1000 list elements appended to it, each with a varying distribution of velocities ranging anywhere from $-3 v_{g,max} d/t$ to $3 v_{g,max} d/t$.

Figure 2. Distribution of particle velocities



As described earlier, each velocity is binned according to its numerical value, and is represented in Figure 3 as the unmodified bin distribution. The bin distribution is later modified as is shown in Figure 4, in order to create a normalized representation of the bin distribution for later comparisons with the Boltzmann distribution.

Figure 3. Bin Distribution

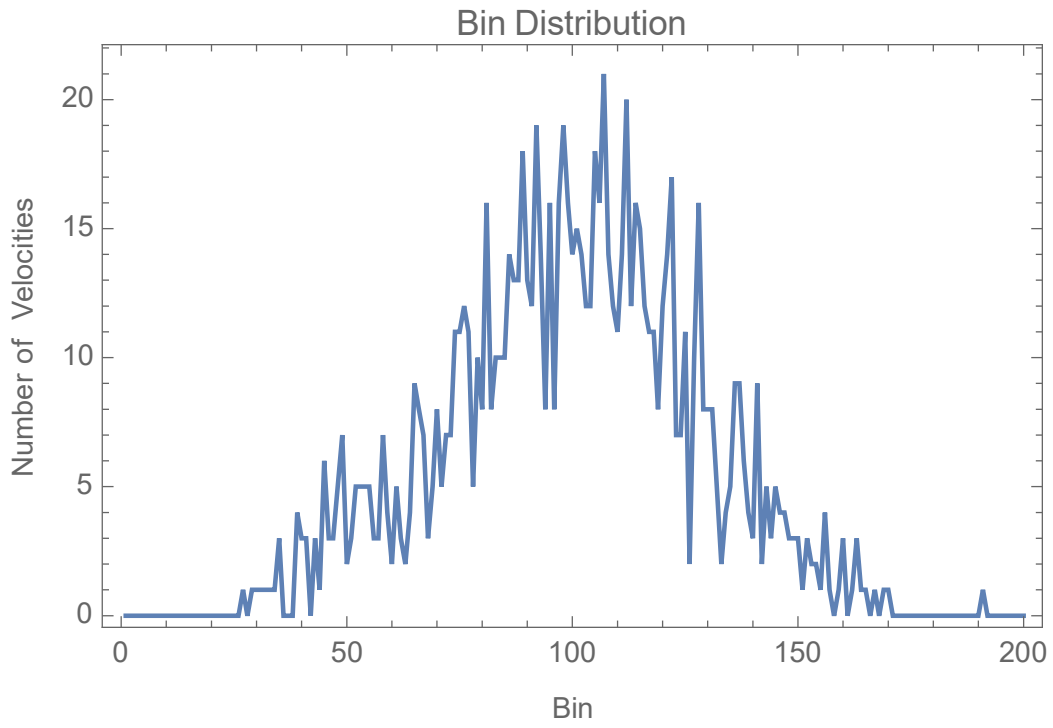
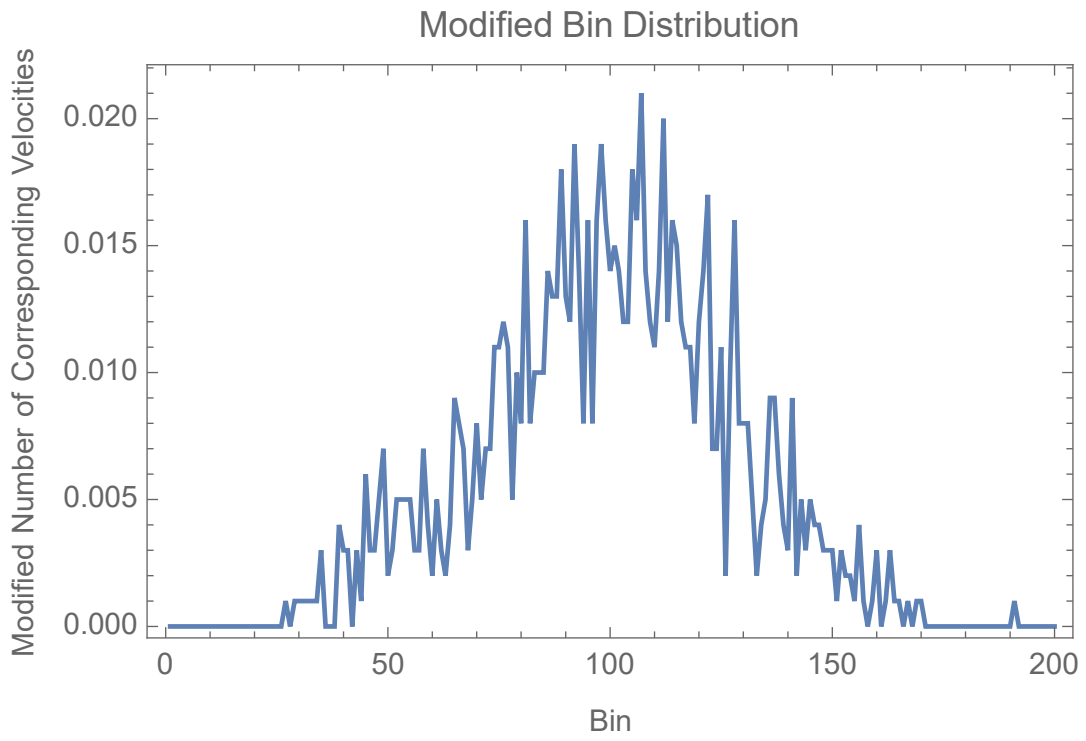
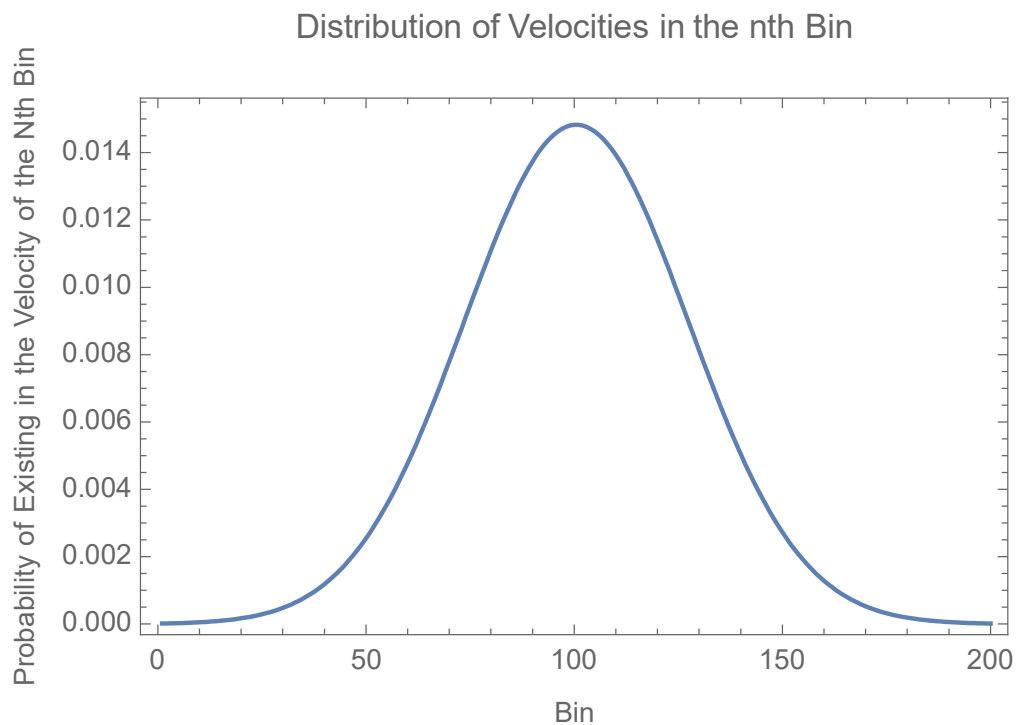


Figure 4. Modified Bin Distribution



The probability distribution described in Figure 5, shows the probability of a gas particle velocity existing in the n th bin. The distribution is modeled after the Boltzmann distribution, we can see that the greatest probability for a gas particle existing in a bin, is in the center bin of the overall distribution. As you go farther from the center bin, the probability decreases and approaches zero.

Figure 5. Distribution of Velocities in the n th Bin



In Figures 6-9, we plot the modified bin distribution together with the probability distribution described in Figure 5 at varying numbers of velocities in order to show analytical convergence. We begin at 1000 velocities in Figure 6 and eventually reaching 1,000,000 velocities in Figure 9. As the number of velocities is increased, we see that the modified bin distribution eventually begins to converge to the probability distribution. This indicates that the binning distribution closely follows a Boltzmann probability distribution, which is the desired

result for this computation. The Boltzmann distribution is the ideal model for the simulation because the Boltzmann distribution describes the behavior of an infinite collection of gas particle velocities. If the simulation was performed to obtain an infinite collection of gas particles, the distribution would ideally perfectly match a Boltzmann distribution.

Figure 6. Analytical Convergence to the Boltzmann Distribution at 1000 velocities

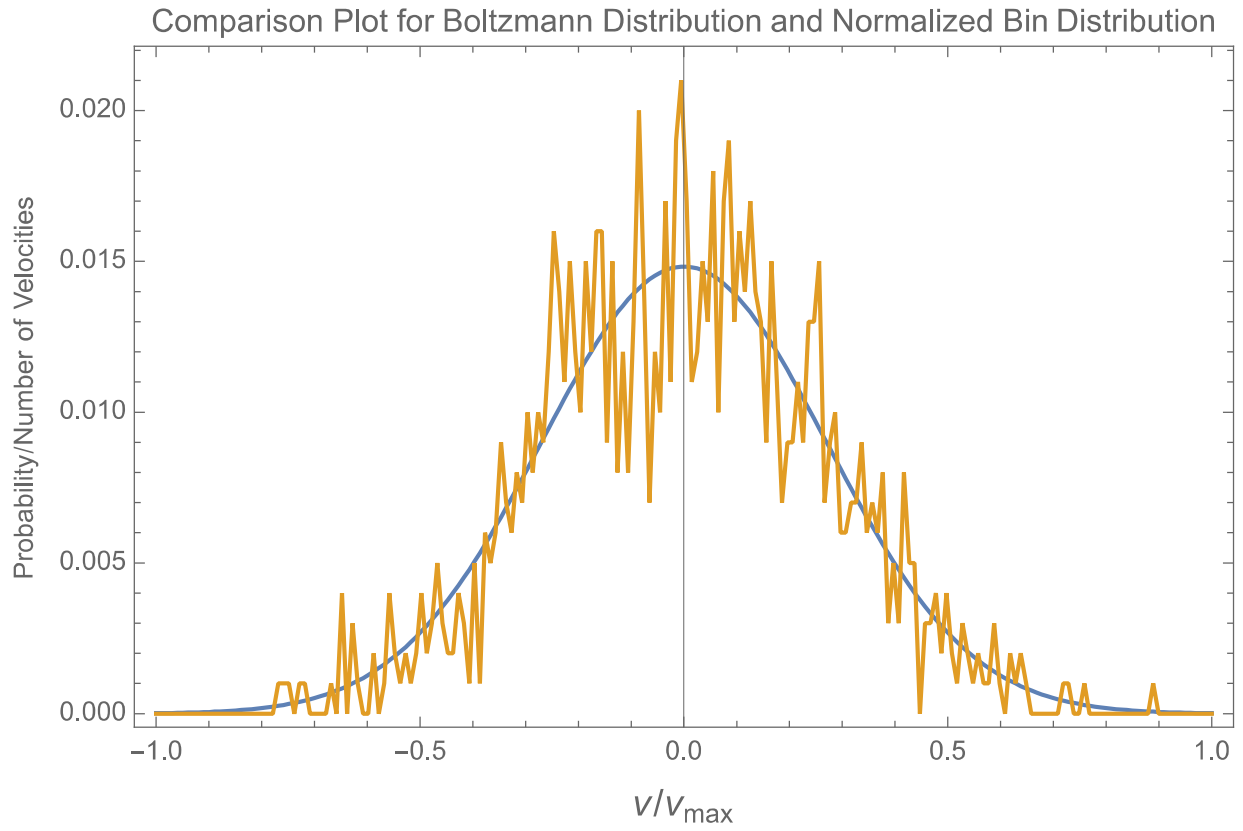


Figure 7. Analytical Convergence to the Boltzmann Distribution at 10,000 velocities

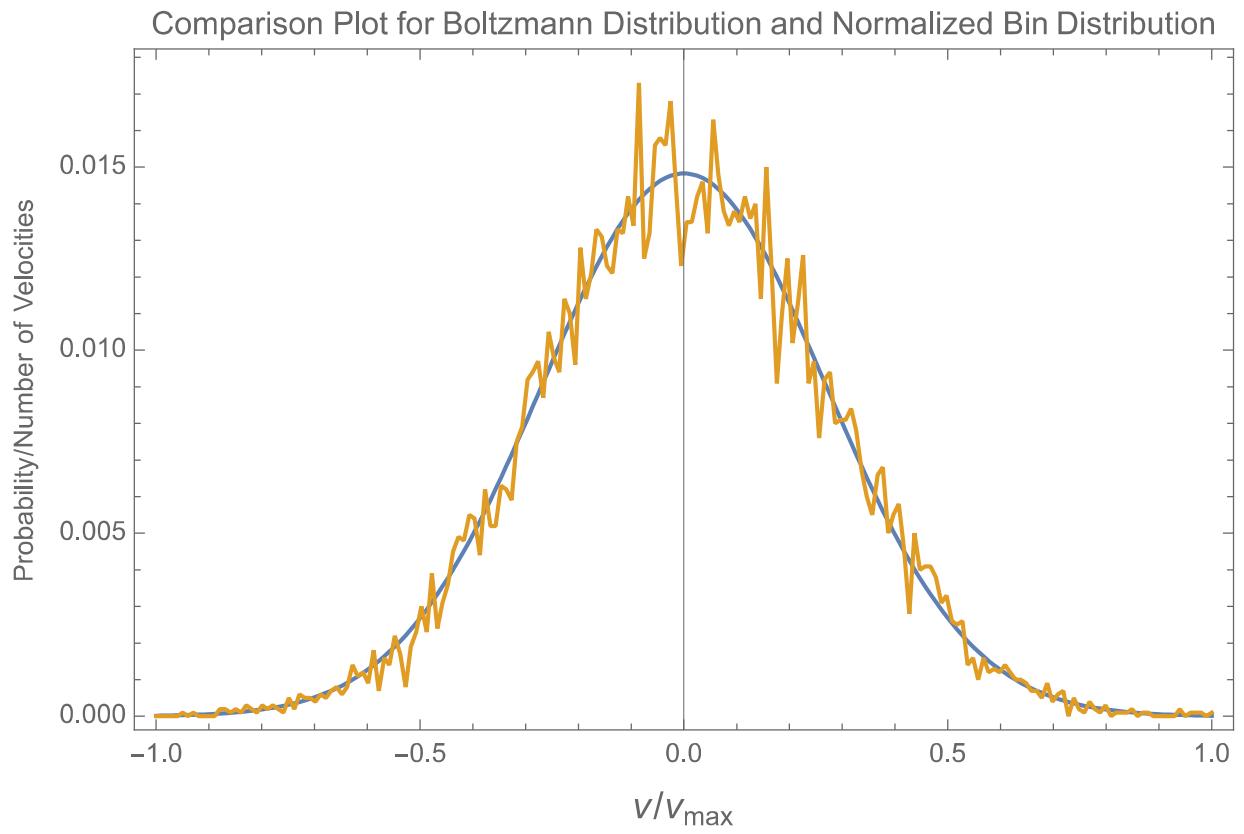


Figure 8. Analytical Convergence to the Boltzmann Distribution at 100,000 velocities

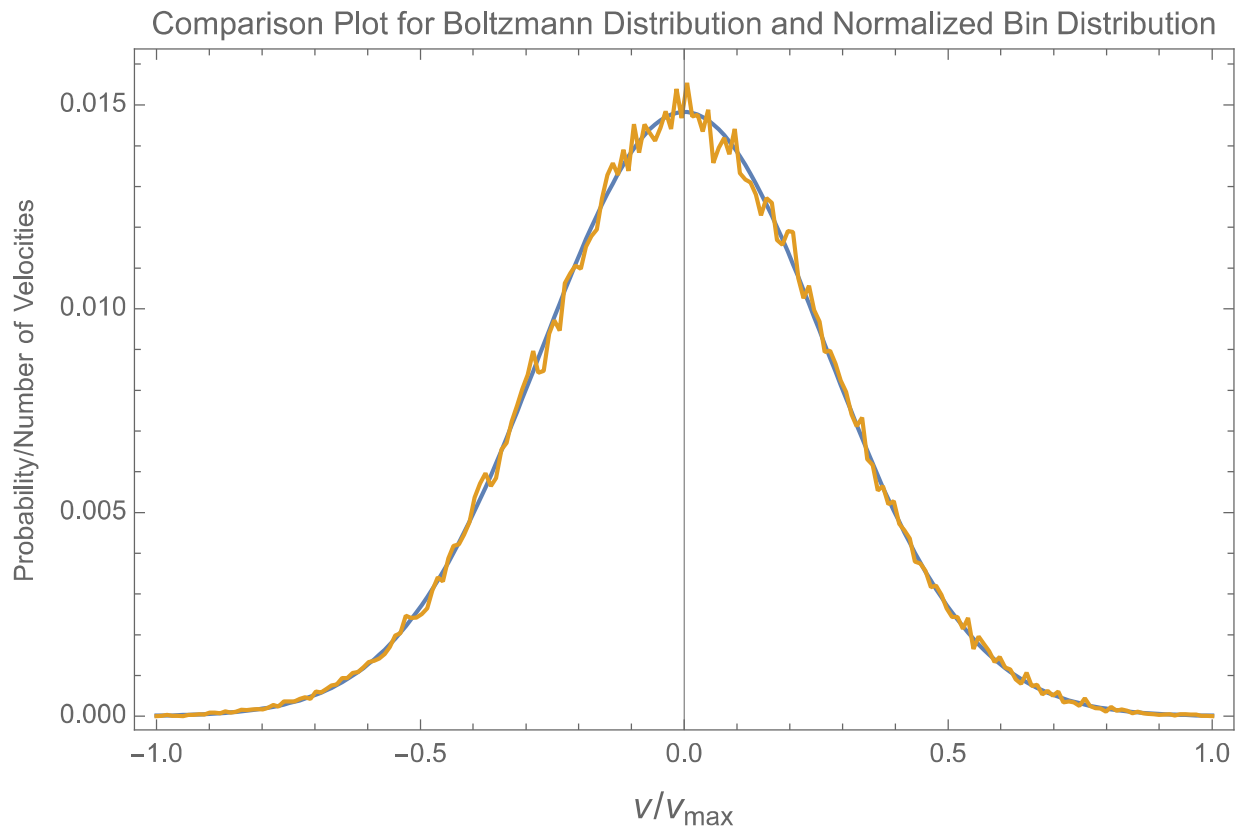
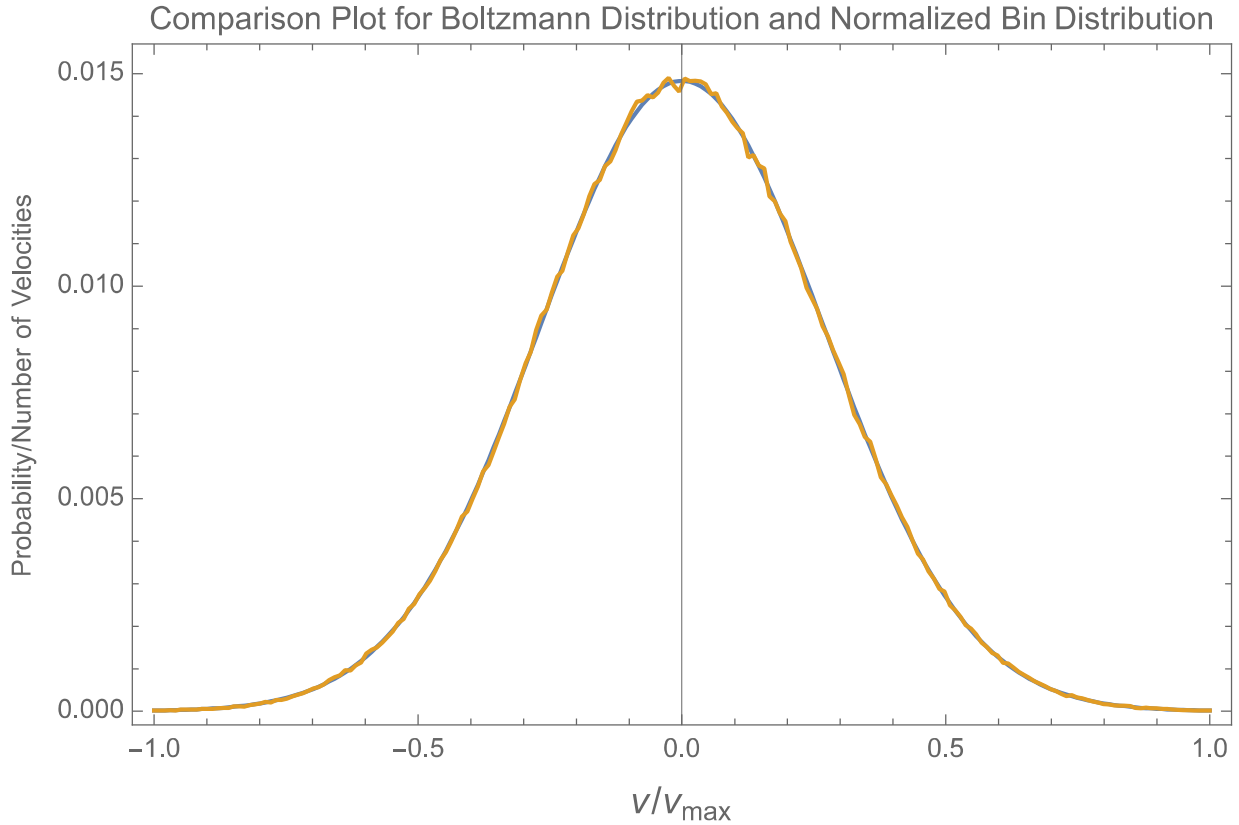
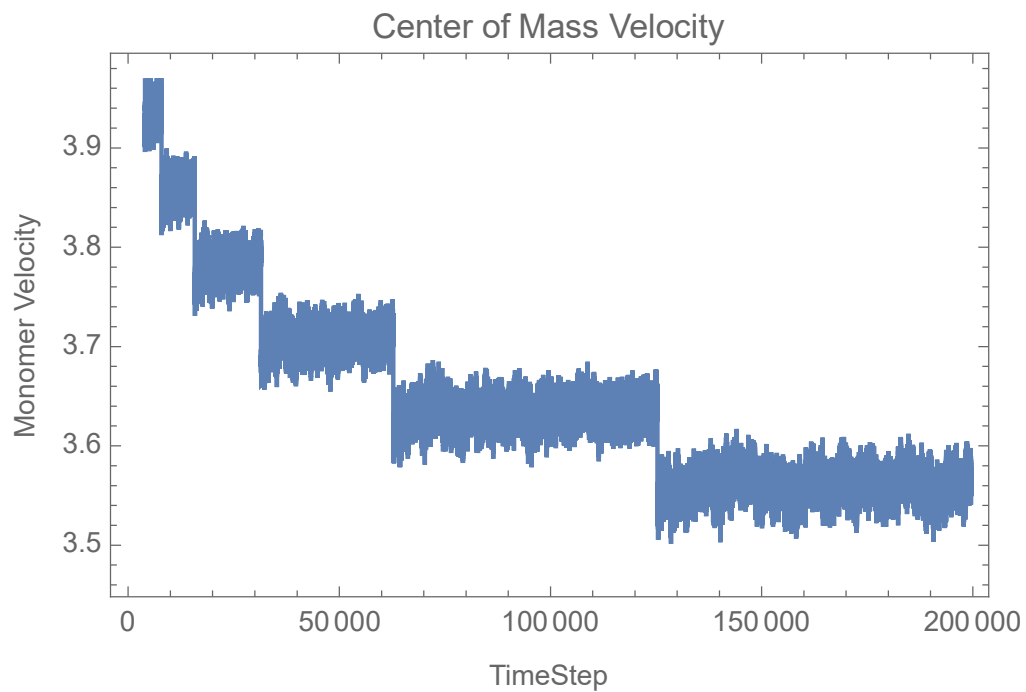


Figure 9. Analytical Convergence to the Boltzmann Distribution at 1,000,000 velocities.



The double-roll procedure described earlier allows us to plot the monomer's center of mass velocity which is graphed in Figure 10. The monomer has an initial velocity of $5.0 d/t$, and within a duration of 200,000 timesteps we see the monomer's velocity slow down to approximately $3.5 d/t$ in a stepwise way. Although it is expected that the monomer's center of mass velocity slows down, the stepwise pattern is unexpected. Initial predictions for the center of mass velocity is that it will slow down in a smooth pattern, but the results show that the velocity oscillates between a specific range of values and then rapidly drops down. For the purposes of this simulation, the current results are valid, but are open to further testing and development to investigate this pattern.

Figure 10. Monomer center of mass velocity



Conclusions and Future Directions

The results gained from this numerical simulation have allowed us to paint an initial picture of what occurs when a single-unit spherical cow interacts with a distribution of gas particle velocities. These initial results indicate that the Monte Carlo procedure and the binning procedure create a desired distribution of gas particle velocities that is modeled after a modified version of the Boltzmann distribution. We can also say that the monomer's center of mass velocity slows down throughout the course of the simulation. The numerical simulation has several future goals that need to be achieved in order to gain a more conclusive picture of particle collisions in a mass spectroscopy experiment. The next step that needs to be taken is a simulation of two harmonically bound particles, and how they experience heating and cooling due to energy transfers from collisions with the gas particles. The end goal of this program is to simulate a spherical cow with many components, in order to get a more conclusive picture.

Appendix

```
in:=
ClearAll["Global`*"]
(*Define a unit of mass*)
m = 1.0;
(*Define a unit of time*)
tau = 1.0;
(*Define a unit of distance*)
d = 1.0;

in:= (*Define the properties of the system*)
mg = 1.0 m;
Tg = 1.0 m * (d / tau)^2;

in:= (*Specify the maximum velocity of the gas particles*)
vmax =  $\sqrt{\frac{2 Tg}{mg} \text{Log}[1000]}$ ;

in:= (*Define a Boltzmann function*)
bz[V_] :=  $e^{(-mg \cdot V^2) / (2 Tg)}$ ;

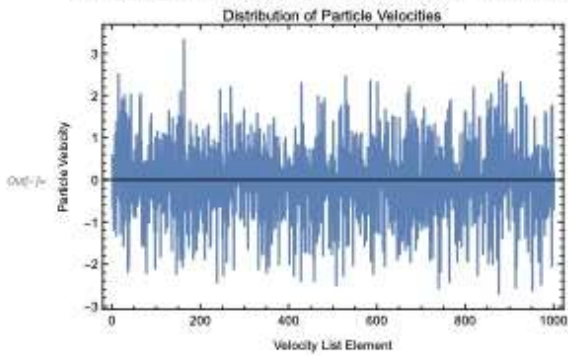
in:= (*Define a list of particle velocities*)
vsb = {};
(*How many elements should the list hold?*)
DS = 1000;
(*Define counter*)
nDS = 0;

in:= (*While Loop that tests each randomized velocity*)
While[nDS < DS + 1,
  VG = (-1)^nDS vmax (RandomReal[]);
  If[bz[VG] > RandomReal[], vsb = Append[vsb, VG]; nDS = nDS + 1]
```

```

In[ ]:=
(*ListPlot vsb to determine distribution*)
ListLinePlot[vsb, Frame -> True,
FrameLabel -> {"Velocity List Element", "Particle Velocity"},
PlotLabel -> "Distribution of Particle Velocities"]

```



```

In[ ]:= (*Bins Procedure*)

(*Define a table with approx B elements*)
B = 200;
vbins = Table[0, {j}, B];

```

```

In[ ]:=
(*Define dv; the width of the bins*)
dv = 2 vmax / B;

```

```

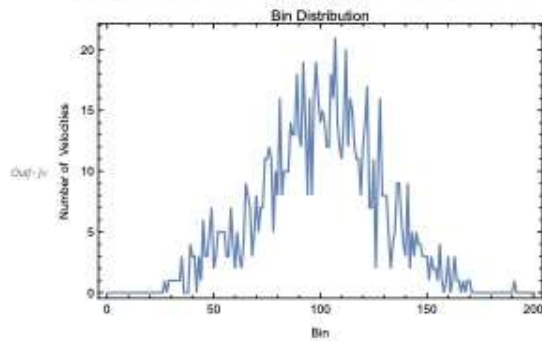
In[ ]:= (*Do Loop to bin each velocity*)
Do[
bins = Floor[(vsb[[nn]] + vmax) / dv] + 1;
vbins[[j] = bins] = vbins[[j] = bins] + 1;
, {nn, 1, DS}]

```

```

In[ ]:= (*Listlineplot the distribution of the bins
for comparison with the Boltzmann distribution*)
ListLinePlot[vbins, Frame -> True, PlotLabel -> "Bin Distribution",
FrameLabel -> {"Bin", "Number of Velocities"}]

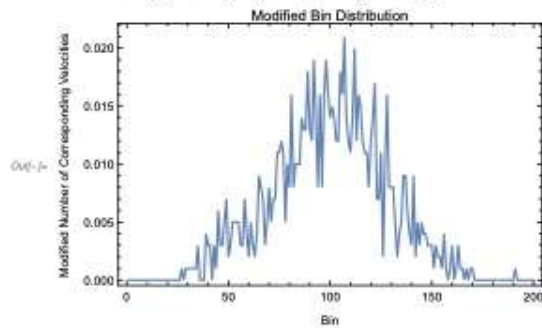
```



```

In[ ]:=
In[ ]:= (*Create a list nbins*)
nbins = Table[{(1/DS) * vbins[[n]], (n, B)}];
In[ ]:= (*Listlineplot table*)
ListLinePlot[nbins, Frame -> True, PlotLabel -> "Modified Bin Distribution",
FrameLabel -> {"Bin", "Modified Number of Corresponding Velocities"},
Ticks -> {{-vmax, 50, 100, 150, vmax}}]

```



```

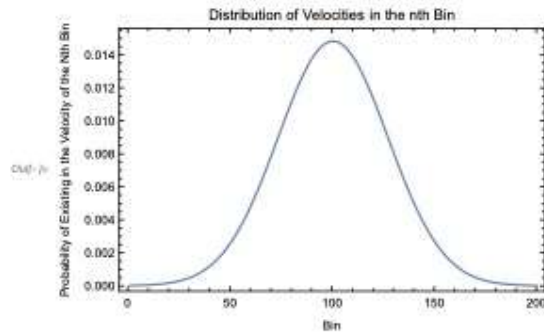
In[ ]:= (*Define a function for middle of the nth bin*)
mnb[n_] := -vmax + dv / 2 + dv (n - 1);
In[ ]:= (*Define a list with functions for the middle of the nth bin*)
plist = Table[ $\sqrt{\frac{m}{2 \pi * Tg}} e^{-\frac{(mnb[n])^2}{2 Tg}} dv, (n, B)$ ];

```

```

In[ ]:= (*Listlineplot p-list*)
ListLinePlot[plist, Frame -> True,
  FrameLabel -> {"Bin", "Probability of Existing in the Velocity of the Nth Bin"},
  PlotLabel -> "Distribution of Velocities in the nth Bin"]

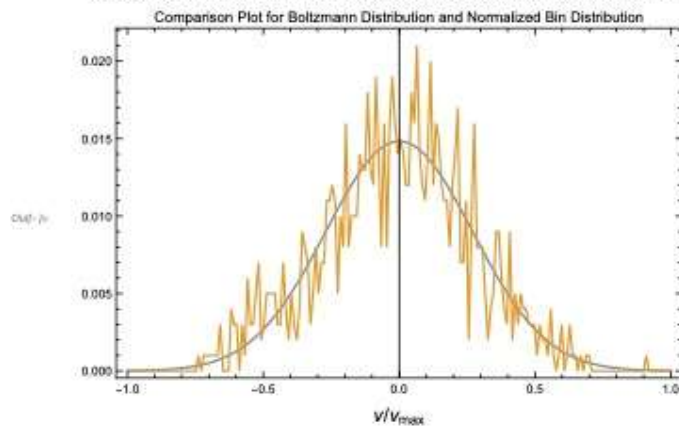
```



```

In[ ]:= (*Listlineplot plist and nbins for comparison*)
ListLinePlot[{plist, nbins}, Frame -> True,
  DataRange -> {-1, 1}, FrameLabel -> {"v/Vmax", ""}, PlotLabel ->
  "Comparison Plot for Boltzmann Distribution and Normalized Bin Distribution"]

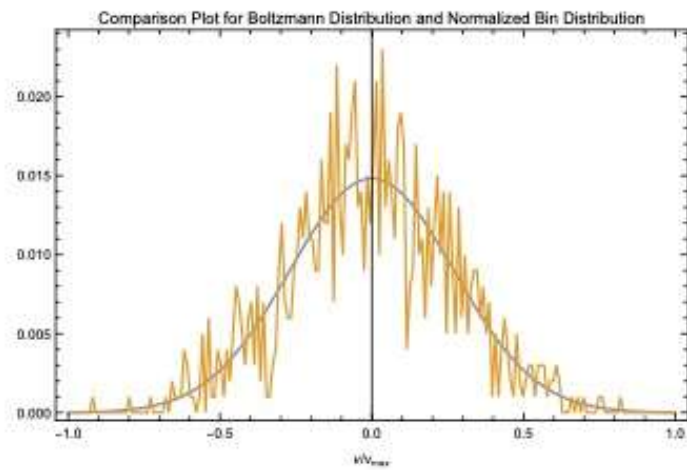
```



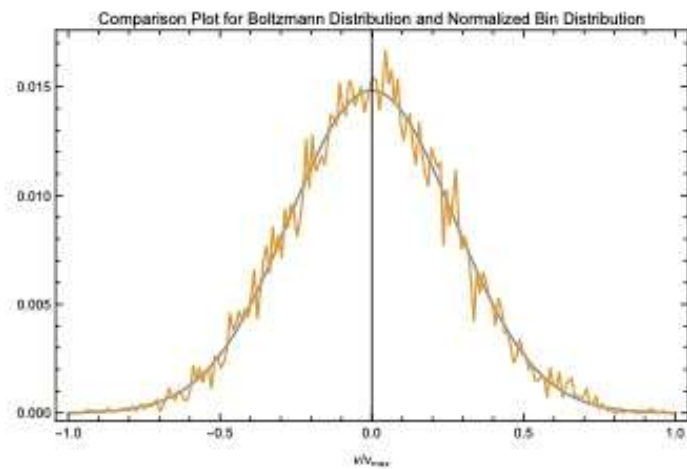
```

In[ ]:= (*Comparison plots of bin distribution and maxwell-
boltzmann distribution for analytical convergence*)
(*1000 velocities*)

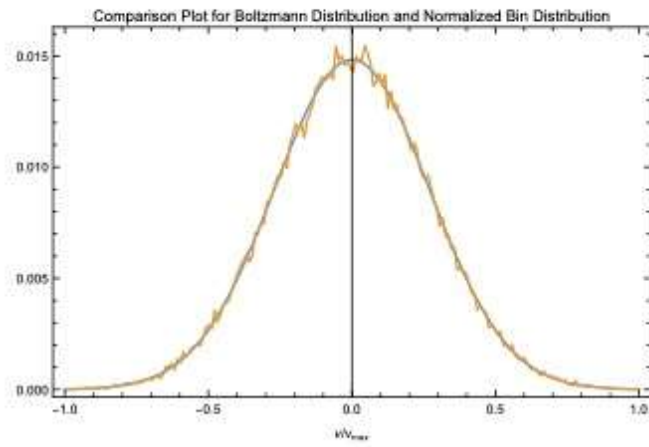
```



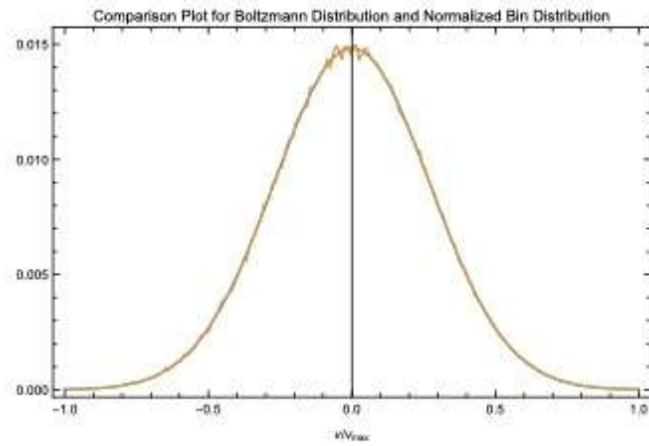
(*10,000 velocities*)



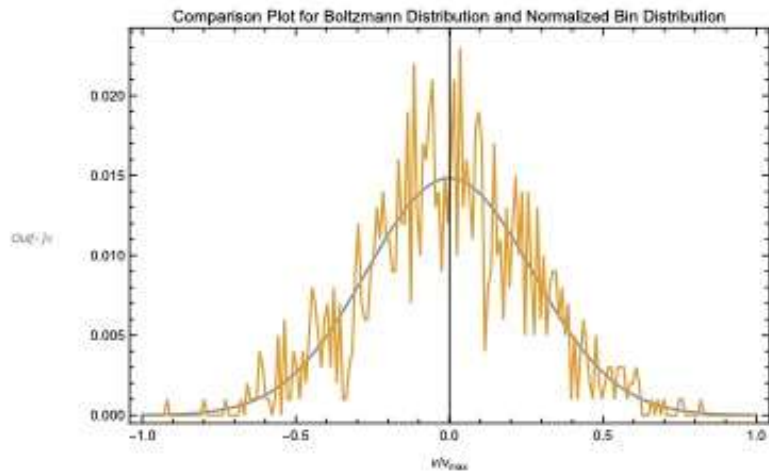
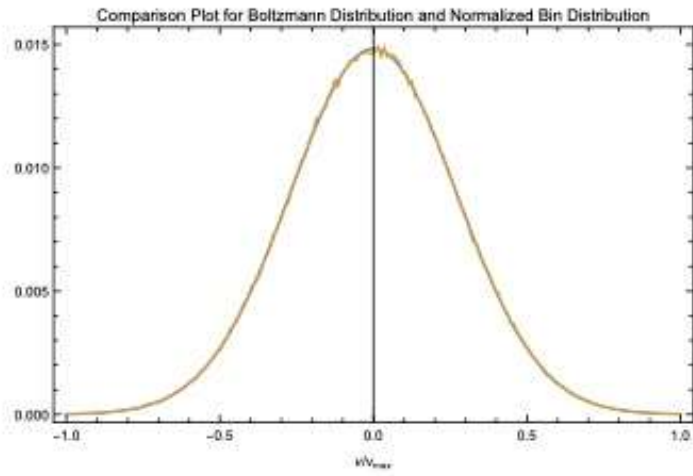
(*100,000 velocities*)

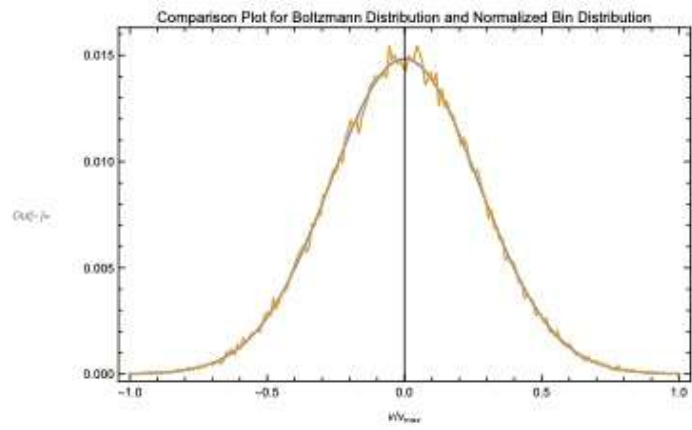
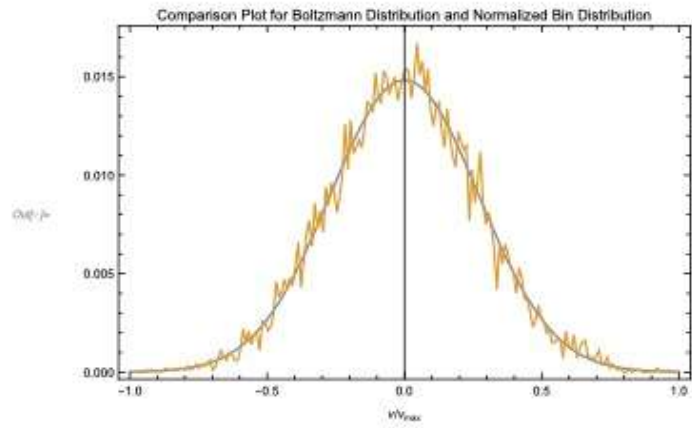


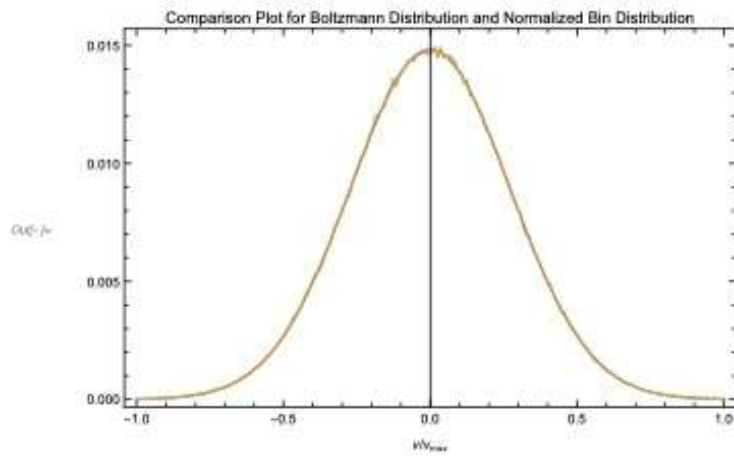
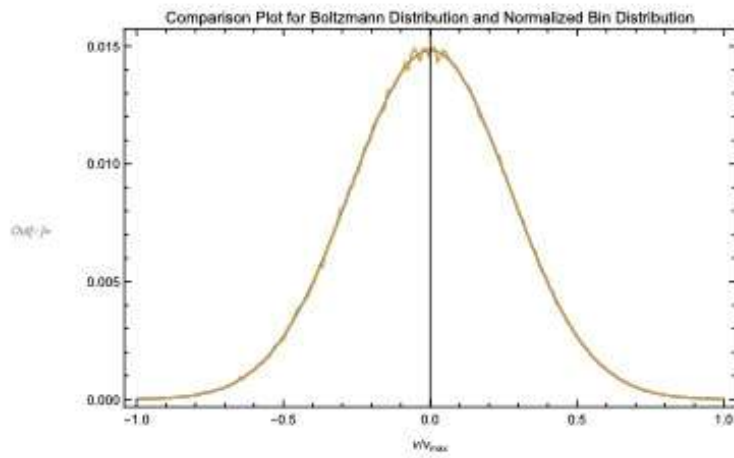
(*500,000 velocities*)



(*1,000,000 velocities*)







```

h[1]-> ClearAll["Global`*"]
      (*Define a unit of mass for the monomer*)
      m = 1.0;
      (*Define a unit of time*)
      tau = 1.0;
      (*Define a unit of distance*)
      d = 1.0;
      (*Define a timestep*)
      ts = tau / 1000;

h[2]-> (*Define the properties of the system*)
      mg = 1.0 m / 100;
      Tg = 1.0 mg * ((0.1 * d) / tau)^2;
      alp = mg / (2 Tg);
      vmi = 5.0 (d / tau);

h[3]-> (*Define a linear density of gas atoms*)
      rho = 1;

h[4]-> (*Define value for vg bar*)
      vgb[vm_] :=  $\frac{1}{2} \left( vm + \sqrt{vm^2 + \frac{2}{alp}} \right)$ ;

h[5]-> (*Define a value for vg double bar*)
      vgbd[vm_] :=  $\frac{1}{2} \left( vm - \sqrt{vm^2 + \frac{2}{alp}} \right)$ ;

h[6]-> (*Specify the maximum velocity of the gas particles*)
      vmax =  $\sqrt{\frac{2 Tg}{mg} \text{Log}[1000]}$ ;

h[7]-> (*Define a probability function that
      states the probability a backwards collision*)
      pr[vm_] := (rho) (ts)  $\left( \frac{1}{2 \sqrt{\pi * alp}} e^{(-alp * vm^2)} - \frac{vm}{2} \text{Erfc}[\sqrt{alp} * vm] \right)$ ;

      (*Collisional probability density for backwards collisions*)
      pd[vg_ vm_] := UnitStep[vg - vm]  $\left( \frac{vg - vm}{vgb[vmf[[j]]] - vm} \right) e^{[alp (vgb[vmf[[j]]]^2 - vm^2)]}$ ;

```

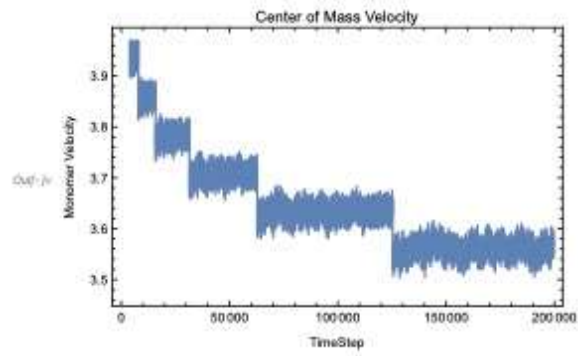
```

Mf:= (*Define a probability function that
states the probability for a head on collision*)
ph[vm_] := (rho) (ts)  $\left( \frac{1}{2 \sqrt{\pi * alp}} e^{(-alp*vm^2)} + \frac{vm}{2} \text{Erfc}[-\sqrt{alp} vm] \right)$ ;
(*Collisional probability density for headon collisions*)
pdh[vG_ vm_] := UnitStep[vm - vG]  $\left( \frac{vm - vG}{vm - vGbd[vmf[[jj]]]} \right) e^{(alp (vGbd[vmf[[jj]])^2 - vG^2)}$ ;
Mf:= (*List of updated monomer velocities*)
vmf = {vmi};

Mf:= (*Loop for determining the type of
collision and its impact on the COM velocity*)
Do[
(*Select collision number for backwards collision*)
If[RandomReal[] < pr[vmf[[jj]]], c1 = 0, c1 = 1];
(*Determine the change in COM velocity*)
If[c1 = 1,
Until[RandomReal[] > pd[VG, vmf[[jj]]], VG = vmax (2 RandomReal[] - 1)];
v1bump =  $\frac{2 mg}{mg + m}$  (VG - vmf[[jj]); vmf = Append[vmf, vmf[[jj]] + v1bump], v1bump = 0.0];
(*Select collision number for head-on collision*)
If[RandomReal[] < ph[vmf[[jj]]], c3 = 0, c3 = 1];
(*Determine the change in COM velocity*)
If[c3 = 1,
Until[RandomReal[] > pdh[VG, vmf[[jj]]], VG = vmax (2 RandomReal[] - 1)];
v3bump =  $\frac{2 mg}{mg + m}$  (VG - vmf[[jj]); vmf = Append[vmf, vmf[[jj]] + v3bump],
v3bump = 0.0], {jj, 1, 100 000}]

```

```
MJ:= (*Plot center of mass velocity*)  
ListLinePlot[vmf, Frame -> True, FrameLabel -> {"TimeStep", "Monomer Velocity"},  
PlotLabel -> "Center of Mass Velocity"]
```



Bibliography

1. Donor, M.; Shepherd, S. O.; Prell, J. S. Rapid Determination of Activation Energies for Gas-Phase Protein Unfolding and Dissociation in a Q-IM-ToF Mass Spectrometer. *Journal of the American Society for Mass Spectrometry* **2020**, *31* (3), 602–610. <https://doi.org/10.1021/jasms.9b00055>.
2. Einar Uggerud, and Peter J Derrick. “Theory of Collisional Activation of Macromolecules. Impulsive Collisions of Organic Ions.” *The Journal of Physical Chemistry*, vol. 95, no. 3, 1 Feb. 1991, pp. 1430–
3. Leonard Kollender Nash. *Elements of Statistical Thermodynamics*. Mineola, New York, Dover Publ., Cop, 2006.
4. Allen, Michael P, and Dominic J Tildesley. *Computer Simulation of Liquids*. Oxford,Oxford University Press, 2017.